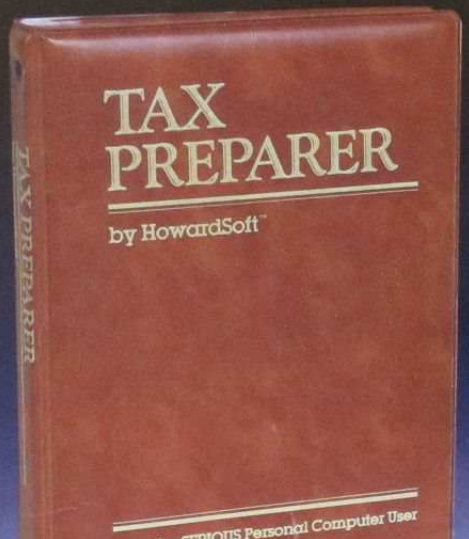
SOFTALK

VOLUME 3 JANUARY 1983 $3.00

THE RETURN OF THE PTERODACTYL

EXEC SUBLOGIC APPLES ON THE PHONE



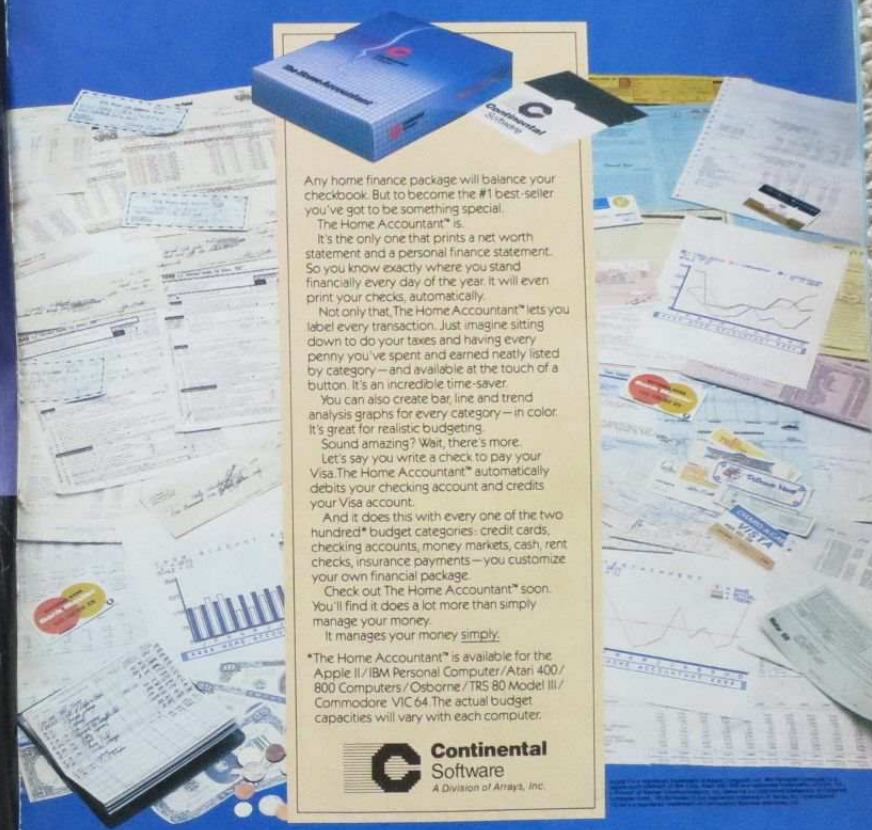
DON'T TAKE CHANCES WITH TAXES! Thousands of tax professionals have used our "Tax Preparer" in the past and have made it #1 in tax software. But we have continually improved the software in response to their "wants" so that NOW it is the most time-saving and error-eliminating tool that any tax preparer could ever own. And it is so fool-proof that even the total novice can now prepare tax returns.

TRULY AUTOMATED TAX RETURN PREPARATION. With low-cost updates as the tax laws change, year-long record-keeping for itemized lists, year-to-year data transfer, instant error-correcting, over 20 forms and schedules built-in, and some state supplements available too, it is the only professional package available at a reasonable price. And most important of all it can be TRUSTED to produce accurate results in a format that is accepted by the I.R.S., because tax-related software is our only business and we take it seriously.

COMPARE FOR YOURSELF at your local dealer. Whether you're a NOVICE OR PROFESSIONAL you'll see why HowardSoft is known for its QUALITY. There may be other tax software, but there is only one "Tax Preparer." (Apple and IBM-PC Editions Available.)

Howard Software Services/8008 Girard Ave., Suite 310/La Jolla, CA 92037/(619) 454-0121

The Home Accountant.™ The #1 best-seller.



Any home finance package will balance your checkbook. But to become the #1 best-seller you've got to be something special.

The Home Accountant™ is.

It's the only one that prints a net worth statement and a personal finance statement. So you know exactly where you stand financially every day of the year. It will even print your checks, automatically.

Not only that, The Home Accountant™ lets you label every transaction. Just imagine sitting down to do your taxes and having every penny you've spent and earned neatly listed by category—and available at the touch of a button. It's an incredible time-saver.

You can also create bar, line and trend analysis graphs for every category—in color. It's great for realistic budgeting.

Sound amazing? Wait, there's more.

Let's say you write a check to pay your Visa. The Home Accountant™ automatically debits your checking account and credits your Visa account.

And it does this with every one of the two hundred* budget categories: credit cards, checking accounts, money markets, cash, rent checks, insurance payments—you customize your own financial package.

Check out The Home Accountant™ soon. You'll find it does a lot more than simply manage your money.

It manages your money simply.

*The Home Accountant™ is available for the Apple II/IBM Personal Computer/Atari 400/800 Computers/Osborne/TRS 80 Model III/Commodore VIC 64. The actual budget capacities will vary with each computer.

Continental Software
A Division of Arrays, Inc.

It sells the most, because it does the most!

Continental Software Co., 11223 South Hindry Avenue, Los Angeles, California 90045 Telephone (213) 417-3003 • (213) 41

VOTES ARE IN;

NUMBERS UNCLEAR

It's My Party. The country is in a serious recession, unemployment is soaring, and future Social Security benefits are being threatened. The only person who could possibly benefit from such tragic conditions with a clear conscience is Edward Radanovich of Bellevue, Nebraska, the winner of Part 5 of the Oracle '82 contest.

Such a dismal economic outlook played an important role in the voters' giving the Democrats a stronghold on the House of Representatives, and Radanovich predicted a year ago that last November's elections would put 219 Republicans and 316 Democrats into Congress.

It wasn't exactly a wild guess. "I went down to the library and looked up past years' elections," says Radanovich. "The pattern seemed to be that whenever there was an off-year election the controlling party lost seats in the House of Representatives."

The actual count was 220 Republicans and 315 Democrats, giving Radanovich a score of −2. But he wasn't the only one with that score.

Tom O'Brien and Marc Epstein, both from Portland, Oregon, submitted predictions of 220 Republicans, 313 Democrats, and 218 Republicans, 315 Democrats, respectively. Because of the three-way tie, Radanovich, O'Brien, and Epstein all slugged it out in the random number generator.

It was neither pretty sight nor fair fight. During the first round, Epstein held Radanovich while O'Brien slugged Radanovich. In the second round, O'Brien held Radanovich while Epstein slugged Radanovich. But during the third round, O'Brien and Epstein held and slugged each other, while Radanovich slipped out the back door.

The prize Radanovich has his eyes on is Sierra On-Line's *General Manager*, and he'll pick it up at his local store, Computer Works.

Don't Forget the Senate. One of the deciding factors was that a lot of you forgot to count the Senate when counting your congressmen. Several contestants (those of you who are kicking yourselves) hit right on the nose or came very close for number of Republicans and Democrats in the House of Representatives. But because you left your senators sitting out in the cold, we had to subtract 100 points from your score. Sorry (heh, heh).

By the way, Epstein and O'Brien are still in the RNG pummeling each other to pieces. Hope they realize the contest is over in time to get back for the final part.

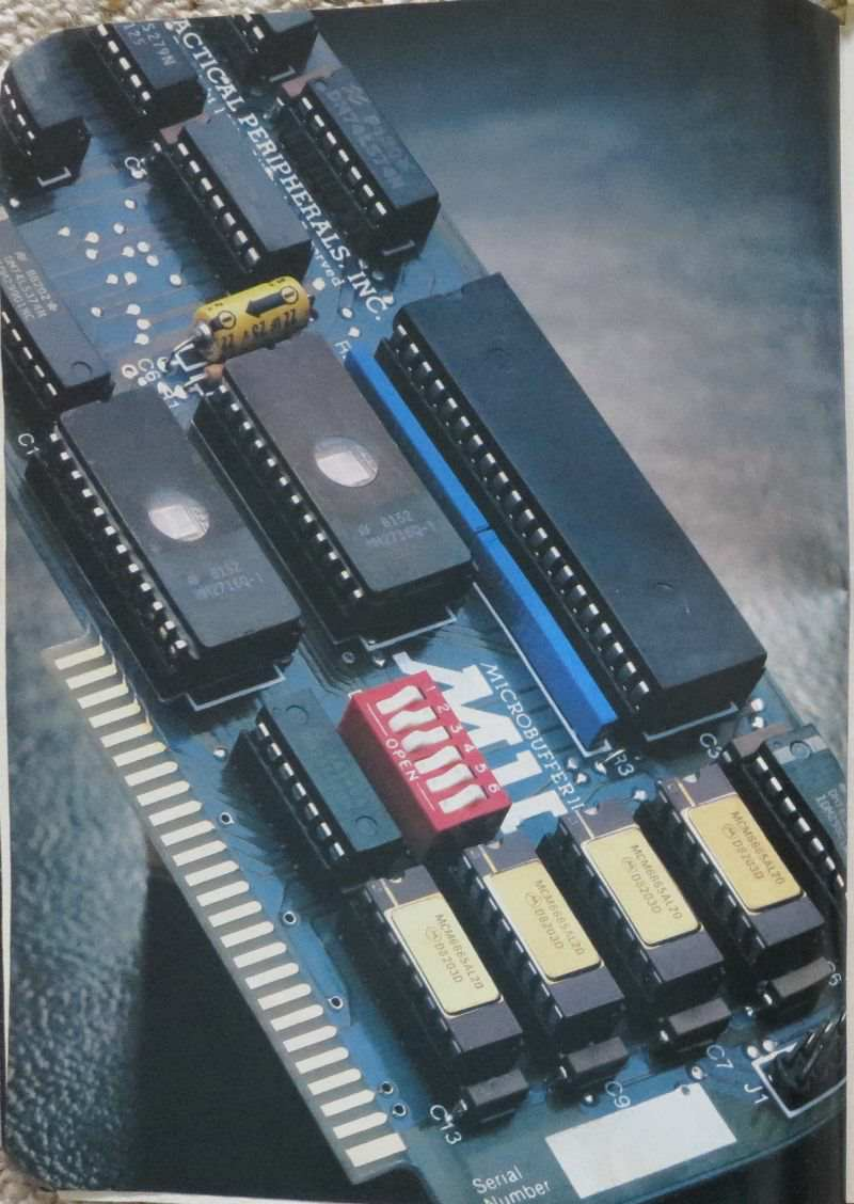
Digging Sig Fig Gig. October's venture into higher mathematics certainly separated the slide rules from the Univacs. A relative handful of entrants came up with the Significant Figure, and half that handful successfully assessed said figure's significance.

The figure in question was, of course, that old favorite, 11616125. For the finicky hair-splitters among you, it was also 11616125.3, and such answers were accepted as correct. (Just keeping you on your toes.) The winner of Part 1 split that hair down to two decimal places, but the random number generator loved him anyway. Larry Houston (Peru, IN) will receive his *Bag of Tricks*, *Diversi DOS*, and copy of *Practical Micro Programming* through Huntington Computing, as there is no dealer within thirty miles of him.

Whiz with decimals though he may be, Larry didn't even take a stab at the significance of the figure, making things that much easier on the excitable RNG in selecting Laura Kirschbaum (Shoreview, MN) as the winner of the contest's second part. As she put it: "Substituting letters of the alphabet for numbers to the left of the decimal yields 1 = A, 16 = P, 16 = P, 12 = L, 5 = E . . . 11616125 = APPLE." Special thanks to Laura for getting us off the hook, decimal-wise, by pointing out that "the 3 to the right of the decimal represents the third digit: 2 (0, 1, 2 . . .). So 11616125.3 = APPLE.2." Absolutely right. Laura wants *Prisoner 2* (appropriately) and *Rendezvous*, washed down with *Raster Blaster*, *Alien Rain*, or *Firebug*, our choice. For being so rude as to correct our spelling of the possessive form of Beatles, she'll just have to remain in suspense as to which one she's going to get.

Contest Notes. Man is a funny animal. Did you ever notice that, when worst comes to worst and life is hanging by a thread, we resort to all sorts of things to make our last gasps of breath valiant ones?

Witness the hapless person caught in a falling elevator, the cable of which has snapped. As he plummets toward the bottom of the shaft, he doesn't collect himself and pray for his salvation after death, nor does his life flash before him. Instead, he tries to time the fall of the elevator just right so he can jump up when the elevator is six inches from the ground. Go ahead and laugh. Ridiculous as it seems, such a silly



MICROBUFFER WILL SPEED UP ANY PROGRAM THAT REQUIRES PRINTING.

MICROBUFFER ALLOWS YOU TO PRINT AND PROCESS SIMULTANEOUSLY.

Now you don't have to wait for the printer to finish before you can use your computer again.

YOU CAN DUMP PRINTING DATA DIRECTLY TO MICROBUFFER.

Unlike your printer, Microbuffer accepts data as fast as your computer can send it. So there's never a bottleneck.

Microbuffer first stores the data in its own memory buffer and then takes control of your printer. This frees the computer for more productive functions.

Additional output may be dumped to the buffer at any time and it will be printed in turn.

THERE IS A MICROBUFFER FOR ANY PRINTER/COMPUTER COMBINATION.

Microbuffers are available in Centronics-compatible parallel or RS-232C serial versions.

FOR APPLE II COMPUTERS, Microbuffer II features on-board firmware for text formatting and advanced graphics dump routines. Both serial and parallel versions have very low power consumption. Special functions include Basic listing formatter, self-test, buffer zap, and transparent and maintain modes. The 16K model is priced at $259 and the 32K, at $299.

FOR EPSON PRINTERS, Microbuffer is $159 in either an 8K serial or a 16K parallel version. The serial buffer supports both hardware handshaking and XON-XOFF software handshaking at baud rates up to 19,200. Both Epson interfaces are compatible with all Epson commands including GRAFTRAX-80 and GRAFTRAX-80+.

ALL OTHER PRINTER/COMPUTER COMBINATIONS are served by the in-line, stand-alone Microbuffers. Both serial and parallel versions are expandable up to 256K. The serial stand-alone will support different input and output baud rates and handshake protocol. The 32K model starts at $299, 64K for $349. 64K add-ons for up to a total of 256K are just $179.

When you think of how much time Microbuffer will save, can you afford to *not* have one?

SIMPLE TO INSTALL.

Microbuffer II is slot-independent. It will fit directly inside the Apple II in any slot except zero.

Microbuffer for your Epson mounts easily in the existing auxiliary slot directly inside the printer.

The stand-alone Microbuffer is installed in-line between virtually any printer and computer.

MICROBUFFER FROM PRACTICAL PERIPHERALS.

Practical Peripherals is dedicated to establishing new industry standards for product performance.

The un-retouched photo at left has been enlarged to demostrate Microbuffer's exact workmanship and precise attention to detail. Specifications demand that each board undergo 36 seperate tests and inspections before it can leave the factory.

Ask your dealer for a demostration of the most practical, most successful new product of the year — Microbuffer.

PRACTICAL PERIPHERALS, INC.
31245 LA BAYA DRIVE
WESTLAKE VILLAGE, CA 91362
(213) 991-8200

move is your best bet when you're about to go under.

Or, how about those cartoon characters who start flapping their arms violently in a vain attempt to fly back to the cliff's edge that they've just run off. Or when massive boulders are about to crush them, they open up umbrellas while gritting their teeth, hoping for the best. We laugh at them because it's silly how our minds work in desperate times.

Now, let's all laugh (in good fun, of course) at some silly antics of contestants in desperation.

Who's Who Here? Hey, kids, is Carl Mueller (Murfreesboro, TN) in your class at school? If he is, tell the teacher to keep an eye on him during the next test. Carl sent in duplicate entries (both wrong)—one for himself and one for a mysterious "Karl Miller." We'll just assume Mueller is going through an identity crisis.

We all generally agree that finding the values of the letters was only slightly challenging. Plugging them into the formula and solving for the value of A was a bit messy, but not impossible for anyone who can operate a calculator. Apparently, some of you didn't even think of that.

Dishonorable mentions go to everyone who submitted entries with 6502 as a wild guess.

Paulette Caswell (Los Angeles, CA) gets eight such mentions. She entered, "The significant figure is: 6502 or 7825 or 4050 or 8100 or 8225 or 3,912.5 or 68,952,523.80 or 7,997.26028. If I win, I'd like to have something that makes my Apple II talk." Considering Caswell's entry, we would have sent her something to make her Apple stop talking.

Then there's Jean Armour (Liverpool, NY) and the gang at the Liverpool Public Library. "Dear guys: Every month we enter, and every month we don't win. We are a worthy cause! Please let us win this time. One of our patrons was a codesigner of the 6502 microprocessor, so we should get extra credit, providing that it is the correct answer." Well, provided that your answer was incorrect, we're not sending you anything. But you *are* a worthy cause, and we know a certain former movie star in Washington that you should write to for extra credit if your public funds ever run out.

Luke Meade (Yorktown Heights, NY), you have only your math teachers to blame for your blunder. Meade told us, "The 'times T' in the middle of your equation eliminates everything preceding it (T = 0). Tricky, but as I am taking math right now I look for that sort of thing." Darn the luck, Luke. That inaccurate math rule also eliminated you from the contest. Send your math teacher to detention hall.

Meade's math teacher sure gets around. Sam Robinson (La Mesa, CA) probably takes classes from him, too. "Since T = 0, the first part of the problem may be left off. You guys made it too easy!" Thanks, Sam. You made it real easy to leave your whole entry off the winners' list!

But we bet Robinson's a lot of fun at parties. After blowing the contest and coming up with 255,600 as his answer, he wrote, "The really significant part is the way you worded the problem. The significant figure is 2 (the significant figure of 255,600). Since rule 4 said the explanation must be more than two words but fewer than A words long, I will not tell you the significance of this number." That's okay, Sam. We won't tell you the price of the 1983 Mercedes you just missed winning.

Essay Test. By far, the most interesting answers came from two contestants who were convinced that their wrong answers were right, and who could probably also convince anyone else they were right. First, there was Steve Emmett (Herndon, VA), who determined his A to be equal to 11,510,789.3. Here's how Emmett figured it:

"The significance is learned by turning the number upside down and using a little imagination to arrive at 'E. Galois II,' for Evariste Galois II.

"As you know, Galois is credited with the founding of modern abstract algebra. [Sure, Steve, anybody knows that!] He gave impetus to the development of the theory of abstract groups. He also developed a general theory indicating the impossibility of finite algebraic formulation of solutions for polynomial equations of all degrees greater than 4."

Well, that's all very interesting, Steve, but because you used the wrong number we can't award you anything. By the way, your zip code turned upside down spells "Ilozz." As you know, that's Bezardian for "contests just aren't smiling upon you these days."

Our final "Gee, that's nice, but you're wrong" goes to Bernard Bopp (Toledo, OH). Bopp got all the clues right, but his computation skills went bopping out the door. He came up with 2,975,376.39 as his figure. So, what does it mean?

"My wife, a librarian, had the flash of insight on this: the number is a patent. It was filed on March 14, 1961, to the Princeton physicist Robert H. Dicke and refers to a special stable oscillator, which subsequently had a great impact on precise timekeeping (an atomic clock). Without precise time, precise navigation is impossible. Precise navigation was and is absolutely necessary for space missions. Orbital docking maneuvers, for example, would be impossible if celestial positions were known to a low accuracy.

"The space program was the prime impetus for miniaturization of electronic components; if the Apollo missions had not taken place, there would likely be no integrated circuits or microprocessors. And, if Dr. Dicke had not developed his stable oscillator, then Apollo, microprocessors, Apple IIs, and, yes, even *Softalk* might not exist."

Wow! What a long way to go just to be told, "Sorry, Mr. Bopp. Not even close." But the speculation of *Softalk* not existing piqued our curiosity. We showed Bopp's contest entry to a local gypsy clairvoyant who told us that if Dr. Dicke had not developed his oscillator, we would still have a magazine—*Softalk for the Smith Corona Adding Machine*. So there. ■

FASTALK

Fastalk is your quick guide to popular, specialized, or classic software. Programs appearing in Fastalk must meet one or more of the following criteria: (1) equal or surpass in sales the least-selling program to appear on any of the current bestseller lists; (2) relate to a specialized subject area and be in general distribution (more specialized packages and areas will be included as Fastalk matures); (3) be new and of professional quality (such programs will be carried for one month only—after that, they must meet other criteria for inclusion); (4) stand out as extraordinary.

Designation as a classic is noted by a bullet preceding a program's title.

Where opinion is expressed, *Softalk* has seen the software in question; the date of *Softalk*'s review, if any, is given at the end of the item.

Softalk may arbitrarily omit any package from Fastalk, whether or not it meets the foregoing criteria.

Adventure

- **Adventure.** Crowther, Woods. The original text adventure, created on mainframe, contributed to by many over a long time. Very logical within fantasy framework, excellent puzzles, maps; complex, convoluted, and great. Several publishers: Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $28.95. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $35. Frontier Computing, Box 402, 666 N. Main, Logan, UT 84321. $10.

Ali Baba and the Forty Thieves. Smith. Fanciful Arabian Nights role-playing game with a sense of humor. Fresh, fast action, challenging options, and secrets that are a joy to discover. Quality, 6660 Reseda Blvd., Ste. 105, Reseda, CA 91335. $32.95. *11/82.*

Cyborg. Berlyn. Text adventure with brief action skill game hidden in plot. As a futuristic cyborg, you're lost in a strange forest, desperately needing food and power. In its realism and use of true plot, it represents one of the most significant advances in adventuring since the orginal *Adventure*. Sentient, Box 4929, Aspen, CO 81612. $32.95. *11/81.*

Deadline. Blank, Lebling. Episode one in a projected series of murder mysteries by the authors of *Zork*. Interrogate, accuse, make transcripts. Includes inspector's casebook, lab report. Infocom, 55 Wheeler St., Cambridge, MA 02138. $49.95. *8/82.*

Escape from Rungistan. Blauschild. A vacation with a vengeance. Get out of jail; battle snakes, bears, and cannibals; acquire skills to get your money refunded. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *8/82.*

- **Hi-Res Adventure #1: Mystery House.** Williams. Whodunit in a Victorian mansion. First adventure with pictures. Vocabulary of more than 300 words. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $24.95.

Hi-Res Adventure #2: The Wizard and the Princess. Williams, Williams. Attempt to rescue princess from vengeful wizard. Features 250 illustrations in full color. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $32.95. *11/80.*

Hi-Res Adventure #3: Cranston Manor. DeWitz, Williams. More full-color adventuring involving the redistribution of wealth. Long on great riddles, short on plot. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *9/81.*

Hi-Res Adventure #4: Ulysses and the Golden Fleece. Davis, Williams. Re-creation of the Greek legend, featuring graphics advances and ability to communicate with the characters. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *12/81.*

Kabul Spy. Wilson. Cold War espionage adventure in which you must slip into Afghanistan to rescue a physicist before the commies make him talk. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $34.95.

Mask of the Sun. A unique animated graphic quest with unusual full parsing. See everywhere you can go as you travel, watch things transform. A professional-looking graphics breakthrough with nice puzzles. Ultrasoft, 24001 S.E. 103rd St., Issaquah, WA 98027. $39.95. *11/82.*

Prism. A forty-thousand dollar, real-life treasure hunt that's also a game. Solve the riddles and puzzles and find the clues to the location of three gold keys actually buried in the continental U.S. International Software Marketing, 120 E. Washington St., Ste. 421, Syracuse, NY 13202. $19.95. *11/82.*

- **The Prisoner.** Mullich. Superb TV series captured in computer game. Escape from an island requires player to solve logical puzzles, overcome obstacles, and answer riddles. Excellent computer fare; nothing else like it. Edu-Ware, Box 22222, Agoura, CA 91301. $29.95. *3/81.*

Prisoner 2. Mullich. Totally relandscaped version of original game: hi-res graphics added, puzzles reworked, obstacles expanded. Sophisticated and difficult exercise in intimidation with elements of satire. Edu-Ware, Box 22222, Agoura, CA 91301. $32.95. *10/82.*

Queen of Phobos. Hi-res treasure hunt. Outwit four opponents on derelict ship in space. Looters after your cookies, too. Phoenix, 64 Lake Zurich Dr., Lake Zurich, IL 60047. $34.95.

- **S.A.G.A. Series.** Adams. Scott Adams's prototypical adventures—twelve in all—spruced up with 100-color graphics and Votrax vocals. Fun, not always logical, very story-oriented series. First to make chance a significant element of play (you can get killed a lot). Each adventure has its own theme; you do a lot of exotic traveling. They map small but score big on imagination. Adventure Intl., Box 3435, Longwood, FL 32750. $29.95 each.

Starcross. Science fiction prose adventure that comes wrapped in a flying saucer. In the year 2186, your mission to harness a black hole takes some unexpected turns. Likable, engaging. Infocom, 55 Wheeler St., Cambridge, MA 02138. $39.95. *11/82.*

Swordthrust Series. Set of adventures, seven so far, that integrate fantasy role playing. Create one character, make new friends in each adventure, battle monsters and achieve goals together. Good stories, fun to map. Vocabulary no mystery but puzzles are. Single character goes through all. CE Software, 801 73rd St., Des Moines, IA 50312. Number 1 prerequisite for rest. Each adventure, $29.95. *8/82.*

Time Zone. Williams, Williams. "Microepic" hi-res adventure featuring ten periods from past and future history all over world and universe on eight double-sided disks. Good puzzles, many dangers. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $99.95. *1/82.*

Transylvania. Antiochia. Best graphics ever in a hi-res adventure. Excellent puzzles and logic—no tricks. Enjoyable. Penguin, 830 4th Ave., Geneva, IL 60134. $34.95. *10/82.*

Zork I. Lebling, Blank. Part one of mainframe adventure; understands complete compound sentences and questions. Simultaneous manipulation of objects. Text. Infocom, 55 Wheeler St., Cambridge, MA 02138. $39.95. *6/81.*

Zork II. Lebling, Blank. *Zork* comes into its own in sequence. Great text adventure technique and communication. Infocom, 55 Wheeler St., Cambridge, MA 02138. $39.95. *3/82.*

Zork III. Lebling, Blank. Text lives! A masterpiece of logic and a grand adventure to revel in. Hard, logical puzzle with unique point system. Infocom, 55 Wheeler St., Cambridge, MA 02138. $39.95. *9/82.*

Business

Accounting Plus II. Software Dimensions. Integrated package: general ledger, accounts receivable and payable, and inventory-purchasing modules. Basic and machine language. Menu-driven; prompting. Systems Plus, 1120 San Antonio, Palo Alto, CA 94303. $1,250.

Apple Plot. Converts numerical data into graphs; stores on hi-res page or prints out. *VisiCalc* interface. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $70.

BPI System. Popular five-module business package; programs also available separately. Includes general ledger (a bestseller), accounts receivable, accounts payable, payroll, inventory control, and job costing. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $395 each. Job costing: $595.

Computer Programmed Accountant. Five-module package: general ledger (very popular), accounts receivable, accounts payable, payroll, and property management. All other modules post automatically to general ledger. Continental, 11223 S. Hindry Ave., Los Angeles, CA 90045. $1,495. Separate modules: $250 each. Property management: $495.

Datadex. General-purpose database manager able to perform specific applications. File generation and report utilities allow definition of file structure and appearance of reports. Information Unlimited, 2401 Marinship Wy., Sausalito, CA 94965. $150. *9/81.*

The Data Factory. Passauer. Database management system allows listing files, getting file statistics, selecting another file, transferring records to new database, and adding fields to update forms. Disk swapping required; excellent product overall. Sev-

Spinnaker presents Delta Drawing.

An introduction to the modern art of programming using the ancient art of drawing.

Kids love to draw.

And with Delta Drawing, your children can have fun creating their own colorful drawings, and at the same time, learn about computer programming.

Delta Drawing is much more than just a fun game. It's an educational tool that helps children develop their creativity, learn to write computer programs and build an understanding of procedural thinking.

Easier to use than LOGO.

Delta Drawing makes it easy for children to use the computer—even if they've never used a computer before.

In fact, it's easier to use than LOGO turtle graphics. And Delta Drawing costs less than half as much as LOGO.

With Delta Drawing, your child controls the Delta cursor, and the Delta cursor controls all the action. So kids can learn at their own pace while they're having fun.

How your children will learn programming.

With Delta Drawing, your child first creates a picture using simple, single-key commands, for example: "D" for draw, "L" for left, "E" for erase.

As the picture is being drawn, the computer keeps track of every keystroke and records it as a program.

Then, by touching the "T" key for "text mode," the child can see the program that was written. As children progress, they can begin to write their programs in the text mode and switch to the graphics mode to see their pictures. They can even switch back and forth between graphics and text modes.

Delta Drawing will make your child comfortable with the computer and helps develop procedural thinking—the kind of step-by-step approach needed to write programs and solve problems with computers.

Get Delta Drawing today.

Delta Drawing can be played on Apple,® IBM® and Atari® computers.

It comes complete with easy-to-follow fast-start cards. And for more advanced users, a completely detailed instruction manual.

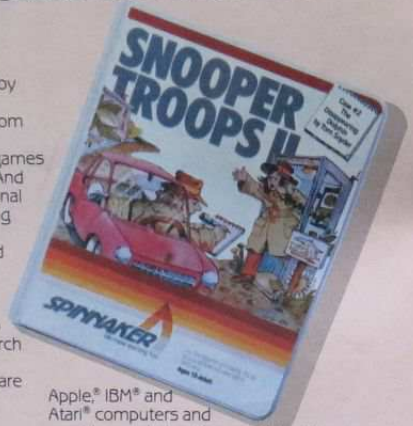
You can save your child's drawings on a blank disk and even print them, if you have a printer with graphics capabilities.

So look for Delta Drawing at your local software store. Or write to Spinnaker, 215 First Street, Cambridge, MA 02142.

You'll find that when we combined learning and fun, we created a work of art.

SPINNAKER™
We make learning fun.

Introducing Snooper Troops™ detective series. Educational games that turn ordinary homes into Sherlock homes.



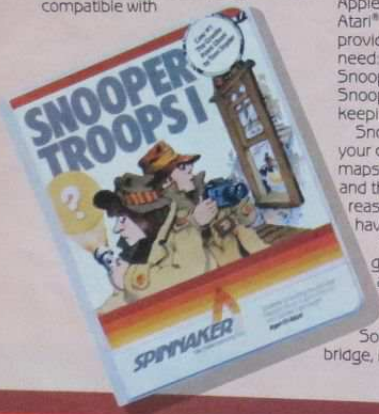
Where can you find educational games that your kids will really enjoy playing?

Elementary, my dear Watson. From Spinnaker.

Our Snooper Troops detective games are fun, exciting and challenging. And best of all, they have real educational value. So while your kids are having fun, they're learning.

As a Snooper Trooper, your child will have a great time solving the mysteries. But it will take some daring detective work. They'll have to question suspects, talk to mysterious agents, and even search dark houses to uncover clues.

The Snooper Troops programs are compatible with Apple,® IBM® and Atari® computers and provide your kids with everything they need: a SnoopMobile, a wrist radio, a SnoopNet computer, a camera for taking Snoopshots and even a notebook for keeping track of information.



Snooper Troops detective games help your children learn to take notes, draw maps, organize and classify information and they help develop vocabulary and reasoning skills. All while your kids are having a good time.

So if you want to find educational games that are really fun, here's a clue: Snooper Troops games are available at your local software store, or by writing to: Spinnaker Software, 215 First Street, Cambridge, MA 02142.

Spinnaker's early learning games will help make your children as smart as you tell everyone they are.

Your kids are pretty smart. After all, they're *your* kids.

Spinnaker can help make them even smarter. With a line of educational software that kids love to play.

Spinnaker games make the computer screen come to life with full color graphics and sound. And they're fun. Lots of fun. But they also have real educational value.

Some of our games help exercise your child's creativity. Others improve memory and concentration. While others help to improve your child's writing, vocabulary, and spelling skills.

And every Spinnaker game provides familiarity with the computer and helps your children feel friendly with the computer. Even if they've never used a computer before.

And Spinnaker games are compatible with the most popular computers: Apple,® Atari® and IBM.®

Our newest game, KinderComp™ (Ages 3-8) is a collection of learning exercises presented in a fun and exciting manner.

Rhymes and Riddles™ (Ages 4-9) is a letter guessing game featuring kids' favorite riddles, famous sayings and nursery rhymes.

Story Machine™ (Ages 5-9) lets children write their own stories and see them come to life on the screen.

And FACEMAKER™ lets your children create their own funny faces and make them wink, smile, wiggle ears (not your kids' ears, the ears on the screen), etc.

And we're introducing new games all the time.

So look for Spinnaker games at your local software retailer, or by writing to: Spinnaker Software, 215 First St., Cambridge, MA 02142. And show your kids how smart their parents really are.

SPINNAKER™

We make learning fun.

eral compatible products available. Micro Lab, 2310 Skokie Valley Rd., Highland Park, IL 60035. $150. *8/81.*

Data Perfect. Assembly language database companion to *Letter Perfect*; compatible with lower case in 40-column, most 80-column boards. Lay out, revise own screen, record design. Excellent built-in editor; ability to be edited by word processor. Searches, sorts, generates reports. LJK, Box 10827, St. Louis, MO 63129. $99.95.

Data Reporter. Allows plotting of data in various charts and graphs; stores data segmented by up to thirty-five fields. Machine language search and sort. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $220.

dBase II. Speedy relational database management system. Requires SoftCard. Ashton-Tate, 9929 W. Jefferson Blvd., Culver City, CA 90230. $700.

DB Master. Comprehensive database management system with password protection, extensive report creation options. Up to 1,020 characters per record. Stoneware, 50 Belvedere St., San Rafael, CA 94901. $229. *10/81.*

DB Master Utility Pak I. Compatible with version III. Translates *DB* files to Apple text, restructures existing files, replicates and merges, and recovers crashed files. Stoneware, 50 Belvedere St., San Rafael, CA 94901. $99.

DB Master Utility Pak II. Accessory disk with label printer, global editor, file merge, reblocker, and forms printer. Stoneware, 50 Belvedere St., San Rafael, CA 94901. $99.

Desktop Planner. Models and analyzes budgets, profits and losses, sales forecasts, cash flow; "what if?" calculations. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $250.

Dow Jones News and Quotes Reporter. With modem, checks latest financial news and stock quotes for more than 6,000 securities from local Dow Jones data bank. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $95. *2/82.*

Executive Briefing System. Nifty business graphics package for preparing color slides, graphs, and charts. Lotus, 55 Wheeler St., Cambridge, MA 02138. $199.

1st Class Mail. Schoenburg, Pollack. Fantastically user-friendly program for specialized database applications. Twelve fields, ability to sort and filter on any field or combination. Continental, 11223 S. Hindry Ave., Los Angeles, CA 90045. $74.95 *6/82.*

FMS 80. Rodman. All-purpose CP/M file management system. Customizable, with extended file management language tutorial for programmers. DJR, 303 S. Broadway, Tarrytown, NY 10591. $990.

General Manager. Database program that allows economic projections, search and select options, and screen formatting for data entry. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $99.95.

Information Master. Database management program that can keep records sorted in five separate orders simultaneously. High Technology, 2201 N.E. 63rd, Oklahoma City, OK 73113. $150.

Infotory. Complete purchase order and inventory system for under 9,999 items of one type. Prints receiving, sales, purchase orders; audit trails available. SSR, 1600 Lyell Ave., Rochester, NY 14606. $295.

List Handler. List-lover's delight. Prints lists, labels, and letters. Handles up to 3,000 records per disk and eight disk drives. Takes requests. Silicon Valley Systems, 1625 El Camino Real, Ste. 4, Belmont, CA 94002. $89.95.

Multiplan. Easy-to-learn electronic work sheet using plain English commands. Powerful modeling and presentation capabilities. For use in analysis, forecasting, technical engineering, and the home. Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $275.

Payroll. Complete payroll system in run-time Pascal. Handles tax for all fifty states, fifteen divisions with up to 300 employees, and any mix of thirty deduction types. Broderbund, 1938 4th St., San Rafael, CA 94901. $395.

Personal Filing System. Page. User controls data in totally unstructured database. Up to thirty-two pages (screens) of information in each record. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $95. *10/80.*

PFS:File. Page, Roberts. User controls data in totally unstructured database. Up to thirty-two pages (screens) of information in each record. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $125; Apple III: $175.

PFS:Graph. Chin, Hill. Works alone or interfaces with files created with *PFS:File* and *VisiCalc*. Produces bar, line, and pie charts merging data from several sources. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $125; Apple III: $175. 5/82.

PFS:Report. Page. Powerful report generator designed for use with *PFS:File*. Sorts, calculates, totals, formats, and prints presentation-quality columnar reports. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $95; Apple III: $125. *6/81.*

State of the Art General Ledger and **Budget and Forecasting Module.** The ledger does twelve-period accounting, two-digit subaccounts; handles up to 470 accounts; enter 100 transactions before updating to permanent files. Budget module extends the account number to nine digits; custom designs reports; does previous year comparisons and two, four, twelve, and thirteen period accounting. State of the Art, 3183A Airway Ave., Costa Mesa, CA 92626. $495; budget module: $395.

VC-Expand/80. Get 80-column *VisiCalc* display; works with Videx 80-column card; expands memory to 128K. Upgrade from previous *VC-Expand* for only $25. Saturn, Box 8050, Ann Arbor, MI 48107. $125.

VersaForm. Business forms generator for invoicing, mailing lists, sales analysis, inventory. Hard disk compatible. Applied Software Technology, 14125 Capri Dr., Los Gatos, CA 95030. $389. *6/82.*

Videx Preboot Apple Writer. Eighty-column display for *AppleWriter II* with upper and lower case input from keyboard Enhancer II and Videoterm compatibility. Videx, 897 N.W. Grant St., Corvallis, OR 97330. $19.

Videx Preboot VisiCalc. Run *VisiCalc* in 80-columns with upper and lower case; see complex formulas in their entirety. Advanced version uses mixture of existing memory cards. Videx, 897 N.W. Grant St., Corvallis, OR 97330. $49. Advanced $89.

• **VisiCalc.** Bricklin, Frankston. Electronic worksheet for any problem involving numbers, rows, and columns. No programming necessary. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $250. *10/80.*

VisiCalc Business Forecasting Model. Seven interrelated Visi templates that provide you with financial information most vital to analysis and planning: income statements, balance sheets, statements of cash flow, and so on. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $100.

VisiFile. Creative Computer, Jameson, Herman. Database management system for organization and retrieval of information, allowing sort and modification of records. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $250.

VisiSchedule. Critical path PERT schedule planner. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $300.

VisiTrend/VisiPlot. Kapor. Combines *VisiPlot* graphics with time-series manipulation, trend forecasting, and descriptive statistics. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $259.95. *7/81.*

Wall Streeter. Collection of stock analysis and management programs that track price, Dow Jones, indices, and advances and declines. Calculates and charts same. Micro Lab, 2310 Skokie Valley Rd., Highland Park, IL 60035. $300.

Communications

ASCII Express II. Blue. Modem software provides automatic redial, individual macro files, and improved file transfer capabilities. Sends any DOS file; uploads one character or one line at a time. Included utilities convert Integer Basic, Applesoft, or binary programs into text files. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $79.95. *9/81.*

ASCII Express: The Professional. Greatly improved version of the original. Supports multiplicity of hardware and prints simultaneously. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $129.95.

Data Capture 4.0. Copyable, modifiable smart terminal program; compatible with Apple III and most lower-case adapters. Southeastern Software, 6414 Derbyshire Dr., New Orleans, LA 70126. $65.

Hello Central! Menu-driven modem software. Upload-download, send-capture, save, retrieve, edit and manipulate files and programs. Advanced Operating Systems, 4300 W. 62nd St., Indianapolis, IN 46206. $99.

Micro/Courier. Electronic mail program. Provides file transfer of any DOS 3.3 file (correspondence, *VisiCalc*, charts) automatically and unattended. Built-in text editor; maintains 100 mailboxes; permits optional clock and calendar scheduling. Microcom, 1400A Providence Hwy., Norwood, MA 02062. $250.

Micro/Terminal. Access and exchange information with mainframes and minis, databases like the Source, and other remote terminals and personal computers. Supports keyboard mapping, upper and lower case, and 80-column cards. Microcom, 1400A Providence Hwy., Norwood, MA 02062. $84.95.

P-Term: The Professional. Supports all Pascal-compatible interfaces, asynchronous serial cards, Apple-compatible modems, and baud rates up to 2,400. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $129.95.

Super Smart. Terminal emulation package to capture, create, edit, print, and save data. Utilizes full capabilities of Hayes Micromodem II; supports full ASCII. Softspoken, Box 7000-863, Redondo Beach, CA 90277. $60.

Transend I, II, and III. Intelligent terminal software with multiple hardware compatibility. Advanced, easy to use. The *I* sends text only; menu driven, limited editor. The *II* sends text and files like *VisiCalc*; verifies transmission. The *III* does both and handles electronic mail with auto-redial, clock calendar, and password protection. Upgrade for only $20; all three get an A+ for error handling. SSM, 2190 Paragon Dr., San Jose, CA 95131. $89, $149, $275. *9/82.*

VisiTerm. Well-planned, comprehensive. Hi-res 60-character display; wide range of protocols for sending text. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $129. *9/81.*

Z-Term. Blue. Flexible, customizable communications software written specifically for the CP/M Apple. A quality package. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $99.95. *5/81.*

Z-Term: The Professional. More than an update. Compatible with a great variety of modems, interface cards, and screen modes. Simple file transfer with integrity. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $149.95.

Fantasy

Apventure to Atlantis. Clardy. The sequel and worthy successor to *Odyssey*. Many refinements including recruitable entourage of wizards with individual attributes. Included cheat sheet is invaluable. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $40. *6/82.*

Aztec. Stephenson. Graphic adventure with animation on-screen throughout. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $39.95.

• **Beneath Apple Manor.** Worth. The original dungeon game for the Apple, created in 1978. Even in lo-res, it still stands up. Quality, 6660 Reseda Blvd., Ste. 105, Reseda, CA 91335. $19.95.

Castle Wolfenstein. Warner. First game to fuse successfully best elements of home-arcade and adventure. Escape from Nazi stronghold, finding and taking secret plans. Room layout changes with each new game. Enemy speaks, in German. Muse, 347 N. Charles St., Baltimore, MD 21201. $29.95. *10/81.*

Knight of Diamonds. Second scenario of *Wizardry*, requiring thirteenth-level characters from the original. Individual quests on each of six dungeon levels. Great. Sir-tech, 6 Main St., Ogdensburg, NY 13669. $34.95. *7/82.*

• **Oddyssey: The Compleat Apventure.** Clardy. Fantasy adventure far beyond one place and one setting. Castles, catacombs, an ocean voyage, and the orb of power. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $30. *10/80.*

Snooper Troops. Snyder. Ongoing hi-res mystery series in the form of educational games. Highly structured; excellent fourth-grade through eighth-grade educational tool, and great fun for adults. Spinnaker Software, 215 First St., Cambridge, MA 02142. $44.95. *9/82.*

Taipan! Canfil. Roam the China Seas as an opium smuggler in this exotic fantasy with a challenging hi-res pirate sequence. Avalanche, 2460 Embarcadero Way, Palo Alto, CA 94303. $39.95. *10/82.*

• **Temple of Apshai.** Lead title in Dunjonquest se-

ries, winner 1981 Academy of Adventure Gaming Arts and Design "Computer Game of the Year" award. Epyx/Automated Simulations, 1043 Kiel Ct., Sunnyvale, CA 94086. $39.95.

Ultima. British. Hi-res color adventure, progressing from Middle Ages to beyond the space age. A masterpiece. California Pacific, 1623 5th St., Davis, CA 95616. $39.95. *6/81.*

Ultima II. British. Faster play in a bigger universe with a time-travel option. Typically British look and feel. Events are much more interdependent; larger realm of fantasy with more transactions available. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $59.95.

Upper Reaches of Apshai. The next four levels (and requires) *Temple of Apshai.* Discover the secret of the monastery, battle giant tomatoes and killer chickens. Epyx/Automated Simulations, 1043 Kiel Ct., Sunnyvale, CA 94086. $19.95.

• **Wilderness Campaign.** Clardy. First fantasy game to leave the dungeon for the great outdoors, first in hi-res, first to bargain with merchants, and more. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $17.50.

Wizardry. Greenberg, Woodhead. Ultimate role-playing fantasy; ten-level maze in hi-res. Generate twenty characters, six at a time on expeditions. Gripping game, superbly produced. Sir-tech, 6 Main St., Ogdensburg, NY 13669. $49.95. *8/81.*

Graphics

Alpha Plot. Kersey, Cassidy. Hi-res graphics and text utility with optional xdraw cursor and proportional spacing. Beagle Bros, 4315 Sierra Vista, San Diego, CA 92103. $39.50.

Apple World. Projects and rotates 3-D color images on screen in true perspective, drawing up to 65,000 points per side. Includes screen-oriented text editor for image formation. United Software of America, 750 3rd Ave., New York, NY 10017. $59.95.

The Arcade Machine. Jochumson, Carlston. Step-by-step arcade game designer—shapes, scoring, sound, and titles. Begin with variations on five games included, then on to your own. Broderbund, 1938 4th St., San Rafael, CA 94901. $59.95. *11/82.*

The Complete Graphics System II. Pelczarski. A wealth of graphics tools at a reasonable price. Make 2-D drawings with game paddles, add text in destructive, nondestructive, or reverse modes, create 3-D figures with a panel module and shape tables with a shape module. Manual features complete outline of command structure. Penguin, 830 4th Ave., Geneva, IL 60134. $69.95; Apple Graphics Tablet version, $119.95. *7/81.*

GPS. Versatile graphics program. Creates, manipulates, and edits images like a word processor. Easy to use; in standard and professional formats. Stoneware, 50 Belvedere St., San Rafael, CA 94901. $59.95, $99.99.

GraForth. Lutus. A graphics language rewritten for maximum speed. Plotting, line, text display, character image, and high speed 3-D graphics, with variety of colors and drawing options. Includes music synthesizer. Insoft, 10175 S.W. Barbur Blvd., Ste 202-B, Portland, OR 97219. $75. *8/82.*

Graphics A2-3D1. High-speed 3-D animation package to guide beginner through scene creation, storage, retrieval, movement, and advanced applications. SubLogic, 713 Edgebrook Dr., Champaign, IL 61820. $59.95.

The Graphics Magician. Jochumson, Lubar, Pelczarski. Outstanding animation package consisting of a picture editor and shape table extender designed to allow programmers to design and store graphics files. Comes with utility program to transfer binary files. Penguin, 830 4th Ave., Geneva, IL 60134. $59.95; Apple Graphics Tablet version, $69.95. *5/82.*

LPS II. Superb hi-res graphics drawing system with light pen. Draw freehand or use circles and lines to create geometric shapes. Fill routine with colors and patterns; fun animation demo; programmable Pentrak driver. Gibson, 23192-D Verdugo Dr., Laguna Hills, CA 92653. $349. *10/82.*

Painter Power. Create hi-res graphics and arrange them in a slide show. Shape, alter, rework, and print hard-copy results. Micro Lab, 2310 Skokie Valley Rd., Highland Park, IL 60035. $39.95.

Special Effects. Pelczarski. Artist's graphic package for creating and enhancing computer graphics. With 108 colors and 96 brushes, magnification and editing point-by-point. Reverse colors, create mirror images, move images around. Penguin, 830 4th Ave., Geneva, IL 60134. $39.95.

Zoom Grafix. Holle. Graphics printing utility allows display of picture on screen prior to print; prints out selected portion at any size. Phoenix, 64 Lake Zurich Dr., Lake Zurich, IL 60047. $39.95. *2/82.*

Home-Arcade

ABM. Atomic war high jinks. Defend the East Coast from Russian nuke attack. Incoming warheads can do splits. Muse, 347 N. Charles St., Baltimore, MD 21201. $25.

• **Alien Rain (Apple Galaxian).** Suzuki. Monsters in this home-arcade classic seem to take it personally when you gun down one of their kind. Broderbund, 1938 4th St., San Rafael, CA 94901. $24.95. *7/81.*

Apple Panic. Serki. Rid a five-story building of crawling Apples and butterflies by running up and down connecting ladders, digging traps in floors, then covering critters over before they devour you. Extremely addictive, excellent hi-res play. Broderbund, 1938 4th St., San Rafael, CA 94901. $29.95. *9/81.*

Bandits. Ngo. Fight off waves of multiple menaces intent on killing you and stealing your supplies. Delirious nonstop action, animated to the hilt. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $34.95. *7/82.*

Beer Run. Turnell. Artesian's delight. Catch falling cans on your way up one building, hop the blimp, and work your way down another. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *1/82.*

Bell Hop. Kitchen, Van Ryzin. Pick up luggage, catch elevators, and get tips for speedy service. Hayden, 600 Suffolk St., Lowell, MA 01854. $34.95.

Bug Attack. Nitchals. Sing along with dagger-wielding ants, blue worms, swarming medflies, a millipede, the 1812 Overture, lots of bright colors, terrific hi-res animation, and bouncy style. Cavalier, Box 2032, Del Mar, CA 92014. $29.95. *11/81.*

Bug Battle. Garden-variety shoot-'em-up that requires careful weeding and no fear of spiders. United Software of America, 750 3rd Ave., New York, NY 10017. $22.50.

Cannonball Blitz. Lubeck. In the cold light of *dawn,* you must find the *key* to victory, no matter how incongruous. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *7/82.*

Canyon Climber. Scale the levels and ladders while avoiding arrows, gorges, and hi-res sheep (no cows). Score by setting explosive charges. Datasoft, 19519 Business Center Dr., Northridge, CA 91324. $29.95.

Choplifter. Gorlin. Fly your chopper into the Bungeling Empire to rescue the sixty-four hostages, avoiding interceptor jets, homing mines, and tanks. Challenging, realistic, and playful. Broderbund, 1938 4th St., San Rafael, CA 94901. $34.95. *7/82.*

County Fair. Illowsky. Shooting gallery with hungry ducks and multiplying rabbits. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95.

Crazy Mazey. Skill and strategy needed in this auto-chase maze game. Not flashy but enduring fun; jazzy sound. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95. *10/82.*

Crisis Mountain. Schroeder. Run, crawl, walk, and leap through mountain maze fraught with rolling rocks, geysers, and chasms; collect nuclear devices. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $34.95. *10/82.*

Crossfire. Sullivan. Aliens come at you from three directions on a grid laid out like city blocks. Each alien has four lives and metamorphoses into its next one when shot. Strategy and intense concentration required. Superb, smooth animation of a dozen pieces simultaneously. One of the great ones. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $29.95. *1/82.*

David's Midnight Magic. Snider. Pinball challenger to *Raster Blaster.* Excellent hi-res graphics and animation. Provision for earning extra balls. Broderbund, 1938 4th St., San Rafael, CA 94901. $34.95. *2/82.*

The Eliminator. Anderson. Pit your hi-res space fighter against numerous adversaries. Plenty of action. Adventure Intl., Box 3435, Longwood, FL 32750. $29.95. *7/82.*

• **Epoch.** Miller. Superbly stylized animation enhances this filmic shoot-'em-up. Tremendous sense of being in space; neat classical music and dramatic time warp sequence. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $34.95. *10/81.*

Falcons. Varsanyi, Ball. A hypnotically good shoot-'em-up with several levels of difficulty. Piccadilly, 89 Summit Ave., Summit, NJ 07901. $29.95. *10/81.*

Fly Wars. Trap fly fighters in your web, score with exploding cocoons. Beware the beetle and bug spray. Simple, addicting. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *7/82.*

Free Fall. Get down, get down! Float through a shower of needles, guns, and bombs on your way to the safety of the holes in the ground. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *12/82.*

Here's to you!

1982
The Year of
Screenwriter II.

You did it! You - our dealers and customers - made Screenwriter II the No. 1 word processor of the year. For that we thank you! And rest assured, Sierra On-Line is continuing to create business products tailored to your needs. Just wait until . . .

1983
The Year of
The General Manager

No personal filing system does more than The General Manager 2.0, and we'll be taking steps in 1983 to show you how easy it is to use. We made The General Manager flexible and powerful. You'll make it the No. 1 information manager of 1983.

Thanks for your continued support and have a happy 1983.

KEN WILLIAMS
Chairman of the Board
Sierra On-Line, Inc.

Sierra ON-LINE INC

Gold Rush. Berlyn, Wilker. Transport the gold from the train through the forest to waiting hoppers, avoiding bears, Indians, bandits, and random troublemakers. Sentient, Box 4929, Aspen, CO 81612. $34.95. *8/82.*

Gorgon. Nasir. Fly over planet shooting and dodging invaders and saving kidnapped inhabitants. Outstanding hi-res graphics, challenging refueling sequence—if you can get that far. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $39.95. *8/81.*

Guardian. Tom & Jerry. Blast your way out of six maze levels surrounded by hostile aliens. Fast and tricky. Continental, 11223 S. Hindry Ave., Los Angeles, CA 90045. $29.95. *7/82.*

High Orbit. Merret. Use lasers to lift modules into place to construct floating space station. Watch out—it's a crowded cosmos. Gebelli, 1787 Tribute Rd., Ste. G, Sacramento, CA 95815. $29.95.

Horizon V. Nasir. Okas follow-up to Gorgon with superb animation, though not much challenge. Gebelli, 1787 Tribute Rd., Ste. G, Sacramento, CA 95815. $34.95.

Human Fly. Bagley. Good crude fun. Climb the C.P.U. building, avoiding apes, nasty birds, and slamming windows. Promises many excruciating falls. C.P.U., 9710 24th Ave. S.E., Everett, WA 98204. $29.95.

Laf Pak by Chuckles. Beuche. Four-game variety disk; a real bargain. *Creepy Corridors* (the best), *Apple Zap*, *Space Race*, and *Mine Sweep*. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *10/82.*

Lemmings. Thompson. Round up mass-reproducing rodents, detaining nonbreeding pairs, before they migrate into the sea. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *6/82.*

Marauder. Weigandt, Hammond. Double duty: bust through force field as a rocket, then switch to man in a maze. Nine mazes with fifteen levels of difficulty. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *9/82.*

•Meteoroids (Asteroids) in Space. Wallace. Making little asteroids out of big ones, plus occasional hostile alien ships. Hyperspace, autobrake, autofire. Quality Software, 6660 Reseda Blvd., Ste. 105, Reseda, CA 91335. $19.95.

Microsoft Decathlon. Smith. Ten standard decathlon events. Hi-res animated athletes, muscle-stirring music; you provide the sweat. Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $29.95. *6/81.*

Ming's Challenge. Dear Flash: Conquer the universe by defeating the deathships, daymen, and cyclops and skirting the black hole. Thanks, Dale. Parameters of the game can be altered. Micro Lab, 2310 Skokie Valley Rd., Highland Park, IL 60035. $34.95.

Money Munchers. Bishop. Yet another eat-'em-up. Scurf the dollars and avoid the title creatures. Completely random mazes. DataMost, 9748 Cozycroft Ave., Northridge, CA 91311. $29.95. *11/82.*

Nightmare Gallery. Aldrich, Clardy. High-moon shot-'em-up. Fast action with ghosts, mummies, and menacing rows of tombstones. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $34.95.

Peeping Tom. Why mom says don't open the window. Shoot-'em-up with shuttered view; at first see only the aliens' ammo. Micro Lab, 2310 Skokie Valley Rd., Highland Park, IL 60035. $34.95.

Pest Patrol. Allen. Where have all the flowers gone? Frenized new bug game with hopping spiders, killer butterflies, and shielding snails—all with dive-bombing capabilities. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $34.95. *10/82.*

Phaser Fire. Salt City. Space shoot-'em-up. Defend vortex from swooping rockets and space junk. Gebelli, 1787 Tribute Rd., Ste. G, Sacramento, CA 95815. $29.95.

Pinball A2-PB1: Night Mission. Artwick. Fantastically realistic and competitive ten-mode pinball simulation, allowing user modification and definition of virtually every aspect of play. SubLogic, 713 Edgebrook Dr., Champaign, IL 61820. $29.95. *5/82.*

•Pool 1.5. Hoffman, St. Germain, Morock. Makes most shots you could on a real table, with the advantages of instant replay and slow motion. Four different games. IDSI, Box 1658, Las Cruces, NM 88004. $34.95. *6/81.*

•Raster Blaster. Budge. Pinball game as good as real ones. *Softalk* readers' Most Popular Program of 1981. BudgeCo, 428 Pala Ave., Piedmont, CA 94611. $29.95. *5/81.*

Russki Duck. Knopp, Merrell. Recover stolen missile plans hidden in fake duck while dispatching enemy agents. Fairly easy. Gebelli, 1778 Tribute Rd., Ste. G, Sacramento, CA 95815. $34.95.

Seafox. A good sub-versus-convoy home-arcader. Variety of vessels, bouncing torpedoes, refueling dolphins (food not included), and intelligent depth charges. Broderbund, 1938 4th St., San Rafael, CA 94901. $29.95. *11/82.*

Serpentine. Hypnotic snake-chase maze game. Clean action, thrills, hairy escapes. Recommended. Broderbund, 1938 4th St., San Rafael, CA 94901. $34.95. *10/82.*

Snack Attack. Illowsky. A three-maze eat-'em-up; starts at any of five speed levels. Nonfattening. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95. *1/82.*

Snake Byte. Arcade action featuring fruit and serpents. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95.

• Sneakers. Turmell. Many-layered shoot-'em-up, one of the best. Stomping sneakers and swarm of other creatures add to the fun. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95. *9/81.*

Space Cadette. Illowsky. Firing, dodging, and running through a square grid maze, trailing shields. Enemy ships have distinct personalities. A little strategy and a lot of action. Keyboard only. Fun-tastic, 5-12 Wilde Ave., Drexel Hill, PA 19026. $34.95. *12/82.*

Spy's Demise. Be the first on your block to run a maze of pile-driving elevators. Fast, frustrating fun with a game show bonus; complete puzzle after all nine levels and win a prize. Penguin, 830 4th Ave., Geneva, IL 60134. $29.95. *11/82.*

Star Blaster. Mines, fireballs, space tunnels, general obstructions and unfriendlies waylay your starship. Piccadilly, 89 Summit Ave., Summit, NJ 07901. $29.95. *8/82.*

Star Blazer. Suzuki. Bomb-run game with five levels, minutely exact animation, and style to burn. A joy. Broderbund, 1938 4th St., San Rafael, CA 94901. $31.95. *4/82.*

Star Maze. Eastman. Hunting for power jewels through sixteen zero-gravity mazes filled with unfriendlies. Uses eighteen hi-res colors and multidirectional scrolling. Mapping advised. A standout. Sir-tech, 6 Main St., Ogdensburg, NY 13699. $34.95. *11/82.*

Succession. Eisnaugle. Real-time maze game. Get the creatures in numbered order as the chaser nips at your heels. Piccadilly, 89 Summit Ave., Summit, NJ 07901. $29.95. *10/82.*

• Super Invader. Hata. The daddy of home-arcades. Still good hi-res, still a challenge. *Softalk* readers' Most Popular Program of 1978—80. Astar Intl., through California Pacific, 1615 5th St., Davis, CA 95616, and Creative Computing, 39 E. Hanover Ave., Morris Plains, NJ 07960. $19.95.

Super Taxman 2. Fitzgerald. Pac up your troubles! Bigger, more complex version of the most perfect extant legal rendition of a certain arcade game. You can look at the cartoons whenever you want. H.A.L. Labs, 4074 Midland Rd., Ste. 23, Riverside, CA 92505. $25.

Swashbuckler. Stephenson. Hi-res swordfighting with realistic pirates, snakes, rats, and other scum. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $34.95. *8/82.*

Teleport. Abbot. Need a job? Learn to stun and bag aliens in your spare time. Maze game with lots of action. Cavalier, Box 2032, Del Mar, CA 92014. $29.95. *10/82.*

Tharolian Tunnels. Nelsen. Shoot-'em-up with several stages of play; on par with *Falcons*. Software Farm, 3901 S. Elkhart St., Aurora, CO 80014. $29.95. *10/82.*

Threshold. Schwader, Williams. Another shoot-'em-up. Hi-res graphics, animation, and accurate collisions. Targets include everything from flying maple trees to Volkswagen Bugs. Frustratingly small fuel supply. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $39.95. *12/81.*

Track Attack. Jochumson. Three-level train robbery chase game requiring considerable dexterity. Broderbund, 1938 4th St., San Rafael, CA 94901. $29.95. *4/82.*

Twerps. Thompson. Plot, elaborate animation, and sound link aspects of several different game styles together. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95.

Wayout. Exciting 3-D maze that moves in perspective as you play. Map displayed at all times. Lots of angles and Cleptangles. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $39.95. *10/82.*

Home/Hobby

Apple-Cillin. Hardware diagnostic tests for all RAM and ROM, plug-in cards, cp registers, disks; nine video test patterns. XPS, 323 York Rd., Carlisle, PA 17013. $49.95.

Apple Logo. Papert. Custom version (by its inventor) of MIT-developed turtle graphics language. First-rate educational tool with graphics, mathematical, even games use. Hefty documentation. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $175.

Apple Mechanic. Kersey. Multiple utility disk with

Format II.™ Word processing so advanced anyone can use it.

Format II was tested for six months in the demanding arena of Wall Street. Distributed by local dealers to law firms, financial institutions, and the like, Format II was used by secretaries and non-technical personnel. Overwhelmingly Format II was recognized as one of the easiest and most effective word processing systems available *in any form.*

Here's why Format II is unique among word processing programs:

What you see is what you get. Format II performs virtually any editing and formatting function you can imagine and displays on the screen the text exactly as it will print out—paragraphs, underlining, justification, page breaks.

Simple to Use. You edit and format text with single key strokes. "D" for delete, "E" for edit, "I" for insert, "J" for justify, etc. Easy-to-remember commands because they make sense.

It supports all printers compatible with the Apple. Shadow printing, boldfacing, proportional spacing, sub- and superscripting are all available.

Includes a sophisticated Mailing List. Stores and retrieves names and addresses which may be printed on labels or incorporated into documents. Powerful "logic" commands allow you to select only those records which match specified criteria.

And more. Format II is lightning fast and menu driven. Add them all together and it's clear why Format II has generated such enthusiasm. Now, at a cost of $250, you *too* can buy the best.

Format II is available from most local dealers. If not, ask them to contact us, or order directly and receive a 30 day money-back guarantee.

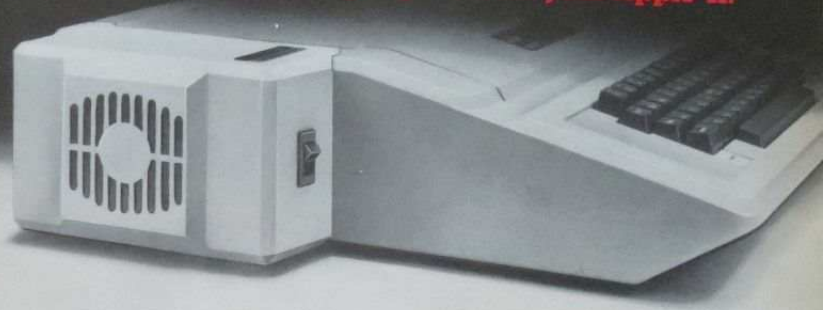
Kensington Microware Ltd., 300 East 54 St., Suite 3L, New York, NY 10022, (212) 486-2802.

KENSINGTON MICROWARE

System Requirements: 48K Apple® II Plus with 80 column card

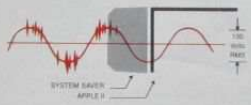
System Saver™

The most important peripheral for your Apple II.



For Line Surge Suppression

The SYSTEM SAVER provides essential protection to hardware and data from dangerous power surges and spikes.

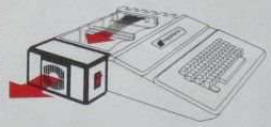


By connecting the Apple II power input through the SYSTEM SAVER, power is controlled in two ways: 1) Dangerous voltage spikes are clipped off at a safe 130 Volts RMS/175 Volts dc level. 2) High frequency noise is smoothed out before reaching the Apple II. A PI type filter attenuates common mode noise signals by a minimum of 30 dB from 600 khz to 20 mhz, with a maximum attenuation of 50 dB.

For Cooling

As soon as you move to 64K RAM or 80 columns on your Apple II you need SYSTEM SAVER.

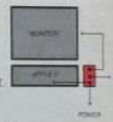
Today's advanced peripheral cards generate more heat. In addition, the cards block any natural air flow through the Apple II creating high temperature conditions that substantially reduce the life of the cards and the computer itself.



SYSTEM SAVER provides correct cooling. An efficient, quiet fan draws fresh air across the mother board, over the power supply and out the side ventilation slots.

For Operating Efficiency

SYSTEM SAVER contains two switched power outlets. As shown in the diagram, the SYSTEM SAVER efficiently organizes your system so that one convenient, front mounted power switch controls SYSTEM SAVER, Apple II, monitor and printer.



The heavy duty switch has a pilot light to alert when system is on. You'll never use the Apple power switch again!

Easy Installation

Just clips on. No mounting or hardware required. Color matched to Apple II.

Compatible with Apple Stand

$89.95 at your local dealer or order direct by phone or mail.

For phone or mail orders include $2.50 for handling. New York State residents add 6¼% sales tax.

VISA and MASTERCARD accepted.

Dealer inquiries invited.

Kensington Microware Ltd.
300 East 54 Street, Suite 3L
New York, NY 10022
(212) 486-2802

KENSINGTON MICROWARE

shape editor, custom typefonts, byte rewriter, and tricks to facilitate music, text, and hi-res generation. Beagle Bros, 4315 Sierra Vista, San Diego, CA 92103. $29.50. 9/82.

Apple Spice. Kosak, Fox. Powerful Applesoft expansion utility using & and *usr* functions. Easily incorporated programming routines. Adventure Intl., Box 3435, Longwood, FL 32750. $29.95. 5/82.

Audex. Collection of utilities to create, edit, and play back your own sounds for your own programs, in Basic and assembly language. Sirius, 10364 Rockingham Dr., Sacramento, CA 95827. $29.95.

Bag of Tricks. Worth, Lechner. Four utility programs for dumping and examining a raw track, sector editing, reformatting tracks, and repairing damaged disk catalogs. Quality Software, 6660 Reseda Blvd., Ste. 105, Reseda, CA 91335. $39.95.

Bug Byter. Screen-oriented mnemonic debugging tool with resident assembler and disassembler. Displays contents of accumulator and X and Y registers. Computer-Advanced Ideas, 1442A Walnut St., Ste. 431, Berkeley, CA 94709. $47.50.

Career Directions. Take a systematic approach to making career decisions. Professionally designed assessment analysis, planning, and exercises. Systems Design, 723 Kanawha Blvd., Ste. 403, Charleston, WV 25301. $59.95.

Chequemate. Home finance package that handles checks, charge cards, cash control, automatic tellers, and more. Reports to screen or printer. A bargain. Masterworks, 25834 Narbonne Ave., Lomita, CA 90717. $39.95.

Crossword Magic. Crosswood puzzle maker. Choose subject, words, and clues; program automatically connects words. Play on screen or make professional-quality printout. L & S Computerware, 1589 Fraser Dr., Sunnyvale, CA 94087. $49.95.

Datafax. Database utilizing unstructured keyword classification system for categorizing and cross-referencing by any method. No programming required; hard disk compatible. Link Systems, 1640 19th St., Santa Monica, CA 90404. $199.

Disk Library. Reads your catalogs and organizes your programs in a printed, sorted list. CP/M and Pascal files entered through keyboard. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $49.95.

DOS Boss. Kersey, Cassidy. Utility to change, shorten DOS commands; customize catalog. Good ideas and witty presentation. Beagle Bros, 4315 Sierra Vista, San Diego, CA 92103. $24. *10/81.*

The DOS Enhancer. New incarnation of the *Universal Boot Initializer*, featuring a speed increase of up to 500 percent and DOS user-definable and data disk init commands. S&H, 58 Van Orden Rd., Harrington Park, NJ 07640. $69.95.

DOS Tool Kit. Excellent utility package; Apple II assembler-editor system and Applesoft tool kit. Edit, assemble machine language programs; write, edit Basic programs. Simplifies graphics, includes character generator. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $75. *10/81.*

Earl's Word Power. Knudson. Educational software providing homonym training in a Shakespearean mode. Late elementary to early high school level. George Earl, 1302 S. General McMullen, San Antonio, TX 78237. $29.95.

Early Games for Young Children. Paulson. Basic training in numbers and letters and the Apple keyboard for children ages two to seven. Has a neat little drawing program. Learning Tools, Ste. 140, Hellard Plaza North, Minneapolis, MN 55426. $29.95. *11/82.*

Electric Duet. Lutus. Two-voice music without hardware. A bit involved, but superb sound quality. Insoft, 10175 S.W. Barbur Blvd., Ste. 202-B, Portland, OR 97219. $29.95. *7/12.*

Facemaker. Early learning introduction to the keyboard and programming, exercising a child's memory, concentration, and creativity. Choose features for a face and give it expression. Spinnaker Software, 215 First St., Cambridge, MA 02142. $34.95.

File Whiz. Goss. Quickly learned database management program with six command modes. Files generated are accessible from Basic programs. Fast, easy, and convenient for home use and users. Soft-House, Box 6383, Rochester, MN 55903. $79. *12/81.*

Financial Management System II. Home finance management; maintains multiple accounts, generates complete audit reports, and stores unlimited files. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $64.95. *5/81.*

Football Predictor 83. Gameforms. Inputs twenty-five sets of statistics on two teams, plays game ten times, predicts winners and point spreads. C&C Software, 316 Nancy Lynn Cr., Ste. 26B, Knoxville, TN 37919. $39.95.

Gertrude's Secrets. Gertrude the Goose teaches four-to-nine-year-olds shape and color relationships. Solve logic puzzles and create shapes. The Learning Co., 4370 Alpine Rd., Portola Valley, CA 94025. $75.

GPLE. Enhanced version of the *Program Line Editor*. Edit everything on a line, line by line, or on a range of lines; plus search for strings. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $60.

Home Accountant. Schoenburg. Thorough and powerful home finance program. Monitors five checking accounts against a common budget, plus credit cards and cash; one-step record of transfer of funds. Continental, 11223 S. Hindry Ave., Los Angeles, CA 90045. $74.95. *4/82.*

The Inspector. Sefton. Fast, flexible utility for examination of disk sectors, directory, and track-sector lists. Salvage blown disks, change data, delete DOS. Omega, 222 S. Riverside Plaza, Chicago, IL 60606. $49.95. *11/81.*

LISA 2.5. Hyde. Long-time popular assembler with extended mnemonics and more than thirty opcodes. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $79.95.

Master Diagnostics. Twenty-four tests that check your Apple for component malfunctions and suggest replacement. Software Source, 17905 Ventura Blvd., Encino, CA 91316. $69.95. *8/82.*

MasterType. Zweig. Learn to type by playing a game; simple and ingenious. Lightning, Box 11725, Palo Alto, CA 94306. $39.95. *4/81.*

Merlin. Does assembly language programming with a dozen editing commands and twenty-eight pseudo-ops. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $64.95.

Micro Cookbook. What's for dinner, Apple? Select recipes, create your own cookbook. Includes hundreds of international recipes, charts, tables, techniques, and a glossary. Virtual Combinatics, Box 775, Rockport, MA 01966. $30.

Mixed Numbers. Readiness skills, addition, subtraction, multiplication, and division of mixed numbers. Grades six through eight. Milton Bradley, 443 Shaker Rd., E. Longmeadow, MA 01028. $44.95.

Multi-Disk Catalog III. Very fast machine language database program for reading and storing file names, types, and sizes. Fast, powerful sort-and-search feature. Sensible, 6619 Perham Dr., W. Bloomfield, MI 48033. $25. *10/81.*

Personal Finance Manager. Gold, Software Dimensions. Handles up to 200 entries a month from maximum of 14 separate accounts. Search-sort-edit routine. Apple/Special Delivery, 20525 Mariani Ave., Cupertino, CA 95014. $75. *11/81.*

• Program Line Editor. Program development and modification program with more than eleven editing commands, listing control, lower case, and programmable cursor control. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $40.

Real Estate Analyzer. Make buy and sell decisions, compare investments, project future sales year-to-year for ten years. File, retrieve, and alter information itemized in tabular form. Howard Software, 8008 Girard Ave., Ste. 310, La Jolla, CA 92037.

$195. *7/81*

The Routine Machine. Meyer. Programming tool that extends Applesoft. Library of routines; install your own; no programming knowledge necessary. Southwestern Data, 10761-E Woodside Ave., Santee, CA 92071. $64.95. *12/82*

SAT English I. Designed to help high school students prepare for the college entrance exam. Covers the verbal half of the test; learn by your own mistakes. Micro Lab, 2310 Skokie Valley Rd., Highland Park, IL 60035. $30. *11/81*.

Statistics with Daisy. Statistics analyzer for business, science, and social use. Hypothesis testing, correlations, multiple regression, and variance analysis. Rainbow, 19517 Business Center Dr., Northridge, CA 91324. $79.95. *10/82*.

•Super Disk Copy III. Hartley. Easy-to-use menu-driven software library utility; transfers all types of DOS files. Sensible, 6619 Perham Dr., W. Bloomfield, MI 48033. $30. *10/81*.

SuperPilot. Courseware authoring language based on Apple and Common Pilot. Twenty-six instructions for creating individually tailored lessons. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $200.

TASC. Peak, Howard. Applesoft compiler. User controls locations of three memory compartments. Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $150. *9/81*.

Terrapin Logo. MIT. The Logo language, using a Terrapin turtle to teach state, control, and recursion. Terrapin Inc., 380C Green St., Cambridge, MA 02139. $149.95.

Typing Tutor. Ainsworth, Baker. Four levels of proficiency; individualized drills created with time response monitoring. Microsoft, 10700 Northup Wy., Bellevue, WA 98004. $24.95.

Utility City. Kersey. Twenty-one utilities on one disk. Beagle Bros, 4315 Sierra Vista, San Diego, CA 92103. $29.50.

VisiDex. Jennings. Electronic index and file-agenda program for spontaneous or structured information entry. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $199.95.

Strategy

Acey Deucey. Okay version of the betting card game. Computer seems to have all the odds. L & S Computerware, 1589 Fraser Dr., Sunnyvale, CA 94087. $29.95. *10/82*.

AirSim-1. Machine language flight simulator in 3-D with six landing fields and optional instrument flying mode. Mind Systems, Box 506, Northampton, MA 01061. $40.

Casino. Five hi-res games, Vegas-style: blackjack, baccarat, keno, poker, and roulette. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $39.95. *10/82*.

•Computer Baseball. Merro, Avery. Remarkable programming feat, simulating individual player abilities from the teams of thirteen famous World Series. Can enter and play teams of your own creation. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95. *9/81*.

Cosmic Balance. Design your own ships and create your own space fleet. Tactical space game that's fast and easy to play. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95. *11/82*.

Cytron Masters. Bunten. Fast-paced strategy in real time; plays much like football. Rates high in excitement and intelligence. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95. *10/82*.

•Flight Simulator. Artwick. Utilizes aerodynamic equations and airfoil characteristcs for realistic simulation of take-off, flight, and landing. SubLogic, 713 Edgebrook Dr., Champaign, IL 61820. $33.50.

Galactic Gladiators. Reamy. Easy and enjoyable tactical simulation. Suit up and arm your fighters for ten different scenarios. High speed for a strategy game. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95. *8/82*.

Gin Rummy. Carpet. Play against computer. Hi-res cards can change position in hand; your entire hand is visible. Space bar allows you to change your mind when discarding. DataMost, 9748 Cozycroft Ave., Chatsworth, CA 91311. $29.95. *6/82*.

Hi-Res Computer Golf. Aronoff. A masterpiece of skill testing, judgment, strategy, and visual acuity. One of the few computer sports simulations that itself requires athletic dexterity. Avant-Garde, Box 30160, Eugene, OR 97403. $29.95. *2/82*.

Ken Uston's Professional Blackjack. Uston. House rules and variations for seventy casinos. Basic to advanced strategies, practice drills; set your own playing parameters. Intelligent Statements, Box 600, Holmes, PA 19043. $89.95.

•Microgammon II. Competition program for learning, practice, and improvement of backgammon skills. Tournament play. Softape, 5547 Satsuma Ave., North Hollywood, CA 91601. $19.95. *2/81*.

Rendezvous. Huntress. Space shuttle simulation in 3-D, created by senior scientist at JPL. Orbit earth, match orbit, and dock with space station. Authentic, demanding. Edu-Ware, Box 22222, Agoura, CA 91301. $39.95. *7/82*.

RobotWar. Warner. Strategy game with battling robots is teaching device for programming. Muse, 347 N. Charles St., Baltimore, MD 21201. $39.95. *1/81*.

•Sargon II. Spracklen, Spracklen. Computer chess game with seven levels of play. Hayden, 50 Essex St., Rochelle Park, NJ 07662. $34.95.

S.E.U.I.S. Combines two game genres. Build the ships and create your fleet like in a strategy game; actual ship-to-ship combat is like a shoot-'em-up arcade game. Variety of scenarios. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95.

The Shattered Alliance. Suit up and arm unicorns, dwarfs, zorks, and centaurs to do battle in seven scenarios, both fantasy and historical. Also called *Chronicles of Ozgoth*. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $59.95. *7/81*.

Singles' Night at Molly's. Two basic hi-res solitaire games with variations. Set your own skill level and strategy; play alone or against others. Various scoring potentials (singles take note). Soft Images, 200 Route 17, Mahwah, NJ 07430. $29.95.

Space Pirates 3000. You're crafty and ruthless, which will you be—businessman or pirate? Purchase a space fleet, trade on various planets, or pirate other people's freighters. With space map and markers. C and C, 316 Nancy Lynn Cr., Ste. 26-B, Knoxville, TN 37919. $29.95.

Space Vikings. Robbins. Subtitled "A Graphic 3D Tactical Simulation of Space Combat in the Distant Future." Raid the planets of twenty star systems in your Imperial battlewagon, gathering loot and establishing bases. SubLogic, 713 Edgebrook Dr., Champaign, IL 61820. $49.95.

Spitfire Simulator. Air flight simulator—Spitfire in combat with German Aces—with 3-D scenery and moving target aircraft. Mind Systems, Box 506, Northampton, MA 01061. $40. *12/82*.

Warp Factor. Space war game featuring twelve starship designs representing five galactic empires, with possible scenarios ranging from skirmishes to galactic war. Extremely challenging. Strategic Simulations, 465 Fairchild Dr., Ste. 108, Mountain View, CA 94043. $39.95. *7/81*.

Word Processing

Apple Writer. The most popular word processing program in town. Type, erase, move words around, save and insert segments from disk, and print out.

Easy to use. Apple. 20525 Mariani Ave., Cupertino, CA 95014. $75.

Apple Writer II. Lutus, Finstead. Written in word-processing language. Additional editing features and functions menu; continuing features and functions menu; continuous readout of character count and length. Apple /Special Delivery, 20525 Mariani Ave., Cupertino, CA 95014. $150.

Apple Writer Extended Features. Malachowski, Cooper. Enables production of multiple copies of *Apple Writer* files and insertion of variables; converts Applesoft programs to *Apple Writer* and vice versa. Brilig Systems, 10270 Fern Pool Ct., Burke, VA 22015. $34.95. *7/81.*

Dictionary. Expandable 25,000-word spell checking program for *Supercribe, Screen Writer, Apple PIE,* and *Apple Writer.* Instant look-ups and corrections. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $99.95.

Easy Writer. Word processor; choose 40 or 80 column version. Information Unlimited, 2401 Marinship Wy., Sausalito, CA 94965. $99.95.

Executive Secretary. Editing, printing, and form letters, plus mail merge and electronic mail system. SofSys, 4306 Upton Ave. S., Minneapolis, MN 55410. $250.

Format II. Word processor with logic-sorting mailing list. Justifies type, wraps text; has one-key editing, menu prompting. Kensington Microware, 300 E. 54th St., Ste. 3L, New York, NY 10022. $375.

Goodspell. Dictionary companion disk to *Apple Writer* with 14,000 words. Flags words not listed when printing out. Apple /Special Delivery, 20525 Mariani Ave., Cupertino, CA 95014. $60.

Letter Perfect. Format-flexible word processor with ability to send control codes within body of program. Works with database files from *Data Perfect.* LJK, Box 10827, St. Louis, MO 63129. $149.95.

Magic Window. Word processing program simulates standard typewriter. 80-column text scrolls across 40-column screen. Three modes of disk file storage. Artsci, 5547 Satsuma Ave., North Hollywood, CA 91601. $99.95.

Magic Window II. Get 40, 70 (in hi-res), or 80 columns in this expanded version. Compatible with Pascal 80-column. With user-tailored, fast menu, underlining; global search and replace. Artsci, 5547 Satsuma Ave., North Hollywood, CA 91601. $149.95.

Perfect Writer. Powerful, easy-to-use word processor. Advanced document design features indents, subheads, footnotes, quotations. Requires Z-80 card and 80-column board. Perfect Software, 1400 Shattuck Ave., Berkeley, CA 94709. $389.

PIE Writer. Business processor that allows 9,999 pages. With word deletion, auto indent, spooling, and typeahead buffer. Hayden, 50 Essex St., Rochelle Park, NJ 07662. $149.95.

PowerText. Does memos, letters, reports, and manus scripts without formatting each time. Good balance of automatic and user-defined functions. Beaman Porter, Pleasant Ridge Rd., Harrison, NY 10528. $199.

ScreenWriter II. Kidwell, Schmoyer. Formerly *SuperScribe II.* No extra hardware for lower case, 70-column display, printer spooling. Edits Basic, text, and binary files; complete search and replace. Sierra On-Line, 36575 Mudge Ranch Rd., Coarsegold, CA 93614. $129.95.

Sensible Speller. Spell-checking program sports listable 85,000 words, extensible up to 110,000 words. Recognizes contractions, gives file word counts, incidence of a single word, and number of unique words. High marks for clear, logically organized documentation and simplicity of operation. Sensible, 6619 Perham Dr., W. Bloomfield, MI 48033. $125. *1/82.*

Super-Text 40/56/70. Zaron. Get 40, 56, or 70 columns without hardware. Design your own character sets. Basics of text editing. Character-oriented, floating-cursor edit with add, change, print, and preview modes. Muse, 347 N. Charles St., Baltimore, MD 21201. $125.

Word Handler II. Elekman. Wonderfully simple program with straightforward documentation. Allows folded paper printout for two-sided printing. Silicon Valley Systems, 1625 El Camino Real, Ste. 4, Belmont, CA 94002. $199. *11/82.*

WordStar. Screen-oriented, integrated word processing system in CP/M. Requires Z-80 card. MicroPro, 33 San Pablo Ave., San Rafael, CA 94903. $495.

Zardax. Philips. Highly recommended. Single program includes all standard word processing features with considerable extras including communication by modem. Computer Solutions, Box 397, Mount Gravatt, Queensland, Australia. In the U.S. Action-Research Northwest, 11442 Marine View Dr. S.W., Seattle, WA 98146. $295. *11/82.*

Apple III

Access III. Communications program for time sharing and standalone tasks; accesses remote information services, minis, and mainframes. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $150.

Apple Business Basic. High-level structured programming language for the III. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $125.

Apple III Business Graphics. Coverts numerical information into charts and graphs; only graphics program to take advantage of the III's capabilities. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $175.

Apple Writer III. Lutus. Uses WPL (word processing language) to automate the process of text ma-

nipulation and document creation. Adjusts print format during printing; translates from typewriter shorthand to English or other language and back again. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $225.

Basic Extension. Extension of Business Basic using disk, array, and utility routines as invokable modules. Machine language; nonmodifiable. Foxware, 165 W. Mead Ave., Salt Lake City, UT 84101. $95.

Data Manager III. Expansion of *Data Factory* allowing 32,000 records per file. Customizable screen display and printing. Micro Lab, 2310 Skokie Valley Rd., Highland Park, IL 60035. $750.

Data Reporter. Flexible database management system. Does form letters, patient files, labels, calculations, inventories, and employment records. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $220.

EASy. Executive accounting system with accounts receivable, accounts payable, and general ledger. Denver Software, 14100 E. Jewell Ave., Ste. 15, Aurora, CO 80012. $749.95.

Hardisk Accounting System. General ledger, accounts receivable, and accounts payable each handle up to 9,999 customers or accounts; inventory features five methods of evaluation. Also payroll, fixed-asset management, and mailing labels. Great Plains Software, 123 N. 15th St., Fargo, ND 58102. $395 to $595 per module.

Lexicheck. Spelling checker that runs from inside *Word Juggler*. Thirty thousand-word dictionary; add your own words. Eight thousand-word legal dictionary disk also available. Quark Engineering, 1433 Williams, Ste. 1102, Denver, CO 80218. $145.

Mail List Manger. Generates, stores, sorts, edits, and prints database files. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $150.

Micro/Terminal. Access any in-house or remote database; set up and log only once. Built-in editor or edit off-line. Microcom, 1400A Providence Hwy., Norwood, MA 02062. $99.95.

Pascal. Program preparer with editor, compiler, disassembler, linker, filer, and system library. Features cursor control; text modeling, and formatting. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $250.

Personal Filing System. Page. Form-oriented information management system allows storage and retrieval of up to 32,000 entries. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $145.

PFS:Graph. Chin, Hill. Works alone or interfaces with *PFS* databases and *VisiCalc* files. Produces bar, line, and pie charts merging data from several sources. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $125.

PFS:Report. Page. Generates reports; performs functions that require sorting, calculating, and manipulating data filed with *PFS*. Software Publishing, 1901 Landings Dr., Mountain View, CA 94043. $125.

Program Writer/Reporter. Basic database code generator that creates interactive, standalone programs. Vital Information, 7899 Mastin Dr., Overland Park, KS 66204. $200.

Quick File III. Personal index card or filing system. Fifteen fields; file as long as disk allows; can be put on ProFile. Apple, 20525 Mariani Ave., Cupertino, CA 95014. $100.

State of the Art General Ledger and Business Modules. Standalone interfaceable modules for twelve accounting periods. *General Ledger* handles 470 accounts and allows entry of 100 transactions before updating files. *Budget and Financial Reporting* module extends account number to nine digits, customdesigns reports, does previous-year comparisons, and handles two, four, twelve, and thirteen period accounting. *Accounts Receivable* prints monthly statements and allows for twenty-five customers and divisional reporting up to six divisions. *Accounts Payable* handles 325 vendors and prints monthly check disbursement register; eight different bank accounts. *Inventory Control* allows up to 500 items per disk; 5,000 total; performs physical inventory count sheet and variant report. State of the Art, 3183A Airway Ave., Costa Mesa, CA 92626. *General Ledger*, $595; modules, $495.

VersaForm. Landau. State-of-the-art business forms processor. Does invoicing, purchasing orders, mailing lists, client billing. Powerful, complex, worth getting to know. Hard disk compatible. Applied Software Technology, 14128 Capri Dr., Los Gatos, CA 95030. $495. *8/82.*

VisiCalc Advanced Version. Designed for corporate-wide modeling applications; allows managers to develop sophisticated templates that can be filled in by novice users. Numerous other new features including on-screen help, IRR and calendar functions, macro facility, variable column widths, locked cell values, and hidden cell contents. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $400.

VisiCalc III. Software Arts, Bricklin, Frankston. Just like it sounds; expanded memory, lower case, 80 columns. Four-way cursor movement. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $250.

VisiSchedule. Critical path PERT schedule planner. VisiCorp, 2895 Zanker Rd., San Jose, CA 95134. $300.

Word Juggler. Gill. Word processor makes use of upper and lower case keyboard, 80-column display, and expanded memory. Printout can be reviewed on screen prior to printing; multiple copies printed of selected pages. Quark Engineering, 1433 Williams, Ste. 1102, Denver, CO 80218. $295. *12/82.*

Word Weaver III. Fast word processor with unique feature: it can print out in shapes. Allows line spillover, does insert and delete, prints multiple files. Synergistic, 830 N. Riverside Dr., Ste. 201, Renton, WA 98055. $99.95.

OPEN DISCUSSION

Open Discussion gives you the chance to air your views and concerns, to seek answers to questions, to offer solutions or helpful suggestions, and to develop a rapport with other readers. It's what you make it, so share your thoughts, typed or printed, and double-spaced (please), in Softalk's Open Discussion, Box 60, North Hollywood, CA 91603. To ensure the inclusion of as many contributions as possible, letters may be condensed and edited.

Finding Our Way—with Your Help

I have watched with interest as your magazine has grown in length from a couple dozen pages to its current few hundred pages. You do a tremendous job, but I do have some questions and suggestions.

I appreciate the index of advertisers, but could you also provide an index of products? I'm sure this would add to your appeal to advertisers, and it would enable us faithful readers to find information about a product without time-consuming searches of our entire *Softalk* library. Most computer magazines contain cards in which the reader circles an advertisement number, mails it in, and some weeks later literature is sent by the given advertiser to the reader. Will *Softalk* consider including such a card in the future whereby readers can obtain more information about the products advertised?

Your Bestsellers list is great, but what rules do you follow when dividing games between adventure, strategy, and fantasy categories? Is it statistically possible for you to provide the dual rating services (a la Nielsen) and report percentage of sales and units sold separately? I like to get my statistics in absolute as well as relative terms. Finally, why, oh why, does *WordStar* consistently rank in the Top Thirty, and usually the top three, in word processors and yet never get more than a passing mention in the accompanying text? I would think *WordStar*'s incredible staying power would be a continuing wonder given that it costs three to seven times more than its competitors, counting required hardware.

How about a section for us modem users? Even though I'm a computer professional, I'm a novice in the area of data communications and have trouble even understanding the terms commonly used. I need help, and I turn to you as a friend who has helped so much in the past.

While you haven't had a lot of trouble in the past, I'd like to issue you a word of warning. Please be very careful when directly comparing products, especially in the form of tables and checklists. Most of those comparisons done in magazines and books are both inaccurate in fact and misleading in presentation. As just one example, both Apple and IBM publish comparison tables and each leads the reader to a totally different conclusion!

I'd love to read a column containing gossip about the industry. No, not Tradetalk, which is too personnel-oriented, but one that deals with new products, trends, and the like. An example is Dvorak's column in *InfoWorld*. Your first topic might be the anticipated Apple Super II and other new Apple computers.

The reviews are okay, but they would be better if a standard rating system was added. A grade for such aspects as performance, documentation, support, and so forth would be helpful. Also, what about a *Softalk* annual, which would contain the best of that year's articles, including all hardware and software reviews? I really would love nothing better than to see these changes in a 500-page *Softalk* that would be the industry standard for computer magazines.

Edward D. Isenberg, Newark, CA

Thanks for the neat letter.

The idea of having an index of products is being considered; a thorough index of products would entail a major staff effort. Know anyone who can reprogram the universe for a thirty-hour day? As for a reader service card, Softalk's research and empirical experience has shown us that advertisers aren't equipped to respond to such cards, and that many readers request information they have no use for. Everyone loses.

Here's how we categorize games: An adventure in the computer-gaming sense is a logic game in which the player must initiate commands to the game and solve puzzles in order to win. Fantasy role-playing games tend to entail the generation of characters that usually move through the game via single-key commands set forth in the instructions; there may or may not be specific puzzles to solve. Strategy encompasses all games in which specific rules must be brought to bear in planning moves in advance—whether on a World War II battlefield or on a chess board.

Clearly, some games cross lines. We then put them where they seem to fit best. Oddly enough, we've been mistaken from time to time. Not having seen Ali Baba, we were misinformed as to its *nature and carried it as an adventure when it should have been a fantasy.* Castle Wolfenstein *seemed to defy categorization. With the arrival of* Aztec*—also difficult to categorize, except it definitely belongs with* Wolfenstein*—we're moving out of strategy and into . . . home-arcade? Fantasy? Good grief!*

Regarding your desire for absolute sales figures, few companies would appreciate having this published, nor do we consider these relevant to the poll. If you know how much any one program on the Top Thirty sold, you can extrapolate any other by reference to the index numbers. The index also gives you a percentage-of-sales estimate.

What did you think of the telecommunications article in this issue by Dale Archibald? Response to it will influence how soon we institute a section devoted to modem use.

Glad you noticed the lack of direct product comparisons. We avoid tables and checklists like the plague. Readers are apt to skip an article when it's supposedly summed up in a table; they might read only the table, and tables alone don't do a thorough, fair job. As for gossip columns—rumors are just that, by nature often untrue, and sometimes damaging. We'll wait for the facts.

Finally, we feel a standard rating of programs falls right in there with tables and checklists. We will lead our readers to water, but they'll have to look to see it—we don't force them to drink.

Binding a Friendship

Since becoming a new Apple owner in the last six months I am very refreshed at all of the user friendliness I have come across in documentation, tutorials, and magazine articles. After putting up with directions that are ambiguous, vague, and demanding, it is a nice change indeed to see such a widespread loosening of these ties. Since I don't know who to thank directly, I am writing this to *Softalk* since many of those responsible will see it here.

Jim Murphy, Cresco, IA

A Paddle on the Back for Service

Sooner or later any Apple II user is going to be in the market for a new set of paddles. The originals supplied by Apple just can't handle heavy-duty workouts that three kids in an *Aliens Typhoon* tourney can generate. My local computer store suggested the TG Products paddles as being able to stand up to demanding use for some time.

I am quite happy with the performance of the TG paddles as far as their comfort and action in the games goes. This letter is not intended to praise the paddles but rather to comment on the excellent service of the TG Products people when I finally did come up with a problem that required repair. I had the paddles for over a year when we started to notice that one potentiometer seemed to fail. Since the paddles were out of warranty, I decided to call TG Products to get an estimate for repairing the set. I was given an address to send the paddles to and an estimate of less than $5. The paddles were mailed that day, and in less than a week they were delivered back to me. There was no charge for the repair.

I think that the prompt, courteous, competent, and conscientious service provided by this company deserves recognition and encouragement. I would have no hesitation to purchase from TG Products again.

Richard S. Jordan, Memphis, TN

Invariably Pleased

I recently purchased *Direct Mail* from Venture Software, which I had seen advertised in *Softalk*. Usually people take the time to complain about problems or their dissatisfaction with something, but I am so satisfied and impressed with the possibilities of this program that I wanted to pass along these thoughts to your readers.

Direct Mail, along with *Apple Writer*, allows you to use embedded variables extensively throughout a form letter, making it very personalized. I realize this isn't the only program available to insert variables into *Apple Writer*, but with the great help of Jim Mechachonis at Venture Software, I was able to use selected records from my existing popular database program, convert them into the *Direct Mail* format, and use them as the variables in the form letter. The results have been outstanding.

The converted variables were all upper case, but again with *Direct Mail* they were easily edited to first letter only with upper case or made fully lower case when desired. It also has the capability of creating master files if you have no formal database. The flexibility, depth, and possibilities with *Direct Mail* are yet to be realized as far as I'm concerned.

Ed Kennedy, Bethel, CT

Top Grade Support

I would like to take this opportunity to make my fellow readers aware of my experiences with High Technology Software in Oklahoma City, Oklahoma.

I purchased their *Information Master* database in December 1981. I intended to use it to store recipes and to keep a membership list for my church. I haven't figured out a good format yet for the recipes, but I'm sure there must be a way. As for the membership directory—it is excellent.

The manual states you can use one or two disk drives. This is true, but one disk drive is practically impossible to use with this program because of the amount of disk access and, therefore, swapping. I now have two disk drives and the program works excellently.

One drawback as far as I am concerned is that the program is copy protected. I would like to change a few items to suit me, mainly printing parameters.

A major consideration is technical support. Here High Technology gets an A-plus. I have called several times and always received prompt, courteous responses to my questions. I recently crashed all of my data disks. I called and then sent them in. My disks were not only fixed, but an extra copy was sent on a free disk!

I feel it is important to give praise or criticism where deserved. In this case, High Technology is to be commended for their excellent support of a good program. The program manual is extensive and well done, with many examples. The program is user-friendly and easy to use. I highly recommend both *Information Master* and High Technology.

Don Hulsey, Olathe, KS

Medical Assistance

This is basically a letter of thanks and appreciation to several companies, all of which have helped me extensively in setting up a program for my medical practice.

Sierra On-Line distributes *The General Manager*, an excellent database package that I used for the core of my program. I have recently acquired the new 2.0 version and I must say it really lives up to its expectations. The manual has been done over with more clarity.

Brillig Systems is the company where this program was originally designed. The authors provided much information that helped me modify the program, another nice feature about this database. You can have user programs and customize your use of the program.

Floppy Disk Services, in New Jersey, is the company that I bought a hard disk drive through when I rapidly exceeded the capacity of two floppies. QCS, also in New Jersey, is the manufacturer of the hard disk drive (ten megabyte, one-year warranty).

The whole system is now working great. More than 1,000 accounts are on it; access time to find a patient is about two seconds. It runs on an Apple II Plus with two floppies from Rana Drives and one printer. All this is thanks to the above companies and their excellent backup support.

Nicholas J. Spagnola, Red Lion, PA

An Interface of Responsibilities

Piracy has always been a heated topic of dialogue among readers of Softalk*, and opinions representing every side of the issue have been aired through Open Discussion. The following two letters are the culmination of such a dialogue.*

It all began back in May 1982 with two letters from J. Barry Smith. One of Smith's points was: "Piracy is a biased word; it implies theft. As they say, it all depends on which ox is being

gored. Mine is my wallet, so my policy on copying is this: If I can get it, I will." Furthermore, he stated: "Copy protection is an indefensible, self-defeating, technically impossible, unenforceable, greedy policy."

A letter of appreciation from Mark Pelczarski—for the positive user response to Penguin Software's decision not to copy protect its products (excluding games)—appeared in Open Discussion, July 1982. This letter contained a reaction to Smith's position. To wit: "Sorry. [Piracy] doesn't imply theft. It is theft. There are no two ways about it. When you take something that doesn't belong to you, it's got a simple one-word definition: stealing. . . . If that's the definition of a pirate, then, yes, you're a thief. And if you steal because you can't afford the software then you had no business buying a computer in the first place."

Mark Pelczarski wants readers to know that he is responding on principle, and that none of J. Barry Smith's comments were directed specifically toward Penguin Software. Now, let us pick up this dialogue where it left off. Here's Smith's response:

Thief? Thief? Who's calling who a thief? A paragraph in an Open Discussion letter starts out with my name, Barry, and later on states, "... yes, you're a thief." I say to the writer, noted publisher Mark Pelczarski of Penguin Software: Them thar's fightin' words where I come from, pardner. I trust your monitor was smiling when it displayed those words.

Let's settle this dispute the modern, 1982 way: Sue me. I'm serious; I'll accommodate you. Here's the deal. You make up some coded program and protect it by scrambling it somehow. Send off your ten bucks to the copyright office with your listings and then sell it to me for eleven dollars. In the presence of you, your lawyer, and a videotape machine, I will break the code, copy it, and give the copy to a friend of mine. You then prosecute me. Let's get this right-to-copy argument out into the open and into the courts.

You say I'm breaking the law and I say I'm not. Let's let the experts decide. Let's set a precedent!

A small item of interest: Did you know that the number-one Apple game program of 1981 was a computer version of the hardwired pinball game, *Firepower*? I believe not one penny in royalties was paid by the programmer or software publisher to Williams Company, the makers of the original game. Maybe we are all pirates trying to keep the bounty as long as we can.

Enough of legal arguments; how about a semantic one? For the record, artificial intelligence exists, has existed, and will exist. I suggest changing the name to reflect accurately what is going on. The results of any program written by humans shall be called distilled intelligence. The results of any program that the computer writes shall be called synthetic intelligence. *Distilled* and *synthetic* sound so much better than *artificial*, and they convey an accurate description of the vague noun *intelligence*.

Shall Franklin Ace owners be welcomed into the *Softalk* fraternity/sorority? I vote yes. The issue is software, not hardware. Maybe competition will get Apple to improve their product as technology improves. Does the Ace with Micro-Sci disk drives really run Apple software?

Two "must" movies for all of us to see are *Blade Runner* and *Tron*. These films ask some very profound questions in an incredibly visual manner.

I tried to run the program of my life the other day and got a "branch error at 29" message. It appears there is a bug in my life master program and I can't find it. It loads okay, runs sporadically, but won't list. Maybe somebody has it copy-protected.

You know, Mark, I was serious when I said "pardner," because we are partners in a sense. What programmers do is fantastic. It's too hard for me; I can't do it. You are entitled to all the cash you can get, at the point of sale. You discovered secrets and then sold them away. You create and sell, and I buy. Without each other we both are not as productive. Where does the money ultimately come from? (It does boil down to money, after all, doesn't it?)

It comes from me, the consumer. Without the consumer spending the cash there is no computer revolution, or at least it will be a very lonely one with the hardware, software, and retail people all standing around complaining about software piracy.

As a consumer, I accuse the hardware manufacturers of selling defective products. As a consumer, I accuse the software creators of selling programs that won't boot, won't run, and don't make sense when they do. As a consumer, I accuse the retailers of being shifty, incompetent, and rude. As a consumer, I accuse consumers of being the stupidest people in the world for being so exuberant in the dream that we put up with it all. We buy hundred-dollar programs without demanding that we be given a demonstration. We send off our money to places unseen for products unproven. We accept lies from salespeople. We make do with peripheral boards that marginally work. And we adapt to programs that are unwieldy and unfriendly. Evidently we believe promises because we want to believe.

J. Barry Smith, Barstow, CA

Mark Pelczarski's reply:

Barry: No, I don't want to fool around with taking you to court. That whole proposition is a little absurd. And, yes, there are already precedents upholding the copyright laws. The courts don't need to be reminded every six months. But mostly I don't want to bother because I not only do this for a living, I do what I do because I enjoy it. It annoys me when I have to take what could be productive time and deal with people who take advantage of what I do by ripping it off. The nice part is that most people aren't like that, and it is truly fun working and dealing with them. Yes, there will always be court cases for those who can't abide by common sense, but those people are in the minority it seems. Most people seem to be teaching their children well.

Your entire letter is put more into perspective by your last paragraph. I assert that there are incompetent people in any field. There are also those who genuinely care about their products and their customers. You are overlooking the fact that you, as a consumer, have a responsibility too. If you just go out and buy anything without looking, you are going to make some decent purchases but also some very foolish ones. As long as people keep buying from those who are incompetent, the incompetents will stay in business. It's your responsibility to yourself to care about what you buy.

Mark Pelczarski, Geneva, IL

Impression Correction

The review of *The Animator* that appeared in Marketalk Reviews (October 1982) did a fine job of explaining many of the features of the system. We would, however, like to correct a couple of misimpressions.

The Animator is designed to produce film strips that contain motion that is more complicated and detailed than what is usually seen on microcomputers. For example, the ballet demo utilizes a figure that has twelve independently moving body parts. The system is not designed to produce the simple, repetitive (but paddle/key controlled) motion found in computer games. *The Animator* will produce titles that move in interesting ways and can be incorporated into video productions. The documentation explains how users can add the animated strips to their own Applesoft programs.

In conclusion, the primary design goal is to provide a means for making intricate motion with as little effort on the user's part as possible.

Ray Balbes, BalbeSoftware Systems,
Saint Louis, MO

III O'Clock High

A thunderous round of applause for John Jeppson's article "It's III O'Clock and All's Well" (November 1982). As manufacturers of the Thunderclock Plus, Thunderware was pleased to see the subject of the Apple III clock covered so lucidly. We would like to take this opportunity to clarify a few points.

On the subject of timekeeping accuracy, the Thunderclock comes factory-adjusted to operate in the average Apple II internal temperature range. This results in typical accuracy of plus or minus a few seconds a week. The user can adjust the Thunderclock for unusually warm or cool environments using the simple calibration procedure in the manual. Temperature is the dominant factor affecting the accuracy of any quartz-crystal-controlled clock. The Thunderclock, the Apple III, and most quartz watches all use the same type of quartz crystal. Quartz wristwatches have the advantage of being maintained at a very nearly constant temperature, that of the wearer's wrist. As a consequence, they can be very accurate timekeepers.

The original SOS driver for the Thunder-

clock Plus was developed and made available by Apple Computer when they could not deliver Apple IIIs with the original system clock. We were pleased that Apple was recommending our product for use in the Apple III even though it was designed for the Apple II. This turned out to be a somewhat mixed blessing. Apple, in their haste to solve their problem, failed to give Thunderware an opportunity to review or comment on their SOS driver and the accompanying documentation. Both, it turned out, were woefully inadequate. Poor documentation, bugs, and, worst of all, the Apple-developed driver that was not designed to time and date stamp disk files plagued most Apple III owners using the clock.

Now, thanks to the efforts of John Jeppson, we have a much improved Thunderclock driver for the Apple III along with complete user documentation. Dr. Jeppson's new driver includes both corrections to the original driver and the update driver discussed in his article. The documentation contains complete instructions for installing and using the Thunderclock Plus with the Apple III.

Thomas W. Petrie, vice president,
Thunderware, Oakland, CA

Froggy Would a'Wooing Go

I find your review of *Frogger* (December 1982) very confusing and frustrating. Confusing because of its rabid inconsistencies, it being a negative review about a product you said many positive things about; frustrating because you have criticized our product simply because you feel it is "graphically no better than others like it on the market," which is akin to saying "She's as ugly as her sister because they're twins."

To expand on the inconsistencies of the review. . . . I contend, as an Apple owner and a software publisher, that a game product review should judge the gaming value of the product in question. (That is, is the game fun? Is it worth playing multiple times? It is easy to understand!) I feel that *Frogger* deserves a *yes!* answer on all those counts, and, from the wording of your review, you agree with me.

Why, then, is the overall tone of your review so extremely negative? Especially when lines like "the game is fun to play" and "*Frogger* is a good game" appear throughout. I have to question the reasoning for a negative review on any game that is "fun to play," "elegant in its simplicity," and "habit forming." (Your words, not mine—although I have to agree.)

The only "real" ground for criticism that you may have of *Frogger* is the graphics—but isn't that blame a little misplaced? As a magazine devoted to supporting the Apple and its owners, you more than anyone should know the extreme limits of the machine. The origin of the "lackluster graphics" you complain of stems from the very hardware that we all know and love—Apple II. That's right, the Apple.

Yet even with its lack of color scrolling abilities and its bothersome habit of making mud of any white object on a colored field, even with the multitude of other problems that Apple graphics present, we made a "fun to play," elegantly simple," and "habit forming" *Frogger* game, complete with blue ponds, colorful cars, female frogs, scrolling logs, and all of the other bells and whistles that people expect from *Frogger*. Granted, Apple *Frogger* isn't Atari *Frogger*—but then Apple hardware ain't Atari hardware either. We did the best implementation that we could with the resources we had available. We are proud of it, SEGA (the maker of the original *Frogger* for coin-ops) endorses it, and we think that Apple owners will enjoy it. We know that we do.

John Williams, Sierra On-Line, Coarsegold, CA

Here's Looking Through You

Since computers and programming are new to me, I find so many ideas in your magazine that it is hard to know what to try first. As a matter of fact, there seems to be more to your efforts than just publishing.

I have an interest in photography also. No doubt several others will ask you, but, in case they don't, I believe you owe us an explanation about the picture to the lower right on page 32 of the September 1982 issue.

Is Mark Overgaard still with us? What he "says" (Pascal) may not be so transparent to some of us, but it appears that he is. Was this a ghost of things to come in October? In case the photographer had a hand in the nondestructive graphics printover, I would like to know the technique or equipment used. That must have been one heck of a blinding flash, or was the photographer using x-ray sources? Well, back to the manuals, while I wait with interest to see your next revelation.

D. Hutton, La Mesa, CA

A Reset That Unsettles

How can I get my Apple to reboot when I type reset in the *Hello* program? If the answer has to do with machine language or anything other than Applesoft, please spell it out for me.

Mike Mahone, Boulder, CO

Smoldering in the Stacks

As a librarian with no more programming skills than understanding that a gosub takes the code somewhere else, I highly resent hot-shot comments like the ones that appeared from some readers in the October Open Discussion concerning Stone's *DB Master* and Sierra On-Line's *ScreenWriter II*.

The micro is the only computer that many small school, public, and special libraries will be able to afford for quite a long time. The value of the machine lies in the fact that available software can be put to use at minimal expense, with no programming expertise required on the part of the library staff. We don't need to change things if we can get them to work for us; in fact, changing code will screw us up a lot more than leaving things as they are.

As for *DB Master* being unfriendly, I admit I use it nearly every day and have yet to discover half of its power. We keep everything from film booking data to bibliographies on it. It crunches data every day and has never failed us in the creation of expansible files. In library work, you don't want something that's going to tell you that you're out of disk space.

ScreenWriter II is also above reproach. *SuperScribe II* wrote my 115-page master's thesis effortlessly. They provided the update in five days for fifteen dollars including a backup that I have never needed to use. At home I use *ScreenWriter II* with a single disk drive; at work, a second drive speeds things up only slightly. The RAM card version is exceptionally faster, but not that necessary. Every time I've called Sierra On-Line, they have solved every problem, save one. We are using *ScreenWriter II* to teach word processing to our students. We wanted fifteen copies of the manual, sparing us a complicated photocopy job, so that each student could have one for reference. They stated that their policy was to sell the entire package only. I can understand why they won't sell extra copies of the manual; with everybody ripping off software like mad, I would hold onto my manuals too. Tough Trotskies—it is still the best word processing package I have found.

For a comprehensive review of word processors, I might suggest the July/August issue of *Peelings II* to the person whose letter, entitled "Pass the Word," appeared in October Open Discussion. In this librarian's opinion, the best software reviews anywhere can be found in this magazine.

Libraries and librarians need to realize that the micro can go to work for them today. Pro-

"NOW WE HAVE IT!"

Introducing better company planning, company-wide, with VisiCalc Advanced Version.

More than 300,000 business professionals have discovered how much help our VisiCalc® program can be in planning, budgeting and forecasting. Now we can help businesses even more.

VisiCalc Advanced Version does it all

VisiCalc Advanced Version makes it easy to extend the problem-solving power, speed and accuracy of the "electronic worksheet" throughout an organization.

With this program, you'll be able to leverage the knowledge of your most knowledgeable people. Make more productive use of more of your staff. And do a better, faster, smarter job.

"I create a complex, yet easy-to-use worksheet..."

The best way to do what every company wants to do better

With VisiCalc Advanced Version, the people and departments in your firm will be able to quickly develop their plans, in a uniform format, for easier consolidation into your overall plan.

And everyone involved in your planning process will be able to make their most productive contribution to the effort.

Sophisticated planning made easy for everyone

Using VisiCalc Advanced Version, knowledgeable individuals within your company can create powerful, preformatted worksheets, or "templates," for nearly anything numerical.

For example, the knowledge of a corporate financial officer can be applied to create templates for complete and consistent financial reporting. Then others in your company without extensive financial or computer expertise can easily use these templates to provide the desired information in a uniform manner.

"Then, doing our department's budget is as easy as filling in the blanks!"

"The program makes it practically impossible to make mistakes!"

"I ask 'What if?' by changing any number... the new results are instantly calculated."

It's so easy. They simply "fill in the blanks" on the worksheet. In seconds, the program performs all the complex calculations built into the worksheet, and displays the answers to problems that could have taken hours to solve by hand.

Everyone can consider "What if...?"

The speed and versatility of VisiCalc Advanced Version make it easy to explore alternatives. Both template builders and worksheet users can change any number in a problem and instantly see the new results. Everyone will be able to analyze the impact of decisions before they make them.

"With just a few keystrokes I consolidate the worksheets..."

Mistakes locked out, help built in

With VisiCalc Advanced Version there's no worry about someone accidentally altering formulas, titles or any other important information. The program can protect all these parts of a worksheet. It can shield confidential data a user need not see and also guide a user step-by-step through a worksheet.

You can also insure that a user can't enter a label when a number is called for — and vice versa.

And if users ever need additional guidance in completing their worksheets, VisiCalc Advanced Version provides comprehensive on-screen help, at a touch of the "?" key.

Multi-sheet consolidation and finished reports, too

VisiCalc Advanced Version makes it easier to consolidate worksheet totals from various groups, departments or divisions into concise, presentation-quality reports.

"And produce presentation-quality reports for my boss."

See for yourself

Visit your nearest computer store for a demonstration of VisiCalc Advanced Version. Or write us at 2895 Zanker Road, San Jose, CA 95134 for a copy of our brochure.

Discover how easy it is to use electronic worksheets in your company for problem-solving, company-wide.

VISICORP™
PERSONAL SOFTWARE™

gramming skills and all the retraining involved in becoming programmers is unnecessary. We have good software available to us now. *DB Master* and *ScreenWriter II* can be counted in this score. For the critics who possess the skill to write better code, go ahead and do it. Then let me see if I can put it to use in my library.

As a postscript, I picked up a copy of *Diversi-DOS* in Rockford, Illinois, last week. (We have to go thirty miles to find a computer store.) I have always stayed away from utilities that change the norm, but this disk is great! We put it on our gradebook program and the staff can't believe how it speeds up the Applesoft code. Kent Welling, our student programmer responsible for revising the gradebook, was equally impressed. I recommend it highly.
Eric S. Anderson, Dakota, IL

Of Rumors and Hints

I was disappointed to hear about Reed Hubbard's rating of Sierra On-Line's *Cranston Manor* (April Open Discussion). My mother and I have completed almost all of Sierra On-Line's adventures with the exception of *Time Zone*, which we are working on now. None of their adventures has disappointed us. That includes *Cranston Manor*. By the way, Reed, the ads state that the ghost of old man Cranston living in his house is a rumor, meaning that it may or may not be true.

I must comment on Sierra On-Line's customer support staff. Howard Luthy, the official hint-giver for the series, provided prompt and efficient help when we needed it. When we played *Kabul Spy*, Sirius Software's staff and even Tim Wilson, the author, helped us. I believe a company should attempt to give such assistance to the consumer when needed.

I am pleased that some companies are converting to nonprotected software. Penguin Software's *Graphics Magician* has helped me with animation and picture drawing and has allowed me to customize the software to my liking. It seems that after a few months Penguin Software is still going strong. Maybe other publishers will borrow the same idea and then truly user-friendly programs will evolve.
Jeff Marraccini, Clarkston, MI

Feedback from the Outback

I noted a letter from Norman J. Wood in Open Discussion (July 1982) describing some failings in the *Home Accountant* that are similar to ones I've also found. I wrote to Continental Software with my questions and got a prompt reply, but the answers were not helpful. I find it most annoying that entries cannot be made to months other than those shown on the screen. When I wanted to go back a few months to the beginning of the financial year, I had to start again and then couldn't get to the budget categories. I tried some text editing and bombed the disk, which isn't write-protected. Luckily I was able to restore an empty text file. Users of the program beware.

One good thing about the program is that the date appears the correct way: first the day, then the month and year. On the other hand, there are limitations to what it allows for in mailing addresses. The state segment is set up for only two characters, but the zip code requires five numbers. Don't they know that the English-speaking world extends beyond the United States?

Readers may be interested that *Home Accountant* works with *Fast DOS*, available from Wytand P/L. This speeds up disk access three to five times, and it's proven to be one of my best investments.
P. N. Benkendorff, Kenthurst, Australia

One Picture Is Worth . . .

We are now moving into a time when more and more families are investing in Apples for home use. Then, the question asked by puzzled parents is, "How do I get my child to do something a little more studious than playing games?" Before buying a personal computer parents usually ask about the academic benefits for their children, only to see their dreams dissolve into the home video arcade.

As director of the Kids Computer Connection I have found the use of graphics to be a very motivating means of introducing children to the world of programming. Graphics are colorful, they allow for self expression and creativity, and they can end up looking like the types of things seen in video games. Do not underestimate the power of graphics. In order to write a workable program, children must understand the whole concept of a program, plot their picture on graph paper, plan ahead, and use a great deal of logical thinking. This easily leads to for-next loops, let, and goto statements.

If the number of students that relinquish after-school time and Saturdays to program is any indication of their willingness to learn graphics on the Apple, then graphics is truly a motivating force for beginning programmers.
Jim Tartaglia, Barre, MA

Innocence Lost

I want to comment on the state of pedagogy for computer innocents who are trying to learn assembly language. I'm one of them, and I'm ready to kill.

First I ordered a copy of Randy Hyde's best seller, *Using 6502 Assembly Language*. Shortly after concluding that the author ought to return to grammar school for a refresher course in English spelling and punctuation, I discovered that I had to purchase an assembler program (I told you I was innocent, didn't I?), and specifically Mr. Hyde's *LISA 2.5* program, to get any benefit out of the book. So I did.

At the same time I ordered a copy of Roger Wagner's *Assembly Lines: The Book*. When it arrived, I determined that Mr. Wagner is at least literate, and that he endorses a variety of commercial assembler programs by name. *LISA 2.5*, however, is not among them despite its presence in your Fastalk listings as a long-time popular assembler. This was not reassuring.

Now I had two texts on assembly language and the manual that came with *LISA*. I went to work, or tried to.

Hyde's text begins with a stultifying discussion of binary and hexadecimal operations. It stopped me cold. This material properly belongs in an appendix, not as the first step to learning assembly programming. I may learn it eventually, or I may simply buy a hand-held converting calculator. Beyond the false start, the innumerable typos, the aforementioned grammar, and the virtual necessity for buying *LISA*, Hyde's book has one fatal defect: the index. The index consists of an alphabetized list of the subchapter headings. That is all it is.

Even at that, it's better than the index to the manual that accompanies *LISA* itself. In this case they simply didn't bother. There's no index at all. On all other counts, as a learning device the *LISA* manual is quite hopeless. Nonetheless, I managed to get *LISA* up and running, and entered and assembled a sample program from Roger Wagner's text. Fine. Then I wanted to save it on another disk and try running it from within a Basic program. It was eight o'clock on a Sunday evening.

By four o'clock on Monday morning I threw in the towel. I had pored over Wagner's text, Hyde's text, the *LISA* manual, and the DOS manual. I had tried every stunt I could think of. Nothing worked. All I got were error messages that the *LISA* manual did not document, or Apple error messages that made no sense. The situation was beyond belief. Nowhere in all this literature was there a simple discussion of how to save and run an assembled program!

I finally got my answer by contacting Sierra On-Line, publishers of *LISA*, later on Monday. The answer was not easy. The answer was not obvious. And the answer was not, in fact, in their manual. Nor was it in Hyde's text. Nor was it in Wagner's text.

None of my friends or acquaintances are computer buffs. I am forced to rely entirely on the printed word, personal experimentation, and—as a last resort, lest I make a nuisance of myself—telephone calls to the software companies themselves. Given this state of affairs, I wonder how any amateur ever manages to get past first base with assembly language.
John W. Field, Fairfax, VA

Viva Pascal!

This is in response to Chuck Walker, whose letter describing Pascal as an abortion appeared in Open Discussion (November 1982).

You admit to being only slightly acquainted with Pascal, then proceed to condemn it. Your acquaintance is surely slight; the ability to list directories to the printer is described in just over two pages of text on the List Directory command. The text includes the specific example of listing to the printer. Based on that kind of knowledge, you then condemn the language for being designed by professional programmers, which makes you "puke."

Be informed that Pascal was designed by one man, Professor Edsger Dijkstra, who is an educator and a well-respected thinker in the computing profession. It was designed to be a language suitable for educating students in the science of programming and for application of the developing scientific principles that the computer profession so sorely needs. Pascal has been accepted as a development language by many professional computing organizations because we find that it helps us write efficient, modular, maintainable, correct programs. It is generally accepted that Basic facilitates attaining none of those goals.

There are many languages used with computers. Some of those available for your Apple are Pascal, Fortran, Forth, Logo, and Lisp. Each of these languages has a reason for existing and a set of tasks for which it is especially well-suited. If you will take the trouble to learn—really learn—a few other languages, you will find that the use of another language can help shape your understanding of a problem and can aid in the solution of it.

As an educator you have a responsibility to teach our children not just facts, but how to learn and how to make decisions based on knowledge. Let me urge you to gain knowledge of Pascal (or of any other subject) before attempting to evaluate it. Let me suggest in evaluating a tool you consider its purpose and the ability of the tool to meet that purpose. Let me urge you to study first, then evaluate. Learn first, then teach. If you will do that, your students may one day be equipped to design a computer language that does not make you "puke."
Ronald E. Jeffries, vice president, Comshare, Ann Arbor, MI

More Than One Way To Skin an Apple

This is in response to the letter "Sour Grapes, or Rather, Potatoes" from Mrs. George Heckler that appeared in Open Discussion (November 1982).

Words of comfort do not have any real effect when a home computer owner becomes frustrated as he or she is learning about their new computer. Feelings such as surprise, confusion, or, worse yet, helplessness can occur even when a user learns about the logistics of a new system or a new software purchase. I can understand how Mrs. Heckler feels—I felt the same way when we purchased our Apple II months ago. Here are some hints on how to get more out of your system based on our experiences:

First, try patience. Learning about your Apple will take time. Plan on that happening. You can reduce that learning curve somewhat by inviting a friend over who has some experience with what you're trying to do. I complained once about how tricky it is to use a word processor program after spending only a few evenings with it. Then I remembered taking fourteen weeks in high school just to learn how to use a simple typewriter! Experience also has no age barrier. The first person to guide me through our Apple was thirteen years younger than myself. So please take your time.

The editing scheme in the Apple leaves something to be desired. The fact that the cursor

GOTO 154

SOFTALK CLASSIFIED ADVERTISING

WOMEN'S GYMNASTICS

Scoring method for a 1 disk drive, 48K Apple + printer. Instant meet results; uses 1982 USGF tie breaking rules; trophy results, team ranking, complete contestant ranking, neutral deductions identified; record and display judges' scores; score Class III through championship Elite meets. 3 years to perfect; used nationwide. $95.00 to D.U. Phillips, 7056 S. Spruce Dr. W., Englewood, CO 80112; (303) 771-0120.

LEARN IBM JCL

On your Apple II, of course. NEW, from the publisher of *Beginning BASIC*, INTRODUCTION TO IBM OS/JCL is a tutorial in IBM Job Control Language for OS. Don't be intimidated by the wizards . . . learn their secrets! Ten lessons teach OS concepts, JCL & IBM utilities. Individual vers. $25; Institutional vers. $50. CATALOG-ONA-DISK for $3. (404) 475-0785. Applied MicroSystems, Box 832, Roswell, GA 30077.

UNIQUE SOFTWARE PRICES

Unbelievable! We sell name brand software and hardware for just $$$ above our cost. For price quotes, orders or more information contact: SPIN-TRONICS, 2490 Channing Way, Suite 303, Berkeley, CA 94704; (415) 540-6345.

THE PREDICTOR

Predict Winners for Professional Basketball. By *Jim Jasper*, author of *Basic Betting & Sports Betting*. 67% accuracy over last two years. $139.95. Req: 48K Apple II or II+ w/Applesoft in ROM. DOS 3.3, 1 disk drive. See your local Apple dealer or write to Pickam Software, 312 S. Los Angeles St., Los Angeles, CA 90013; (213) 687-9530.

APPLE PASCAL PLOTSTUFF

A Library unit for *your own Programs!* Well-labeled screen with user titles and colors. Plots Y versus X with a choice of Log, dB, or Linear axis, using lines, dots, and/or characters. *Save and display plots in 1.5 seconds from disk!* $29.95. VISA & MC ORDERS CALL (408) 629-1251.
ANALOG DESIGN
7109 Via Carmela
San Jose, CA 95139

PHONE DIALER

Don't have your phone daconnected because of unapproved dialers. DIAL-LOG is a hardware and software phone dialing system for the Apple II Plus. Its databasing capabilities make dialing 1,000 numbers as easy as dialing 1.
Completely FCC approved. Guaranteed. $44.95. CATALOG-ONA-DISK for $3. (404) 475-0785. Applied MicroSystems, Box 832, Roswell, GA 30077.

TANK PLATOON!

Players control infantry, tanks, leaders and personnel carriers, with artillery, machine guns, anti-tank missiles, wounding, panic. This is war. 48K DOS 3.3. $29 to Dataworks, P.O. Box 236, Deep River, CT 06417.

** GENERATION STATION **

Writes your Apple II programs for you! MENU DRIVEN, LISTABLE. Search, Custom Input, Screen & Dollar Format Routines. Full Sequential & Random File Systems. $95. (c) Martin Hardee, 1800 Sans Souci Blvd., #208, N. Miami, FL 33161.

FAILING EYESIGHT?

With PEACHY WRITER BOLD you can compose text in double size letters on the screen and print them out in regular or double width. Disk includes PEACHY WRITER, our easy to learn, easy to use text editor. (See review Sept. '82 *Softalk*, page 131.) $24.95. Free catalog. VISA/MC. Cross Educational Software, Box 1536, Ruston, LA 71270. Phone (318) 255-8921.

Develop Professional Video Input/Output Displays in a matter of minutes. Requires Applesoft in ROM, Microsoft's TASC compiler runtime routines, DOS 3.3 & 48K. Price $39.95. Order from: Ed Minkler, 32327 Gibraltar Dr., Riverside, CA 92506.

WIZARDRY PLAYERS

Modify your characters with this utility program. Resurrect the dead or lost. Alter characteristics, spells, hit, experience & gold points. CHEAT! Make yourself a 1000th level super hero for Knight of Diamonds. Disk, 48K & Applesoft. Includes maps. $20.00. California residents include 6½% sales tax. ARS Publications, 3710 Pacific Ave. #16, Venice, CA 90291.

SOFTWARE SALES FOR THE APPLE II

RAINBOW SOFTWARE
9528 Miramar Road, Suite 26
San Diego, CA 92126

POST HOLIDAY SPECIALS

We've taken some of our best selling, low priced items and **lowered** their prices even more. Our combination of **super-low prices** and **unbelievable specials** can't be beat. Call or write for our free price list and our specials card. **Bytes & Pieces**, P.O. Box 525, Dept. S, East Setauket, NY 11733; (516) 751-2535, Source TCP 637.

CHILDREN'S STORIES

Some of Aesop's favorite fables told by Spiral Dan. Pushing single keys lets children interact with the stories. They are delighted by the animated color graphics. Price $26 + $1 postage. Requires Apple II/II+ with Applesoft in ROM. Solar Software, 4102 Laurel Fork Ct., Kingwood, TX 77339.

THOUSAND OAKS

Extensive Apple software. Extremely low prices. AUTHORIZED RANA DRIVE DEALER
The Computer Workshop
One W. Thousand Oaks Blvd., Suite 28-B
Thousand Oaks, CA 91360
(805) 496-7996

EPSON OWNERS! HERE IS PRINT+APPLE!

If your Epson printer has Graftrax-Plus, with PRINT+APPLE you may print address & file labels, stationery, addresses on envelopes, recipes, poetry, line paper, darken word processing, disable paper-out sensor and more! Applesoft/disk/DOS 3.3/48K. Send $24.95 to:
MICROWEST SOFTWARE
868 N. 2nd St., Suite 100
El Cajon, CA 92021

FREE SOFTWARE

Join The Big Red Apple Club, a national Apple user's group with benefits including monthly newsletter and a large library of free software. Annual membership $12. Sample newsletter $1. BIG RED APPLE CLUB, 1301 N. 19th, Norfolk, NE 68701; (402) 379-3531.

MASTER DIAGNOSTICS +

For the Apple II/II+. Keep your Apple in top-notch operation with 24 tests, incl. RAM, ROM, drive speed. Comes with a complete cleaning kit and head cleaner. Great review in August Softalk (p.118). $69.95 + $2 shipping. California residents add 6½%. M/C and Visa.
The Software Source
17905 Ventura Blvd., Encino, CA 91316
(213) 705-4445

TRANSFORM ANY DESK INTO YOUR CUSTOM WORKCENTER

* Handcrafted SOLID OAK system * Modular * Grows with your needs * Unique designs * Top quality * Guaranteed * Random access disk storage * Saves time * Efficient * Houses drives, CRT, manuals, joysticks, etc. * From $31.95 * FREE INFO * **VENICE WOODWORKING CO.**, 12810 Venice Blvd., Suite 426, Los Angeles, CA 90066; (213) 390-4885.

BIG PROFITS TRADING

stocks and stock options. Small investment with low risk. Sophisticated money management system. Write free details. Softsystems, 2005 W. Balboa Blvd., Suite 309F, Newport Beach, CA 92663.

!LOW SOFTWARE PRICES!

No Membership Fees—No Clubs to Join
WRITE FOR FREE BROCHURE
VALLEY SOFTWARE
4974 N. Fresno St., Suite 132
Fresno, CA 93726

APPLE SOFTWARE DIRECTORY

Over 500 Apple programs with full program descriptions, prices, and ordering information. Available at Apple dealers, bookstores, or send $4.95 to: The LAMA Software Directory, Box 201, Bloomingdale, IL 60108. Next deadline for program listings is March 31, 1983.

"HELP" IS IN STOCK

at Leigh's Computers where we offer support before and after the sale. We carry Apple, Atari, Franklin, TI, Vic-20, Commodore 64, and Timex hardware/software; peripherals including printers, modems, and monitors; Atari, Intellivision, ColecoVision game units & cartridges. Stop in today and get our expert advice and friendly service. Leigh's Computers, 212 E. 85th St., New York, NY 10028; (212) 879-6257.

INEXPENSIVE TEXT EDITOR

Search-replace, word wrap, right justification, block text movement, centering, and preview of printed page. Also upper/lower case, alternative keyboards (Dvorak Simplified, one-hand, and others) without hardware. Requires 48K Apple II+, DOS 3.3. Fully copyable. 100% fast machine language. A STEAL AT $30. Cheapsuit Software, Box 923, Iowa City, IA 52244.

BUDGETMATE

A practical solution to computerizing your budget. Easy expense entry and maintenance for up to 250 categories. Provides daily and monthly graphs or summary totals on both screen or 80-column printer. Apple II, 48K, one-disk drive and DOS 3.3 required. Send $20 to BYKE SOFTWARE, P.O. Box 15311, Cincinnati, OH 45215. Ohio res. add sales tax.

VERBATIM

Datalife minidisks for your Apple. Reinforced centers, certified 100% error-free. Only $27.95/box of 10. Packed in plastic file box $2 more. **THE SOFT SPOT, INC.**, Dept. S4, P.O. Box 212, Corbett, OR 97019. Please include $2 shipping/handling. Send for our catalog. Visa/MasterCard now accepted. Verbatim Datalife and Apple are registered trademarks.

CLOCK III KIT

For Apple III. Includes documentation and batteries. Will not void warranty. SOS will date all files automatically. $60 + $2 shipping. Calif. residents add 6%. M/C & Visa OK. Apex Business Services, Box 11109, Dept. C, San Francisco, CA 94101.

GOLD OX DELIVERS!

And at no charge, too. Gold Ox Software sells name brand software at tremendous low prices. Send today for our *free* catalog. Gold Ox Software, 1 Woodcrest Drive, Scotia, NY 12302.

PSYCHOLOGICAL PSOFTWARE

A new name in software introduces four exciting programs: PERSONALITY PROFILE, ASSERTIVENESS TRAINING, STRESS MANAGEMENT, and THE DREAM MACHINE. For more information about these outstanding and reasonably priced programs, write: PSYCHOLOGICAL PSOFTWARE, 4757 Sun Valley Rd., Del Mar, CA 92014.

WARGAMERS!!

Programs now available to assist play of popular board wargames. Game assistance programs increase enjoyment and playability by eliminating bookkeeping & endless die rolling. *Send now for a free catalog of programs in stock.* TRILOBYTE SOFTWARE, INC., 7820 Exeter St., New Orleans, LA 70126; (504) 733-3195.

GET TRW CREDIT REPORTS

With MICROMODEM II, this system replaces TRW terminal for business and consumer credit reports. Data entry is simplified and system is easy to customize. $50.00.
PYRAMID COMPUTER SYSTEMS, INC.
18040 Sherman Way #508
Reseda, CA 91335
(213) 705-8259

LOW SOFTWARE PRICES!

Check out our fantastic prices. Write for our **FREE** price list! KERR SOFTWARE, Box 5301-ST, Long Beach, CA 90805. Or call (213) 428-8193.

QUALITY MICRO PRODUCTS

Apple, TRS-80, Xerox 820. Over 75 manufacturers, *special prices!* For free catalog write or call ODYSSEY DATA PRODUCTS, 9014 W. Broad St. Rd., Richmond, VA 23229; (804) 270-1992.

STOCK MARKET SOFTWARE

Increase your profits with OPTIONX, the ultimate option analysis program for the Apple computer. Price $145.00 including a 65-page manual. In CA add 6% tax. Send for free brochure describing OPTIONX and other sophisticated investment software.
CRAWFORD DATA SYSTEMS
350 N. Lantana, Box 3000-561C
Camarillo, CA 93011

TABULYZER

Analyze survey data. Print 2- or 3-way report-ready tables, frequencies, percent, statistics, marginals. Quickly enter/revise data, table titles, column headings, row labels. Rearrange, combine, delete rows—automatic recalculation. Verify, search, filter, audit. For Apple II+, 48K, 2 drives (3.3), printer. $395. Manual alone, $35. BUSINESS RESEARCH & SURVEYS, 50 Greenwood Ave., West Orange, NJ 07052; (201) 731-7800.

HISTOGRAM PLOT

will print histograms on an 80-column printer, lo-res screen or text screen with machine language speed. Save, edit data files on disk. Printout: mean, median, standard deviation, cell limits, actual and expected frequencies, relative skewness and chi-square goodness of fit test. Compiled version for hi-speed. Applesoft version for user changes. $39. Andent Inc., Dept. J, 1000 North Ave., Waukegan, IL 60085.

— SAILMAKERS — IOR RATING ANALYSIS

This program calculates the IOR rating (R or RA) for any yacht. Sensitivity analysis shows the effects upon any variable of varying any measured variable. Penalties can be displayed. Twenty yacht database. Certificates printed on an Epson printer. 64K Apple required. Contact: Kurt TeKolste, 5612 Harding Ave., Alexandria, VA 22311.

THE EASY BOARD: Serial and Parallel interface for Apple II or II+. FEATURES: Operates in any slot; menu-driven I/O in ROM; programs set I/O with 1 byte; input-serial/keyboard; output-screen/serial/parallel/any combination; external or Apple terminal in Full/Half duplex. Only $99.95. Introductory price good through Apr. 83. Dealer inquiries invited. E.T. COMPUTER PRODUCTS, 6523 Briar Glade Dr., Houston, TX 77072.

If you like **FIRE ORGAN** (FREE from most Apple dealers), you'll *love* **SPARKEE**, the second "album" of dynamic (ever-changing) visual compositions from Vagabondo Enterprises. The 24 **NEW** works of art on this disk (for your 48K Apple) are **GUARANTEED** to delight you with their brilliant colors and graceful flowing patterns of continually moving shapes, lines, and curves. **SPECIAL OFFER:** Send just $20 (PA residents add $1.20 sales tax) to Ken Sherwood, 117 N. 25th St., Reading, PA 19606.

SUPPORT YOUR HABIT

Picture yourself in your own home . . . using your talents in a money-making opportunity with unlimited sales potential. Sell state of the art software and accessories for the Apple Computers. If your forte is people and your habit is Apple, then send for more information immediately on one of the most significant acts of your life. No franchise fees. No inventory investment. Write Wallace Micro-Mart, 2619 N. University, Peoria, IL 61604.

Softalk's classified advertising section offers a considerably less expensive way than display advertising to reach tens of thousands of Apple owners.

Classified advertising space is available at the rate of $10 per line for the first ten lines, with a five line minimum. Each line over ten lines is $25 per line. Ad copy should be received no later than the 10th of the second month prior to the cover date of the issue in which you want the ad to appear. Payment must accompany ad copy.

The publisher reserves the right to reject any advertising that he feels is not in keeping with the publication's standards.

Heads will be set in 10 point bold face, all capitals only. Italics are available for body text only; please underline the portions you would like italicized.

The body text of the ad will hold roughly 45 characters per line. Spaces between words are counted as one character. Heads will hold roughly 24 characters per line, with spaces between words counted as one character. Please indicate if you would like the head centered or run into the text.

Please write or call for additional information.

Softalk Classified Advertising
11160 McCormick
Box 60
North Hollywood, California 91603
Attention: Linda McGuire
213-980-5074

GOTO 46

TRADETALK

☐ **Jim Nitchals**, cofounder of Cavalier Computer and coauthor of *Bug Attack*, *Microwave*, and others, has left the firm to freelance. Currently, he has a contract with **Electronic Arts** (Menlo Park, CA) as an independent game author. He will produce one game for EA immediately, and the company will have first refusal rights on future games.

☐ **Orange Micro** has moved to new corporate headquarters in Anaheim, California. The new facility will have more than fourteen thousand square feet of space. "The incredible success of the Grappler Plus interface card and the Bufferboard has certainly played a major role in Orange Micro's surging development," says company founder and president **Art Scotten**. "This success, coupled with the success of our retail computer stores, has led to expansion of market reach as well as facilities." Orange Micro's new address is 1400 North Lakeview, Anaheim, California 92807; phone (714) 779-2772.

Paul Chaison, head of IUS research and development.

☐ **Information Unlimited Software** (Sausalito, CA) has appointed **Paul D. Chaison** vice president of research and development. He spent more than twenty years with IBM and was in charge of overseeing vendor-supplied applications for the IBM pc. Chaison oversees technical writing and programming as well as manufacturing functions.

The company, publisher of *EasyWriter* and related programs, has assigned its advertising account to **D'Arcy-McManus and Masius** of San Francisco, who will develop national consumer and trade advertising for IUS as well as collateral material and other communications projects.

☐ A U.S. District Court of San Francisco has ruled in favor of plaintiff **Sirius Software** (Sacramento, CA), prohibiting **Sirius Systems Technology** (Scotts Valley, CA) from using the name *Sirius*. It was ruled that the Sacramento software company had superior rights to the name and was demonstrably in direct competition with other personal computer manufacturers, having introduced its new color digitizing microcomputer at last November's COMDEX.

☐ Advertising piracy is on the rise! An employee of **Software Distributors** (Culver City, CA) attending the Hanover Industrial Fair in Germany discovered an advertisement for Northamber Limited, an English distributor of computer hardware, virtually identical to Software Distributors's "1-Upmanship" ads. Half a dozen words in the ad copy had been changed to refer to hardware and peripherals instead of software. Software Distributors has not yet contacted the English firm regarding the matter.

Software Distributors has expanded its offices to a five thousand square foot facility at 10023 West Jefferson Boulevard in Culver City. The building will house administrative, sales, and service departments. The company's original quarters at 9929 West Jefferson Boulevard will be used for warehousing and marketing support. "These new quarters will allow Software Distributors to continue with its present rate of expansion," says **Linda Johnson**, vice president of administration. "In addition it will provide retailers with a greater software selection and faster delivery time."

☐ "The Personal Computer Show," produced by **Solutions Incorporated** (San Antonio, TX), is up and running on the cable Satellite Program Network. The show, sponsored by **Apple Computer** and **Sierra On-Line**, is a "video magazine" utilizing demonstrations and interviews to illustrate different applications and benefits of personal computers. News, business, education, games, and weekly feature segments spotlight new products and developments in the industry.

☐ *Softline* magazine, born in a log cabin in Coarsegold, California, as the *On-Line Newsletter*, has been purchased from **Sierra On-Line** by **Softalk Publishing** (North Hollywood, CA) for a six-figure sum. Softalk had taken over publication of the newsletter in September 1981 and turned it into a nonpartisan computer games magazine, with On-Line staying on as financial partner. As the software house has now gone on to other ventures, Softalk has taken over all aspects of the magazine's publication. With full autonomy, *Softline* will expand in size, featuring contests, special games, and a classroom participation column, and will "generally be more fun than most reasonably sane people can stand," say informed sources.

☐ **Programs Unlimited** (Jericho, NY) has granted franchisees exclusive rights to develop seventeen territories nationwide. Under terms of the agreement, franchisees may retain their exclusivity by opening sixty-one new computer stores in the next thirty-four months.

☐ All Apple software from **Micro-Sparc** (Lincoln, MA) is now available to dealers through **Softsel** (Inglewood, CA) and **Software Distribution Services** (Buffalo, NY).

☐ The **Burroughs Business Forms Division** (Phoenix, AZ) is working with business applications software companies to provide a forms support program. Burroughs will supply the necessary business forms, reports, and statements required in the ongoing use of a business software application program. **Spectra/Soft** (Chandler, AZ), makers of *COMMA*, a medical management software system, has been the first company to take advantage of the program, in which Burroughs provides related supplies and equipment for computer installations and national marketing representation. Approximately eight hundred Apple dealers have been contacted by Spectra/Soft about the *COMMA* software and forms support program. Company treasurer/controller **Brian Sheppard** has termed dealer response "extremely favorable."

☐ **Computerland** (Hayward, CA) has announced an "early bird" software program for its nearly three hundred stores worldwide, giving participating stores product return privileges and the opportunity to experiment with software at no risk and providing product selection and evaluation by Computerland's newly formed Software Products Committee. "We want to provide a forum at corporate through which to channel the expertise of the network and the vendors," says software department manager **Ed Murphy**, "providing follow-up support to the stores, so they can educate the customers about what the software does, answer their questions, and help solve their problems."

Computerland has signed an agreement with **Simple Soft** (Elk Grove, IL) to distribute the *Quickcalc Real Estate Investor*.

☐ **W. G. Zirneklis**, president of **MCE** (Kalamazoo, MI), formerly Interpretive Education, has announced the sale of the company's entire line of multimedia products. MCE will dedicate itself exclusively to the production of high technology products. "This move represents a major repositioning of our company to meet better the current and accelerating future needs for microcomputer-based programs for schools, adult-training, and home education," says Zirneklis. MCE is also expanding its research and development efforts following two years of development and testing for educational microcomputer programs.

☐ **Evotek** (Fremont, CA), manufacturer of 5¼-inch hard disk drives, has named **Russ Bishop**, **Peter A. Tegan**, and **Darwyn F. Kelley** as regional sales directors. Bishop, former manager of product marketing for Honeywell, will handle Evotek's sales in the eastern United States. Tegan, formerly western region manager for ECS Microsystems, will fill the same role at Evotek. Kelley, southern regional sales manager, was previously branch manager and regional manager for Honeywell.

☐ **Don Mandell** has been appointed vice president of sales and marketing for **Axlon** (Sunnyvale, CA). He will be responsible for all Axlon domestic and overseas marketing efforts. Previously, Mandell was western area director and regional general manager for Commodore Business Systems and Northern California sales manager for Wang Laboratories.

Don Mandell, Axlon's vice president of sales and marketing.

☐ **Pat Bachelder** is the winner of the first annual national postal *RobotWar* tournament. His robot, *Stalker*, overwhelmed **Richard Fowell**'s *Norden2*, winner of the *Computer Gaming World* magazine *RobotWar* tournament. The other three semifinalists were **Norman Lowe**'s *Remover*, **Allan Turoff**'s *QX4*, and *Bell* by **Carlos Escobar**. There were seventeen entries, two from Canada. Novice robots will not have to face *Stalker* again unless they want to or until they have won a tournament and also become master robots. For information on future tournaments, write to **Frank Krogh**, Box 5337, North Hollywood, CA 91616.

☐ The **Erin Computer Learning Center** (New York, NY), a part of Erin Computer Distributing Corporation, provides training to businesses and individuals on a variety of microcomputer systems and software, focusing on word processing, accounting, and financial applications. The center offers one-day intensive courses for executives and managers, half-day training courses for office staff in aspects of computer-based office work, a posttraining annual support contract for individual students, and ongoing training arrangements for businesses.

☐ **Corvus Systems** (San Jose, CA) has again expanded its facilities, bringing the firm's total facility to one hundred thousand square feet. Part of the expansion will be dedicated to additional shipping capabilities for Corvus networks and Winchester disk drives. The rest will be used by the expanded marketing department. The firm has also completed the offering of 2,750,000 shares of common stock at $13 per share.

☐ Wells Fargo Bank has extended an unsecured line of credit in the amount of $500,000 to **Corona Data Systems** (Westlake Village, CA). The funds will be used as working capital and inventories in current production of hard disk peripherals. The company has appointed **Loren Eltiste**, former financial vice president, treasurer, and secretary for PERCI, Incorporated, a Los Angeles based computer peripheral manufacturer, as finance vice president and chief financial officer.

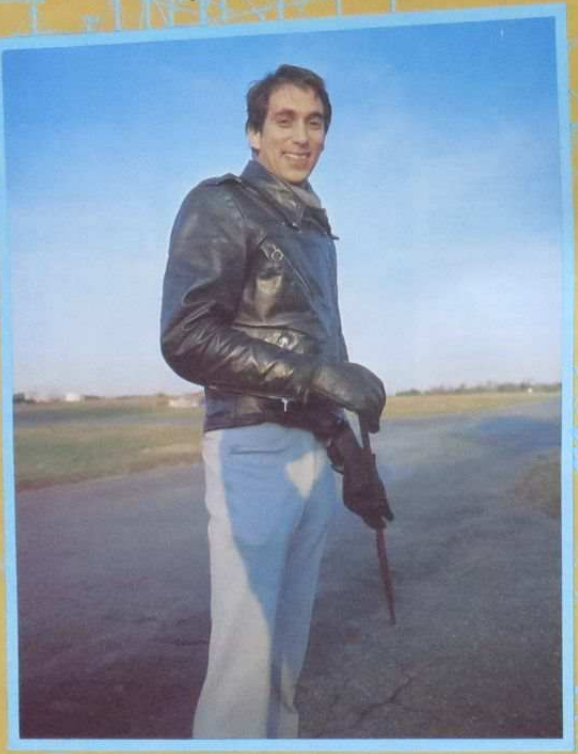
☐ **Software Dimensions**, producers of microcomputer accounting packages, has moved to new thirty-five thousand square foot corporate headquarters in the Lake Forest Technical Center in Folsom, California. **Ronnie Green**, president of the two-year-old firm, cites accelerated research and development and personnel additions as reasons for the expansion. "Software Dimensions's growth has surpassed all our expectations, and there is every indication that we will continue this growth rate in the future," says Green. "We expect to hire one hundred fifty people in our first three or four years in the Folsom area." The company's hiring needs are in research and development, production, and marketing, with priority given to the local labor pool.

☐ **Mark G. Kallin**, former president of Precision Products of Aptos, California, has been named manager of business development for **Ultra Magnetics Technology** (Watsonville, CA). He will be responsible for the company's expansion into new markets for 5¼-inch and 8-inch floppy disks. "We want to explore a number of business development opportunities," says Ultra Magnetics president **John A. Buchanan**. "Kallin's engineering and operations background with companies like Ampex and Memorex, and his sales management skills, provide a unique combination of talents and versatility."

☐ **Prentice-Hall** (Englewood Cliffs, NJ) announces the appointment of **Dr. Lance A. Leventhal** as editor of the company's new personal computing series. His duties will include editing, reviewing, and developing outlines for the two hundred books in the series. Dr. Leventhal will consider books accompanied by software for inclusion in the project.

☐ **Stellation Two** (Santa Barbara, CA), manufacturers of *The Mill* and *Pascal Speed Up System*, have entered into a sales agreement with **T&W Systems** (Fountain Valley, CA), creators of the *CadApple* drafting system, allowing T&W to package the *Speed Up System* with every *CadApple* program.

☐ The starting dealer discount from retail price on products from **Hayden Software** (Lowell, MA) is now 40 percent on a minimum order of $100 at list. The new discount structure, which has a maximum discount of 50 percent, includes an additional 3 percent discount on prepaid or COD orders. In a departure from industry policy, the discount applies to both dealers and distributors. Additionally, dealer credit limits have been increased and a new order processing system installed to reduce turnaround time. "Our goal is twenty-four-hour turnaround on orders," states company president **Oscar Ray Rodriguez**.



Exec SubLogic: On Course and Flying High

BY DAVID HUNTER

How fast is too fast?

How big is too big?

Ask the ghosts of Elvis, Marilyn, and James Dean. Ask the ghosts of Sutter, Keats, and Alexander the Great. While you're at it, ask T. E. Lawrence, Amelia Earhart, and Dostoyevsky.

Elvis became a king and a slave to his subjects before he knew what hit him. In three years Keats wrote more good poetry than most poets write in a lifetime and then he died. Alexander the Great conquered most of the known world before reaching the legal drinking age in present-day California.

The Harder They Fall. Speed is exhilarating. Uncontrolled growth is intoxicating. As long as youth can dream, life will never move fast enough.

Whether it's motorcycles, cars, planes, skiing, volleyball, or assembly language, Bruce Artwick likes speed. He likes Winchester disk drives, BMWs, zooming through undergraduate and graduate school in four years, and tearing down the Angeles Crest Highway on a Suzuki at a dangerous clip. The president of SubLogic Communications Corporation, Artwick is a tall, quiet, twenty-nine-year-old bachelor. He possesses a remarkable mind, which has created several of the finest programs to grace the Apple's RAM.

Contrast Artwick with Stu Moment. Outgoing, of medium height, and possessing an exceptional love of flying, Moment is SubLogic's chairman of the board. A businessman, Moment has steered the company to

Opposite page: SubLogic's chairman of the board, Stu Moment, calls himself a "crazy aviation fanatic." This page: top, SubLogic's president and chief software engineer Bruce Artwick at ease in his office—christened Lazer Bay One from the movie *Tron*; bottom, Moment and Artwick pose with the faithful company plane, a Cessna 150.

Both Artwick and Moment are licensed pilots and fly all over the country when the need arises or the spirit moves. Above, Moment listens for important information on the radio and scans the sky for other planes while approaching the University of Illinois Willard Airport.

its present course, complementing Artwick's superior software engineering talents with organizational and financial skills. He's even picked up some modest programming skills, designing a system for logging flight hours at a fair-sized flying institute.

Redford and Newman. Lewis and Clark. Laurel and Hardy. Jobs and Wozniak. Artwick and Moment. The grand adventurers riding the hard trail, living and playing at lives larger than life. It's an old story.

Leaving hyperbole for the poets, the truth is that Artwick and Moment are two thoroughly likable individuals who confess to hiding out in central Illinois, seemingly far away from the spotlights and hubbub. Unfortunately, spotlights and hubbub are notorious for seeking out the famous, no matter how hard they try to hide. Not intentionally being mysterious, the company SubLogic is nevertheless something of an enigma. Who are they? Where did they come from? Where are they going?

One thing is known. Artwick's *Flight Simulator* for the Apple was a monster success in the first two years of its release. It placed second on *Softalk*'s first Top Thirty poll in October 1980. It was last seen on the Top Thirty in May 1982's poll at number twenty-seven. *Flight Simulator* has been pulling strong on the Strategy Five for sixteen straight months, usually second or third. Including the TRS-80 version, Artwick's *Flight Simulator* has sold more copies than any other flight simulator of any kind in the world.

"And It Has Always Been Attributable to Human Error." Champaign is a farm community and a college town—home of the Fighting Illini and of the third busiest airport in the state and birthplace of the most famous fictional computer, HAL, in Arthur C. Clarke's *2001: A Space Odyssey* and *2010: Odyssey Two*. It all started here.

Artwick and Moment first met in the early seventies when they were attending the University of Illinois. They roomed together with a dozen other spirited students in a big old house dubbed Gamma Ray Zappa. The house was evenly split up among AV (aviation) jocks, parachuters, and electrical engineers.

Moment was one of the AV jocks. Rakish looking in his black flyer's jacket, Moment flashes a winning smile and recalls the true roots of his love of flying: "My mother was six months pregnant with me and she flew in a Piper Cub through turbulence. I was born a month early, feet first like a paratrooper. I first flew in a small plane at fifteen. I was really into engines and decided I wanted to be an airplane mechanic."

Moment enrolled at the university's Institute of Aviation and began his flying career. He recalls fondly his student days, a past that both Artwick and Moment are very much in touch with.

"I was like any good college student at the time. I protested and had a guitar," says Moment. "I finally gave that away to our vice president of marketing, Norm Olson, a year ago."

Gamma Ray Zappa was not a real fraternity, though fraternities were a big thing around campus. Still, the ad hoc brotherhood included a pretty crazy bunch of guys.

Bruce Artwick was one of the electrical engineers. "We agreed that Stu would teach me to fly if I taught him about digital electronics," he remembers.

When Artwick wasn't entrenched at the university's digital computer labs, he was doing the things only college students and devil-may-care parachuters can think up. He enjoyed the vibrant atmosphere and the chance to learn about flying.

Rebel without a Cause. An electrical tinkerer for years, Artwick surprised no one when he started to build his own computer. On his homemade BACS 1000 (Bruce Artwick Computer System), Artwick did some "3-D stuff and even a real-time runway. I just had to have my own real-time 3-D graphics."

Through most of his college career, Artwick was hardware oriented, particularly interested in computer design. Then his thesis project called for creating a real-time flight simulator on the PDP-11. Graphics soon became a major concern and hobby of Artwick's that would figure significantly in the genesis of SubLogic.

Artwick completed his bachelor's and master's degrees in electrical engineering in four years. When he started college, Illinois-born Artwick says it was a fifty-fifty chance that he would study either music or engineering. Once he made the choice, thanks to the excellent curriculum at the university, Artwick pursued his passion for computers ferociously. When he had finished, he had no trouble landing a job at Hughes Aircraft in Culver City, California.

While he was at Hughes, several interesting things happened to Artwick. Twice, midair collisions between small planes occurred close to his house; one time debris landed in the swimming pool. He learned to love the Southern California mountains and their winding roads that beat the cornfields of central Illinois for riding motorcycles. Artwick also began dreaming of founding a big company with hundreds of employees.

Back in Champaign, flight instructor and business major Moment shared some of Artwick's dreams. In early 1977, they decided to form a company. Back at the university's digital computer labs, Artwick worked on devices known as sublogic circuits, and that's where the company name comes from.

"We've had people think we make logic units for submarines. One or two people have called thinking we were a mental institution," Moment recalls.

South by Southwest. In early 1977, Artwick wrote an article on 3-D graphics for *Kilobaud Microcomputing* and SubLogic sold its first product, a 3-D graphics package for Southwest Technical Products's 6800 processor. In those early days, neither Artwick nor Moment had any real expertise in marketing, though their job wasn't that tough. There were only about two hundred computers with that processor.

Other early products from SubLogic were graphics drivers for S-100 boards, drivers for Vector graphics, and television dazzlers. Artwick's megacompany dream started slow, but the two fighting Illini gained considerable knowledge about the market with those early forays into hardware and software.

Artwick was still working at Hughes and Moment was still plowing through college when the two decided to publish programs for machines with a wider circulation. In 1978, that meant TRS-80s and Apples. But first Artwick had to return to Illinois.

Besides pursuing the dream of heading his own big company, Artwick chose to leave Hughes because the rewards for good work were too

few. "Only generals would see the results," he explains. "I also wanted to leave Los Angeles because of the high cost of operating SubLogic from two locations."

Anatomy of a Dream. One of Artwick's heroes is Seymour Cray, who left Control Data in the mid-seventies to start his own company. Cray discovered a niche in the computer industry and formed Cray Research to fill that hole. Supercomputers are Cray's forte, and Artwick looked for a similar niche in the microcomputer market. He found it in graphics and simulations.

In June 1979, Artwick packed up a U-Haul and moved back to Champaign. Four months later, SubLogic published its first Apple program, *A2-3D1 Graphics*. Three months after that SubLogic launched *A2-FS1 Flight Simulator* for the Apple. A TRS-80 version of the flight simulator, *T80-FS1*, was released simultaneously. Unlike the typical first-time-in-business Apple software publishers—Sirius, HowardSoft, Synergistic, Sierra On-Line, and many others—SubLogic diversified into the Apple market instead of out of it.

Artwick took advantage of the Apple's larger memory to add lots of little improvements impossible in the TRS-80 version of the flight simulator. The animation is 25 percent faster and the instruments are closer to those of a real small airplane. On cassette at first, as were many Apple programs in those days, *A2-FS1* soon became available on disk and sales took off.

In June 1980 SubLogic got its own Cessna 150. The recession had already hit private aviation, but the eighteen-year-old aircraft had been equipped in the factory with long-range gas tanks at the request of its original owner. This made cross-country flying economically feasible. Moment and six-foot two-inch Artwick crammed into the tiny two-seater and flew all across the country. They would take test flights to see if Artwick's mathematical simulations were accurate. It was very gratifying to fly in the company plane and see Artwick's software engineering coincide with the real thing.

Real Men Don't Eat Applesoft. Artwick prefers assembly language because of its high performance characteristics and the results, which run very fast. Higher level languages like C and Pascal are 75 percent as fast as assembly language, and that extra 25 percent, Artwick and Moment believe, is what makes SubLogic competitive. You need good tools to work in assembly language, because it's not a nice structured format. Still, that difference of three or four frames per second in the animation is what the competition has to contend with.

In 1980, SubLogic hired its first full-time employee, to answer the phone. Rhonda Fultz filled the role until she got married and quit. Julie Newman took her place and is still with the company, though she now performs in a different capacity. Newman is distribution manager and the major force behind SubLogic's superb dealer network.

Soon after Newman took over Fultz's position, SubLogic hired Teresa Evans to handle the shipping. A key person for the last year and a half has been vice president of marketing Norm Olson. An old college chum of Moment's, Olson had a motley college career that included psychology, premed, and rhetoric. When Moment and Artwick were looking for a marketing person, they chose Olson for his "fantastic" background.

Artwick and Moment tried to create an image for SubLogic. Particularly in the Apple market, they didn't want to be seen as just another software publisher. Caution is perhaps the key thing here. Artwick and Moment believe in putting out a few quality products and providing strong service for those products.

"Money is slightly secondary to us," explains Moment. "We've made plenty for where we live. In central Illinois there are no Joneses to keep up with."

It's cheap to run a business in Champaign and it's cheap to live there. How cheap?

"This spring I'm buying five acres at $4,000 a piece," says Moment. "I'm building a big house for my family and the entire thing will cost around $80,000."

The Luck of the Irish. Nestled in the center of Champaign is a local institution that has played no small part in the SubLogic story. Artwick describes Murphy's Pub as the kind of place where "all the engineering majors go to drink green beer on Saint Patrick's Day." Warm, dark, intimate, and supplied with a well-stacked jukebox, Murphy's Pub is the site of "some of the biggest decision making" for the SubLogic executives.

Above, view of the University of Illinois Willard Airport—third busiest in Illinois—from about two thousand feet. Artwick and Moment dream of moving SubLogic's offices out to the airport—they really like the flying environment.

For almost two years, SubLogic's only Apple products were the *A2-FS1 Flight Simulator*, *A2-3D1 Graphics Package*, *A2-3D2 Enhancement*, and *A2-GE1 Graphics Editor*. Early in 1981, Artwick and Moment decided to start experimenting with a slightly more diverse product line. In May 1981, calling themselves the "engineering and graphics experts," SubLogic released *Saturn Navigator*, by Wesley Huntress, author of EduWare's current hit *Rendezvous*.

A hi-res adventure/simulation, *Saturn Navigator* is a thinking-person's arcade game, a good example of the SubLogic image. You don't have to contend with invading aliens and you're not searching for a long lost monolith or the secret chamber of Urbana. You have to navigate through the rings of Saturn and negotiate an orbit around the massive gas giant. It's not easy, but it's an educational and rewarding simulation.

Another good example of SubLogic's elusive image is their advertising. Like many a young company that sprouted up at the start of the microcomputer boom, SubLogic was at a complete loss and searched for an agency to handle this all-important task.

"We looked for one in the Yellow Pages," remembers Moment. "The first guy we went to see was choking on a tie but didn't impress us. Then we found Bob Chapdu of ADLIB who ran a creative and artistic place, but he wasn't trying to be Madison Avenue." He was also sympathetic to SubLogic's conservative image.

Most growing companies eventually have to move up to color, and late 1982 saw Chapdu and ADLIB launch SubLogic's first big advertising campaign—for Artwick and Moment a momentous, almost scary, step. They wanted to be colorful but different. The result is a group of distinguished looking full-page ads that aren't misleading. Unembellished screens from the programs are featured prominently, giving potential buyers an idea of what they'll be getting.

It's not a new approach, particularly for business programs, but in the world of entertainment software it's a welcome change from the usual airbrushed extravaganza. SubLogic, with the help of Chapdu, has taken the idea and used it in packaging as well. Hanging on the rack in the computer store, a SubLogic product always has a sample screen to persuade the buyer and make the salesperson's job a little easier.

"Pinball Wizard, There Has To Be a Trick." When *Raster Blaster* was released, Budge's masterpiece seemed to be the last word on the subject. Few people would have predicted that two other pinball programs would achieve even a modicum of success a year later. *Raster Blaster* unseated *VisiCalc*; it sold systems; it was a phenomenon.

Never underestimate Bruce Artwick's talent. *A2-PB1: Night Mission Pinball* was released in late spring 1982 and climbed to number five on the Top Thirty in July of that year. The reviews were staggering. How had Artwick done it? The animation was so smooth, the configuration of features so realistic and competitive. The Cosmic Mode was the rave of the day.

What Was That?

The blank space to the left of this column of type ran on page 290 of the December issue.

Softalk's art director disclaims that space as a representation of all he's been able to accomplish with the Apple's hi-res page.

Softalk's publisher disclaims that space as his collected wit and wisdom.

Those readers of keen insight and a puzzle-solving bent probably referred to the Ad Index, where they ascertained that the space belonged to the Keyboard Company.

Let it hereby be made known that the Keyboard Company had much more to say than the space on the left indicates. In the production process, we muted the Keyboard's voice.

Astute readers may recognize the Keyboard Company name as belonging to Apple. The old Keyboard Company was folded into Apple and dissolved. The new Keyboard Company picked up the remnants of that process but feared the name might cause confusion.

So now they're the Accessory Products Company, and what they had to say as the Keyboard Company is the essence of the Accessory Products ad to the right of this column.

With a new name and a restored advertising voice, Accessory Products is off and running. Which may be more than can be said for the collected wit and wisdom of the publisher.

If Broderbund's also highly praised *David's Midnight Magic* hadn't come out several months earlier, *Night Mission Pinball* probably would have unseated *VisiCalc*. Well, maybe.

In the last year, a number of different kinds of programs have emerged from SubLogic. Artwick and Moment decided to experiment with the educational market, releasing *Whole Brain Speller* from the team of Susan Campanini, David F. Manton, and Joe Weintraub and *Music Maker* by Jim Baldridge. It's still too early to tell, but SubLogic is not dissatisfied with the results and may offer more educational products.

Another new area was opened up with the hi-res arcade/adventure *Space Vikings* by Mitch Robbins. "We wanted something that would take people all winter to play," recalls Moment. Multifaceted and challenging in all the right ways, *Space Vikings* is another perfect SubLogic product. It doesn't have to sell thousands and thousands of copies. It has to stimulate the imagination and exercise the intellect.

The latest experiment is in the familiar area of strategy games, or simulations. SubLogic's New Wave Strategy line is just now reaching the market. All three games, *Zendar*, *Roadblock*, and *Front Line*, are authored by Terry Eagan and fit into the SubLogic mold. Moment is a big fan of strategy games and even did some of the finishing touches on the programming of *Front Line*. He's hoping the games will be well received, though that is no condition on the success of the experiment.

"Junior! Come Down Here!" SubLogic's next move will most likely be into hardware for the Apple. Artwick and Moment are developing what they call a "universal simulator." Not just another joystick, this multiplexed, seven-channel contraption will give a more realistic aspect to the flight simulator, possibly including foot pedals, a steering wheel, a separate throttle, and other features.

It sounds like Artwick and Moment should just build an FAA-approved flight simulator and get it out of their systems. They're smarter than that. Enough companies are out there making the real thing that SubLogic isn't interested. Nonetheless, their idea for the new controller would come darn close to having all the features of a real simulator.

Naturally the universal simulator will adapt to other kinds of programs. Artwick believes it could add a whole new aspect to computer gaming. One exciting feature would be motion feedback, a software/hardware coordination allowing more precise control and faster, more accurate response.

All this experimentation is meant to define a niche for SubLogic in the world of microcomputers. Artwick turns again to Seymour Cray and his success story in the extremely narrow market of supercomputers. SubLogic would like to be king of simulation and concentrate on putting out the best products of this kind. One at a time, if need be.

Even though he wanted a big company at first (SubLogic has had close to 400 percent growth each year of its existence), Artwick is now content to stay relatively small for a while. He's learned something else from Cray, whose company grew so fast it almost caused Cray to leave. Artwick feels that when a company gets too big it gets impersonal. To maintain control over a large company, its leaders often become egotistical dictators, Artwick believes, and he doesn't want any part of that.

Artwick and Moment like to keep their fingers in all the workings of SubLogic, dealing with customers, making the everyday decisions, and sharing the long-term decisions with the likes of Norm Olson and Julie Newman. Artwick and Moment personally talk to new and old customers every day, coming to know many pretty well. Newman and Olson also have a terrific rapport with the loyal SubLogic customers and dealers.

"There are a lot of people out there who've bought all our programs," says Moment. "They're fiercely loyal and we'd never let them down." That's why SubLogic still carries products for the S-100 computers, even though they represent a rapidly dying market. That's why all SubLogic's Apple products come with a lifetime five-dollar replacement policy when the bad disk is returned.

All Along the Control Tower. Moment says that few flight instructors like to fly all day. He, on the other hand, couldn't get enough of it. "At the end of a long day, I'd be tired, yeah, but I felt great. I got along with the super hippie types and the military jocks."

Moment got his own instruction at the institute from a retired Marine commander. An active flight instructor from 1973 to 1980, Moment is proud to say he's on the Aviation Institute's Incomplete List of Excellent Instructors.

At the same time, he calls himself a crazy aviation fanatic and a super aviation head. "I fly all week and on Saturdays. On Sundays I'm out there in the fields with my daughter flying a radio-controlled model airplane." He routinely flies across the country, has logged in 2,500 total flight hours, and performs maintenance on the plane regularly.

"Even though I do it full-time, it doesn't matter how many hours you rack up," he explains. "Each time it's a new experience. Most flights have something new."

Artwick is not as fanatical about flying as Moment, but he's a good pilot nonetheless. Artwick's true love is motorcycles. He currently owns three and in his lifetime has had a total of fifteen. Why so many? "It was unintentional. I sell them. I don't crash them."

Artwick is currently working on a book about interactive computer graphics as well as on an ambitious computer-aided design project with associates in Amsterdam. The CAD project is very close to his heart and Artwick considers it his most important and complex creation to date. It's not for the Apple market, unlucky us.

Airport 1982: Hijacking Software. The University of Illinois Willard Airport lies just off Highway 45, and, if Artwick and Moment get their way, it will be the future home of SubLogic. They like the flying environment and would love to be close to the Institute of Aviation and the company plane. There may even be good business reasons for such a move. The only major problem comes with building their own offices—the university owns all the land and would own the offices.

Comfortably spread out in three locations at present, SubLogic is constantly looking to the future—uh, cautiously. You know, conservative-like. Artwick and Moment recently took a flight in a helicopter and it's not hard to guess why. Yes, Virginia, there may be a future simulator there.

Oh, and about Artwick's love of motorcycles. . . . But there's no reason to go too far, too fast. All good things happen in their time.

You may not have heard this one, but you've probably got a similar Apple horror story to tell. There was this small business owner who bought himself an Apple and a simple accounting package to use in his one-man shop. Ignoring the pleas of his family, he stayed up for many a night entering three years' worth of past records, making those records balance, and then printing out reports that looked just like those he'd been paying an accountant to do.

Finally, the man felt confident enough to put his current records "on the computer," as he now knew how the jargon went. All went well for about a year. Then, on a dark, stormy January night at 2:17 a.m. while he was in the process of entering his post-Christmas sales and inventory data, it happened.

He had just finished entering the sale of an alligator shirt, matching belt, and three pairs of socks to credit card number 18723-865-09543-0081-2 (exp. 5/84) when, without warning, his Apple went into a coma.

He could see that it was still alive. Its power light was on and warm to the touch. Its monitor glowed green and displayed the last entry. But the machine was totally insensate. It simply wouldn't pay any attention to anything he typed at the keyboard. Finally, in frustration, he powered down for the night and went to bed.

The next day, of course, our friend successfully entered the data and balanced his books. A couple of weeks later, the Apple again went into shock. And then, a few days later, it happened again. Figuring that the machine might be overheating, the man bought a fan. This helped for a while, it seemed, and then it didn't.

Well, to make a long story short, the man finally gave up on his experiment in computerized accounting and returned to his old manual system (which, if the truth be known, allowed him to finish his books each night in time for the late news and then to get a good night's sleep). The Apple became a great game machine for his kids and seemed ideal for that purpose; when used for games, it never showed the slightest sign of brain damage or toxic shock.

All would have been well for the kids if their father hadn't walked into a computer store at the wrong time one day, shortly after having reinitialized and reused all those disks that he'd spent so many hours saving his accounting records on. As he stood at the counter waiting to buy a new home-arcade game for the kids, he overheard the clerk telling a customer about something he called "garbage collection."

Our businessman listened in ever-growing horror. Then he ran out of the store, jumped in his car, laid rubber as he peeled out, and. . .

There's no need to go through the gory details. Suffice it to say that the man's kids are again enjoying mumblety-peg, hopscotch, and kick-the-can. They're out of doors, full of vitamin D, and ever so much tanner. The computer store clerk who originally sold our businessman his system and taught him how to use it but *forgot* to mention garbage collection is recovering nicely.

So there's something called "garbage collection" that you didn't know about, right? Well, you'd better learn. If not, someday you may find that after you've been entering data for half an hour or so your Apple, without warning, will appear to hang up and won't respond to anything except a reboot. Actually, if you wait a long while—perhaps as much as several minutes—your Apple may reawaken and pick up where it left off. But if you don't know what's happening you'll probably blame the problem on a program bug, swear a bit, hit reset, and start over, long before the necessary time has elapsed.

Unless there actually is a program bug (pr#8 will do nicely), what's happening when the Apple appears to hang after a lot of string data entry is that it's erasing the data that's no longer being used to make room for new data. String variables are stored differently than numeric variables. In numeric variables, the values of the same two eight-bit bytes can be changed repeatedly to represent any integer number from −32767 to +32767. Applesoft's numbers are more complicated, but the principle remains the same. A string variable, on the other hand, can be anything from a single character to 255 characters in length.

If the Apple had to set aside 255 bytes for each string no matter how short the string in question might be, an awful lot of memory would be wasted. Instead, each input string is stored in order of entry. You can't cram in a long string where a short one used to be, and you can't be sure that any new strings will be the same length as the discarded ones; so Applesoft simply puts the new strings right after the old ones without bothering to erase the old ones. If, for example, you had an address list in which at first X$ = "101 5th Street" and later X$ = "25032 Washington Boulevard," both addresses would be retained, but a pointer would skip over the first after the second one "replaced" it. After a while, of course, those unused strings would start to fill up the Apple's memory.

Garbage collection—removal of strings that are no longer used—can be done periodically from inside a program. If it isn't, however, Applesoft will do it automatically, but only when all available memory is used up. At that point it checks to see which strings are still in use and removes only those that the program has discarded. The slang term *garbage collection* is apt.

Go to the Source. When you run across something new and confusing, it's always a good idea to look in your Apple's memory to see what's going on. In the process you'll also pick up a smattering of machine language and have a better idea of just how the computer really works.

To start, type *call −151*. As you may already know, the Basic command *call* is a form of goto. They're the same in that both jump to a line number and then begin executing the program they find there. The difference between the two commands is that goto goes to a Basic line, while a call goes to a memory location and then executes the machine language program that begins there.

As you probably also know, there are 65,536 bytes or memory locations in the chips inside your Apple. Because the old and virtually obsolete Integer Basic couldn't handle numbers higher than half that (32,767), it became customary to refer to numbers between 32,767 and 65,536 as the numbers you actually wanted minus 65,536. Thus 65,385 becomes −151. Although Applesoft doesn't have the same limitation and can find a location using either the actual number or its negative complement, using the latter makes for less typing and has been retained as an option.

In both cases, what you get is an asterisk prompt instead of a bracket. This signifies that you've entered the Monitor, that mysterious land where you can read your Apple's mind byte by byte. In the Monitor, you can even make your Apple change its mind—directly and without laborious peeks and pokes.

From the Monitor type *0G* (that's zero G). This should put you back in Applesoft. If it doesn't, type *3D0G*.

The Monitor uses hexadecimal numbers whose digits go from zero through F; that leaves the letter G free to be used as a command. When you're in the Monitor, G means goto, and what we've done in this instance is to tell your machine to go to the beginning of its memory (location zero), which contains the Applesoft command to switch back to Basic. If something's been changed there so 0G doesn't work, try location $3D0 (976), which normally contains the corresponding DOS command. Because this location is in a part of memory that you and your programs have access to, it might no longer contain this command. In that case you'll need to hit reset or even the off/on switch.

In fact, if you've run a commercially produced program, 3D0G probably won't return you to Basic because the 256 bytes from $300 to $400 (768 to 1024) are a favorite place to tuck short machine language programs so they'll be out of the way of those accumulating Basic strings.

So, call −151 is the Basic command to jump into the Monitor while 0G and 3D0G are the Monitor commands for jumping back into Basic.

Go back into the Monitor again. Don't worry about the fact that all numbers here are in hex. Just follow the instructions.

Type *9500.950F*. When you're in the Monitor, typing a location number is the command to list its contents. (You already know what happens in Basic if you try to list a line by typing just its number, don't you?) For a range, type the beginning number and the ending number with a period between them. You'll see two lines, one beginning with 9500 and the other with 9508. The first line contains the two-digit numbers in memory locations $9500 through $9507 and the second the numbers in locations $9508 through $950F. If you've just booted your system, these will probably alternate between 00 and FF, the lowest and highest numbers that can be represented by a single eight-bit byte (and the initial numbers in RAM on power-up).

Let's write a very short machine language program. Begin with *9500:*. The colon after the location number is the instruction to treat the numbers that follow as a new program to be stored in memory beginning with that location. Type:

```
9500:20 DD FB 60
```

Check your typing by listing the line using the command *9500.9503*. The eight characters form a complete program. In machine language $20 acts like gosub in Applesoft, although it is usually referred to as *jump to subroutine* and abbreviated *JSR*, while $60 is similar to the command return. The memory location $FBDD (64477) contains a short program to ring your Apple's bell once.

All locations entered in hex via the Monitor require two bytes or four hex digits. These must be entered with the second two in correct order followed by the first two in correct order. Thus, location $1234 would be entered into machine language as 34 12 (not completely backward as in 43 21). It's done this way so that the higher byte will contain the higher places of the number. The program will read the lower byte first, since machine language programs, like those in Applesoft, execute the instructions in the lowest numbered locations first and then progress to the higher ones. In summary, then, our four-byte machine language program goes to location $FBDD and executes the subroutine it finds there.

To see if it works, type *9500G*. You should hear the bell, but, this time at least, it shouldn't be accompanied by an error message. Now return to Basic. Type in this little program; in line 30, the number 38144 is the decimal equivalent of $9500.

```
10  INPUT A$
20  IF A$ = "N" GOTO 99
30  CALL 38144
40  GOTO 10
99  PRINT "END": END
```

Each time you get the input prompt, type in a letter or a number. Do this ten times or so, and, when you're satisfied, enter N to end the program. Whenever the response is not N, this Basic program calls your machine language subroutine located at $9500, which then calls the machine language subroutine at $FBDD. That routine rings the bell and returns.

turns to your machine language program, which returns to your Basic program when it reaches the command $60.

Let's move your machine language program. If DOS isn't in, you'll need to boot it now and retype your Applesoft program. Enter the Monitor and type *95F0:20 DD FB 60*. Return to Basic and change line 30 in your Basic program to *call 38384*, the decimal equivalent of $95F0. This puts the program almost at the top of usable memory. Although the user-available memory goes from location 0 to 49151 ($BFFF), DOS takes up all the space from there down through 38400 ($9600). Your four-byte program then runs from 38384 to 38387 ($95F0 to $95F3), leaving only twelve bytes empty between the end of your work and the beginning of DOS.

Now return to Basic and run your program with the new Monitor location in line 30. First, enter the letter A three times and then stop with N to make sure the program doesn't have any bugs. Run it again, slowly entering the letter A over and over again. What happens the thirteenth time you enter A? The system hangs, right?

Let's find out what happened. Hit reset, enter the Monitor, and list locations $95F0 to $9600 by typing *95F0.9600*. The byte at $9600, the last location, is the first instruction in DOS. Leave it alone; it bytes. Working backward from $95FF, you should see 41, the ASCII code number for the letter A, once for every time you entered A. Remember that Applesoft doesn't replace old input strings with new ones; it merely starts a new string in the next location—working downward.

Take a look now at your machine language program. The first three bytes are okay, but a usurping 41 has taken the place of your return code 60. Strings always trail downward. When they hit your Basic program, which builds up from memory location 2048 ($800), Applesoft will appear to go into a coma. It has begun automatic garbage collection.

Unfortunately, strings creeping down over a machine language program simply overwhelm it and take over its locations. If the machine language program is one that isn't needed again, that's okay. If it is, too bad; it's been clobbered. In this case your twelfth A took the place of your return code 60, and the instruction to return to your Basic program after executing the machine language subroutine was lost. So the system hung. If you had just painted a hi-res picture (using locations 8192 through 16383 or 16384 through 24575) and the strings dangled that far down in memory, they'd mess it up, too.

There are three primary ways of preventing accumulated strings from interfering with machine language programs and hi-res pictures. Please don't protest that, since you're a beginner and not yet writing in machine language, you don't need to know about them. After all, a number of useful packages written in machine language can be used right now to enhance your Basic programs. These include graphics programs that allow your own programs to draw hi-res pictures, music synthesizers that let your programs play music through the Apple's speaker or through your own stereo, and programs to let you hear the sound of lasers zapping and aliens frying.

We've mentioned the first way of preventing accumulated strings from interfering with your other programs. This method involves finding an empty hole that nothing else will use. About the only one of any size is page 3, memory locations $300 through $3FF. But a number of commercial packages place their short routines here on boot.

The second method is to initiate garbage collection routinely without waiting for Applesoft to realize the need for it. This is done with the command *X = FRE(0)*. This command does two things. If you print X now, it will tell you the number of free bytes between the top line of Basic and the bottom string; unfortunately, it counts places occupied by hi-res graphics and machine language programs as also being free because Basic itself isn't using them.

The other thing the command X = FRE(0) does is to initiate early garbage collection. Go back into the Monitor and type *95F0.9600* to list that range. Repair your machine language program and zero out the dangling strings with:

```
95F0:  20 DD FB 60 00 00 00 00 00 00 00 00 00 00 00 00
```

To proofread what you've just done and to make sure that you didn't change location $9600, list it by typing *95F0.9600*. Then return to your Basic program and insert the line:

```
35 X = FRE(0)
```

Run the program again, entering A at least a dozen times. It won't hang. Hit N to end it and then go back into the Monitor and look at locations $9F50 to $9600. Most of your zeros are still there because your Basic program now starts garbage collection after every string, which effectively starts each string from the top of available memory rather than from the end of the last string.

We spoke earlier of erasing old strings but, as you can see in the Monitor, garbage collection doesn't necessarily erase the unused strings. In this case, all it had to do was reset the input string pointer to the top of available memory so that new strings could overwrite old ones as needed.

And, finally, the third way of avoiding strangulation is to redefine available memory. This is done with the Basic command *himem:n*, where *n* must be a decimal number. Remove line 35 from your Basic program and add the line:

```
5 HIMEM:38144
```

This command starts the string formation at 38143 and protects any machine language programs stored in locations 38144 and beyond. Run your program, entering A more than twelve times before you quit with an N, and then go into the Monitor. Examine locations $95E0 to $9600. There is your machine language program from $95F0 to $95F4. The new strings, ASCII code $41 for A and $4E for N, dangle from $95EF downward.

On booting, DOS sets himen at 38400 ($9600), which is just low enough to protect itself but not low enough to protect anything else. If you change this location, make sure you always know where you've put it or else have all your programs relocate it as needed. Himem is *not* reset by typing *new* or *run*. Nor does it help to hit reset or to remove the program line that set himem after you've run the program. If himem is not where you want it for the next program, you've got to move it yourself unless you know that the next program will.

One last word of warning. Don't set himem above the bottom of DOS. Your garbage can garble your disk operating system just as effectively as it can zap the machine language programs you create. ■

NEW APPLE UTILITIES FROM BEAGLE BROS

And a Game Disk Too!

NEW!

Alpha Plot

Hi-Res Graphics/Text Utility
by Bert Kersey & Jack Cassidy

Here are a few of Alpha Plot's useful graphics features. Compare with other graphic utilities at any price—

HI-RES DRAWING: Create hi-res pictures and charts with text, on both pages, all **appendable to your programs**. Optional Xdraw cursor (see lines before drawing). Mix colors & Reverse (background opposite). Circles, Boxes, Ellipses, filled or outlined. **Compress Hi-Res to 1/3 disk space.** Relocate any portion of an image anywhere on page. Superimpose two & convert hi-res to lo-res for colorful abstracts!

HI-RES TEXT: Beautiful upper/lower case with descenders (no hardware required). Color and reverse characters positionable anywhere (no vtab/htab limitations). Professional-looking **proportional spacing** and adjustable character height and letter spacing. Sideways typing for graphs too!

$39.50 ☐ Unprotected disk (48K min.) ☐ Beagle Bros Apple Tip Book #4 ☐ Peeks & Pokes Chart

Frame-Up

Graphics Display Utility
by Tom Weishaar

Frame-Up is a very-high-speed Apple "slide projector" utility that lets you create professional-looking displays of intermixed hi-res, lo-res and text pages **on any Apple.** Frame-Up is very easy-to-use and above-all **FAST**, allowing you to load hi-res pictures, for example, in **8 1/4-seconds**; that's three-times faster than normal! Paddles or keyboard are used to change images in forward or reverse order, skipping pages if you want. OR, presentations may be left unattended, with **each page individually timed** to appear and remain on the screen from 3 to 99 seconds, as you choose.

Frame-Up includes a sophisticated black and white **text screen editor** that lets you create text "slides" as part of your show. You can even add type "live" on the screen during your presentations. Up to 17 hi-res or 136 lo-res/text pages may be stored per disk. One or two drives are supported. The order and timing of your graphics and text images may be easily (and instantly!) arranged and rearranged. Frame-Up includes **a display module which may be copied** and distributed to your associates so they can run your display, as you designed it, on their Apple or ANY Apple.

Frame-Up is ideal for store displays, presentations to the boss, club programs, trade show booths, product demos, promotions, seminars, conventions, classes, and so on.

$29.50 Machine language. Unprotected. 48K minimum. Peek/Poke Chart included.

NEW!

Beagle Bag!

12 Games on One Big Disk
by Bert Kersey

Twelve great games from our classic Beagle Bros collection—TextTrain, Slippery Digits, Wowzo, Magic Pack, Bumzword... Almost all of our "Game Pack" games have been updated and re-released on this jam-packed unprotected disk! **ALSO INCLUDED** is our "Beagle Mania" greeting program (description under "Typefaces" disk on this page).

Compare Beagle Bag with any 1-game locked-up disk on the market today!

All 12 games are a blast, the price is right, the instructions are crystal clear, AND the disk is copyable! You can even list the programs to see what makes them tick!

$29.50 Unprotected. Paddles NOT required. Beagle Mania works with all normal Peek/Poke disks. Includes Peeks/Pokes Chart.

Apple Mechanic

Shape Writer/Byte-Zap Utility
by Bert Kersey

Another best-selling multiple-utility disk—Nine useful, listable, copyable and customizable programs—

SHAPE EDITOR: Put professional hi-res animation in your programs. Keyboard-draw any shape and let your Apple write a shape table and store it on disk. Design large and small **custom typefaces** too, with special characters. 6 fonts on the disk. LISTable demos show how to use shape tables to animate games, graphic displays, and attractive Charts & Graphs. A valuable time-saving utility/learning tool.

BYTE ZAP: A MUST utility. Rewrite any byte on a disk by loading a sector onto the screen for inspection. **Hex/Dec/Ascii** display optional. Examine bytes via cursor control, enter hex, dec or ascii to change. Create illegal filenames, restore deleted files, change greeting program names, repair/protect disks, change DOS, examine program files. Clear illustrated instructions show how disk data is stored and how to access it. Very educational.

MORE: A disk PACKED with useful music, text and hi-res tricks **for use in your programs.** A great demo-writer program, useful hi-res utilities and educational, entertaining documentation.

$29.50 ☐ Unprotected disk (48K min.) ☐ Beagle Bros Tip Book #5 ☐ Peeks & Pokes Chart

NEW!

Typefaces

for Apple Mechanic

Here are more hi-res fonts for Apple Mechanic's Xtyper and Hi-Writer programs—26 of them at last count, both large and small, all **proportionally-spaced** and positionable anywhere on either hi-res screen. Most are **full 96-character fonts** many with special graphic characters. Each character ("!" to "~") of every font ("Ace" to "Zoo-loo") is, of course, editable with Apple Mechanic's Font Editor.

BONUS: Here's BEAGLE-MENU! A unique greeting program that displays **only the catalog file names you want** on the screen (for example, only locked-Applesoft files, or only Binary files) for one-key cursor selection. Just hit Return to Run, Brun or Exec the program at the cursor. Many other features—Space-on Disk, Load/Bload option, forward and backward catalog "scrolling" for easy file location, and optional sector number elimination. PLUS the ability to **swap file names** in your catalog!

$20.00 Unprotected Beagle Bros' Apple Mechanic disk is required to utilize the type fonts. Beagle Menu works with all normal DOS 3.3 disks.

Beagle Bros MICRO SOFTWARE

4315 Sierra Vista, San Diego, Ca 92103
714-296-6400

If you don't find our products at your Apple Dealer, tell him to phone Beagle Bros 714-296-6400. **OR** his favorite software distributor.

NEW!

Flex Text

70-Column Text Utility
by Mark Simonsen

Flex Text is a unique utility that lets you print variable-width text on Apple's hi-res screens in normal 40-column format, 80-column expanded or 66- and **70-column condensed** characters. Character widths may be mixed as you like for emphasis. Flex Text understands **normal Applesoft** Basic commands, including Home, Inverse, Normal. Vtab 1-24 and **Htab 1 through 70**. It also supports text window pokes and scrolling, so you can program normally, but with the **ability to add text to graphics**, or graphics to text! You can even run your existing programs using these features!

FLEX TEXT IS COMPATIBLE WITH DOS TOOL KIT® FONTS

Enter up to NINE font names in Flex Text's boot-up program for easy ctrl-command access. **Upper & lower case** in any character-width without hardware. All characters redefinable with a **text character editor**. Toggle between "normal" text screen and both hi-res pages. Compatible with Neil Konzen's Program Line Editor™ and GPLE®.

$29.50 Machine language. Unprotected. 48K min. Peek/Poke chart included. Condensed characters require 80-column monitor (instead of a tv) for best results.


```
10 REM HI-RES NUMBER GENERATOR
20 SIZE=5: SCALE=SIZE: REM NUMBER-HEIGHT
30 HGR: HOME: POKE 232,0: POKE 233,3: ROT=0
40 FOR A=7*8 TO 830: READ B: POKE A,B: NEXT A
50 N=N+1: N$=STR$(N): X=99: Y=0
60 FOR A=1 TO LEN(N$): HCOLOR=0: DRAW 8 AT X,Y: HCOLOR=3: DRAW VAL(MID$(N$,A,1)) AT X,Y: X=X+SIZE+SIZE: NEXT: A: GOTO 50
70 DATA 20,0,24,0,27,0,31,0,35,0,39,0,44,0,49,0,52,0,57,0,53,62,38,0,49,38,0,53,55,61,0,53,23,37,0
80 DATA 46,38,52,0,61,46,62,5,0,61,54,37,7,0,53,38,0,54,37,60,46,0,53,39,53,62,5,0
```

DOS Boss

Disk Command Editor
by Bert Kersey & Jack Cassidy

A classic Apple utility you will ENJOY! Rename DOS commands ("Catalog" can be "Cat", etc.). PROTECT PROGRAMS; any unauthorized save- attempt produces a "Not Copyable" message. Also: **List-prevention** and 1-key program-run from catalog. Custom catalogs: Change Disk Volume message to your title; Omit or alter file codes. Rewrite error messages: "Syntax Error" can be renamed "Oops!!" or anything you want! **Two books included**— Fascinating documentation and hours of good Apple reading!

Dos Boss's change features may be appended to your programs so that anyone using your disks (booted or not) formats DOS as YOU designed it.

$24.00 ☐ Unprotected disk (32K/48K) ☐ The Dos Boss Book ☐ Beagle Bros Apple Tip Book #2 ☐ Peeks & Pokes Chart

NEW!

ProntoDOS

High-Speed DOS Utility
by Tom Weishaar

ProntoDos is FAST, saving you time where it counts the most. This comparison with normal Apple-DOS speaks for itself—

FUNCTION	PRONTO	NORMAL
BLOAD HI-RES IMAGE	3 sec.	10 sec.
BSAVE HI-RES IMAGE	3 sec.	12 sec.
LOAD 60-SECTOR PROGRAM	4 sec.	16 sec.
SAVE 60-SECTOR PROGRAM	9 sec.	24 sec.
BLOAD INT/LANGUAGE CARD	4 sec.	13 sec.
TEXT FILES		(in dump)

MORE DISK SPACE: Booting ProntoDos frees up 18 extra sectors of Disk Space, almost a full track. To speed up your Apple, just boot ProntoDos or any disk you have updated with ProntoDos, and you're in business. You can even create new ProntoDos disks with Apple's normal INIT command. ProntoDos is compatible with ALL commands and performs normally (but FAST) with almost ALL programs.

$29.50 Machine Language. Unprotected. Peeks & Pokes chart included. All normal 3.3 disks are updatable.

Tip Disk#1

100 Tip Book Tips on Disk
by Bert Kersey

100 programs from Beagle Bros' Tip Books 1, 2, 3 and 4— Fascinating tricks to make your Apple do things it's never done before! All 100 programs are listable, copyable and changeable, and each teaches another fascinating Apple programming technique.
Two different charts are included.

$20.00 ☐ Unprotected (32K/48K) ☐ Peeks & Pokes Chart ☐ Apple II Command Chart

Beagle Bros MICRO SOFTWARE

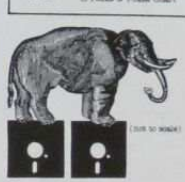
4315 Sierra Vista, San Diego, Ca 92103
714-296-6400

Utility City

21 Utilities on One Disk
by Bert Kersey

A best-seller since it hit the market, and a MUST for your program-development library. Take a look at the features—
List Formatter makes properly-spaced & indented listings with page breaks, each statement on new line, if-thens and loops called out, a great de-bugger. **Multi-Column Catalog** in any page-width to any printer or CNT. Auto-post Run Number and last-used Date in programs. Put INVISIBLE working commands in your listings. Access program lines in memory for repair & illegal alteration. Alphabetize & show info on disk. Run any program while another stays intact. Renumber to 65536. Save inverse, trick and **invisible file names**. Convert dec to hex & binary, or Integer to FP. Append programs. Dump text screen to printer.
21 LISTABLE UTILITIES TOTAL!

$29.50 ☐ Unprotected disk (48K min.) ☐ Beagle Bros Apple Tip Book #3 ☐ Peeks & Pokes Chart



GOTO Your Apple Dealer.

Most dealers carry our software. If yours doesn't, he can have our disks in his store for you **within 2-3 days** by calling us or his favorite software distributor.

Or Order by Phone:

24 Hour **Toll-Free** Order Desk
Visa/MasterCard/COD* orders, call
Nationwide **1-800-854-2003** ext.327
California **1-800-522-1800** ext.327
Alaska/Hawaii **1-800-854-2822** ext.327
(ORDERS ONLY. PLEASE) *COD add $3.00

OR ORDER BY MAIL—

RUSH! the disk packages checked below plus the free 11x17 Peeks & Pokes Chart

☐ Alpha Plot	$39.50	☐ Frame-Up	$29.50
☐ A.Mechanic	$29.50	☐ ProntoDos	$29.50
☐ Beagle Bag	$29.50	☐ Tip Disk#1	$20.00
☐ DOS Boss	$24.00	☐ Typefaces	$20.00
☐ Flex Text	$29.50	☐ Utility City	$29.50

(Add $1.50 Shipping, any size order. California, add 6% tax.)

NAME: _____
ADDR: _____
CITY: _____
ZIP# _____

VISA MASTERCARD Include Card Number & Expiration Date

Mail VISA/check, money order or Visa/MasterCard numbers to:

BEAGLE BROS, DEPT. S
4315 SIERRA VISTA
SAN DIEGO, CA 92103

All Orders Shipped Immediately.

Please add $4.00 for shipping outside North America. COD orders add $3.00. California residents, add 6%.

THE THIRD BASIC

by Taylor Pohlman

Exploring Business Basic, Part 16

This month's column features our most ambitious program yet (at least in size). It's a database manager that shows how binary tree data structures can be used for data access. If you have been following along, you'll recognize several old and many new techniques used in this example program. Even though it's a pretty long program, it's only the bare bones of a database program, giving you plenty of opportunities to add your own wrinkles.

Remembrance of Things Past. If you haven't read last month's article, you really should become familiar with the topics we discussed therein before perusing this month's missive. Remember that in a binary tree structure each value has associated with it a pointer to values less than the value (a "left" pointer) and a pointer to values greater than the value (a "right" pointer). Because there is no way of predicting the sequence in which values will be added to the tree, each individual branch may or may not contain both left and right pointers. For example, consider how you would arrange the following list into a binary tree structure:

1 Johnson	5 Williams
2 Baker	6 Douglas
3 Phillips	7 Connor
4 Jones	

The tree would look like figure 1.

As a table of pointers, the same data would look like table 1.

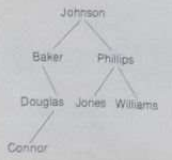


Figure 1

		Left Pointer	Right Pointer
1	Johnson	2	3
2	Baker	0	6
3	Phillips	4	5
4	Jones	0	0
5	Williams	0	0
6	Douglas	7	0
7	Connor	0	0

Table 1

The binary tree (usually called b-tree) structure offers several advantages over other sorting methods. As you can see, it doesn't take many tests to find out where a value goes. This is true even if the tree is very large, as long as it's reasonably well balanced (with most branches of roughly even length). Also, as we saw last time, it is easy to construct a sorted list from the b-tree structure, even without reference to the original values. Another advantage is that we can keep as many b-trees around as memory will allow. The *hash* method we used some months back to make a database had the disadvantage that only one field could be a *key* field. Since we can keep multiple b-tree structures around, it is possible to have many different keys in the same file. The b-tree has some problems as an access method, however, and we'll cover those as we get into the program.

Bird's-Eye View of Our Tree. Our b-tree database program uses two files. One contains the actual data records, implemented as a single string in a random access textfile. The single string textfile approach was chosen to keep things simple. You could easily change it by modifying the file read and write routines. Associated with the main file is a key file, with its name formed by appending .key to the end of the main file name. Information about the structure of the database is kept in the key file, along with the actual key values and b-tree pointers. If at program start-up you request a file that doesn't yet exist, the program allows you to define it, including the names of the fields, which fields are to be key values, and where in the output record the field is to go. If the key file does exist, then all the required information is read from it, and the main file is opened for access.

One of the interesting things about the program is that it allows you to specify whether keys must be unique or not and checks when you enter a key value to be sure. The program requires at least one key value to be unique and uses the first such value as the key for deleting records. Examples of unique keys are social security numbers or employee numbers. With some programming effort you could change this to allow choosing the record to delete from several different fields, but the technique shown is simpler and safer. Other general capabilities include getting simple lists based on key or non-key values. The program automatically knows when your search field is a key and uses the fast key look-up routine. For nonkey fields, the program scans the whole file looking for a match.

One last thing before we get started. The program as it stands keeps all key information in memory. This makes it fast, but it limits the number of records that the program can handle. Fortunately, the Apple III has lots of memory, but even the biggest Apple III can run out of memory if you have lots of records and keys. With a little effort, and a tradeoff of size versus performance, part of the key arrays can be kept on disk in a random access file. You should still keep as much of the first part of the key array in memory as possible, to reduce the number of disk accesses.

Now for the program:

```
15 DIM item$(99),ib%(99),le%(99),
   ik%(99),id%(99)
20 DIM dup%(1000)
25 z=0:o1=1:o2=2:bell$=CHR$(7)+
   CHR$(7):b20$="
   b20$ is 20 spaces
28 blank$=b20$+b20$+b20$+b20$+b20
29 blank$=blank$+blank$+MID$
   (blank$,1,55)
30 TEXT:HOME:PRINT"Database program
   with BSAM"
40 PRINT:INPUT"Name of file to access ";f
45 IF a$="" THEN 400
50 IF LEN(a$)>11 THEN PRINT"File name
   must have a maximum of 11
   characters":GOTO 40
60 file$=a$
70 GOSUB 1000
80 IF errorcode=1 THEN 40
85 IF errorcode=2 THEN RUN
```

The opening of the program lines do some

tialization and request the file name for access. Then a gosub to line 1000 does the file initialization or creation as required. Explanations of the arrays declared in lines 15 and 20 will be handled when the initialization routine is covered.

```
90 TEXT:HOME:PRINT"Database: ";file$
95 WINDOW 1,3 TO 80,24:HOME
100 PRINT:PRINT"Functions:"
110 PRINT:PRINT"   1 - Add a Record"
120 PRINT"   2 - Delete a Record"
130 PRINT"   3 - Find a Record"
140 PRINT"   4 - List all Records"
200 PRINT:PRINT"                Your choice: ";
202 no.error=0
205 INPUT"";a$:a=CONV(a$)
210 ON a+1 GOSUB 400,2000,3000,5000,
    8000
212 IF no.error THEN 90
215 msg$="Choose a value from 1 to 4 or
    press RETURN to exit":GOSUB
    900:PRINT CHR$(12);:GOTO 100
400 TEXT:PRINT:PRINT"End of program."
410 GOSUB 1500
420 CLOSE:END
```

This next section of the program displays the main menu once the file has been initialized. Note that the list of functions was kept simple. It is easy to add additional routines to the menu by modifying a few lines. Note also the *window* statement. This method will be used extensively to keep header information on the screen during times when the display normally scrolls upward. The gosub in line 410 (gosub 1500) references the routine that saves the changes made to the file during a program run. This routine will be covered in more detail later.

```
500 FOR key=0 TO num.key-1
510 testrec=-1
540 IF sort$(key,rec)>=sort$(key,testrec)
    THEN 570
550 IF sortpl%(key,testrec) THEN testrec=
    ABS(sortpl%(key,testrec)):GOTO 540
560 sortpl%(key,testrec)=rec:GOTO 590
570 IF sortpr%(key,testrec) THEN testrec
    =ABS(sortpr%(key,testrec)):GOTO 540
580 sortpr%(key,testrec)=rec
590 NEXT
595 RETURN
```

The routine in lines 500 through 595 should look familiar to those of you who were tuned in last time. Last month's version handled only one key, while this one uses a two-dimensional sort value array and pointer arrays to update multiple keys. Another change will probably be of interest to you sharp-eyed routine watchers. Note that lines 550 and 570 assign the absolute value (ABS) of the pointer array to the variable testrec. This precaution was taken because, as we shall see later, a negative pointer is used as an indication that the given value has been deleted, even though the value itself must remain to complete the b-tree.

```
600 testrec=1:dup=0:errorcode=0:del.rec=
    (sortpl%(key,0)<0)
610 IF key$>=sort$(key,testrec) THEN 630
615 del.rec=(sortpl%(key,testrec)<0)
620 IF sortpl%(key,testrec) THEN testrec=
    ABS(sortpl%(key,testrec)):GOTO 610
625 RETURN
630 IF key$<>sort$(key,testrec) THEN 640
635 IF NOT del.rec THEN dup=dup+1:
    dup%(dup)=testrec
640 del.rec=(sortpr%(key,testrec)<0)
645 IF sortpr%(key,testrec) THEN testrec=
    ABS(sortpr%(key,testrec)):GOTO 610
650 RETURN
```

The routine from 600 to 650 is a variation on the binary tree search routine in the previous example, except that its sole function is to assemble a list of record numbers whose key values match the variable key$. These are stored in the array dup%. Note that the variable del.rec is used as a flag to ignore a record if its pointer is negative (deleted). This routine is used by the program's Find function to scan the file for matching key values and return all records that apply.

```
700 testrec=1:errorcode=0:del.rec=
    (sortpl%(key,0)<0)
710 IF key$>=sort$(key,testrec) THEN 730
715 del.rec=(sortpl%(key,testrec)<0)
720 IF sortpl%(key,testrec) THEN testrec=
    ABS(sortpl%(key,testrec)):GOTO
    710:ELSE:RETURN
730 IF key$=sort$(key,testrec) AND NOT
    del.rec THEN errorcode=1:RETURN
735 del.rec=(sortpr%(key,testrec)<0)
740 IF sortpr%(key,testrec) THEN testrec=
    ABS(sortpr%(key,testrec)):GOTO
    710:ELSE:RETURN
```

The routine from lines 700 to 740 is the most specialized of all. Its sole function in life is to check to see if a given key value has a duplicate value already in the file. It is used to ensure that the keys marked "no duplicates" are in fact unique.

```
900 VPOS=21:HPOS=1:INVERSE:PRINT
    msg$;CHR$(31);:IF beep THEN PRINT
    bell$;
905 VPOS=line:HPOS=col:NORMAL:
    RETURN
910 VPOS=21:HPOS=1:NORMAL:PRINT
    CHR$(31);:VPOS=line:HPOS=
    col:RETURN
930 FOR i=1 TO delay*60:PRINT
    CHR$(22);:NEXT:RETURN
```

The short routines in lines 900 through 930 are utilities that are used throughout the program. Lines 900 and 905 put a message in the message window and restore the cursor. Line 910 clears the message window, and line 930 creates a delay by printing screen sync characters (one-sixtieth of a second each). The length of the pause in seconds is determined by the value of the variable delay. Remember that delays on the Apple III should be programmed like this, rather than with for-next loops only. Because the Apple III is interrupt-driven, it is really impossible to tell exactly how long a given routine will take to execute.

Now for the fun stuff:

```
1000 REM initialize file
1005 errorcode=0
1010 OPEN#2,file$+".key"
1020 READ#2,0:IF TYP(2)<>1 THEN 1100
1030 READ#2,num.rec,num.key,pl.rec,
     pr.rec,sort.rec,num.item,item.rec,
     rec.len,tot.rec
1032 OPEN#1,file$,rec.len
1033 DIM sortpl%(num.key-1,1000),
     sortpr%(num.key-1,1000),
     sort$(num.key-1,1000)
1035 IF num.rec=0 THEN 1092
1037 IF TYP(1)<>8 THEN PRINT"Your file
     has been damaged.";bell$:errorcode=
     2:IF TYP(1)=0 THEN DELETE
     file$:RETURN:ELSE:RETURN
1040 READ#2,pl.rec
1050 FOR i=1 TO num.rec:FOR j=0 TO
     num.key-1:READ#2;sortpl%(j,i):
     NEXT:NEXT
1060 READ#2,pr.rec
1070 FOR i=1 TO num.rec:FOR j=0 TO
     num.key-1:READ#2;sortpr%(j,i):
     NEXT:NEXT
1080 READ#2,sort.rec
1090 FOR i=1 TO num.rec:FOR j=0 TO
     num.key-1:READ#2;sort$(j,i):
     NEXT:NEXT
1092 READ#2,item.rec
1095 FOR i=1 TO num.item:READ#2;item$
     (i),ib%(i),ie%(i),ik%(i),id%(i):NEXT
1097 RETURN
```

You guessed it, the initialization routine! The first step is to open the key file, formed by adding .key to the file name in line 1010. If it exists and contains data, then initialization proceeds. If it doesn't exist (this is checked for in line 1020), then the program jumps to line 1100, where the new file is created. Line 1030 reads a number of important variables from the key file. Most of them are self-explanatory.

Variables ending in .rec point to the beginning record numbers in the key file where the associated arrays are to be found. Thus pl.rec is the record number where the sortpl% array is to be found and item.rec points to where the lists of data item definitions start. The exception to this rule is tot.rec, which is simply the total of valid (undeleted) records in the file. This is different from num.rec, which is the total number of physical records.

After reading these variables from the key file, the main file is opened in line 1032 using the record length that was read from the key file. Line 1033 then dimensions the appropriate arrays according to the number of keys defined in the key file. Note that this is an extremely powerful capability not found in many versions of Basic. The arrays have been arbitrarily defined to be 1,000 records long. If you have a 128K system and use lots of keys, you may want to reduce this total number.

The rest of the routine determines whether it is necessary to read in the key data and, if it is, does so in lines 1050 through 1095. We'll cover the meaning of the arrays in line 1095 in a moment when we talk about the creation routine:

```
1100 PRINT"The file ";file$;" is not a
     database file"
1110 INPUT"Do you wish to make it a
     database file? ";a$
1120 a$=MID$(a$,1,1):IF INSTR("Yy",a$)
     THEN 1200
1140 DELETE file$+
     ".key":errorcode=1:RETURN
```

The section from lines 1100 through 1140 prepares for the creation of a new file. Note the use of INSTR in line 1120. It substitutes for *a$="Y" or a$="y" then 1200*. Another way to write line 1120 is *if INSTR("Yy",MID$(a$,1,1)) then 1200*.

```
1200 HOME:PRINT"Database setup -
     Record definition:"
```

```
1210 PRINT WINDOW 1,3 TO 80,24:HOME
1230 FOR i=1 TO 99
1235 IF VPOS=22 THEN PRINT:VPOS=21
1240 PRINT USING "245";"line= ";VPOS
1245 IMAGE:"Item ";2#;" -- Name: "
1250 INPUT"";item$(i):IF item$(i)="" THEN
     1370
1252 HPOS=17:VPOS=line:PRINT
     MID$(item$(i),1,16);:PRINT CHR$(31);
1255 IF LEN(item$(i))>16 THEN PRINT
     bell$;:HPOS=1:GOTO 1240
1260 VPOS=line:HPOS=34
1270 INPUT"begin: ";a$
1280 ib%(i)=CONV%(a$)
1285 IF ib%(i)<1 THEN PRINT bell$;:GOTO
     1260
1290 VPOS=line:HPOS=41:PRINT
     USING"2#";ib%(i);
1300 INPUT" end: ";a$
1310 ie%(i)=CONV%(a$)
1315 IF ie%(i)<ib%(i) THEN PRINT
     bell$;:GOTO 1290
1320 VPOS=line:HPOS=50:PRINT
     USING"2#";ie%(i);
1330 INPUT" Key? ";a$
1340 ik%(i)=(INSTR("Yy",MID$(a$,1,1))>0)
1350 VPOS=line:HPOS=59:PRINT MID$
     ("NY",ik%(i)+1,1);:PRINT CHR$(31);
1351 IF NOT ik%(i) THEN
     id%(i)=1:PRINT:GOTO 1360
1352 INPUT" Duplicates? ";a$
1355 id%(i)=(INSTR("Yy",MID$(a$,1,1))>0)
1357 VPOS=line:HPOS=74:PRINT MID$
     ("NY",id%(i)+1,1);:PRINT CHR$(31)
1360 NEXT i
```

This rather elaborate input routine prompts for each field name, gets beginning and ending columns, and asks whether the field is to be a key. If the field is a key, the routine asks whether duplicate values are allowed. Note the extensive use of vpos and hpos to facilitate editing, and the use of INSTR and MID$ in lines 1340 through 1357 to save time and program size. As can be seen by examination, item$ holds the individual field names, ib% and ie% hold the beginning and ending field positions (and thus the maximum field size), and ik% and id% hold the flags for key fields and duplicates allowed. If memory size is a problem, these two arrays could be combined with a minimal amount of programming effort.

```
1365 msg$="Initializing file '"+file$+"'.":
     GOSUB 900
1370 num.item=i-1:num.key=0
1375 FOR i=1 TO num.item:IF ie%(i)>
     rec.len THEN rec.len=ie%(i):NEXT
     ELSE:NEXT
1380 FOR i=1 TO num.item:IF ik%(i) THEN
     num.key=num.key+1:NEXT:ELSE
     NEXT
1385 num.rec=0:pl.rec=100:pr.rec=200
     sort.rec=300:item.rec=10:
     rec.len=rec.len+1
1390 WRITE#2,0;num.rec,num.key,
     pl.rec,pr.rec,sort.rec,num.item,
     item.rec,rec.len,num.rec
1395 WRITE#2,300;0:REM establish end of
     file
1400 READ#2,item.rec
1410 FOR i=1 TO num.item:WRITE#2;
     item$(i),ib%(i),ie%(i),ik%(i),id%(i);:
     NEXT
1420 msg$="File '"+file$+"' is initialized":
     GOSUB 900:delay=2:GOSUB 930
1430 OPEN#1,file$,rec.len
1435 DIM sortpl%(num.key-1,1000),
     sortpr%(num.key-1,1000),
     sort$(num.key-1,1000)
1440 TEXT:HOME:RETURN
```

Lines 1365 through 1440 take the information from the creation routine and write it to the key file, open the main file, and dimension the appropriate arrays for use by the program.

```
1500 WRITE#2,0,num.rec,num.key,
     pl.rec,pr.rec,sort.rec,num.item,
     item.rec,rec.len,tot.rec
1510 IF num.rec=0 THEN 1600
1520 READ#2,pl.rec
1530 FOR i=1 TO num.rec:FOR j=0 TO
     num.key-1:WRITE#2;sortpl%(j,i):
     NEXT:NEXT
1540 READ#2,pr.rec
1550 FOR i=1 TO num.rec:FOR j=0 TO
     num.key-1:WRITE#2;sortpr%(j,i):
     NEXT:NEXT
1560 READ#2,sort.rec
1570 FOR i=1 TO num.rec:FOR j=0 TO
     num.key-1:WRITE#2;sort$(j,i):
     NEXT:NEXT
1600 PRINT"File '"file$"' updated. There
     are "tot.rec" records in the file."
1610 RETURN
```

Lines 1500 through 1610 do just the opposite, storing away all the current data about the key file onto the appropriate records.

```
2000 TEXT:HOME:PRINT"Add a Record to
     file "file$"."
2010 PRINT
2012 WINDOW 1,3 TO 80,24:HOME
2015 rec=num.rec+1:key=-1:line$=MID$
```

```
      (blank$,1,rec:len-1)
2020  FOR i=1 TO num:item
2022  beep=1:IF ik%(i) THEN key=key+1
2025  field:len=ie%(i)-ib%(i)+1
2035  IF VPOS>20 THEN PRINT:PRINT:
      VPOS=20
2040  PRINT"(";i;") ";item$(i);": ";
2045  line=VPOS:col=HPOS
2050  INPUT"";a$
2060  IF a$="" AND i=1 THEN 2200
2070  IF LEN(a$)>field.len THEN
      msg$="Entry is too long":GOSUB
      900:GOTO 2050
2075  GOSUB 910
2080  IF NOT(ik%(i) AND NOT id%(i)) THEN
      2100
2085  key$=a$:GOSUB 700
2090  IF errorcode THEN msg$="Entry must
      be a unique value in this field":GOSUB
      900:GOTO 2050
2095  GOSUB 910
2100  SUB$(line$,ib%(i),field.len)=a$
2110  IF ik%(i) THEN sort$(key,rec)=a$
2115  PRINT
2120  NEXT i
2130  msg$="Record being added.":beep=
      0:GOSUB 900
2140  PRINT#1,rec;line$
2150  IF rec>1 THEN GOSUB 500
2155  num.rec=rec
2157  tot.rec=tot.rec+1
2160  GOSUB 910
2165  PRINT:PRINT
2170  GOTO 2015
2200  IF VPOS>20 THEN PRINT:PRINT:
      VPOS=20
2202  PRINT"End of Add. ";tot.rec;" records
      now in file ";file$;"."
2205  msg$="Press return to continue: "
      :GOSUB 900:GET a$
2210  no.error=1:TEXT:RETURN
```

This Add routine is long but relatively straightforward. The only real twist occurs in lines 2080 to 2095, in which the check for unique values is performed. Line 2080 checks to see whether or not a given field is supposed to be unique, skipping to line 2100 if it's a non-unique field. If the field is supposed to be unique, a gosub to the routine at 700 is performed and the data is checked against all previous entries.

Notice that in line 2100 the record is built up by using SUB$ to insert fields into the line$ string. After all fields are entered, a gosub to line 500 is performed to add all the keys to the pointer arrays.

```
3000  TEXT:HOME:PRINT"Delete a Record in
      file ";file$;"."
3010  PRINT
3020  WINDOW 1,3 TO 80,24
3025  unique=0
3030  FOR i=1 TO num.item:IF NOT ik%(i)
      OR(ik%(i) AND id%(i)) THEN
      NEXT:ELSE:unique=i
3035  HOME
3040  PRINT:PRINT"Records are deleted by
      using the ";item$(unique);" field"
3050  PRINT:PRINT item$(unique);": ";
3060  INPUT"";a$
3070  field.num=unique:key=-1:FOR i=1
      TO field.num:IF ik%(i) THEN
      key=key+1:NEXT:ELSE:NEXT
3080  IF a$="" THEN 3400
3090  HOME
3100  msg$="Searching for "
      +item$(field.num)+": "+a$:line=
      VPOS:col=HPOS:GOSUB 900
3105  key$=a$
3110  GOSUB 600
3115  IF NOT dup THEN 3550
3120  rec=dup%(1)
3125  GOSUB 5620
3130  IF VPOS>17 THEN PRINT:PRINT:
      PRINT:PRINT:PRINT:VPOS=17
3135  PRINT:PRINT"The record is:"
3140  GOSUB 5650
3142  IF VPOS>19 THEN PRINT:PRINT:
      PRINT:VPOS=19
3145  PRINT:PRINT"Delete? ";
3150  msg$="Type 'Y' to Delete, any other
      key to Retain:":line=VPOS:col=
      HPOS:GOSUB 900
3155  INPUT"";a$
3160  IF NOT INSTR("Yy",MID$(a$,1,1))
      THEN 3300
3165  msg$="Deleting the Record":line=
      VPOS:col=HPOS:GOSUB 900
3170  IF rec=1 THEN FOR i=0 TO
      num.key-1:sortpt%(i,0)=-1:sortpt%
      (i,0)=-1:NEXT:GOTO 3200
3175  FOR i=0 TO num.key-1:FOR j=1 TO
      num.rec
3180  IF sortpt%(i,j)=rec THEN sortpt%
      (i,j)=-rec:GOTO 3195
3185  IF sortpt%(i,j)=rec THEN sortpt%
      (i,j)=-rec:GOTO 3195
3190  NEXT j
3195  NEXT i
3200  GOSUB 3500
3205  msg$="Record Deleted":line=
      VPOS:col=HPOS:GOSUB 900
3210  delay=2:GOSUB 930
3215  tot.rec=tot.rec-1
3220  GOTO 3000
3300  msg$="Record not Deleted.":line=
```

```
    VPOS:col= HPOS:GOSUB 900
3310 delay=2:GOSUB 930
3320 GOTO 3000
3400 no.error=1:RETURN
3500 PRINT#1,rec;" "
3510 RETURN
3550 msg$="Record not found":line=
    VPOS:col= HPOS:GOSUB 900
3560 delay=2:GOSUB 930
3570 GOTO 3000
```

This Delete routine is much tougher technically. Because the b-tree depends on an ordered structure of key values, it's not possible simply to blank out a value in sort$ and zero out the pointers in the arrays. The full solution is more complex than it's worth delving into here (read about "balanced b-trees" and "b-splat" trees in references). To keep things simple, we just negate the pointers and go on. This is done in lines 3170 through 3220.

```
5000 TEXT:HOME:PRINT"Find Records in
    file ";file$;""
5010 PRINT
5020 WINDOW 1,3 TO 80,24
5025 unique=0
5030 FOR i=1 TO num.item:IF NOT ik%(i)
    OR(ik%(i) AND id%(i)) THEN
    NEXT:ELSE:unique=i
5035 HOME
5040 PRINT:PRINT"Functions:"
5050 PRINT:PRINT" 1 -- Search on a single
    field value"
5070 IF unique THEN PRINT" 2 -- Find a
    record using the ""item$(unique)""field"
5080 PRINT:PRINT"          Your Selection: ";
5090 INPUT"";a$
5110 a=CONV(a$)
5120 ON a+1 GOTO 5900,5200,5500
5130 msg="Choose a number from 1 to 2 or
    press RETURN to exit":GOSUB
    900:PRINT CHR$(12);:GOTO 5040
```

This is the start of the rather long Find routine, which gives the option of searching on an individual field or using the first unique field, the same one used by Delete. This second option was put in for convenience, since the same thing can be accomplished with option 1 and a little more effort. An interesting option that could be added would be the ability to search on combinations of fields.

```
5200 HOME
5205 PRINT:PRINT"Search on a single field
    value":PRINT
5210 FOR i=1 TO num.item
5215 IF VPOS>20 THEN PRINT:PRINT:
    VPOS=20
5220 PRINT USING 5225;i,item$(i)
5225 IMAGE "(",2#,") ",2x,16a
5230 NEXT i
5240 PRINT:PRINT"          Your Selection: ";
5250 INPUT"";a$
5260 a=CONV(a$)
5270 IF a=0 THEN HOME:GOTO 5040
5280 IF a<1 OR a>num.item THEN
    msg$="Field number invalid":GOSUB
    900:PRINT CHR$(12);:GOTO 5205
5282 select.all=0
5283 field.len=ie%(a)-ib%(a)+1
5284 field.num=a
5285 PRINT:PRINT"Field value: ";
5286 msg$="Use '=' for all, '>' for all
    nonblank":line=VPOS:col=
    HPOS:GOSUB 900
5287 INPUT"";a$
5288 IF a$="" THEN PRINT CHR$(12);:
    GOTO 5205
5289 IF MID$(a$,1,1)="=" THEN select.
    all=1:ELSE IF MID$(a$,1,1)=">"
    THEN select.all=2
5290 IF LEN(a$)<field.len THEN value$=
    a$+MID$(blank$,1,field.len-LEN
    (a$)):ELSE value$=MID$
    (a$,1,field.len)
5291 rvalue$=a$
5292 HOME
5295 IF ik%(field.num) AND NOT select.all
    THEN 5400
5300 msg$="Scanning the file":line=
    VPOS:col= HPOS:GOSUB 900
5305 rec.found=0
5307 IF tot.rec=0 THEN 5360
5310 FOR rec=1 TO num.rec
5320 GOSUB 5600
5325 IF no.rec THEN 5350
5330 GOSUB 5475
5335 IF VPOS>19 THEN
    PRINT:PRINT:PRINT: VPOS=19
5340 IF select THEN GOSUB 5650
5350 NEXT rec
5355 IF rec.found THEN msg$="No more
    records. Press RETURN to
    continue":GOSUB 900:GET a$:GOTO
    5035
5360 msg$="No records found":GOSUB
    900:delay=2:GOSUB 930:GOTO 5035
```

Lines 5200 through 5360 handle the case of searching on a given field. The options include selecting all records, or all records with non-blank fields. In addition, line 5295 checks to see if the field is a key field and, if it is, jumps to this next routine (lines 5400 through 5450), which does a fast scan of the key file in memory. Notice also that all actual I/O is done through subroutines in lines 5600 and 5650, to facilitate changing file structures with a minimum of effort.

```
5400 key=-1:FOR i=1 TO field.num:IF
    ik%(i) THEN key=key+1:NEXT:
    ELSE:NEXT
5405 key$=rvalue$
5410 msg$="Scanning the Key file":line=
    VPOS:col= HPOS:GOSUB 900
5415 GOSUB 600
5417 IF NOT dup THEN 5360
5420 FOR i=1 TO dup
5425 rec=dup%(i)
5430 GOSUB 5620
5432 IF no.rec THEN 5445
5435 IF VPOS>19 THEN PRINT:PRINT:
    PRINT:VPOS=19
5440 GOSUB 5650
5445 NEXT i
5450 GOTO 5355
```

This is the routine used to scan the key file for a value. Notice that it uses the subroutine at line 600 to pull duplicates of a given value. Then the dup% array is used as the record list.

```
5475 select=0
5480 IF select.all=1 THEN select=1:
    RETURN
5485 IF select.all=2 AND field$>"!" THEN
    select=1:RETURN
5490 IF field$=value$ THEN select=1:
    RETURN
5495 RETURN
```

The routine in lines 5475 through 5495 is used by the search routine to determine whether the field meets the search criteria.

```
5500 field.num=unique:key=-1:FOR i=1
    TO field.num:IF ik%(i) THEN key=
    key+1:NEXT:ELSE:NEXT
5505 PRINT:PRINT"    "item$(field.num)":";
5510 INPUT"";a$
5515 IF a$="" THEN PRINT CHR$(12);:
    GOTO 5205
5520 HOME
5525 msg$="Searching for "+item$
    (field.num)+": "+a$:line= VPOS:col=
    HPOS:GOSUB 900
5530 key$=a$
5535 GOTO 5415
```

Line 5500 sets up the search for the first unique field and then uses the regular key-search routine to complete.

```
5600 INPUT#1,rec;line$
5601 no.rec=0
5602 IF LEN(line$)<rec.len-1 THEN
    no.rec=1:RETURN
5605 field$=MID$(line$,ib%(a),field.len)
5610 RETURN
5620 INPUT#1,rec;line$
5622 no.rec=0
5625 IF LEN(line$)<rec.len-1 THEN
    no.rec=1:RETURN
5630 RETURN
5650 PRINT"("line$")"
5655 rec.found=1:RETURN
5900 no.error=1:RETURN
```

The preceding routines are general purpose. The search routines and other parts of the program use them to perform actual read operations on the files.

And now, at long last, we come to the final routine (at least for this article!).

```
8000 TEXT:HOME:PRINT"List all records in
    file ";file$;""
8005 WINDOW 1,3 TO 80,24:HOME
8007 IF tot.rec=0 THEN 8035
8010 FOR rec=1 TO num.rec
8015 GOSUB 5620
8017 IF no.rec THEN 8030
8020 IF VPOS>19 THEN PRINT:PRINT
    PRINT:VPOS=19
8025 GOSUB 5650
8030 NEXT rec
8035 msg$=CONV$(tot.rec)+" records
    listed. Press RETURN to continue":
    GOSUB 900:GET a$
8040 no.error=1:RETURN
```

Lines 8000 through 8040 provide a quick list of all records using the previously defined read routines.

There—more than anyone wants to know about b-tree access methods in Basic! Pursuing among you will note that a lot has been left to the imagination. For example, what happens when the key array fills up with deleted records? How fast will this method add records when there are lots of records and lots of duplicates? These are real questions, and they are solvable with effort and cleverness.

It was not the intention of this article to give you a working, general-purpose database program. There are plenty of those on the market for the Apple III. Rather, the program has given us a chance to explore programming techniques that may prove very useful in special tasks and should enrich your knowledge of programming in general. It is therefore with mixed eyes that we bid databases a fond, albeit temporary, farewell.

Part 1: Labels, Legends, and Keys

SoftGraph

BY DAVID DURKEE

Since the dawn of history humankind has longed for ways to convey profit and loss ratios at a glance. Early cave drawings found in the south of France are now thought to be primitive attempts at expressing proto-economic trends, but, as the culture that produced them had no concept of numbers greater than three, these early graphs were completely lacking in modern refinements. In modern times, of course, graphs are frequently, even casually, used on a daily basis by sociologists, businessmen, students, professors, and especially politicians. Your Apple, whether you know it or not, is uniquely qualified to help you partake of this age-old ritual. So put on your programming hat and we shall forge together into this uncharted territory.

Our goal for this endeavor is singular, but ambitious. We propose to develop a unified graphing system. Though the system will be composed of several Applesoft programs as well as a few machine language routines, the individual components will act as a unit. One program will accept, store, and edit data from which the others will be able to create Cartesian coordinate charts, which combine line graphs and bar graphs on a two-dimensional grid, and pie charts.

Line and bar graphs allow you graphically to display prices, values, production levels, or other dependent variables as they vary over the range of values of a single independent variable, most commonly time. Pie charts are useful for showing how individual components of a system relate quantitatively to the whole. Both types of charts have an advantage over the data from which they are created. Where a huge array of numbers is often confusing, the charts that can be made from them usually simplify things immensely.

Now, if you're not completely lost so far, you should have no difficulties in the pages, and installments, to come. Compared to the last two paragraphs, the rest of the series will be a breeze.

The first actual graph program won't come until next month; there are foundations to be laid down first. A major stumbling block to creating professional looking graphs on the hi-res screen is labeling them. Using what Apple has given you, you can have four lines of text at the bottom. This is totally inadequate, not only because any graph using Cartesian coordinates has two axes to label, but also because the text and hi-res displays are in different areas of memory, complicating the task of saving and printing out your work.

Hi-Res Writing Resolved. Without going into all the other possible ways of putting text on the hi-res screen, we'll jump right into shape tables. Listing 1 is a nice long program that you'll only run once. Seem like a waste of time? Not so. The program creates a shape table for forty-nine ASCII characters; not all the characters on the Apple keyboard are represented, but it should serve our purposes. Once we've run the program, we'll save the table to disk so we can retrieve it quickly when we need it. Save the Applesoft program too, though. If we have to add any more symbols to the table, that's where we'll be starting.

The shapes are organized in a logical sequence based on their ASCII values. Table 1 shows the characters, their ASCII values and shape num-

Characters	ASCII	Shape Number	Data in Line
space	32	blank left in text	
#$%	35-37	1-3	170-190
( )*+,-./	40-47	4-11	200-270
0	48	same as letter O	
1 through 9	49-57	12-20	280-360
;	61	21	370
?	63	22	380
A through Z	65-90	23-48	390-640
^	94	49	650

Table 1. Characters recognized by HGR printer.

bers, and the data lines they can be found on. Typing in such a large program, especially with so many data statements, you are likely to make a few typographical errors. The table and the program in listing 2 should allow you to pinpoint exactly where any errors are.

```
10  SL = 24576:NB = 480
20  POKE 233, INT (SL / 256): POKE 232,SL - (INT (SL / 256) * 256)
30  FOR CT = 1 TO NB
40  READ DT: POKE SL,DT:SL = SL + 1
50  NEXT CT: END
```

Lines 100 through 160 are data for the shape index.

```
100  DATA 49,0,100,0,113,0,122,0,131,0,136,0,141,0
110  DATA 149,0,154,0,158,0,162,0,165,0,172,0,177,0
120  DATA 184,0,191,0,198,0,205,0,212,0,218,0,228,0
130  DATA 235,0,241,0,248,0,255,0,8,1,16,1,24,1
140  DATA 33,1,40,1,49,1,58,1,65,1,71,1,80,1
150  DATA 86,1,98,1,109,1,117,1,125,1,134,1,143,1
160  DATA 151,1,158,1,166,1,175,1,186,1,198,1,207,1,218,1
```

Punctuation characters.

```
170  DATA 49,54,46,248,63,14,196,33,188,13,13,6,0
180  DATA 57,39,214,14,173,30,230,55,0
190  DATA 27,77,241,30,30,30,77,49,0
200  DATA 23,54,14,6,0
210  DATA 171,54,30,6,0
220  DATA 26,21,21,196,249,18,6,0
230  DATA 50,62,104,6,0
240  DATA 146,30,6,0
250  DATA 145,63,55,0
260  DATA 146,50,0
270  DATA 9,23,23,23,23,6,0
```

The numbers 1 through 9.

```
280  DATA 43,54,54,6,0
290  DATA 43,21,30,30,46,53,0
300  DATA 43,21,30,14,30,55,0
310  DATA 51,46,45,196,182,54,0
320  DATA 57,55,46,14,30,55,0
330  DATA 191,54,14,45,224,55,0
340  DATA 43,53,30,54,6,0
350  DATA 21,30,21,30,63,96,28,12,6,0
```

The program in listing 1 to poke in the shape table is surprisingly simple. The variable SL in line 10 is the location in memory where the table will begin. The 24576 might not seem like a very round number, but it's the same as $6000 in hexadecimal. This location is immediately above

```
360  DATA 191,14,45,48,246,55,0
```

Equal sign and question mark.

```
370  DATA 17,63,183,45,53,0
380  DATA 191,77,30,30,22,6,0
```

The letters A through Z.

```
390  DATA 191,54,102,173,36,52,0
400  DATA 21,23,111,242,63,36,36,53,0
410  DATA 17,28,191,54,14,45,48,0
420  DATA 21,54,30,63,36,36,53,0
430  DATA 57,63,110,58,55,46,45,6,0
440  DATA 57,63,118,57,55,54,0
450  DATA 17,28,191,54,14,45,60,6,0
460  DATA 49,54,38,24,183,35,36,52,0
470  DATA 57,175,54,14,63,6,0
480  DATA 49,54,23,231,6,0
490  DATA 241,30,14,21,223,36,36,6,0
500  DATA 27,54,54,45,53,0
510  DATA 137,146,36,36,23,23,199,57,48,54,54,0
520  DATA 9,54,54,199,57,56,56,48,54,54,0
530  DATA 21,54,30,63,32,100,6,0
540  DATA 21,30,191,34,36,44,6,0
550  DATA 21,246,30,13,223,32,100,6,0
560  DATA 21,30,175,21,223,36,36,53,0
570  DATA 57,191,14,173,30,63,6,0
580  DATA 9,63,63,141,54,54,0
590  DATA 49,54,62,63,32,36,6,0
600  DATA 9,54,30,30,199,57,32,52,0
610  DATA 9,54,246,199,33,214,199,33,36,6,0
620  DATA 9,30,30,30,30,77,225,28,24,28,6,0
630  DATA 9,30,30,199,57,168,145,54,0
640  DATA 27,45,45,30,30,30,30,45,45,6,0
```

The up-arrow.

```
650  DATA 186,23,77,225,6,0
```

Listing 1: Shape table poker.

hi-res page two. We'll be using the second hi-res screen for display to allow our programs to be longer. Because Applesoft programs ordinarily start at 2048 (in hex, $800) and move up from there, and the first hi-res page begins at 8192 ($2000), Applesoft programs that use hi-res page one are limited in memory use to 6K, and some of that space is used for variables. Using page two instead allows us a generous 14K. Anything we can put into specific areas of memory, like the shape table, will go above hi-res page two.

Going back to line 10, NB is the number of bytes in the table. It's right up front there to remind us to change it if we add any shapes to the table. Line 20 sets the Apple's shape table pointer. When the draw command is executed in Basic, the computer looks to this pointer to find out what to draw. Finally, lines 30 through 50 read the data and poke it into memory.

```
10  TEXT : HGR
20  FOR Y = 0 TO 40 STEP 10
30  FOR X = 0 TO 9
40  IF Y + X = 0 THEN 60
50  XDRAW Y + X AT X * 10 + 20,Y + 20
60  NEXT X,Y
```

Listing 2. Shape table test.

But What Do All Those Numbers Mean? The first seven data statements (lines 100 through 160) are the index to the table. Be sure you get those right or nothing else will work. After line 160, each line of data represents one shape. If, when you attempt to draw a shape, the computer displays some sort of incoherent garbage, you will know that the error was in the index, the data line for that shape, or any of the data lines before it. Any error in the data for one shape is likely to affect all the shapes after it. Displaying all the shapes should show you where the error originated. Listing 2 is a quick program to test the shape table. It assumes, of course, that the table is in memory.

When you're convinced the table is all there in working order, save it on disk with the command:

```
BSAVE SHAPEFILE, A24576, L480
```

Whenever you want to use the shapefile, you can get it back with the commands:

```
BLOAD SHAPEFILE
POKE 232,0: POKE 233,96
```

Now let's do something with our shape table. Listing 3 is a routine we'll use as a print statement in our later programs. The calling program must define five variables, X, Y, D, F, and W$. The word or phrase to be printed goes to W$. The direction is indicated by the value of D. If D is 1

then it prints left to right. Top to bottom printing is accomplished by setting D to 2. Right to left is 3, and bottom to top is 4. These options might seem unnecessary, but they'll help if you need to make a graph that is taller than it is wide. With the alternate print directions available, you'll be able to generate it sideways, print it out, and present it right side up on paper.

The variable F controls which way the characters are facing. If F is 1, then they'll face in a direction perpendicular to the writing direction. In other words, the letters will be stacked above one another. This is primarily for labeling vertical axes. If F is 2, the letters will face in the same direction as the line, for normal printing.

Finally, X and Y indicate where the word will begin. The range of values for these variables corresponds exactly to the dimensions of the hi-res screen.

```
9000  C = 6: IF D > 2 THEN C = - 6
9010  ROT= 16 * (D = F + 1)
9020  FOR CT = 1 TO LEN (W$)
9030  L = ASC ( MID$ (W$,CT,1))
9040  IF 64 < L AND L < 91 THEN SH = L - 42: GOTO 9150
9050  IF L > 48 AND L < 58 THEN SH = L - 37: GOTO 9150
9060  IF L = 32 THEN 9160
9070  IF L > 39 AND L < 48 THEN SH = L - 36: GOTO 9150
9080  IF L = 48 THEN SH = 37: GOTO 9150
9090  IF L > 34 AND L < 38 THEN SH = L - 34: GOTO 9150
9100  IF L = 61 THEN SH = 21: GOTO 9150
9110  IF L = 63 THEN SH = 22: GOTO 9150
9120  IF L = 94 THEN SH = 49: GOTO 9150
9130  PRINT "ERR-- NO SHAPE FOR CHARACTER: "; CHR$ (L);
      CHR$ (7)
9140  GOTO 9160
9150  DRAW SH AT X,Y
9160  IF D / 2 <> INT (D / 2) THEN X = X + C: GOTO 9180
9170  IF D / 2 = INT (D / 2) THEN Y = Y + C
9180  NEXT CT
9190  RETURN
```

Listing 3: Hi-res print routine

Lines 9000 and 9010 deal with print direction and character facing, which we just discussed. Then the word is compared one character at a time with all the possible character values to determine the shape number to draw. Lines 9040 through 9120 are the comparison tests. The most likely conditions are tested first, minimizing the number of comparisons made and thus optimizing the speed of execution. Line 9150 draws the character and 9160 and 9170 determine the location of the next character based on the printing and facing directions selected.

The Portable Printer. You can use this routine in your own programs without much difficulty; just make sure to set the variable values before calling the routine. If your programs use a separate shape table, you will be okay as long as the two tables are in different places and you reset the shape table pointer when you want to switch tables.

Also remember that the routine assumes that the scale is 1. If your program changes the scale, be sure to set it back before calling this routine. Similarly, the routine sets the rotation based on the values of D and F, so if your program uses rotation you'll have to reset it after calling the hi-res print routine.

Listing 4 is a final quickie to help demonstrate and test the hi-res print routine. Type it in with the routine and run. It simply inputs the values of the necessary variables and calls the hi-res printer.

Next month we'll use the shape table and the hi-res printer in a pie chart program. A moratorium on "easy as pie" puns is hereby called until March.

```
10  HGR : HCOLOR= 3: SCALE= 1
20  HOME : VTAB 21
30  INPUT "LOCATION?";X,Y
40  INPUT "DIRECTION (1=R; 2=D; 3=L; 4=U): ";D: IF D <> INT
    (D) OR D > 4 OR D < 1 THEN 40
50  INPUT "LETTERS FACING (1=V; 2=H): ";F: IF F <> INT (F) OR
    F > 2 OR F < 1 THEN 50
60  INPUT "WORD?";W$
70  GOSUB 9000
80  GOTO 20
```

Listing 4: Hi-res writer

THE BASIC Solution

By Wm. V. R. Smith

With the holidays finally over and the last assaults against your 1982 budget behind you, the time has come to plan for the new year. With the economy as it is today, you naturally want to eke as many pennies out of your savings as possible. To help in this pursuit, this month's Basic program will help you calculate compound interest.

Annual interest is calculated by multiplying the principal (in this case, your savings) by the interest rate once a year. Except it isn't usually that simple. Most banks calculate interest more than once a year. Quarterly or monthly interest accrual is not uncommon, and many banks now calculate interest on a daily basis.

This means that whenever the interest is calculated the principal is not actually multiplied by the annual rate but by the annual rate divided by the number of interest periods in a year. This affects your financial situation in two ways. First, with daily calculation you stand to lose less interest by withdrawing part of the principal at an odd time of the year than you would lose with yearly, quarterly, or monthly calculation. Second, as interest earnings are plowed right back into the principal automatically, with interest accrued daily, the interest you earned on Monday starts to earn you more money on Tuesday.

This sort of interest calculation affects you in one more way: it makes it really complicated to figure your interest income on your ten dollar calculator.

There are actually two ways to determine compound interest. One is to plug the values for principal, interest rate, number of periods per year, and time into a formula. The other method is brute force: calculating the interest for each period in a repetitive loop. We'll be using the second method, which offers the advantage of allowing us to keep a running balance.

Enter this month's program and run it. As an opening example, try calculating the interest on one hundred dollars at 10 percent with interest calculated yearly (one period per year). Then try the same example again with interest calculated daily (365 periods per year).

There is a difference in the results.

There is a lot more that can be done with interest calculation than we have covered here. For instance, you might make provisions for entering deposits and withdrawals and recording the dates of transactions. Using text file read and write routines, which we have covered in several different contexts in the Basic Solution, you could keep all of your financial transactions on disk and always be one step ahead of your bank statement.

Good luck this year with your savings and programming.

```
10 HOME : PRINT : PRINT : PRINT "ENTER
   AMOUNT OF PRINCIPAL";: INPUT P
20 PRINT : PRINT "ENTER INTEREST
   RATE:": PRINT "5 1/4 PERCENT = 5.2
   5": PERCENT = 10": INPUT "INTEREST
   RATE IS ";SI:SI = SI / 100
30 PRINT : PRINT "HOW OFTEN IS
   INTEREST COMPOUNDED?": PRINT
   "DAILY = 365; MONTHLY = 12": INPUT
   PERIODS
40 PRINT : PRINT "CALCULATE HOW
   MANY YEARS";: INPUT Y
50 HOME : PRINT "INTEREST ON $";P
   PRINT "GENERATING INTEREST AT ";S
   * 100;"% PER ANNUM": PRINT
   "CALCULATED ";PERIOD;" TIMES A
   YEAR": PRINT "FOR ";Y;" YEARS."
60 VTAB 9: HTAB 1: PRINT "IS THIS
   OKAY?";: GET A$: IF A$ <> "Y" AND A$
   <> "N" THEN 60
70 IF A$ = "Y" THEN 90
80 GOTO 10
90 VTAB 6: HTAB 1
100 PRINT "PERIOD INTEREST TOTAL
    PRINCIPAL"
110 POKE 34,7: REM SET NEW TOP OF
    SCREEN
120 HOME
130 R = SI / PERIODS
140 FOR D = 1 TO PERIODS * Y
150 I = P * R
160 TI = TI + I
170 P = P + I
180 PRINT D;
190 HTAB 10: PRINT INT (I * 10000) /
200 HTAB 24: PRINT INT (TI * 100) /
210 HTAB 32: PRINT INT (P * 100) /
220 NEXT
230 PRINT : PRINT
240 TEXT : VTAB 22: END
```

MARKETALK News

Unless otherwise noted, all products can be assumed to run on either Apple II, with 48K, ROM Applesoft, and one disk drive. The requirement for ROM Applesoft can be met by RAM Applesoft in a language card. Many Apple II programs will run on the Apple III in the emulator mode.

☐ What's new from **Apple Computer** (20525 Mariani Avenue, Cupertino, CA 95014; 408-996-1010)? Three years of research and development by Apple has produced mass storage systems for Apple III users. UniFile contains a high-density disk drive, and DuoFile holds two of them. Each drive provides 871,424 bytes of on-line storage to complement the 140K Disk III drive that is built into the Apple III. DuoFile contains two drives, for a storage capacity of 1.7 megabytes. Both UniFile and DuoFile use Fileware disks which contain 62½ tracks per inch, giving you six times the capacity of the conventional single-sided 48-track

disk. UniFile, less than $1,000; DuoFile, less than $1,700. Apple also has two colorful guides for those who want to learn more about personal computers. The first is the *Personal Guide to Personal Computers*, an introduction for people who would like to have a personal computer for home use. The guide explains how computers are used, how they work, and how to shop for one. Fifty pages. $1.95. *Personal Computers in Business* introduces managers, professionals, and small business owners to the personal computer as a business tool. It's chock-full of information that explains how computers are used in large and small offices, how they work, how to plan for them, how to evaluate different systems, where to shop, and how to estimate costs. Fifty pages. $2.95.

☐ Is anybody listening? Probably. **Century Systems** (12872 Valley View Avenue, Suite 11B, Garden Grove, CA 92645; 714-895-3381) takes care of nosy people with *ComCrypt*, a program that combines computer communications with encryption and decryption to send and receive CP/M files with complete security. *ComCrypt* helps to prevent unauthorized eavesdropping on computer communication that is going over telephone lines or by direct wire between terminals. The encrypted chat mode allows two users of the program to have completely secure communications between themselves over modem or on timesharing systems. CP/M is a must. $250.

☐ **Maxell** (60 Oxford Drive, Moonachie, NJ 07074; 201-440-8020) strengthens its commitment to quality by announcing a new warranty policy. They now guarantee that all their floppy disks will be free from manufacturing defects in materials and workmanship for the lifetime of the original purchaser. Any defective disk will be replaced free.

☐ As an upgrade to their *Palantir* word processor, **Designer Software** (3400 Montrose, Suite 718, Houston, TX 77006; 713-520-8221) has added *Mailout*, which gives *Palantir* the ability to generate automatic form letters. *Mailout* gives you full variables and conditionals as well as the ability to include information from a data or text file from an external document file. It also has four-function integer math, an essential for report generation. The new version of *Palantir* is $450. Current *Palantir* owners may upgrade for $25.

☐ No more delays; now you can connect your computer directly to the New York Stock Exchange Ticker line. **First Flight Data Systems** (Box 555, Kitty Hawk, NC 27949; 919-441-6480) gets you there with *The Monitor*, a ticker tape and portfolio monitoring system that eliminates delays and long-distance phone calls. The system package includes software, interface circuitry with connecting cable, and a NYSE stock guide. You get ticker tape, a portfolio monitor, advance/decline indicator, tick index, stock prices, and data files. Requires NYSE line. $1,800.

☐ **John Wiley and Sons** (605 Third Avenue, New York, NY 10158; 212-850-6336) has two book-and-disk combinations as their entries into the software market. *Apple Basic: Data File Programming* shows how to get maximum use from the Apple for data management. It teaches how to program and maintain data files for billings, catalogs and lists, numerical and statistical data, and more. $32.90. *Golden Delicious Games for the Apple Computer* features a book that teaches how to create games. The package comes with prepared games and some game subroutines that can be added. $47.90.

☐ *Rank Master* is a tool from **Exemplary Software** (725 Cowper, Suite 46, Palo Alto, CA 94301; 415-328-0545) that allows decision makers to consider relevant qualitative variables in the decision-making process. To set priorities, you create a description of the problem, specify criteria that establish how alternatives affect the goal, and synthesize the criteria to develop a ranking of alternatives. $195.

☐ When you have printing to do, the Bufferboard from **Orange Micro**

(3150 East La Palma, Suite G, Anaheim, CA 92806; 714-630-3620) frees your computer to do other things. You just dump into the buffer what is to be printed, and the printer can print at its own speed while your computer is free for other activity. With the standard 16K buffer, the Bufferboard holds about five pages of text. When expanded to 64K, it can store up to twenty pages. It features Orange Micro's interface docking system, which means that no new interface is needed; the Bufferboard fits right in the computer and docks onto all popular parallel printer interfaces. $120.

☐ Here is the address of **Pro/Pac,** makers of *Financial Management Models for the Service Firm* (Marketalk News, November 1982): Box 219000, Houston, TX 77218; 713-496-1179.

☐ New releases from **The Professor** (Box 301, Swanton, VT 05488): *The Apple's Core* is an introduction to Applesoft programming. One disk contains fourteen teaching modules that explain and give examples of programming statements. A second disk has eleven learning tutorials that complement the corresponding teaching modules. The more than thirty-five hands-on programs include an errorproof entry system so you can concentrate on what you're learning and not on typing. $49.95. *DOS Helper* lets you change DOS commands, modify error messages, alphabetize catalogs, restore deleted files, purge files, find sectors on the disk for any file, and read or modify the contents of any sector on a disk. Makes you the boss of DOS. $29.95. Finally, *Counting Plus* is an educational series that develops a knowledge of numbers and how to work with them. Sixteen modules teach counting, addition, and addition with carrying. Positive feedback and encouragement to participate make this a useful program at the preschool and primary levels of schooling. $34.95.

☐ The JSK Lower Case Adapter from **John S. Kallend and Associates** (10 West 33rd Street, Chicago, IL 60616; 312-567-3054) is a direct replacement for the character generator in recent Apple IIs and II Pluses (revision 7 or later). The adapter features the full ASCII character set and lower case with true descenders. Optional software provides listings to allow entry of lower case in Basic programs and supports shift key modification. $29.95; with software, $34.95.

☐ PrinterMaid is a 64K buffer from **Personal Micro Computers** (475 Ellis Street, Mountain View, CA 94043; 415-962-0220) that accepts data at up to 3,600 bytes per second and transfers data at 50 to 19,200 baud. Interfaces from the computer to the PrinterMaid and from the PrinterMaid to the printer can be either parallel or serial. $399.

☐ *Repton* is a fast action game from **Sirius Software** (10364 Rockingham Drive, Sacramento, CA 95827; 916-366-1195) in which you are the star fighter Armageddon, and you must prevent the Quarriors from completing a base on the planet Repton. It's you versus Nova Cruisers, Dyne-beam shooters, Draynes, and other undesirables. $39.95.

☐ *Delta Drawing*, an introduction to programming for children of ages four to fourteen, is being sold by **Spinnaker Software** (26 Brighton Street, Belmont, MA 02178; 617-868-4700). With *Delta Drawing*, children create colorful drawings on the screen using single keystroke commands to control the cursor. Also, complex pictures, patterns, and designs can be built from simple drawings because drawings are stored as programs and can be used to create complex and sophisticated graphics. $59.95.

☐ *KSAM* is a keyed sequential access method, *Edvance* is an advanced system editor, and *Print Spooler* is a print spooler. And they're all from **SofTech Microsystems** (16885 West Bernardo Drive, San Diego, CA 92127; 619-451-1230) to enhance the capabilities of the UCSD p-System. *KSAM* supports sequential and keyed retrieval of data and is designed for use in applications where fast access is a necessity. $200. *Edvance* incorporates a wide range of enhancements over the standard p-System screen-oriented editor. $200. *Print Spooler* sends one or more text files to a printer while you continue normal p-System operations. It queues as many as twenty-one files, deletes files from the queue, displays names of files on queue, and stops printing temporarily or permanently, letting you resume printing the current file if necessary. $50.

☐ **Opportunities for Learning** (8950 Lurline Avenue, Chatsworth, CA 91311; 213-341-2536) offers its *Selected Microcomputer Software* catalog, which features software in mathematics, science, language arts, computer programming and literacy, logic and simulations, social studies,

foreign language, business education, school business management, games, music, and art. It's free.

☐ **Psychological Psoftware** (4757 Sun Valley Road, Del Mar, CA 92014; 619-755-3161) enters the market with four programs. *Personality Profile* is a self-analytic program that's meant to determine your personality dynamics—gain insight into who you really are. Comes with tutorials on understanding personality growth and development. $29.50. *Assertiveness Training* gives you the help you need to become the person you've always wanted to be. Teaches you to say "no" with fortitude. $29.50. *Stress Management* defines what stress is and what causes it, measures the stress in your life, graphs it, and helps you deal with its causes. $29.50. *The Dream Machine* is designed to analyze and evaluate your dreams. There are tutorials, a dream dictionary with more than two hundred listings, and a dream log that allows you to file your dreams. Included is an illustrated book, *Beyond Dreaming*. $49.50.

☐ From **Sybex** (2344 Sixth Street, Berkeley, CA 94710; 415-848-8233) comes *The Apple Connection*, a book that shows how to interface the computer to home appliances and devices. The book describes the hardware and software necessary for simple control applications. The hardware described consists of widely available integrated circuits, and the software consists of simple Basic control programs. The text explains such simple but essential concepts as input/output and analog-to-digital conversion in enough depth to give the reader a solid understanding of the techniques presented. 288 pages. $14.95.

☐ **Kate's Komputers** (Box 1675, Sausalito, CA 94965; 415-332-9434) has released an enhanced version of *Analy$t*, their graphics system for technical analysis of the stock and commodities markets. With the new version, the user can create and display any technical formula desired and build a relational screening map that will work with the custom formulas and will indicate when changes take place in the markets. If you have a clock card, you can update the database automatically during trading hours. Update intervals may be as small as two minutes. $600.

☐ *Space Cadette* is the debut product from **Funtastic** (5-12 Wilde Avenue, Drexel Hill, PA 19026; 215-622-5716). It's a space battle simulator in which players are pitted against each other in a fight for stellar superiority. The game features six levels of play and progresses in difficulty as you progress in skill. A one-player mode pits you against the computer. $34.95.

☐ *Data Base Management Systems* explores sophisticated software for serious business applications. In this book from **Osborne/McGraw-Hill** (630 Bancroft Way, Berkeley, CA 94710; 415-548-2805), the author defines the capabilities of three categories of database systems: file, relational, and network/hierarchical. Provided are standards for evaluating database management software and examinations of several packages such as *dBase II*, *Pearl Level 3*, *DataStar*, and others. 256 pages. $16.95.

☐ *Cypher* is an algebraic calculator from **Cache Data Systems** (19112 Carp Circle, Huntington Beach, CA 92646; 714-968-0270) that allows you to set up and execute sequences of equations without resorting to formal programming languages. Models are established by keying in related formulas and can then be stored and recalled as text files. $75.

☐ Here are a few new ones from **Cortland Data Systems** (Box 14414, Chicago, IL 60614; 312-929-7727). *Chicago Strip Search* is an adult board game in which one to six players encounter a variety of fun and mature adventures. Get to the airport with your clothes on, and you win. The game is modifiable for customized adventures. $39.95. *Family Pak I* is fun for all. Educational games range from *Hangman* to *Go Fish*. This one is also modifiable. $29.95.

☐ Quit fighting over the printer! **Digital Laboratories** (600 Pleasant Street, Watertown, MA 02172; 617-924-1680) is introducing a buffered multiport spooler that lets up to five computers share one printer (or many printers share one computer) simultaneously to cut processing time. The MultiSpool is a 64K buffer that features an automatic memory technique for allocating storage between devices. It sequentially scans each port, permitting simultaneous print functions instantly. Models range from two to six ports; prices from $595 to $995.

☐ Get out a pencil. The correct address of **Compu-Home Systems,** the makers of TomorrowHouse, is 3333 East Florida Avenue, Denver, CO 80210; 303-777-6600. TomorrowHouse, as you remember, is the computerized home monitoring and control system that will watch everything from electrical appliances to your hot tub.

☐ Say goodbye to *Firebug* and hello to *Firefly* from **Muse** (347 North Charles Street, Baltimore, MD 21201; 301-659-7212). It's the same game with a different name. In this new version, you pilot the firefly through the mazes of an electric trap. Rack up the points for destroying the trap by picking up and releasing water drops to short-circuit it as you move along. It's a game of chain reactions and also a game of imagination. $24.95.

☐ Here's a system that was made to crash. **Portware** (5724 Tucker Lane, Edina, MN 55436; 612-933-3510) is offering its *Portware* system in a trial offer called the Portware Option. The Option gives investors hands-on use of the system to manage their portfolios, letting them then decide whether or not to purchase the system. After a fixed number of uses, the system will no longer function. At that point, the user has the option to purchase it at the full price. $440; trial offer, $49.50.

☐ The third edition of the *CP/M Software Index* from **Small Systems Group** (Box 5429, Santa Monica, CA 90405; 213-392-1234) is a comprehensive directory of 1,688 professionally supported programs offered by 507 vendors. The vendor's name, address, and phone number; a brief description; price; and operating system versions are shown for each package. $10.

☐ *Amperware* is a software package from **Scientific Software Products** (3171 Donald Avenue, Indianapolis, IN 46224; 317-299-0467) that significantly extends the capabilities of Applesoft. *Amperware* offers upper and lower case character entry without additional hardware; accepts commas, colons, and quotation marks; and intercepts illegal characters. The utility allows you to read and write to disk up to twenty times faster than you can using DOS. Arrays can be written in single statements, saving you up to 75 percent on disk storage space. $49.95.

☐ A new concept in print buffers is available from **Interactive Structures** (146 Montgomery Avenue, Bala Cynwyd, PA 19004; 215-667-1713). The IS PipeLine with Random Access Printing allows you to select sentences, paragraphs, graphs, or pictures from different programs, even from different computers, in random fashion to compose a single document. The PipeLine includes a "first in, first out" print operation; data is loaded into the buffer at the computer's speed and is then printed at the printer's speed. The PipeLine is compatible with the PKASO printer interface or with any Centronics parallel computer-printer connection. Memory ranges from 8K to 128K; prices from $195 to $405.

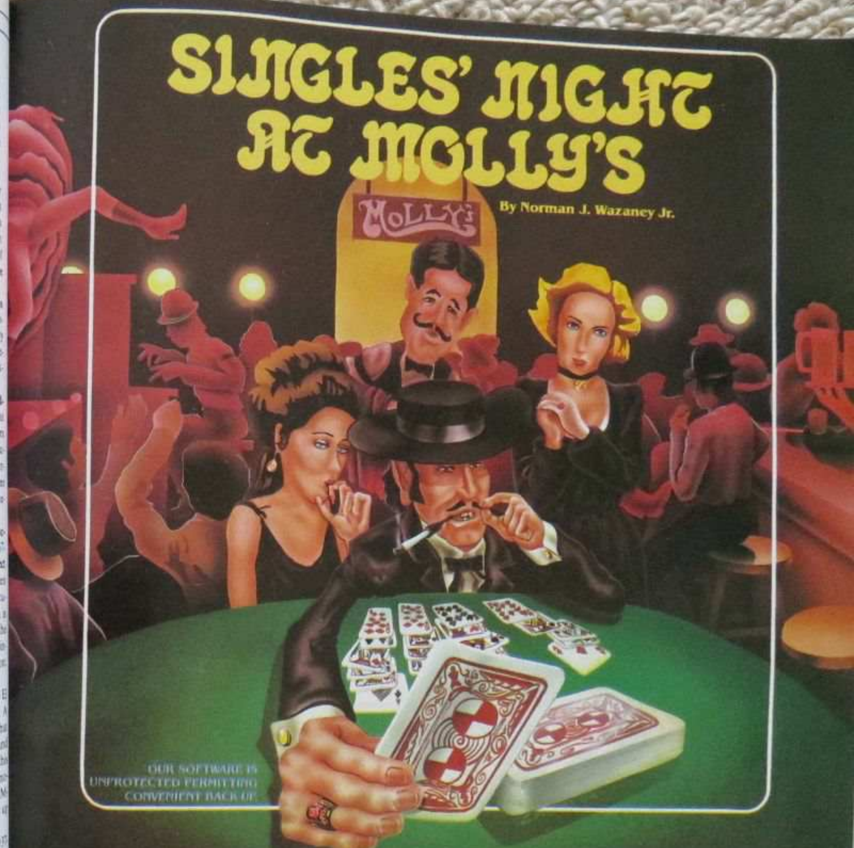
☐ Two memory expansion products from **Mountain Computer** (300 El Pueblo Road, Scotts Valley, CA 95066; 408-438-6650) have arrived. A 32K RAMplus card expands memory to 80K. Included is a disk that serves as a language expander, providing the user with both Integer and Applesoft Basic. No need to remove any chips for installation. With the card, DOS and other programs are automatically moved to high memory. $219. The *VisiCalc Expander* is a disk that, when used with the RAMplus card, can expand the memory available for *VisiCalc* calculations up to 145K. $100.

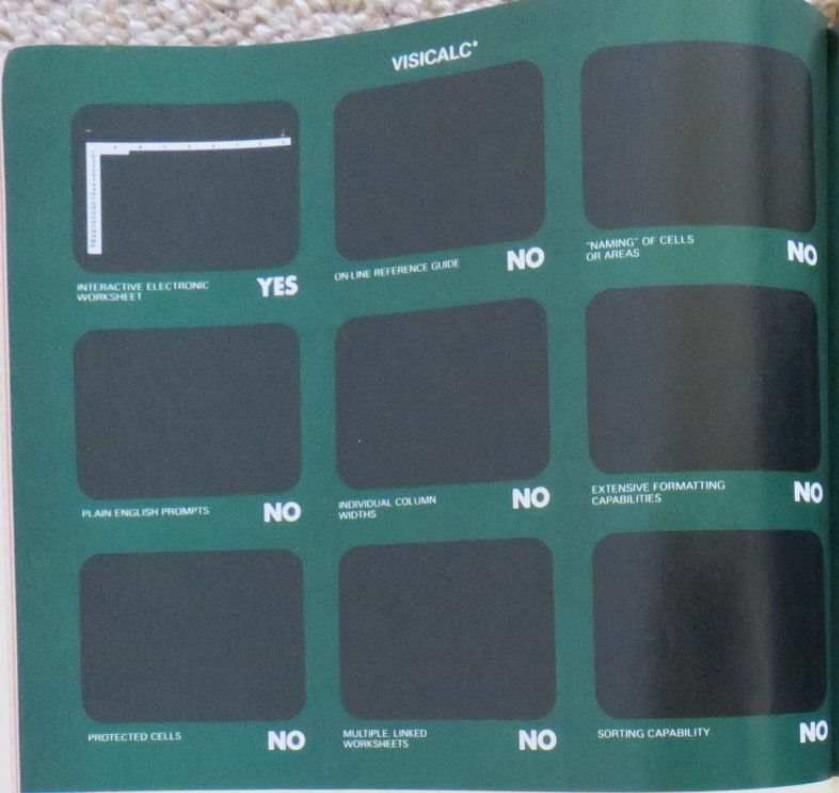
☐ **Hayden Software** (600 Suffolk Street, Lowell, MA 01853; 617-937-0200) brings in *Shapes in Color*, a Basic precision shape-drawing program. It enables the user to create and compile shapes that can be drawn on a medium-res grid in various colors, sizes, and angles. With it, you can design shapes ranging from custom fonts to animation. Backgrounds and shapes can be saved to disk for use in other programs. $49.95.

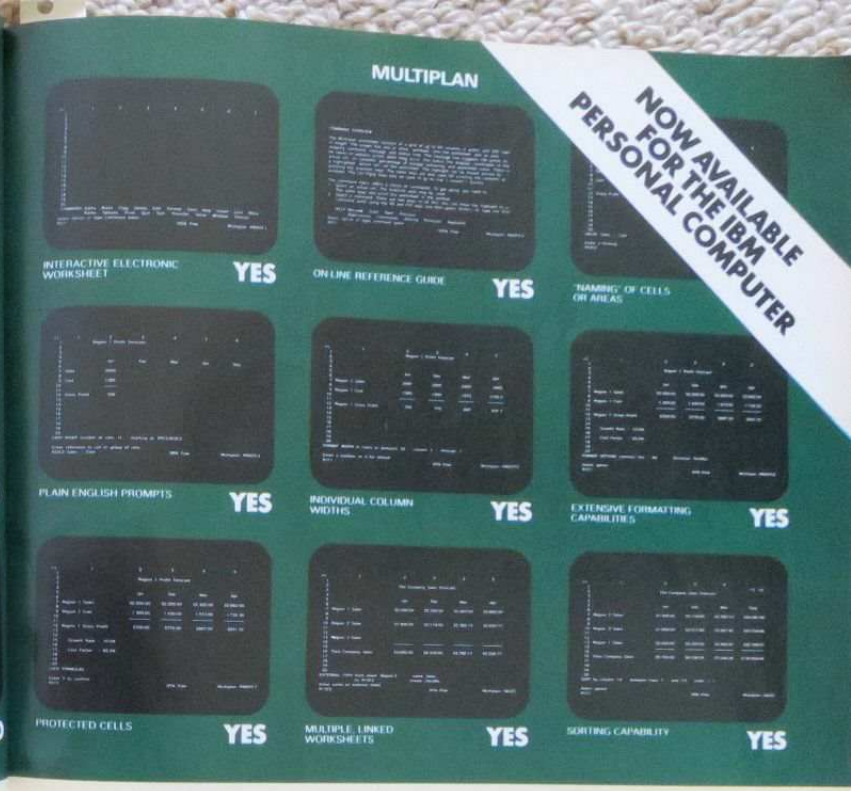
☐ **Urban Pacific Data Services** (1320 South Baldwin Avenue, Arcadia, CA 91006; 213-574-8591) has announced a disk process that will all but eliminate piracy and unauthorized copying. Their new disk prevents copies of programs in an operable form; copies can be made, but they won't run. Initial production is for CP/M systems. Quantity prices vary.

☐ *Geography Scramble* is a game that combines geography with a graphic sliding-block puzzle. In this game from **Notable Software** (Box 1556, Philadelphia, PA 19105), players must unscramble maps of various parts of the world in the fewest number of moves. It features detailed hi-res maps of both the United States and the world. Scrambling speed and number of puzzle pieces can be varied, and ratings are provided after each game. A new twist for the cube fanatics. $18.

☐ A communications package is available from **Fountain Computer Products** (1901 North Kipling, Lakewood, CO 80215; 303-232-







VisiCalc® was a swell idea for then.

The next generation. First generation electronic worksheets were a good idea. They were early software management tools that could eliminate a lot of hours with a spreadsheet, calculator, pencil and eraser. Enter Multiplan, the next generation electronic worksheet that's as easy to use as it is useful.

Make comparisons. Compare Multiplan to any of the earlier electronic worksheets. We've given you some "prompts" above.

Compare learning time. Multiplan's tutorial book brings you up to speed. Fast. But Multiplan doesn't stop there. Multiplan's On-line Reference Guide gives you instant help if you have questions. It knows where you are in Multiplan and offers information related to your problem, right on the screen.

Compare ease of use. All Multiplan prompts are full length words or phrases. And Multiplan provides "naming," the ability to assign a plain English name to any cell or area. "Gross Profit = Sales—Cost" rather than "AA44=AZ23—BK154." Which means you can work more intuitively. And faster.

Compare utility. Multiplan lets you link related worksheets so that information is transferred between them automatically. For instance, you can keep regional sales forecasts on separate sheets but link them with your overall company forecast. Then, just change the forecast for any region, and the company forecast sheet is updated automatically. Something you can't do with first generation worksheets.

Compare reports. Not just the work you can do, but the way you can present it. Multiplan's flexible formatting options allow you to produce presentation-quality reports. And its sorting capability lets you sort by either alphabetic or numeric order. So a sales manager who normally lists sales regions alphabetically could sort by amount sold and conveniently rank by sales performance. The result is a more meaningful and useful presentation of data.

Compare availability. With Multiplan, you're not limited to a single range of computers. Multiplan is available for Apple® and for microcomputers that run MS™-DOS, XENIX™ or CP/M-80® operating systems. Multiplan supports both 40- and 80-columns on the Apple II.

Compare the source. Microsoft was the world's first microcomputer software company. Today, Microsoft software is running in well over a million installations, worldwide. Languages. Utilities. Business programs. All, maintained at the state-of-the-art.

Compare for yourself. Drop into your computer store. Compare Multiplan's powerful, user-oriented features to any electronic worksheet on the market. If you've been using VisiCalc, Multiplan's ability to directly utilize your VisiCalc files lets you easily upgrade to Multiplan. And that's just another of the many features designed to make Multiplan the electronic worksheet for now. And years from now.

Multiplan™ is a great idea for now.

BETTER TOOLS FOR MICROCOMPUTERS

MICROSOFT®

Microsoft is a registered trademark, and Multiplan, XENIX, and MS are trademarks of Microsoft Corporation.

VisiCalc is a registered trademark of VisiCorp.

CP/M-80 is a registered trademark of Digital Research, Inc.

Apple is a registered trademark of Apple Computer, Inc.

Tekterm combines the features of an intelligent terminal, a graphics workstation, and a telecommunications program. The program is menu-driven and features three character sets and two modes of operation. In graphics mode, *Tekterm* features two separate graphics screens, variable speed playback, and a high-density character set (140 characters per line). In terminal mode, the program offers standard capture and transmit functions and operates in speeds up to 9600 baud. $90.

☐ The Missouri Computer-Using Educators Conference brings together educators of preschool through college level to share ideas and successes. The conference will be held February 11 and 12, 1983, at the Ramada Inn in Columbia, Missouri. Participation is limited. Reservations and registrations will be accepted on a first-come, first-served basis. Details are available from the **Missouri CUE Conference** (Dr. Cameron Pulliam, 35 Sandalwood Court, Route 5, Columbia, MO 65201; 314-445-4309).

☐ *AE* is a new game! **Broderbund Software** (1938 Fourth Street, San Rafael, CA 94901; 415-456-6424) makes you the leader of the Earth Defense System in a peaceful city in the distant future. Once under attack by a squadron of giant stingrays, your job is to drive the invading forces out of the solar system. $29.95.

☐ **SoftCorp International** (229 Huber Village Boulevard, Westerville, OH 43081; 614-890-2820) premieres *SpeedStat*, a program for statistical analysis. It has a capacity of more than ten thousand data points and more than thirty different statistical measures. Designed for small business and professional users, the program makes it easy to perform statistical analyses of demographics, product testing, and market research data. You need two disk drives for this one. $250.

☐ A memory-expansion system is available for the Apple III in the RAMdisk 320K Memory System from **Axlon** (170 North Wolfe Road, Sunnyvale, CA 94086; 408-730-0216). This product provides access speeds previously unavailable to III users. Boasting 320K, the RAMdisk functions like a single eighty-track or two forty-track disk drives. The system includes software for diagnostics, fast load and copy routines, and some business applications. A rechargeable battery system provides three hours of power back-up. $1,395.

☐ *Comprehensive Medical Management for the Apple (Comma)* is a complete, flexible medical management system from **Spectra/Soft** (Box 277, Chandler, AZ 85224; 602-963-6380) for practices of up to nine doctors. Features include total accounts receivable control with variable period aging reports, delinquency notices, daily cash reports, statements, a journal, and a day sheet. Patient accounts can be entered as families or as individuals. *Comma* also provides complete insurance forms handling with American Medical Association, Blue Shield, Medicare, Medicaid, Medi-Cal, and Champus forms. Included is a recall appointment scheduler with recall report, mailing notices, and mailing labels. Requires three disk drives or Corvus hard disk. $1,495.

☐ **Innovative Computer Products** (18360 Oxnard Street, Tarzana, CA 91356; 213-996-4911) introduces their PerfectData Type Element Cleaning Kit for daisy wheel printers. The kit consists of the proprietary cleaning unit, a cleaning pad, and a bottle of cleaning solution. $19.95.

☐ **Atsuko Computing International's** (303 Williams Avenue, Suite 1132, Huntsville, AL 35801; 205-533-7590) *Taxman-83* is an electronic worksheet overlay to be used with *VisiCalc* or *SuperCalc* that prepares and prints 1982 individual income tax returns. Multiple overlays consider all tax alternatives and compute the lowest tax based on the chosen rates for persons who are married, single, or married filing separately. Sales tax tables for each state are included, and deductions for sales tax are computed. All forms and schedules are included, calculated, and printed. This year's model has the following improvements: more input storage space, faster data input, faster computation, faster amending, and comparison of results with the prior year. Requires CP/M. $95.

☐ Knowledge Index is an information service that lets you use your computer to search through more than four million descriptions of articles, reports, and books from more than ten thousand journals and other publications. The descriptions include author, title, publisher and keywords, and a brief summary. Information covers computers, electronics, engineering, law, medicine, agriculture, business, psychology, and education. Subscribers can enter orders for printed copies of the full articles they want while on-line with Knowledge Index. Accessible by modem, the service lets you find information that would be almost impossible to find by browsing through magazines or printed indexes. Available through **Dialog Information Services** (3460 Hillview Avenue, Palo Alto, CA 94304; 415-858-3785, 800-227-1927). $24 per hour. The complete contents of each day's issue of *Commerce Business Daily* are available for on-line access from Dialog. Users can obtain information at midnight on the day of publication, which could be several days before subscribers receive their copies. All words in the *Commerce Business Daily* file can be used for an information search. In addition, you can retrieve records by subject file, abstract, sponsoring agency, special category, or publication date of each entry.

☐ **Software Publishing** (1901 Landings Drive, Mountain View, CA 94043; 415-962-8910) introduces *PFS: Graph* for the Apple III. *Graph* can work alone or directly with *PFS: File* or *VisiCalc* to produce bar, line, or pie charts of presentation quality. Line and bar graphs can be mixed and matched, and up to four graphs can be displayed on a single set of axes. Bar graphs can be stacked or comparative. Other features include automatic formatting, scaling, legend labeling, and pattern fill. $175.

☐ Extend-a-Slot from **Southern California Research Group** (Box 2231, Goleta, CA 93118; 805-685-1931) acts as an extension cord to allow the user to change peripheral cards without having to enter the computer. An eighteen-inch flexible cable allows placement of the card in a convenient location. Connectors are gold plated for reliability. $29.95.

☐ The *DIAB Point-of-Sale System* from **Micro-Logic Applications** (1029 Cote du Beaver Hall, 5e etage, Montreal, Quebec H2Z 1R9; 514-866-3606) allows up to three cashier stations and a control center to operate at one time with the control center on-line for up-to-the-minute information. The reports that are generated include an activity summary that shows stock number, description, price, quantity on hand, quantity sold, and current inventory. It will accommodate up to ninety salesman codes and up to nine hundred customer codes. $1,200.

☐ The *TRW Credit Reports* system from **Pyramid Computer Systems** (18040 Sherman Way, Suite 508, Reseda, CA 91335; 213-705-8259)

is a complete substitute for the Teletype terminal system. The system, operating with a Hayes Micromodem II, matches the TRW system and can send from one to twenty requests in a single session. A user profile is generated to include the telephone number, passwords, and slot assignments. There are separate programs for business and for consumer reports. Also included is a modification for accessing the TRW system via Tymnet. $50.

☐ Patch Computer Bug Spray will not fix any bugs in your system. It won't do anything at all, except clean your CRT screen. This novelty item from **LoTech** (1550 California Street, San Francisco, CA 94109; 415-673-6362) comes complete with tongue-in-cheek directions, cautions, and "disk-laimers." A gift idea for government agencies, phone companies, the post office, and other institutions that need debugging. At last, a pet rock for your computer. $5.95.

☐ This won't hurt a bit. **Raynbow Data Systems** (20353 Seagull Way, Saratoga, CA 95070; 408-253-9250) announces *Dental Practice Management System* for CP/M systems. The package is a versatile accounts receivable and practice analysis system for use by single or multiple doctor practices. All programs are accessible by menu and provide full-screen data entry and edit with on-screen displays and error trapping. The system prints all statements, insurance billings, and reports. Up to ninety-nine different insurance forms may be used by the system if desired, and co-insurance billings may be performed. Standard American Dental Association service code tables are provided. Requires at least two disk drives or at least one hard disk. $995.

☐ **Syntauri Corporation** (3506 Waverly Street, Palo Alto, CA 94306; 415-494-1017) has unveiled *Composer's Assistant*, a tool for composers that features automated transcription and hard copy score printing. The composer performs on the alphaSyntauri keyboard. The music is stored in up to sixteen separate polyphonic tracks and may be viewed track by track on the screen or printed on paper. The software tool can adjust for performance timing variances and ignore accidentally hit wrong notes. *Composer's Assistant* also includes sixteenth-note resolution, special triplets mode, variable accidentals, and measure ties. A built-in text editor lets the user add expressions, special instructions, lyrics, and chords. Requires the alphaSyntauri digital synthesizer system. $295. Syntauri also announces two new developments for the alphaSyntauri synthesizer system: sync-to-tape and drum interfacing capabilities. Both have been integrated into their sixteen-track digital recording system, Metatrak II. The drum synthesizer interface offers tempos of 1 to 150 beats per minute. Metatrak II generates a synchronized signal that's written to tape. Once recorded, the signal can control the playback speed of the alphaSyntauri digital recording system. An update is free to Metatrak I owners. The price of Metatrak II is $275.

☐ **Strategic Simulations** (465 Fairchild Drive, Suite 108, Mountain View, CA 94043; 415-964-1353) announces the arrival of two strategic war games: *Battle for Normandy* and *Germany 1985*. *Battle for Normandy* takes players to the northern coast of France to re-create the D-Day invasion. The game is programmed for a 25-day time passage (June 6 to 30, 1944) in which time the player must penetrate as deeply as possible into the continent and capture the port of Cherbourg and the towns of Saint Lo and Caen. The player is the Allied commander of thirty-three American and British combat formations. Weather conditions, leadership quality, and soldier fatigue levels are all variables. $39.95. *Germany 1985* is the first in a series of 1985 games, SSI's catastrophic vision of World War III. Soviet battalions have invaded the southern center of West Germany. The game provides several variables to challenge the player in this confrontation between East and West. The battle will be decided by how well you can utilize smoke screens, take advantage of air superiority, and command various sized troops. There is also a solitaire option in which the computer can play either side. Comes with hi-res map for each scenario. $59.95.

☐ *How To Use Your Apple in 10 Easy Video Lessons* is available for both the II and III. It's a set of video tapes from **Stoneware** (50 Belvedere Street, San Rafael, CA 94901; 415-454-6500) designed to be used with video cassette recorder and CRT placed side by side. Lessons cover computer fundamentals, assembling your system, operating the machine, using disks and utilities, using files, and programming in Basic. Also includes a software overview. $120.

☐ **Agile Corporation** (25 East Trimble Road, San Jose, CA 95131; 408-946-7576) has developed a series of interface devices that permit your Apple II or III to communicate directly with an IBM mainframe. The Agile 5278 consists of a 4-inch by 12-inch circuit board that plugs into a slot, a software package that runs under the standard operating system, and a keyboard overlay. $1,595.

☐ **Atlantic Cabinet** (Interstate Park, Box 100, Williamsport, MD 21795; 301-223-8900) has some new stuff to add to their line of computer furniture. Roll-a-Drawer is a locking mobile two-drawer cabinet. $165. Roll-a-Store is a locking mobile storage cabinet. $115. Their 21-inch printer stand is designed for small printers. $67. The Heavy Duty Printer Stand is meant for high-speed, bottom-feed printers. $115. And, to hold all that paper, the Wire Printer Basket is just $13.

☐ *Acculink-RX* is another in the series of error-free file transfer programs from **IE Systems** (Box 359, 112 Main Street, Newmarket, NH 03857; 603-659-5891). Designed for file transfer between mainframes and personal computers, *Acculink-RX* is controlled via terminal port by your computer running any of IE's micro-based asynchronous communications packages. Upon error detection, the program will automatically attempt retransmission. $1,980.

☐ *Quic-N-Easi AG* is an applications generator and database manager from **Standard MicroSystems** (136 Granite Hill Court, Langhorne, PA 19047; 215-968-5966) for the novice computer user. Without having to code and with no previous programming experience, you can produce professional-looking applications. Using the applications generator, you just fill in the blanks with your answers to its questions to format things the way you want them. With the report generator, you can use up to six applications-generated files simultaneously. You can scroll three hundred columns horizontally and a thousand lines vertically. $295.

☐ Hey, investors, meet the *Option Analysis System* (Oasys) from **Software Resources** (186 Alewife Brook Parkway, Cambridge, MA 02138; 617-497-5900). With it, you can determine fair market values, test trading strategies, examine price spreads, measure leverage potential, compute volatilities, and replace costly advisory services. $395. Also from

Software Resources is the *Securities Market Analysis, Reporting, and Transactions* (SMART) system. It handles automatic charting, daily updating of your portfolio, gives numeric analysis of statistics, monitors accounts, and is fully expandable to work with other programs or in a network or hard disk system. $1,950.

☐ **Softprotex** (Box 271, Belmont, CA 94002) sells *Crossword-Scrambler* for the Apple III. Lets you interface with a computer rather than subject yourself to one and features graphics with audio output. $39.95.

☐ Science instructors, take note: **Focus Media** (135 Nassau Boulevard, Garden City, NY 11530; 516-248-8799) has several programs that teach various subjects in an interactive game format. *Your Body Series I* covers the digestive and circulatory systems. $119. *Your Body Series II* covers the muscular, skeletal, endocrine, and brain and nervous systems. $119. *Matter & Energy* takes students through elements, compounds, mixtures, physical and chemical changes, potential and kinetic energy, and energy changes. $159. *Water & Weather* teaches the water cycle, humidity, clouds, and precipitation. $89. *Energy* covers heat energy and sound energy. $69. Finally, *Chemistry Series I* gets involved with states of matter, atomic structure, and acids and bases. $89.

☐ **Avion Video Computers** (22916 Lyons Avenue, Suite 2A, Newhall, CA 91321; 805-259-2910) presents two self-paced videotaped computer courses prepared by a layman for laymen. *Computer Fundamentals* requires no previous computer knowledge and contains fundamental principles of all microcomputers and beginning Basic programming. *Programming in Basic* is for the person who has some familiarity with a microcomputer and wants to customize programs for his or her own purposes. Tapes are in both VHS and Beta formats. $68.95 each.

☐ **Dickens Data Systems** (Suite A, 3050 Holcomb Bridge Road, Norcross, GA 30071; 404-448-6177) sells the *Wall Street Plotter*. The program provides the individual investor with visual tools designed for use in technical analysis of financial securities, commodities, and market averages. With it, you can generate high-low-close volume price plots, standard price plots with a moving average, and trend analysis plots. Included is a line-oriented text editor for manual data entries, deletions, or additions. $125.

☐ What the ComRiter CR-1 from **Comrex** (3701 Skypark Drive, Suite 120, Torrance, CA 90505; 213-373-0280) offers is letter perfect printing at a low price. It prints 17 characters per second and 132 to 198 characters per line. The carriage offers bidirectional printing with sixty positions per inch. Compatible with Qume, Centronics, and serial interfaces. $1,099 to $1,199.

☐ **Applied Creative Technology** (2723 Avenue E East, Suite 717, Arlington, TX 76011; 800-433-5373) has come out with the Printer Optimizer, a device for either your printer or modem. It has a memory capacity of 65,000 to 256,000 characters, which can be loaded from the computer, then sent to the printer or modem at the appropriate speed. More than just a spooler or buffer, the Printer Optimizer features a keypad and display that enable you to enter instructions on exactly how you want it to print. $495.

☐ Dramatic price slashings have taken place in **Ellis Computing**'s ([illegible] 41st Avenue, San Francisco, CA 94121; 415-751-1522) CP/M based software series. *Nevada Cobol*, *Nevada Fortran*, *Nevada Pilot*, and *Nevada Edit* have all been reduced to $29.95.

☐ The second edition of the *Stock Portfolio System* from **Smith Micro Software** (Box 604, Sunset Beach, CA 90742; 213-592-1032) expands the capabilities of its predecessor. Complete margin accounting is provided for up to three margin accounts. Multiple cash accounts such as money market, credit union, or bank accounts may also be included. Security quotes may either be entered manually or be retrieved automatically from the Dow Jones News and Retrieval Service. A terminal mode for gaining access to other Dow Jones services is provided. $185.

☐ Micro ID plugs into your computer to provide complete security for your software. This device from **Delta Digital Design** (Box 15952, New Orleans, LA 70175; 800-535-1814) gives your computer a unique identity, making it possible to construct a program disk that will operate only on your computer. $100; quantity prices available.

☐ From **Great Plains Software** (123 North 15th Street, Fargo, ND 58102; 701-293-8483) comes the *Hardisk Accounting Series, Version 2.0*. This is the latest release of *General Ledger*, *Accounts Receivable*, and *Accounts Payable* modules. The *Version 2.0* release includes larger data fields, additional reports and printouts for analysis, enhanced data entry features, faster program manipulation, and improved documentation. *Payroll* and *Inventory with Point of Sale, Version 2.0* are two new modules of the *Hardisk Accounting Series*. *Payroll* handles up to 9,999 employees and supports hourly, salary, commission, tips, and piecework compensation. Also supports up to twenty different types of payments and twenty deductions and allows for up to five local tax deductions and eight different payroll periods. $495. *Inventory* handles up to 32,767 parts, accommodates serial numbered inventory items, allows sales history by individual item, supports five price levels, and includes a point-of-sale program that calculates sales tax, prints a sales slip, and automatically updates inventory. $595.

☐ *CApple* from **Computer Automation** (2181 Dupont Drive, Irvine, CA 92713; 714-833-8830) connects your Apple with mainframes and SNA networks through SyFA network processors. The software allows you to use the Apple as a functional SyFA workstation. Additional function key emulation is provided to allow the Apple to function as an IBM 3270 when operated with IBM 3270 emulation software. Screen and keyboard enhanced features are required. $2,500; enhancement features, $300 to $500.

☐ February 1, 1983, is the deadline for article submissions to **The Writing Instructor** (The Freshman Writing Program, University of Southern California, Los Angeles, CA 90089; 213-743-5672) for their summer issue. They're looking for articles, reviews, and exercises that relate computer to writing instruction.

☐ Educational fun for children is what **Edu-Ware** (Box 22222, Agoura, CA 91301; 213-706-0661) offers in *Spelling Bee Games*. Four colorful games, *Squadron*, *Skyhook*, *Puzzle*, and *Convoy*, strengthen spelling and reading skills as well as your hand-eye coordination, memory, and motor skills. Picture menus are available for tots with limited reading ability. For bees of ages four to seven. Requires game paddles. $29.95.

VOTE! For the MOST POPULAR PROGRAM of 1982!

The Apple software market has come a long way. The winner of *Softalk*'s First Annual Most Popular Software Poll, which covered Apple software released prior to January 1, 1981, was the venerable *Super Invader*. Last year's winner, from a field of contenders released from October 1, 1980, to January 1, 1982, was *Raster Blaster*, Bill Budge's pinball masterpiece. Now it's time to cast your ballot to put a third program in our software hall of fame.

Though the first two winners were both games, keep in mind when you vote that we're looking for the best programs regardless of category—business, word processing, education, home, and hobby as well as games. Apple III owners are encouraged to vote too. When all the votes are tallied, the results will be set up like our monthly bestsellers, with a top thirty for all programs and top fives or tens in each major category.

So tell us what programs you use and like, and we'll spread the word. We welcome your comments as well; maybe they'll shed some light on how a turkey piece of software can become a bestseller or how a real gem can languish on the store shelf gathering dust.

In the first two years of the poll, the term *state of the art* has come to have a whole new meaning in Apple graphics. The standards for other aspects of program quality have gone up as well. By telling the market what you want and what you like, you the consumer are a major impetu behind the rising expectations of what makes a good program.

This is why *Softalk* asks its readers to speak out en masse once a ye about the software they like best. Factors like flashy packaging and clever advertising campaign can sometimes contribute more to sales fi ures than program quality. Reviews look at software products in dept but it's always possible for a reviewer's analysis to miss the mark. The in dustry gets some of its best feedback on software quality through th mechanism of this poll.

A hotly debated issue, at least in some circles, is just how far state o the art can go. Some folks say that the best Apple program has alrea been written. Others cite the fact that people have been surprising Jo and Wozniak with what the Apple is capable of for years now and be lieve that they'll continue to do so. This controversy may never be re solved to everybody's satisfaction, but, to fan the flames a little, *Softalk* introducing a new element to the Most Popular Software Poll. We'd li you to name, in addition to your ten favorite programs from this yea your number one program of all time. With that we'll see how this yea models stack up against the classics.

Rules. Only one ballot per Apple user will be accepted; this mea

that a family-owned Apple might generate four or five ballots—one for each member of the family. Where there are two ballots from one person, only one will be counted if the choices are the same and neither will be counted if the choices differ. Your ballot must reach *Softalk* by February 15, 1983.

For your convenience and ours, you can use the attached card to vote. If there are many eligible voters sharing one magazine, or if you picked up this *Softalk* in the dentist's office and the ballot has already been ripped out, just send a numbered list of your ten favorites with your name, address, and comments. Mail it to Softalk Pops, Box 60, North Hollywood, CA 91603.

The ballot has spaces for your ten favorite programs. This is a weighted vote—give us your top ten in order and we'll give the most points to the ones that head the list. No fair voting for a program twice. Any program that was released between October 1, 1981, and January 1, 1983, is eligible. The reason for the three-month overlap is that deserving programs released late in the year often don't have sufficient chance to be noticed for that year's poll; so they're eligible in the following year as well.

The list given here contains a lot of eligible programs, but by no means all of them. Use it to jog your memory, not to direct your vote.

When casting your vote for the most popular program of all time, choose only one program. Any program. It can be a program that was released this year or a program that appeared years ago, as long as it runs on an Apple.

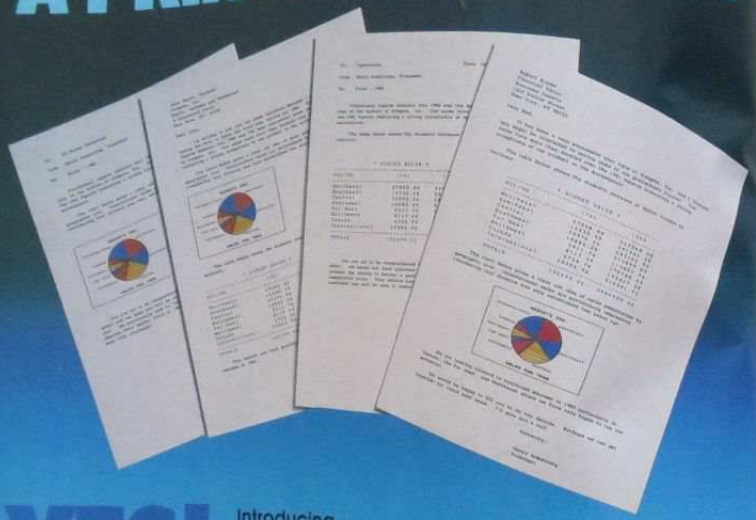
The two sections of the poll are being tabulated separately, so if your all-time favorite is a recent release you'll have to list it as the first program on your top ten list for it to be credited for this year's most popular program.

The votes will be tabulated by members of the *Softalk* staff, who, while they aren't Price Waterhouse, are reasonably scrupulous. The results will be hermetically sealed and the winner will be announced at the West Coast Computer Faire, where the author and the publisher of the winning program will be presented with the coveted award. Even if you can't make the fair, you'll be able to read about the poll results in the April issue.

This is a list of many of the programs that are eligible for this year's Most Popular Software Poll. The list was compiled from entries in Marketalk News, Marketalk Reviews, *Softalk* Presents The Bestsellers, and Fastalk from our last fourteen issues. It is not meant to be all-inclusive, however, and you may vote for any programs published between October 1, 1981, and January 1, 1983.

Accounting Plus
Accurec
AccuShapes
Acey Deucey
Agenda Files
AgriCalc
AIRSIM—1
Alexander the Great
Algebra 2
Ali Baba and the Forty Thieves
Alkemstone
Alpha Plot
Amper Memory Program
Ampersoft
Anova II
Apple-Cillin II
Apple Flasher
The Applegrator II
Apple Logo
Apple Mechanic
Apple Speller
Apple Spice
Apple VisiCalc Info Printer
Apple Writer II
Appointment at Aldebaran
Apventure to Atlantis
Arcade Machine
Arith-Magic
The Artist
ASCII Express Professional
Assembler Teacher
Assembly Language Development System
Bag of Tricks
Bandits
The Battle of Shiloh
The Battle of the Bulge: Tigers in the Snow
Battlesight
Beer Run
Bez-MX
Birth of the Phoenix
Borg
BPI Accounts Receivable
The Budgeteer
The Budget Planner
Bug Attack
BusiComp
Business Bookkeeping System
Cannonball Blitz
Casino
Castles of Darkness
Cermac
Ceiling Zero
Choplifter
The Christmas Story
Computer Acquire
Computer Football
Computer Stocks and Bonds
Conglomerates Collide
Congo
Cops and Robbers
The Cosmic Balance
The Count
Counting Bee
Crazy Mazey
Crop Management
Crossfire
Crush, Crumble and Chomp
Curse of Ra
Cyborg
Cyclod
Cytron Masters
Danger in Drindisti
Dark Forest
DataFax
Datalink
Data Perfect
The Data Reporter
David's Midnight Magic
dBase II
Deadline
Deadly Secrets
Death Race '82
Derby
Dictionary
Diet Analysis
Directory Master
The Disk Labeler
Disk Library
Dnieper River Line
Dogstar Raider
Dow Jones Market Analyzer
Early Elementary I
Early Games for Young Children
EASy
Easy Mailer
Easy Writer Professional
Editrix 1.0
Electric Duet
The Electric Semicolon
Elementary, My Dear Apple
Eliminator
Empire Builders
Empire I: World Builders
Escape from Rungistan
The Estimator
Executive Briefing System
Executive Secretary
Facemaker
Federation
File Whiz
Filewriter
Firebird
Firebug
First Class Mail
Flight Simulator
The Flockland Island Crisis
Fly Wars
Fore!
Frazzle
Fred III
Free Fall
Frogger
Galactic Gladiators
General Manager
Genetic Drift
The Ghostwriter
Gin Rummy
Global Program Line Editor
Gold Rush
GPS
The Graphic Composer
Graphic Writer
Graphics Magician
Graphics Package 42—3D1
Guadalcanal Campaign
Guardian
Guns of Fort Defiance
Gusher
The Gutenberg Word Processor
Hadron
The Harmonic Motion Workshop
Hello Central!
Hi-Res Adventure #4: Ulysses and the Golden Fleece
Hi-Res Computer Golf
Home Accountant
Honambo Warrior
Horizon V
Horse Racing Classic
The Illustrator
Jawbreaker (Lubeck)
Jawbreaker II (Chuckles)
Jellyfish
Job Cost II
Juggler
Kabul Spy
Keno Master
The Keys of Acheron
Knight of Diamonds
Labyrinth
Laf Pak
The Liberator
The Linguist
Locksmith 4.1
Lunar Leeper
Magic Spells
Mail Mate
Marauder
Market Advisor
Market Tracker
Mars Cars
Mask of the Sun
MatheMagic
Menu Generator
The Menu II
Merlin
Micro-Courier
Micro-Deutsch
Microfinesse
Micro Golf
Microwave
Mike Caro's Video Poker
Milestones
Minotaur
Mouskattack
Mr. Speller
Multiploy
Mummy's Curse
Napoleon's Campaigns: 1813 & 1815
Neptune
The New Step by Step
The Night Before Christmas and Christmas Songs
Norad
Odin
Oil Rig

RANDOM ACCESS IN A PRINTING BUFFER?



YES!

Introducing . . .
The IS PipeLine™ Random Access Printing Buffer.

Insert pictures, graphics or spread-sheet data into reports. Duplicate form letters—automatically changing addresses on each. Now, all your programs can work together to produce printed output.

For the first time ever, here is a buffer that not only frees your fast computer from your slow printer but also allows you to rearrange, compose and copy your data on its way to the printer.

- Random Access Printing—stores paragraphs or pictures for printing in any order—any number of times.
- FIFO Printing—conventional first-in first-out operation.
- Compression of data for efficient utilization of memory space.
- Ability to interrupt long-term buffer operations for straight-thru short-term printing.
- Simple Erase feature to clear buffer.
- Automatic duplication capability.
- Easily expandable, by you, from 8K Bytes to 128K Bytes.

The IS PipeLine is Universal—it works with any parallel (Centronics*—style) computer/printer combination. A special version is available for PKASO** Printer Interfaces.

The IS PipeLine is a self-contained unit with operating manual, cables and power supply included.

For more information on the truly revolutionary IS PipeLine Random Access Printing Buffer, call us today.

Interactive Structures Inc.
146 Montgomery Avenue
Bala Cynwyd, PA 19004
Telephone: (215) 667-1713

*Centronics is a trademark of Centronics Data Computer Corp.

The IS Pipeline™ Random Access Printing Buffer is patent pending

ABOUT BUYING A HERE'S SOME HELP.

Good software keeps the complications in the computer, where they belong. And keeps the capability at your fingertips. It's that simple.

You simply have to see for yourself.

You can read any number of interesting books and magazines about personal computers. You can ask friends who have them. You can look at all the sales literature you can get your hands on. And you should do all those things before you decide to buy.

But as helpful as all that can be, there really is no substitute for a real, live demonstration.

When you do go out shopping, we recommend you take a look at the PFS® Family of Software.

The PFS family is designed the way we think all software should be: simple, straightforward and powerful.

Currently, three products make up the family. PFS:FILE, PFS:REPORT and PFS:GRAPH, with more programs on the way. Here's a little more about each of them.

PFS:FILE. The simplest way to get organized.

Basically, FILE works like a paper filing system, without the paper. So you can record, file, retrieve and review information in a fraction of the time it takes with a conventional filing system.

FILE lets you arrange your information in "forms" you design yourself. So you can get at and really use your information in ways never before possible.

What's more, FILE lets you change the original form without having to redo the information on it.

PFS:REPORT. Making the most of your information.

REPORT summarizes the information on your forms so you can use it to analyze, plan and make better-informed decisions.

With REPORT, you get presentation-quality reports—sorted, calculated, formatted and printed—automatically, in seconds.

PFS:GRAPH. Instant pictures.

GRAPH gives you presentation quality bar charts, line graphs, and pie charts, in black and white or color, on paper or the computer screen. To get a clearer picture of things and spot trends instantly, you simply enter your information and specify the kind of graph or chart you want. GRAPH does the rest.

You can also mix and match line and bar graphs, or even stack or compare up to four bar graphs simultaneously.

And GRAPH will work with PFS:FILE, VisiCalc® files, or data entered directly into the computer.

Best of all, compared to the cost of hand-drawn graphics, GRAPH can save you enough money over the course of a few months to pay for the computer it runs on.

The PFS Family of Software. Simple and powerful.

Send for our Free PFS SOFTWARE CATALOG.

It'll tell you more about the PFS Family of Software and how to use it.

It's free. And all you have to do to get one is return the coupon below, or see your participating PFS dealer.

The PFS Family of Software. We've already made computers simpler to use.

Now we're making them simpler to buy.

pfs:

FREE PFS SOFTWARE CATALOG

Name _____
Address _____
City _____
State _____ Zip _____
I plan to use a personal computer:
☐ at home ☐ at work ☐ both.
Mail to: PFS, 422 Aldo, Santa Clara, CA 95050

371/83

*PFS is a registered trademark of Software Publishing Corporation, 1901 Landings Drive, Mountain View, CA 94043. The PFS Family of Software currently runs on all Apple® II and Apple III computers. *Apple is a registered trademark of Apple Computer, Inc. *VisiCalc is a registered trademark of VisiCorp Personal Software.

All About Applesoft

by Doug Carlston

Instead of using Applesoft to manipulate information in the computer, as we have in the past, this month we're just going to look into ways of organizing data and storing it so that we can get it back whenever we want. Don't let the phrase *data storage and retrieval* put you off. It's what computers are really all about.

Let's imagine that we want to create a personal Christmas card list with names and addresses of all the people we know. It may seem late for this, but think of it as starting early for next year. For each address in the list, we'll want to store four items: the name of the person, the street address, the city and state, and the zip code. (If you want to talk like a computer person, you can call each line of the address a *record*.)

It would be possible to set up an array and load into that the information on each person, one person after another. But variables lose their values when the computer gets turned off, and it may not be practical to keep the same program running year-round. Therefore, we have to find a way to store this information in a more permanent fashion.

We are going to discuss three methods of permanent storage. The first method is the least practical but the only one possible if you are one of those poor benighted souls who are still saving everything on cassette tape. This method involves storing data inside the program itself in the form of data statements. You save your information by saving the program.

Here's a short program that uses data statements to store information. This program will search through its data to find an address if you enter the name (or any portion of it) of the party you are interested in:

```
10  HOME: INPUT "ENTER NAME OF PERSON: ";A$
20  RESTORE
30  READ N$,AD$,ST$,ZIP$
35  IF N$ = "END" THEN 80
40  L = LEN (N$) - LEN (A$)
50  FOR X = 1 TO L
60  IF A$ = MID$ (N$,X, LEN (A$)) THEN X = L: NEXT : GOTO 100
70  NEXT: GOTO 30
80  PRINT "NOT FOUND": GOTO 110
100 PRINT N$: PRINT AD$: PRINT ST$: PRINT TAB (9);ZIP$
110 PRINT : PRINT : INPUT "ANOTHER? ";A$: IF LEFT$ (A$,1) =
    "Y" THEN 10
120 END
130 DATA THE ALAMO, ALAMO PLAZA, "SAN ANTONIO, TX",
    78205
140 DATA THE WHITE HOUSE, 1600 PENNSYLVANIA
    AVENUE,"WASHINGTON, DC",20500
150 DATA BILBO BAGGINS,1 BAGSHOT ROW,HOBBITON,THE
    SHIRE
160 DATA TOBY SACKTON,HOME FOR UNWANTED
    CHILDREN,DAR ES SALAAM,TANZANIA
900 DATA END,END,END,END
```

Let's see how this works. The first line clears the screen and then asks you for the name of the person you are trying to look up (actually, the first name alone or the last or any unique part will do, as we will see in a moment). Then line 20 tells your Apple that when it reads a line of data it should start with the first item in the first data statement. Line 30 is the read statement—it picks off the first four pieces of data and stores them in four different variables: N$, AD$, ST$, and ZIP$.

Note that all items in the data statement are separated by commas. However, if we want to include a comma *inside* an item, we can do that by enclosing the whole item in quotation marks. You can see examples of this in lines 130 and 140.

The next thing we want to do is compare the string that we are searching for with the one that is now called N$. If any part of N$ is the same as A$, then we have a match.

We do this test in lines 40 through 70. To find a substring (A$) within a string (N$), we line up A$ against the first part of N$ and then gradually slide it to the right. Imagine, for example, that we had said we were looking for "White." When we get to the name "The White House," your Apple first compares the string "White" against "The W"—the first five characters in The White House. Then it checks House against each subsequent group of five letters until it finds a match, like this:

```
The W     = White?
 he Wh
  e Whi
   Whit
   White = White
```

The for-next loop in lines 50 through 70 does this test. The MID$ statement in line 60 picks from N$ the five characters to be compared against A$. If it finds a match, program control jumps to line 100; otherwise, it goes back to line 30 to examine another data statement.

The advantage of this kind of storage is that your data is stored right along with your program, and it can be saved right along with the program. One disadvantage is that every time you want to add, delete, or change a name you have to get into the program and alter it. Another disadvantage is that you cannot store more data than will fit in memory at any one time (a considerable amount, to be sure, but still a real limiting factor).

Those of you with disk drives can get around these limitations through the use of text files. Text files come in two flavors—random access and sequential. Both are designated by the letter T in disk catalogs. Sequential files tend to be much more compact than random files, permitting far more names per disk. However, random access files allow much easier access to records than sequential files. You can modify a record in a random file without worrying about overwriting other records.

Both kinds of text files are handled with the same commands. First, you *open* a file and give it a name. Then you *write* your data into the file, and, finally, you *close* it. If you are creating a sequential file, the data is packed in, one piece right behind the last, like this:

```
<THE ALAMO><ALAMO PLAZA><SAN ANTONIO, TX>
<78205><THE WHITE HOUSE><1600 PENNSYLVANIA
AVENUE><WASHINGTON, DC><20500>
```

and so on.

The trouble with this, as we mentioned, is that if you're looking for a particular record you have to begin at the beginning and search your way through the whole file. Since each record is a different length, there is no easy way to jump directly to the record you want.

Random access files take care of this problem. When you first open a random access file, you must specify a record length. Then, when you write data to the file, each record starts at an even multiple of the record length, like this:

```
<THE ALAMO                >
<ALAMO PLAZA              >
<SAN ANTONIO, TX          >
<78205                    >
<THE WHITE HOUSE          >
<1600 PENNSYLVANIA AVENUE>
```

```
<WASHINGTON, DC>
<20500>
```

Because the file is organized this way, it is relatively easy to figure out just how many characters into the file you will have to jump in order to reach the address you are interested in. The disadvantage, as it's plain to see, is that you will waste a great deal of space on your disk saving blank spaces at the end of each record. In the example, each record is twenty-five characters long. Since zip codes are only five characters long, all zip code lines waste twenty characters (bytes) of space in the file.

The easiest way to see the difference between the two text file types is by writing a couple of short programs and comparing them. So let's first write one that uses sequential files. We'll enter data into the program from the keyboard and save it in an array. When we've finished, we will press return without typing anything, and the program will take this as a signal to write the array out to the disk as a sequential file.

In order to get a feel for proper programming style, we will try to give these programs a proper finished polish. A little additional vocabulary is used, so we'll dissect the programs line by line.

First, here's the new vocabulary:

WAIT	ON ERR GOTO
CHR$(4)	CHR$(34)
POKE 216,0	PR#n
OPEN file	DELETE file
CLOSE file	WRITE file

Strictly speaking, the last five commands are part of your Disk Operating System, not part of Applesoft at all. However, you need them to write this sort of program, so there's no point in being too much of a stickler about technicalities.

Now here's the program:

```
10  DIM A$(3,99):D$ = CHR$ (13) + CHR$ (4)
20  HOME : PRINT TAB( 15)"MAILING LIST"
30  GOTO 500
40  REM ********************
50  REM     SUBROUTINES
60  REM ********************
69  REM
79  REM
80  REM *** GET INPUT LINE ***
81  REM
82  A$ = CHR$ (34):B$ = A$
83  GET C$: IF C$ = CHR$ (13) THEN PRINT C$;:B$ = B$ + A$:
    RETURN
84  IF C$ = CHR$ (8) AND LEN (B$) > 1 THEN PRINT C$;:B$ =
    LEFT$ (B$,LEN (B$) - 1): GOTO 83
85  IF C$ = CHR$(8) THEN 83
86  B$ = B$ + C$: PRINT C$;: GOTO 83
89  REM
90  REM *** PAUSE ***
91  REM
92  PRINT : PRINT TAB( 8)"PRESS ANY KEY TO CONTINUE";
93  WAIT - 16384,128
94  POKE - 16368,0: RETURN
500 REM ********************
510 REM       MAIN MENU
520 REM ********************
550 VTAB 5: HTAB 8: PRINT "E(NTER NAMES"
560 VTAB 7: HTAB 8: PRINT "P(RINT LIST"
570 VTAB 9: HTAB 8: PRINT "Q(UIT"
580 VTAB 12: HTAB 8: GET A$: IF A$ = "E" THEN 1000
590 IF A$ = "Q" THEN HOME : END
600 IF A$ = "P" THEN 2000
610 GOTO 500
1000 REM ********************
1010 REM      INPUT DATA
1020 REM ********************
1030 PRINT : PRINT "NAME (<RET> IF DONE): ";
1040 GOSUB 80: IF LEN (B$) < 3 THEN 1090
1045 A$(0,K) = B$
1050 FOR X = 1 TO 3: PRINT " ADDRESS LINE ";X;": ";:GOSUB
     80:A$(X,K) = B$
1060 NEXT
1080 K = K + 1: GOTO 1030
1090 PRINT D$"OPEN MAILING LIST"
1095 PRINT D$"DELETE MAILING LIST"
1100 PRINT D$"OPEN MAILING LIST"
1120 PRINT D$"WRITE MAILING LIST"
1130 FOR X = 0 TO K - 1
1140 FOR Z = 0 TO 3
1150 PRINT A$(Z,X): NEXT : NEXT
1160 PRINT D$"CLOSE MAILING LIST"
1170 RUN
2000 REM ********************
2010 REM       PRINT LIST
2020 REM ********************
2025 HOME : VTAB 5:P = 0
2030 PRINT TAB( 8)"S(CREEN": PRINT : PRINT TAB( 8)
     "P(RINTER": PRINT : PRINT : PRINT TAB( 8);: GET A$: IF A$
     = "S" THEN 2050
2040 IF A$ <> "P" THEN 2025
2045 PRINT D$"PR#1":P = 1
2050 PRINT D$"OPEN MAILING LIST"
2060 PRINT D$"READ MAILING LIST"
2070 ONERR GOTO 3000
2080 FOR X = 0 TO 99
2090 FOR Z = 0 TO 3
2100 INPUT A$(Z,X): NEXT : NEXT
2110 FOR X = 0 TO K
2120 FOR Z = 0 TO 3
2130 PRINT A$(Z,X): NEXT : IF NOT P AND (X + 1) / 3 = INT ((X + 1)
     / 3) THEN GOSUB 90: PRINT
2135 PRINT : NEXT
2140 PRINT D$"PR#0": PRINT : IF X / 3 <> INT (X / 3) THEN
     GOSUB 90
2150 RUN
3000 POKE 216,0:K = X - 1: GOTO 2110
```

Take a look at line 10. We define the string D$ as a CHR$(13) plus a CHR$(4). CHR$(13) is the same as a return, so what we are really doing is defining D$ as a control-D preceded by a return. The reason we have done this lies in the difference between Applesoft and the Disk Operating System (DOS for short). As we mentioned, some commands in this program are not Applesoft commands at all. If they ever got to the Applesoft interpreter in your computer, they would merely generate syntax error messages. Therefore, we need some way to intercept these commands in the program before Applesoft gets a chance to look at them.

The way we do this is by printing DOS commands and by preceding them with the invisible control-D. Whenever an Applesoft program prints anything, your Apple checks the first character in the line to see if it is a control-D. If it is, the Apple says, "Aha! What follows will not be sent to the screen at all. My beloved user merely wants to speak with DOS." It then immediately takes whatever is printed and, instead of printing it on the screen, sends it off to DOS to try and puzzle out.

The CHR$(13) part of D$ is not necessary as a rule. However, if you are using the get command in Applesoft, you will find that it has an unfortunate propensity to swallow the first character in a subsequent print statement. In addition, frequent use of the semicolon to continue printing on the same line could lead to a situation where the control-D is not actually the leading character in a line, in which case it won't work.

Look through the program and note which commands are preceded by D$. These are the DOS commands, most of which deal with text files. Then let's look at the rest of the program.

The program structure is fairly traditional. Room is left for subroutines near the beginning of the program (although in a program this short it won't make any appreciable speed difference). Rem statements are used to set off the major subdivisions of the program. The main menu starts at line 500. The input routine is at line 1000, and the print routine starts at line 2000.

After we have dimensioned the array that we will store our addresses in, cleared the screen, and printed the header, we jump to the main menu. There we print our two principal choices—to enter names or to print the list—and use a get statement to accept a single character of keyboard input. Two if statements evaluate the response and branch to the appropriate line numbers in the program.

If we press E to enter addresses, program control jumps to line 1000. Line 1030 prompts the user to enter the first name on the list or to hit return if he or she is finished entering the list. Line 1040 then jumps to the input subroutine at line 80. A brief inspection will show why we used this subroutine instead of a more straightforward input statement (which would, admittedly, be a whole lot less trouble).

The subroutine uses three string variables. A$ is set equal to CHR$(34), which (as you'll see if you look it up) is the same as a quotation mark ("). B$ is the variable that eventually becomes our output temporary string. We use the get statement to load it with characters from the keyboard, one by one, examine them, and then add them to the end of B$. If, however, C$ comes back from the keyboard input loaded with a CHR$(13), then we know that the user has pressed return and is finished with the input, so we add a final quote mark to B$ and exit.

What does this routine have that an input statement does not? The main advantage is that it permits the user to include commas in the string. If we tried to enter *Dubuque, Iowa* with an input statement, we would get an "?extra ignored" message, and the portion after the comma would be lost. (Line 84 is necessary to handle backspaces—look at it carefully and you can figure out how the line works.)

Line 1040 tests B$ to see whether we entered any data. If the length of B$ is less than 3 (that is, just a pair of quotation marks), then control jumps to line 1090, which is where we create the text file. Otherwise, B$ is loaded into the string array A$(0,K), not to be confused with the string variable A$, which still contains a quotation mark. Arrays are very special animals and are not kept anywhere near mere variables.

The K is our counter. It tells us which set of addresses we are working on. Each time we add a four-line address, the value of K is increased by 1 (in line 1080). Line 1050 loads up the other three lines of the address and stores them in the array.

Once we have typed in all the names in our mailing list, we are ready to create the text file and save it off to disk. Lines 1090 to 1160 do this like so. First line 1090 sends out a DOS command to open a file, giving it the file name Mailing List. Then lines 1095 and 1100 send out the DOS commands to delete that file and then open the file all over again.

This may seem a bit peculiar to the uninitiated. But the only really convincing way to prove that this piece of redundancy is, in fact, quite necessary is to delete lines 1095 and 1100 and then run the program a couple of times with lists of differing lengths (not too long, as you'll have to do them over again once the point has been made). The reason for the two-time opening will emerge, as they say. So go ahead and try it.

The next DOS command is a write command. From this point on, anything printed via the print command will be written into the text file rather than printed to the screen. The only way to return to screen printing is to close the file with a DOS command, as is done in line 1160 after the array has been completely written out.

That's the end of the Enter Names main menu item. The second choice is Print List. This section starts at line 2000. The first thing the program does in this section is ask whether the list should go to the screen or to a printer. If you have a printer, you can send the output to the slot your printer card is in by inserting the DOS command *PR#n*, where *n* is the number of the slot (usually slot 1).

In this program, we also set the variable P equal to 1 if output is to be directed to the printer. We keep track of where the output is going so that, if it has been directed to the screen, we can pause every few names so the list can be read. Otherwise, the list will whip by at a speed to defy even the quickest eyes.

We do have one major problem in taking this file off the disk. That is, we don't know how long the file is, so we don't know how many names to read in. Why don't we just look at the value of K, you may ask. A very good question. But imagine, if you will, that after entering your data you turned off your Apple and went off to do something boring. Now you've returned and are ready to print your list. On goes the computer; in goes the program. But who was keeping track of variable values? Not I, says the Apple; please refresh my memory. Not I, says your program; it's not in my contract.

The problem is this. The loop between lines 2080 and 2100 reads in the data from the text file (input statements get their input from the text file instead of from the keyboard after a DOS read command). However, sooner or later, your Apple is going to try to input a piece of information and your data file is going to be all used up. Ordinarily, at this point you would hear a beep, the program would break, the Applesoft cursor would appear on the screen, and the message "end of data" would sit there staring you in the face.

A sad state of affairs. However, all is not lost. We can tell the computer that, if it finds an error, we would prefer to handle the problem from within the program. Look at line 2070. This line says, "If you should run across an error, please direct your attention to line 3000, which will tell you what to do."

Line 3000 starts out with another one of those pokes that makes for an oversight in Applesoft. All this poke does is turn off the on err command. After all, there are all sorts of possible error messages, and we don't want the program branching to line 3000 any time it feels the least bit cantankerous.

Line 3000 assumes that the reason program control jumped to it was that you ran out of data in your address file (there is a way to check this, but we'll take that up another time). Therefore, it is now possible to know the number of addresses in the file and set K equal to that number. Then program control is returned to line 2110, where we can get ready to print what we have read.

If we are printing to the screen and not to the printer, line 2130 includes a test that will pause the program after every three names. Can you figure out how the test works? The pause routine itself is a subroutine that starts at line 90. The only difference between this pause routine and a simple get statement, which would work almost as well, is that this routine does not put a blinking cursor on the screen. Instead, it uses a very obscure and seldom used Applesoft command called *wait*.

Wait tells your Apple to stop doing anything at all until something changes. *Wait −16384,128* means, "Stop all execution until memory location −16384 becomes greater in value than 128." As you may remember, −16384 is the location of the keyboard strobe, which is set to 128 plus the ASCII value of whatever key you press whenever you press it. The poke on line 94 is used to reset the keyboard strobe.

The final line of the print routine includes the DOS command PR#0, which tells the computer to direct all output from then on to the screen.

Next month, we will build a similar program using random files and add utilities to append names, to delete and edit entries, and to sort the list by last name or zip code. ■

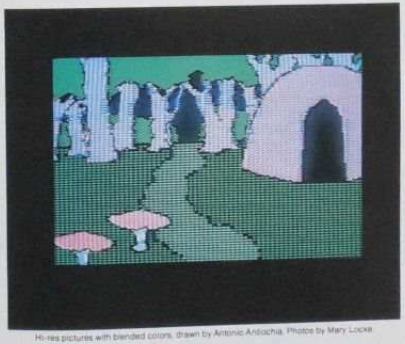
GRAPHICALLY SPEAKING

by Mark Pelczarski

This time around we'll take a look at hi-res colors on the Apple. There are only six: black, white, green, violet, blue, and orange. Sometimes you may hear that there are eight, but that's when you count two blacks and two whites. The extra black and white may be stored differently inside, but they look alike.

So how's it work? Some publishers of graphics utilities advertise that they can give you anywhere from twenty to more than a hundred colors. Well, what they're really doing is combining the existing six colors in various patterns to make it look as if there are more. It's effective. There are hi-res pictures done in several shades of a single color alone that look real nice. It's not the same as having a computer with a few hundred pure colors built in, but it's definitely stretching your Apple to its limits. The pictures in this article were created using this technique. To see how it's done, we can do a couple experiments. Then, compliments of Eagle Berns, we'll have a fairly simple machine language fill routine for you next issue.

First, drag out the machine language plot routine from the November 1982 issue along with the hi-res look-up table. (It was a twenty-three byte routine, and listing 5 showed how to enter it directly into memory.



Hi-res pictures with blended colors, drawn by Antonio Antiochia. Photos by Mary Locke.

Listing 2 created the look-up table.) Now, to fill the screen with black, all we have to do is put either $00 or $80 in every byte of the screen. These numbers give the two "different" blacks. In one, all the bits are off, and, in the other, all bits except the high bit (color flag) are off. See figure 1 for this and other color patterns.

Similarly, to fill the screen with white, you need to store $7F or $FF in every screen byte. Listing 1 shows a Basic routine to fill the screen with

```
5  PRINT CHR$ (4);"BLOAD LOOKUP"
6  PRINT CHR$ (4);"BLOAD PLOT"
10  HGR
20  FOR Y = 0 TO 159
30  FOR X = 0 TO 39
40  C = 255: REM 255 = $FF
49  REM COLOR IS POKED IN AT 24592 ($6010)
50  POKE 24576,X: POKE 24577,Y: POKE 24592,C: CALL 24578
60  NEXT : NEXT
```

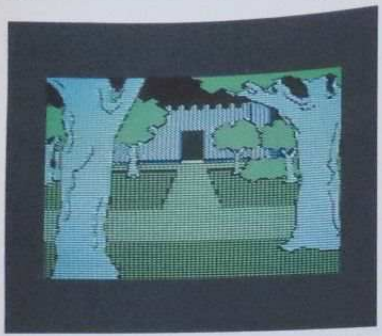
Listing 1.

```
10  HGR
20  FOR Y = 0 TO 159
40  HCOLOR= 7
50  HPLOT 0,Y TO 279,Y
60  NEXT
```

Listing 2

```
10  HGR
20  HCOLOR= 7: HPLOT 0,0
30  CALL 62454
```

Listing 3.



		Even Bytes	Odd Bytes
		0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Black	Hcolor=0	0 0 0 0 0 0 0 0 = $00=0	0 0 0 0 0 0 0 0 = $00=0
Black	Hcolor=4	0 0 0 0 0 0 0 1 = $80=128	0 0 0 0 0 0 0 1 = $80=128
White	Hcolor=3	1 1 1 1 1 1 1 0 = $7F=127	1 1 1 1 1 1 1 0 = $7F=127
White	Hcolor=7	1 1 1 1 1 1 1 1 = $FF=255	1 1 1 1 1 1 1 1 = $FF=255
Green	Hcolor=1	0 1 0 1 0 1 0 0 = $2A=42	1 0 1 0 1 0 1 0 = $55=85
Violet	Hcolor=2	1 0 1 0 1 0 1 0 = $55=85	0 1 0 1 0 1 0 0 = $2A=42
Orange	Hcolor=5	0 1 0 1 0 1 0 1 = $AA=170	1 0 1 0 1 0 1 1 = $D5=213
Blue	Hcolor=6	1 0 1 0 1 0 1 1 = $D5=213	0 1 0 1 0 1 0 1 = $AA=170

Figure 1: Standard Apple color patterns. Note that the bytes are displayed (and listed here) with bit 0 at left and bit 6 at right. Bit 7, the color flag, is not displayed. Since the bits are displayed in reverse order, the order must be reversed again to convert to hexadecimal—or you can turn the page upside down.

black or white, using the plot routine and the look-up table. Note that the value being poked into the hi-res memory is set in line 40. Because the routine returns to Basic after each byte, it is slow. In fact, filling the screen can be done even faster directly from Basic. This way, however, gives you more room to experiment later. To fill the screen by drawing parallel lines, instead of poking values into specific bytes, see listing 2. And for a real quick version of these first few examples (using the pure Apple colors) you would more likely want to use the method in listing 3.

Now, to get a color other than black or white, you may recall that only half the dots are used. Blue and violet have only the dots in the even columns set. Orange and green have only the dots in the odd columns set. Blue and orange require the high bit (color flag) to be set. Violet and green require that it be off. Look at figure 1 again to see the patterns.

One catch is that there are only seven dots displayed per byte. That means that if the leftmost byte on a screen line (byte 0; even) has only bits 0, 2, 4, and 6 set, it will show as violet (since 0, 2, 4, and 6 will fall on

```
5  PRINT CHR$ (4);"BLOAD LOOKUP"
6  PRINT CHR$ (4);"BLOAD PLOT"
10 HGR
20 FOR Y = 0 TO 159
30 FOR X = 0 TO 39
39 REM CHECK IF X IS EVEN
40 IF X / 2 = INT (X / 2) THEN C = 85: GOTO 50
44 REM X IS ODD
45 C = 42
50 POKE 24576,X: POKE 24577,Y: POKE 24592,C: CALL 24578
60 NEXT : NEXT
```

Listing 4.

screen columns 0, 2, 4, and 6). But for the next byte (byte 1), bit 0 corresponds to column 7 on the screen—an odd column. So, if the same pattern (0, 2, 4, 6) is used, that byte will show as green! (Bits 0, 2, 4, and 6 in that byte correspond to columns 7, 9, 11, and 13.) To continue with violet across the screen, byte 1 has to have a different pattern; bits 1, 3, and 5 should be set, with the others off. Moving to the right, byte 2 would go back to the original pattern: 0, 2, 4, and 6. Byte 3 would go back to 1, 3, 5, and so on.

Listing 4 uses the plot routine to put the correct byte values on the screen for violet. By changing the even and odd values to correspond with the other patterns in figure 1, you could also do green, blue, or orange. Of course, you could also still take the Basic shortcuts from listings 2 and 3.

Figure 1 shows the patterns for the standard colors. You can, of course, make up your own patterns. Try using different numbers in listing 4. Make up some; anything from 0 to 255 is legal. The key to a good-looking color is a regularly repeating pattern. The next example gives another hint for good patterns.

Here's where we play some tricks on your eyes. We're going to do different colors on alternate rows of colors. Not much explanation is needed; just watch what happens. Listing 5 uses the plot routine, mixing the colors orange and green. You can change the color patterns easily in the program to try others. Listing 6 does the same thing strictly in Basic and with only the six Apple colors. It's set up in a loop so that it will cycle through every combination of the six. You may even try playing with the color intensity control on your monitor or television. Tweaking it up more exaggerates the effect considerably.

Eagle Berns has written a color fill routine and an explanation of its algorithm for Graphically Speaking. You'll see his work in next month's issue.

```
5  PRINT CHR$ (4);"BLOAD LOOKUP"
6  PRINT CHR$ (4);"BLOAD PLOT"
10 HGR
20 FOR Y = 0 TO 159
30 FOR X = 0 TO 39
31 REM IF Y IS EVEN USE ONE PAIR; IF ODD, USE ANOTHER
32 IF Y / 2 = INT (Y / 2) THEN 40
34 IF X / 2 = INT (X / 2) THEN C = 42: GOTO 50
36 C = 85: GOTO 50
39 REM CHECK IF X IS EVEN
40 IF X / 2 = INT (X / 2) THEN C = 170: GOTO 50
44 REM X IS ODD
45 C = 213
50 POKE 24576,X: POKE 24577,Y: POKE 24592,C: CALL 24578
60 NEXT : NEXT
```

Listing 5.

```
20  FOR C1 = 1 TO 6
30  FOR C2 = C1 TO 6
40  HGR
50  HOME : VTAB 21: PRINT "COLORS ";C1;" AND ";C2
60  FOR Y = 0 TO 159
70  IF Y / 2 = INT (Y / 2) THEN HCOLOR= C1: GOTO 90
80  HCOLOR= C2
90  HPLOT 0,Y TO 279,Y
100 NEXT Y
105 PRINT "PRESS ANY KEY": GET A$
110 NEXT C2: NEXT C1
```

Listing 6.

Mind Your Business

BY PETER OLIVIERI

Happy New Year! Here it is, 1983. Since this is the time for making resolutions, let's make some for somebody else. Here are a few resolutions for the vendors of hardware and software:

1. Be it resolved that all products offered for sale to the public will include thorough, readable, and error-free documentation.

2. Be it resolved that product prices will appropriately reflect both the value of the product in question and the ability of the consumer to pay for such a product.

3. Be it resolved that the user will be provided with backup copies of each product purchased.

4. Be it resolved that we'll work diligently to make the computer easier to use and that we'll devise many more built-in features that smooth the way for the user.

5. Be it resolved that the manufacturers of printers will make available a letter-quality printer that prints amazingly fast for a price under $1,000.

6. Be it resolved that, when releasing a new or upgraded product, vendors will consider ways for people who own the earlier version to trade up to the newer one.

You can probably think of many more resolutions that could be added to this short list. But to include them all would consume an entire column. Since it's not unrelated, however, here are a few general comments about things it would be nice to see in the future.

The Apple II and the Apple III are wonderful machines, providing tremendous computer power to many people who may not have been able to take advantage of the technology before. Yet, regretably, these machines are not very easy to use. So-called ease of use is relative, of course, and certainly the Apple computers available now are easier to use than the earlier versions were. But, in all honesty, there's still room for improvement.

Picture, for example, the situation that executives or small business owners face when it comes to computers. These busy people have a time-consuming task ahead of them.

First, they must learn to operate a new computer. Now that's not all that hard if the machine comes with adequate documentation. Indeed, learning to use any new tool requires the investment of some time and effort. For business people, however, the work is just beginning. Now they must learn to use various applications packages.

The happiest computer users seem to be the ones who use the computer almost exclusively for a single task—word processing or spreadsheet analysis, for example. Where business people are concerned, the computer's main strength lies in its capacity to facilitate five main areas. These are word processing, database management, spreadsheet analysis, graphics, and the standard business functions (accounts payable, accounts receivable, and so on).

Many of the software packages for these different areas can "talk" to one another. It may be possible, for example, to use a graphics package from one vendor to display a portion of the table you developed using another package (*VisiCalc*, perhaps).

There are excellent spreadsheet packages, superb graphics packages, truly nice database management programs, and powerful word processors. But it is not always easy to integrate these packages together in order to take better advantage of the computer's power. And learning to use even one new program can be a lot of work.

Some software vendors have tried to do some of the work involved in integrating the various kinds of packages. The problem is that some of the programs that are part of these integrated packages aren't as good as other similar programs that are available separately. This puts users in a real dilemma.

Here are some recommendations. It's important that the computer be easy to use, and this implies menu-driven programs that, indeed, may not even need much separate documentation. All you'd need to know would appear on the screen in front of you. Perhaps printed documentation could be reserved for technical notes or examples demonstrating how the program works.

This ease of use would come about in two ways. First, programs themselves would be easier to use. It's intriguing to envision a single program (at least from the user's viewpoint) that, once loaded into the machine, would allow the user to do word processing, database management, spreadsheet analysis, and graphics. This would be a sort of "one-stop shopping" type of program. This is certainly not a thought that hasn't been thought before—several vendors are already working on such a beast. The intent of such a program would be to make the computer easier to use. And, since it's the software that makes the computer do what it does, an integrated program would achieve part of that goal.

It's not only the software that needs to be improved. From the user's point of view, the keyboard is the single most important aspect of the hardware. Apple Computer's addition of up and down arrows and a numeric keypad to the Apple III keyboard were significant improvements.

And much more can be done here. For example, a lot of letters come in from Business User Group members who rave about their use of word processing. Indeed, word processing is truly a marvel; typewriters, in the traditional sense, may soon be gone. And yet, word processing on a microcomputer is not easy. For one thing, the keyboard wasn't designed for word processing. On most standalone word processors, the keys are clearly labeled so that it is obvious which function each performs. There's likely to be a key labeled "delete a word," another key marked "underline," and so on.

These same handy features ought to be incorporated into the keyboard of the microcomputer. In the short run, this could be accomplished by means of stick-on labels. Or, perhaps some vendor will come out with a keypad that plugs into the game port and has on it only the word processing symbols for a particular word processing package.

It would be nice, too, if computer vendors would design the "ultimate" keyboard. Such a keyboard would take into account the fact that database management, word processing, spreadsheets, and graphics were the major uses to which the computer was being put. The keyboard would be appropriately designed—perhaps it would even have two keypads or interchangeable ones. The main keyboard might be much bigger than the ones Apples have now and might have plug-in modules. The design is, of course, up to the person who creates and markets it. The intent is clear.

These two factors—integrated software and improved keyboards—would enhance the ease of use of these computers tenfold. Imagine how great it would be if there were a button on your Apple keyboard marked *bar chart*. When pressed, this button would make the computer print, or ask you about, a bar chart of the data you'd been working with. It'll happen one day.

One matter of some concern is what will happen to Apple II and III users once the hardware and software reflect the kinds of changes we've been talking about. Well, perhaps some imaginative entrepreneurs will open up Apple II Trading Posts where users can trade in their old equip-

ment much the way they would a used car. Or perhaps the newer versions of the Apple will be priced so low that we won't care whether we can arrange for a trade-in. We'll just have to wait and see.

We've rambled at great length. It really is time now to get to the business at hand and proceed with the column.

Apple III Revisited. As you already know, the first version of the Apple III had some problems. The "new" Apple III has none of the problems of the original and is, in fact, an excellent machine. As was mentioned earlier, the Apple III keyboard was redesigned to include some desirable features. Now, if the ProFile hard disk that Apple Computer produces just had a more effective means for providing for backup of files, the Apple III would indeed be a remarkable computer system.

The Apple III's only real drawback is some of the manuals that go with it. It's not that they're poorly written, it's just that some of the manuals aren't very easy for a first-time user to get through, especially when they start discussing such things as "device drivers" and "configuring your system." This is good stuff, but it doesn't make it easy for the new user to get up and running.

Some people say that when they read the Apple III manuals they start feeling as though they need to become programmers in order to understand what's being said. What they're getting at is how frustrating it can be when you have to learn a whole lot of new terminology before you can begin to use a program.

Much that's currently included in manuals could be left unsaid. For many users, it's just not necessary to know everything there is to know about a program. One way of making things simpler would be to have two sets of manuals. The first set would allow the prospective business user to get up and running quickly without having to know a great deal about the inner workings of a program. The second set would be for people who want to "get inside" of things—to learn to perform special tasks, write programs, customize applications, and so on. In many cases, one manual just can't be all things to all people.

The preferred Apple III business system would include an Apple [illegible] with built-in disk drive, a ProFile hard disk, a Monitor III, and a printer. Apple Computer provides manuals for each component of this system and also offers a variety of Apple III applications packages. We'll discuss some of these packages now and will continue to review Apple [illegible] applications packages in future columns.

Lots of manuals are provided with such an Apple system. The two main ones are the *Apple III Owner's Guide* and the *Standard Device Drivers Manual*.

The *Owner's Guide* describes the Apple III in depth, explains how to put the system together, outlines some of the fundamental steps involved in operating the machine, and provides more detailed information about the machine itself. The sections on putting the machine together are well done and include pictures to help the uninitiated user connect all of the components properly. A disk that gives a helpful demonstration of some of the Apple III's features is also provided.

Next the manual describes SOS, the Sophisticated Operating System. This system is, in many ways, comparable to CP/M. The manual covers such topics as file-naming conventions, selecting and identifying different devices that are attached to your machine, and how directories are set up. (While all this is very useful, it's an example of what we were talking about before—information that may not be needed by all users.)

The *Owner's Guide* then describes the utilities disk that's supplied with the system. This disk allows you to work with files (to transfer, copy, and so on), explains how to configure your system (tell the Apple all the parts you are using), and discusses how to format and use disks. The appendices describe error messages, how to make the Apple III think it's a II, the specifications of the input and output ports on the back of the machine, and how to care for disks.

The Standard Device Drivers Manual is a somewhat more technical document. It begins by describing what device drivers are. Essentially, *device drivers* are programs that communicate with the external devices that are connected to your Apple. When device drivers are used, a wide variety of equipment can be connected to the machine fairly easily. The manual also contains sections on using the console, with its associated screen control commands, and on using graphics, audio, and the RS-232 standard interface. It is, really, a reference manual for the Apple III.

The *ProFile Owner's Manual* is short but complete. As is the case with most manuals that are supplied with a piece of hardware, the first sections of the manual deal with how to install the equipment. In the case of this disk drive, you must install the hardware, which simply means inserting a card into one of the Apple's slots, and you must install the software so that the Apple knows that a ProFile is attached.

There's a very nice, though short, section on how the drive actually works. With the exception of a few appendices that give some operating characteristics and some Pascal information, that's all there is to the manual. The interesting thing is that not much more is needed. The ProFile is extraordinarily simple to use. You don't really need a manual.

In addition to these manuals, Apple Computer has manuals for specific products. For example, there are two volumes on Apple Business Basic. If you're doing any of your own programming (or you expect to), you may find these books quite helpful. The same is true for the Apple III Pascal manual.

Then, of course, there are the special applications packages. Next month we'll be looking at the support materials for *VisiCalc III*, *Apple Writer III*, *Business Graphics*, and the Apple III mailing list program.

Graphics. Clearly, most games are more exciting when they include graphics. In a similar vein, graphic presentation of business information can be more interesting. In a business setting, the most common means of presenting data is in the form of a table. People who need to examine the data, draw conclusions from it, or make decisions based on it usually have to wade through a lot of numbers in a table in an attempt to find the relationships that are important. That very same data expressed as a graph can make the process of finding relationships and examining them a good deal easier and more effective.

In the business setting, the most common graphs are a bar chart (sometimes referred to as a histogram), a pie chart, and a line graph. In a bar chart, data is displayed on a standard two-dimensional axis, with the

height of the bars representing the magnitude of a particular number (for example, dollar sales in 1982). A pie diagram looks like a pie (probably apple). Each slice represents a percentage of the whole pie. Thus, if a pie chart were to be used to represent a firm's total expenses, one slice might be the proportion of the total that was allotted to travel. A line graph is simply one or more lines plotted on a graph. These are the graphs most people are familiar with. Line graphs do a great job of displaying growth trends.

One of the nicest graphics packages we've seen is *Apple II Business Graphics*, developed by Business and Professional Software and available through Apple Computer. The highlight of this package is how easy it is to use.

Picture this: After you load the disk, a prompt symbol appears on the screen. If you wish to enter data, you simply type the word *edit* and then begin entering the data. Let's say you wanted to enter ten years' worth of data on sales. You'd type the data as follows:

```
"1972",2000
1973,5000
1974,10000
1975,4000
1976,7000
1977,12000
1978,9000
1979,15000
1980,17000
1981,20000
END
```

When you type *end*, the data you've entered is stored on your disk. If you want a bar chart of your data, all you do is enter the command *draw bar*. Lo and behold, after a slight pause, a beautiful bar chart will be drawn on the screen (see photo 1). You can control the color of the bars and how many bars appear for each year (which makes it possible for you to display two or three years of data simultaneously if you want to). You also control the titles of the charts.

Photo 1.

Type *set title* and a title can be typed on the graph. If you enter *set vertical title*, a vertical title can be entered, while typing *set horizontal title* allows you to move a title all around a graph until it's positioned exactly where you want it to be. (By the way, you can enter all of these commands by using only two letters. If you want to save your picture, you merely type *save*. If you'd like to see what the same data looks like in the form of a line graph, type *draw line*. And (you guessed it!), if you want to see a pie diagram, you simply type *draw pie*. It's precisely this simple vocabulary that makes this package so easy to use.

Apple II Business Graphics requires an Apple II with 48K of memory, at least two disk drives, and either a language card or a 16K add-on memory card. A color monitor is quite useful and, of course, having a printer allows you to make copies of your charts. The package works quite nicely with a variety of printers.

The user manual is clearly written. It guides you through the development of several graphs as a means of teaching you how to use the program, which turns out to be an effective way of managing the material. In addition, a variety of help commands can be used. Thus, if you weren't sure what to do at a certain point, entering *help titles* would get you an explanation of how to use titles and their appropriate commands.

This is the kind of package you can get started with immediately. With very little instruction, you can prepare graphs, display them, save them, and print them. And, if you wanted to use the package for more complex applications, you could learn to do this by studying the sections of the user manual that deal with more sophisticated tasks (manipulating data, using sine functions and logarithms, using some of the math functions in the package to alter data, and so forth).

Furthermore, *Apple II Business Graphics* "talks" to files from other sources. You can draw graphs from data found in various sources, including DOS 3.2.1 text files, DOS 3.3 text files, Pascal/Fortran text files, *VisiCalc* print files, *VisiCalc* DIF files, and *Apple Plot* data files.

Slide Shows. As if all this weren't enough, various companion packages can be purchased that work in combination with *Apple II Business Graphics*. One of the first you're likely to consider is *Screen Director*. *Screen Director* essentially enables you to be the director of your own slide show. The package allows you to put the pictures (graphs) you've created into a "slide tray" for later viewing.

The slides in the tray are given various characteristics such as display time, fade-in, or flash-on. The tray can include "text screens" that you can create from within *Screen Director*. You can select from a variety of font sizes and colors and then create introductory or comment slides for inclusion in your tray. Slide shows can be set to run automatically and continuously, and trays can have slides added to them or taken out.

You don't have to have *Apple II Business Graphics* in order to use this package. *Screen Director* will group together high-resolution graphic images from a variety of sources, including *Apple Plot*, *VisiPlot*, and *The*

Coloring Board. You then use short commands to indicate how these pictures are to appear. To present a group of screens, you can use either the keyboard (arrow keys), the game paddles, or an adapted carousel-type projector button to go forward or backward through your tray.

You do need the same equipment as was mentioned earlier, including a language card or a 16K card, but such an addition gives your Apple more power for a variety of chores.

Business and Professional Software offers a wide variety of user services. For example, there is a *Printer/Plotter Installation Kit (PIK)* available that allows you to tie various packages to a large number of peripheral devices. Furthermore, it's possible to send your slides over the telephone to the company's offices and have computer-enhanced 35mm slides prepared from your originals.

B.U.G. Membership in the Business User Group continues to grow. Names of members have been entered into a database/mailing list package, but don't worry; you won't be getting any junk mail from this list. It never leaves the office; it's for B.U.G. use only.

We haven't yet done anything as formal as a meeting, but that may come someday. Right now we just share our successes and failures with one another and try to help each other out. If you have something to share or a problem you'd like solved, write to B.U.G. in care of *Softalk*. In fact, how about offering some suggestions as to what roles the B.U.G. might play. Could we have a collective impact on Apple? On some of the vendors of software we all use?

Think before Buying. You'll recall that we've talked in this column about how important it is to be able to try out software before you buy it. Well, Robb Jacobs, a B.U.G. member who read that particular column, is opening a store in Bloomington, Minnesota, where people can do just that.

A franchise of Software Centres International (Culver City, CA), the store will be called The Software Centre and will sell business packages, utilities, education packages, and entertainment programs. You'll be able to try out software before you buy. This means, for example, that you could look at, use, and ask questions about three or four different word processing packages in the store before deciding on which one is right for you.

Trying before buying is a great idea that's well worth supporting. It requires some investment of time, effort, and money on the part of the supplier of such a service and can certainly help users out a lot.

Help B.U.G. Out. Some readers have questions they want to ask of other readers. If you can help out with any of these, please send along your comments, noting whom they are for. Comments will be forwarded to the right person or mentioned as part of a column. Now for some questions.

William Burkhard of Woodstock, Georgia, asks, "Has anyone had experience with the low-cost letter-quality printers on the market? How are they standing up?"

"I need the protocol setup to allow file transfer between an IBM Displaywriter and the Apple III using *Access III*," says M. Ken Hockabout of Alameda, California. "Also, how do you send Displaywriter stuff to the III?" He also wonders, "What is interlacing and how does one get nicer screen output (nonsegmented letters)?"

"Has anyone used *ScreenWriter II* in conjunction with the *General Manager*?" (Peter French of Dayton, Ohio, wants to know.)

And John P. Schnell of West Olive, New Jersey, writes to inquire, "Is there any way to print superscripts using *Apple Writer III*? Subscripts are easy, but we find that control-V escape-S followed by anything only prints subscripts. We would be grateful for any help on this."

Well, this column has been a bit longer than usual this month; there was a lot to say. We'll just have to have a special magazine for business users (just kidding). Take care, everyone. Look forward to hearing from you and to having you back next month. ■

Apple Computer, 20525 Mariani Avenue, Cupertino, CA 95014, (408) 996-1010
Business and Professional Software, 123 Binney Street, Cambridge, MA 02142, (617) 491-3377.

And It's Up We Go!

BY MELISSA MILICH

We are only left on the ground if we choose to be.

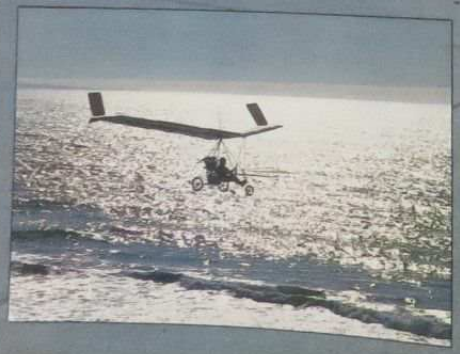
"Clear prop!"

Nothing in sight except dirt runway, hay-mown fields, china-blue sky. No kids, dogs, or other ultralight aircraft.

"Clear!"

The pilot pulls the starter rope. It doesn't start. He yanks it again and the sputter makes contact with the engine. The throttle is pushed, and the machine, a strange reincarnation from Kitty Hawk 1903, plows off, shaving the runway in one smooth stroke.

Jonathan Livingston Pterodactyl. And then the miracle of liftoff—little more than an equation, lift versus weight and thrust versus drag, that few people other than physicists and aviation experts ever completely understand.



In the Ultralights Raised on Apples

Look! Up in the sky! It's a bird, uh, a plane, uh. . . . Ah heck, whatever it is, it's weird. Actually, it's just ultralight guru Jack McCornack in one of his latest creations buzzing over the California coastline near Watsonville. Neither bird nor conventional plane, McCornack's experimental Apple-enhanced Pterodactyl is testimony to the recent explosion in private homebrew aviation. McCornack and others are picking up where the Wright Brothers left off, flying for the fun and the thrill of it.

The mathematics works, and machine and pilot are airborne, no longer tied to the ground.

These ultralight aircraft are strange birds indeed. They are engines attached to hang-gliders, motorcycles with wings, souped-up lawnmowers. However these new ultralights are described, the new pilots, veterans of Boeing 707s, DC airglides, antique biplanes, and torn parachutes, say flying an ultralight is as close as you can come to being a bird.

And it is these same veteran pilots who confess they'd never felt acrophobia until they'd flown an ultralight.

There's not much metal up there. From the ground, looking up a thousand feet, the ultralight seems as if it could be held together with toothpicks. But stainless steel cables, aluminum tubing, shock absorbent wheels, and strong dacron sailcloth-covered wings in rainbow colors make up a deceptively sturdy body.

Even so, say the pilots, you are *absolutely naked* up there. There are no brakes. You don't need brakes, in theory, on something this light. You don't even need a pilot's license—the Federal Aviation Administration hasn't yet attempted to control ultralights. As for parachutes, nobody seems to bother; there's hardly enough room for one.

This particular ultralight is called a Pterodactyl—the same name as the flying reptile that became extinct fifty million years ago.

No brakes, no license, no parachute, and a Pterodactyl. What is there in this barebones aircraft to hang on to for even the least sense of security?

It was designed with the help of an Apple II.

"Computers can't do everything!" Jack McCornack, designer of the Pterodactyl, says loud and clear.

Prehistoric Pilot. McCornack hovers and swoops over the field adjacent to Pterodactyl Limited, his two-year-old company by the sea in northern California. He lands on one wheel, spins around, and zips off again, sending a haze of dust floating over the small group of earthlings watching him with envy.

"Jack might not look sane, but he knows exactly what he's doing," says fellow ultralight pilot Jim Campbell.

Campbell is certainly qualified to talk about sanity; he's often been charged with its opposite lately. Campbell and a friend, Pat Trusty, are planning to travel around the world by Pterodactyl, a trip they figure should take them about a year and a half to complete. In theory, such a trip is possible; the farthest their ultralights will have to fly over the ocean in between land on their easterly route is four hundred miles, and a gas-saving Pterodactyl gets forty-five miles per gallon on a ten gallon tank. It's called the Blue Goose Route, and it will take them over Maine, Newfoundland, Greenland, Iceland, the Faeroe Islands, and Scotland, then across the English Channel to central Europe, over Siberia, and, finally, across the Bering Strait to Alaska. Among other things, says Campbell, he wants to touch a pyramid and sleep next to the Great Wall of China.

When people buck the comfortable, common life like that, they're laying themselves open to a lot of criticism, and Campbell and Trusty have had their share. But they don't think they're missing anything. Are they crazy? A lot of people think so; Campbell and Trusty disagree.

In Watsonville, California, perhaps, lies the answer.

"Watsonville? What's that?" That's what the travel agent said, and, when a travel agent doesn't know, it makes you wonder.

Watsonville is on the map. Well, on some maps. Approximately sixty miles south of the Silicon Valley and not quite five hundred miles north of Disneyland, it's there—a pinpoint. The Pacific Ocean and some of the most fertile agricultural land in the world surround it on the west and east. A nice place to drive through on your way to San Francisco, but rather inconsequential compared to all the booming metropolises nearby.

That's what outsiders think. But the local chapter of the Friends of the Library won't let you forget that John Steinbeck wrote about Watsonville. And the chamber of commerce still remembers when *The Dating Game* sent a lucky couple here on a dream date. Not to mention that Kim Novak stayed at the Hotel Resetar on Main Street in 1954.

Watsonville is the center of apple production in California, and in this part of the country that means ice-cold red Delicious, sun-warmed golden Delicious, crunchy pippins, McIntosh, Bellflowers, and Granny Smiths.

Now it's the home of some electronic Apples and their lighter than air product with the Pterodactylous name. Watsonville finds itself torn between its wholesome, home-loving apple growers and its foolhardy, risk-taking Apple users.

Jack McCornack is a little bit of both.

This Apple's Got Guts

Take an Apple up in one of those things? You've got to be kidding. What do you do with it? Play *Phantoms Five*? Write airborne poetry with *Screen Writer II*? Balance the checkbook with *Home Accountant*?

Take a good look at the picture. That's an Apple II Plus, monitor, disk drive, and Apple Juice power supply bolted to a wooden mount with foot-long bolts and protective pads. On the two-seater Pterodactyl pictured, the Apple sits where the passenger normally would. The control stick for the canard and winglets is managed with the right hand while the pilot reaches over with the left hand to type on the keyboard.

Jack McCornack of Pterodactyl does not play games or write poetry or manage finances while he is soaring and swooping through the air. He uses the Apple, which runs off the engine with the help of the Apple Juice, to document test flights under different conditions.

McCornack is continually experimenting with different designs and is particularly interested in the two-seater's performance during varying degrees of wind turbulence. Instead of risking two people, he chooses to take the Apple along—and it's useful, to boot.

The more you increase the payload of an ultralight, the more crucial certain factors become. McCornack is thinking about expanding the wingspan of the Pterodactyl and these test flights give him valuable information to start basing designs on. The software McCornack uses is generally the work of his employees and students, nothing off the shelf.

Another scheme McCornack has in mind involves the use of ailerons, the surfaces on the outside trailing edge of the wing that control roll. Up to this time, McCornack's Pterodactyls have been limited to the canard (the elevator mounted on the front of the plane) and the winglets or rudders (the control surfaces at the wingtips). These are fine if there is not much crosswind or turbulence.

Ailerons on a conventional aircraft and other brands of ultralights provide lateral control that would come in handy on a Pterodactyl. McCornack is working toward a version of the two-seater in which you can control ailerons with a joystick hooked to the Apple. In a normal plane, ailerons are controlled by the stick and the rudders are controlled by foot pedals.

The computer would control servo motors that activate the ailerons. The computer would also act like a flight instrument giving McCornack information on control force, effectiveness, and efficiency, avoiding the more traditional seat-of-the-pants decision making.

McCornack is currently investigating navigational software written by a programmer in Washington. While flying, McCornack can determine the proper heading to achieve better fuel economy. He'll be able to do what they call inflight planning, taking advantage of shortcuts. "It's off-the-shelf software, but you won't find it in your local computer store," McCornack says.

Once again the Apple proves it's an incredibly durable machine. Both takeoff and landing in a Pterodactyl are bumpy, like riding a bicycle on a dirt road at twenty miles an hour. It's a good test of mettle even for the most ruggedly designed microcomputers. McCornack has had few problems, other than the unavoidable hazards like landing and then rolling into a mud puddle. Needless to say, his airborne Apple may be the first that positively has no need for a cooling fan.

McCornack stresses that the two-seater with the Apple is just an elaborate testing device for Pterodactyl's research and development department, and he has carefully embossed the side of his flying machine with the label "experimental."

In the meantime, he has something to do while he's zipping around the skies of Watsonville.

"I've gotten real good at flying with one hand and typing with the other," McCornack boasts in his usual unimposing manner.

Chariots of the Clouds. Although it's hard to get to know him when he's up there and you're down here, McCornack believes in moderation. A balance. It's not always apparent, but it comes through at the important places and always on time. It's true, ultralights can be daredevil, harum-scarum bucking broncos, but they can also be gazelles in motion, Adonis riding Pegasus, and the Lone Ranger on Silver. It's all in the way you take control.

The only thing you can rely on in a one-seater model is yourself. Take comfort in the fact that ultralights are well designed, but, as far as security goes, that's it. You can kill yourself in a minute.

Consider McCornack, the pilot who designed the thing, an ex-motorcycle racer with speed still in his blood, and a philosopher. "There is no excuse for trusting your engine to keep you alive. Figuring an average lifespan of seventy years, you've got less than one chance in a million of dropping dead in any given hour. Why hitch your life to a piece of equipment with less than a thousandth of your durability?"

So you've got to keep your wits about you in case the engine conks out while you're three miles in the air. You'll come down for sure, but managing the aircraft sanely increases your chances of landing safely and walking away.

"The basic goal of ultralight pilots," says McCornack, "is to be able to tell our grandchildren about it when we're old and gray and feeble. That means that for years and years and years while we're getting old and gray and feeble we can live our lives with a bang."

Although the FAA is seriously considering changing this little loophole, present regulations state that, as long as an aircraft can be foot launched, a pilot's license is not required. Foot launching is possible in a single-seat ultralight, but you practically have to run with each foot flying over your shoulder—fast, huge strides—somewhat like a goose. If the aircraft is capable of taking off after that, you don't need a license to fly it.

"Look, Ma, No License!" Pilots don't need to prove this capability when they buy a standard model ultralight, and, unfortunately, they don't need to prove any ability at all. That's why the FAA is starting to worry about the increasing popularity and availability of one-person ultralights.

To fly a two-seater Pterodactyl, you need at least a student pilot's license, because the planes can no longer be foot launched. And, of course, there are other necessary steps a pilot must take to fly in certain areas. Any time you enter controlled air space, you must have proper radio equipment for communicating with air traffic controllers.

People are attracted to one-person ultralights because no license is required (yet), they're not too expensive (you can take home a Pterodactyl for about three thousand dollars), and they're very portable (an ultralight packs up easily and fits nicely atop a Volkswagen Bug). Besides all that, a person who has never flown before can learn to fly an ultralight in a week.

"They're too easy to fly," says McCornack; "that's what makes them so dangerous.

"You can learn to fly an ultralight from a book, but there really is a need, even for a *brilliant* person, to have supervision."

What's it like learning to fly an ultralight? Imagine learning to ride a bicycle on extremely thin ice. Some folks say ultralight flying is vaguely like driving a car, once you get used to having no brakes. Occasionally, it takes experienced pilots the longest to master the concepts; they have to unlearn a lot of reactions ingrained by conventional aircraft flying.

As in most things, an instructor is involved in the educational process, but the learning is done by the student and the teaching is done by the aircraft. McCornack has written a flight manual and developed a step-by-step method for learning to fly one of his Pterodactyls. A student starts learning as soon as she or he starts exercising judgment.

"You can have the best instructors in the world, but those instructors have no control over your destiny," McCornack explains. "They can tell you what to do and you'll agree to do it, but once the instructors step out of the way and the plane taxis down the runway you're on your own."

All ultralight aircraft are unforgiving of incompetence; but a Pterodactyl offers a lot of crash protection.

"This sport is as dangerous as you want to make it," McCornack says. "In my book, a recreational aircraft should be very resistant to crashing. If it does crash, everything possible that could absorb energy by its own destruction should do so. In other words, if it does break, it doesn't break you."

Twelve O'Clock High and Designing. And this is where the sit-at-homer, stay-at-homer part of McCornack comes into play. He designs these aircraft on his Apple computer, diligently, to make the most efficient aircraft that will give the pilot the best possible chance in case something goes wrong.

The plane is designed to absorb a lot of impact. Rough field landing gear, aluminum tubing axles intended to bend, and wheels with extra suspension take the shock first. The pilot's lower body is surrounded by a cage, and the power plant structure is directly behind it for further protection. A compression system keeps the engine intact so it doesn't break loose and smash into the pilot. A harness keeps the pilot inside the cage. McCornack says a helmet is essential.

The speed of a Pterodactyl is another built-in safety feature. At top speed, a Pterodactyl Fledgling can go fifty-five miles per hour, but pilots usually fly them at thirty-five, a healthy bicycling clip. The pilot in an ultralight has a better chance of surviving a crash than someone traveling in a much faster vehicle.

The last thing McCornack wants to do is soft-pedal the danger; the fatality rate for Pterodactyl pilots is roughly equal to that for pilots of conventional small airplanes.

You can take a fifty pound bag of dog food, put it on the seat of a Pterodactyl, push the throttle all the way, follow it in another Pterodactyl till the first one runs out of gas, and ninety-nine times out of a hundred, McCornack says, you'll find it sitting on its wheels without a scratch.

Nine times out of a thousand you'll find the aircraft with a few bent parts, a wheel broken off it, and the dog food bag lying twenty feet away. One time out of a thousand, you'll find a heap surrounded by an acre of dog food.

"A severe gust of wind or just a thermal barrier can catch it in the wrong place at the wrong time," shrugs McCornack. "On that one time out of a thousand, if there's no pilot there to correct the plane's sudden change of altitude, just to make a simple turn . . . it's just *so* easy to make hamburger out of something."

McCornack has investigated a few accidents in order to increase the safety of his own models. "As a consumer I know what I would like to have, and I make what I want and see if anybody else wants it."

The computer is gradually replacing pencil and paper at Pterodactyl Limited. McCornack designs, plans, and rearranges his ultralights on the Apple.

"I use *VisiCalc* for this much the way business people use it for trends," he says, that is, to get immediate information on such things as what effect changing the thickness of the tubing will have on the expense and labor as well as the strength and weight of the aircraft.

McCornack adds here that, by the way, he didn't invent the Pterodactyl—the Wright Brothers did. "Designing an airplane is mostly a matter of taking available knowledge and cataloguing and correlating it against possible design goals.

"You want to have low drag, better fuel economy, more excess power available for speed changes and climbing, and the ability to carry larger payloads. One way is to increase the wingspan; but then, to balance it out, you have to decrease the weight of the rest of the aircraft. When I'm trying to find the optimum wingspan for minimum drag, I can enter the variables into *VisiCalc* and see the results," McCornack explains.

The Plain Truth. Not a computer expert when he started and not one now, McCornack has had some bad luck and some good luck with the Apple. On the bad side are his early dealings with a local computer store and his never-ending search for a good graphics program. Rarely has he found a package that delivers what the advertising promises. McCornack particularly berates the manuals that accompany several of the most popular graphics packages currently available, saying that they're great if you're as smart as the programmer.

On the lucky side are his experiences with a digitizer, which he used for producing the structural diagrams in the Pterodactyl's owner manual. A digitized image of an assembled joint is stored in the computer and then printed out. A clear piece of acetate is placed over the printout and the parts are outlined. He can even speed up the process a tad by tracing the image directly off the monitor.

McCornack and his wife Toni James also use *Apple Writer* for producing the assembly instructions in the owner manual and for writing a

quarterly newsletter for Pterodactyl owners called *The Ptimes*. Apples are also used in the business side of McCornack's company, managing the inventory, the mailing list, and the accounting.

The story of Pterodactyl Limited and how it got one Apple, two Apples, and eventually seven Apples is an Algerlike success story—but not without problems. McCornack and James's main gripes are directed toward the computer industry in general. To understand how the problems began, you need to visualize these two pioneers.

Meet Jack McCornack: Pinstriped overalls, a bright T-shirt with "Have You Hugged Your Pterodactyl Today?" scrawled across it, and blue running shoes. He also has a beard, mustache, and red hair, the kind that would be unruly if he allowed it to get much longer. Not too many pretensions here, by McCornack's own admission.

James wears the same type of clothes, but somehow on her they look more like a coordinated outfit. Neither is sloppy; just casual.

Even so, the first time they walked into a computer store in a neighboring town, both McCornack and James were shunned by the employees.

"Like we were a couple of hippies looking for a handout," recalls McCornack.

With cash in their pockets, they were trying to buy a power supply for their computer and were ready to be talked into buying an Apple III. But the store's employees wouldn't give them the chance, didn't even look at them.

So McCornack and James left, made a lot of excuses for the store's behavior, and returned another day to try again.

"But still," says McCornack, "we didn't even get to first base. Someone tried to sell us a game but didn't take us seriously when we expressed interest in buying something more complicated."

Snobs and Slobs. McCornack believes snobbishness still runs rampant in the computer industry. And he and James agree that it didn't help them get taken seriously when they informed the store that James was the head of the company. (McCornack does all the design work, but James quietly does everything else.)

"Computer stores are going to have to get used to the fact that the best customers are on the leading edge of the wave of change. Customers should be allowed to dress any way they please."

At that point the couple could have said "forget it" and ordered their equipment through a catalog, but Pterodactyl wanted to establish a good working relationship with a retail store.

"You're buying a lot of expertise along with that equipment and it's very worthwhile to expend the effort to get person-to-person contact. A lot of software is unintelligible unless somebody is there to help you."

So, on their third try, they not only acceded to the store's values, they threw them at it—arriving purposely ostentatiously in their vintage Lotus 1966 and wearing their most establishmentlike clothes.

It worked. The two from Pterodactyl were waited on immediately. In succeeding trips, they got everything else they wanted: modem, camera clock, digitizer, *Wizardry*, *Zork*, *Crossfire*, and seven complete Apple systems.

McCornack didn't see their actions as giving in.

"Sometimes playing their game is fun. You get the fun of beating the system. Of course, it's a stupid system."

And that's an important part of balance—knowing you can't be one way all the time. Sometimes you have to bend a little, but still, in all fairness, the bending should go for both sides. In the future, it will, and people like McCornack and James will do fine being themselves.

James is excited about the future. As she sees it, our grandparents had a lot of changes to get used to: television, jet-powered cross-country flight, high-tech medicine.

"While people older than thirty are suddenly finding it necessary to get used to computers, today's children are growing up with them," James observes.

"You've got to wonder what changes in technology the kids today are going to see in their lifetime."

Prehistorically Modern. Pterodactyl Limited seems to qualify as a new-age company. There is a woman at the head of it making all the decisions, computers behind the scenes doing all the tedious work, and those Pterodactyls—they get high mileage per gallon of regular gasoline, and McCornack has started to experiment with alternative fuels. Are ultralights the transportation of the future?

Most of the Pterodactyls sold are for recreational purposes, but a few consumers get tax write-offs when they claim them for business purposes—pulling banners and crop dusting are frequent uses—and a lot of new-age farmers are using them for cattle counting forays or checking on the wheat rust.

McCornack hears about some new creative use for them everyday, but he doesn't believe ultralights will ever replace the automobile.

"Aviation won't tolerate the sort of behavior that takes place in an automobile. There's no such thing as a fender-bender in aviation accidents."

McCornack knows some people he wouldn't trust near an ultralight; to others he offers free flying lessons. "I don't want to put everybody in the air, but sometimes I get a strange satisfaction when I meet somebody who wants to do it."

Morning dawns early in Watsonville. The big Bud Antle trucks rattle by on their way to the cold storage buildings; people with brown faces under straw hats are already out in the fields cutting lettuce; and a large group of men and women carrying empty lunch boxes are just now punching out at the canneries while the next shift punches in.

It's sometime after seven at Pterodactyl. Inside, the Apple computers are chugging out the mailing list and spitting out the inventory. Good computers. But let them be. Outside, the field is wet and the worms are out.

It's lesson three: Taxiing, takeoff, and landing. The speed isn't much more than thirty miles per hour, and the plane was never more than a couple feet off the ground, but the burst of cool air fresh in the face and the thrill of seeing the ground move by underneath brings a roller coaster grin to the new pilot.

McCornack looks worried. "Wait. There's something on your teeth," he says.

In the excitement, an airborne worm landing right in the mouth is no big deal. Amelia Earhart probably went through all this and more when she was learning to fly.

Grin and Bear It. There are definitely hazards involved in flying—like smiling too much. So why do people persist in doing it?

Somewhere, sometime, you may have learned and recited out loud the venerable Robert Louis Stevenson poem, *The Swing*.

> Up in the air and over the clouds,
> Till I can see so wide,
> Rivers and trees and cattle and all,
> Over the country side. . . .

Stevenson wrote *The Swing* in 1879, long before the Wright Brothers came along. Wouldn't it have thrilled him to ride in an ultralight! Poets, some say, have a different view of the world, they see different things. Flying in an ultralight can make anyone a poet.

The most thrilling thing that ever happened to round-the-world pilot Pat Trusty was a pigeon coming along and flying for several miles right along with her. In 1979 when McCornack and his friend Keith Nicely were flying in Pterodactyls across the country, Nicely saw a couple of mountain climbers scaling the Grand Tetons. So he swooped down and treaded air while he had a conversation with them. It's a different world up there.

But it's not exactly a surrealistic experience. Before you can even get off the ground in a Pterodactyl, you must adjust the swing seat so you are perfectly balanced in it according to your weight. (Again, this balance—it keeps cropping up.) But, if you keep the equilibrium, anything is possible—even a takeoff—in these amazing flying machines that weigh less than most of the pilots who fly them.

You don't have to be an aviation expert to learn to fly a Pterodactyl; still, you must exercise caution and common sense. McCornack is something of a daredevil himself, but he's an excellent teacher emphasizing both the dangers and the joys of flying an ultralight. He wants to see more people in the air flying well-designed, reliable machines in a safe and sane manner.

The High and the Flighty. Once you're up a thousand feet, you can start to see things more clearly or surrealistically, depending on the poetry you write. But you can't forget the wonder of human flight. It still seems impossible.

Back to earth, McCornack helps make it all more real with a bit of aviation history and some of its surprises. "World War I messed up aviation," he says, "*really* messed up aviation."

In the early days, McCornack recounts, aircraft served no practical purpose; they were purely for recreation, and the developmental work on aviation took very creative avenues.

"But then World War I came along and people discovered that airplanes were useful for killing people. After that, high-status planes were fast, powerful, maneuverable—all factors that made them more efficient for killing.

"And when Johnny came marching home again after the war, the high-status pilot was rugged, daring, craving excitement, and cool under pressure—efficient for killing."

No balance here. A shifting to one side. It was fifty years, says McCornack, before people were willing to risk developing aircraft solely for recreational purposes again.

Other people besides the military are attracted to aircraft for dangerous purposes. For instance, ultralights are very good tools, perhaps the best, for daredevils, says McCornack, a former daredevil himself.

The Devil and Jack McCornack. A weekend motorcycle racer from not so many years back, McCornack remembers the psychology of the daredevil fairly clearly.

"What daredevils want is public recognition for being braver than the public that's recognizing them. I can't explain that attitude, even though at one time I had that feeling. One of the symptoms of being a daredevil is claiming, 'I have such high standards of courage that what you see is remarkable is nothing to me.' "

But daredevils, says McCornack, are likely to be very undisciplined students, and he doesn't particularly want to teach them. He's more interested in giving lessons to people like fifteen-year-old Lee Raymond who had to ask his mother first.

McCornack does, however, credit his racing background with giving him a good awareness that sometimes things don't go as planned—a handy thing for all pilots to know. So at least half of the landings in Pterodactyl are made with the engine shut off to simulate a forced landing.

McCornack seems to take the most daredevilish type of risks now only when he's playing *Crossfire*. He may be the only player in the world who attempts the game with the monitor turned off.

His high score, sans monitor, is 480. He plays by the sounds, listening to the delays and counting the shots, and when the game is over he turns on the screen to see how he did.

McCornack says *Crossfire* is the only thing that enables him to relax after his "brain drain" at work. "*VisiCalc* is useful to me as a designer, but I would have to say *Crossfire* has been a more effective business program."

To some people, McCornack probably seems to do things in extremes. Maybe he shouldn't go around calling other people daredevils when he himself shuts off his monitor to play computer games and then later goes riding with the birds.

But take a minute to look beyond the man in the Pterodactyl, the big spender in his yellow Lotus, the coffee shop psychologist. Balance works in McCornack's life. He knows how to construct a safe aircraft, how often to practice forced landings in his machine, when to quit work, and how to relax with a computer game.

Then take his and James's marriage. Toni James doesn't fly—you don't have to fly to have a stimulating amount of risk in your life. She takes plenty of chances everyday with the business, and she was first to explore the possibility of computers at Pterodactyl. Theirs is not so much a well-balanced marriage as it is two well-balanced individuals who happen to be married. In between, they manage to run a growing company successfully.

McCornack and James also live in a town some people might turn up their noses at, but Watsonville swings like a pendulum, maintaining a balance that tends to be jolted in larger metropolises.

It's this balance that's important, and calling someone else "crazy" because they dare to do something different is the most dangerous game to play.

Adventure International. By now, Campbell and Trusty are nearly ready to leave on their round-the-world adventure. Again, they don't think they're any crazier than the next person. In fact, they've painted "Kindred Spirit" in big block letters on the lower wing of the aircraft—because they believe we're all alike.

We all have our extreme sides. So we're either all crazy or none of us is, and, as the old adage has it, anything in moderation doesn't seem to hurt. The balance, Pterodactyl shows, is the important thing.

Pterodactyl Limited, Box 191, Watsonville, CA 95076; (408) 724-2233. Not all Pterodactyls remain in Watsonville, however; one Pterodactyl Fledgling resides in the permanent collection of the Smithsonian Air and Space Museum in Washington, D.C.

Left: Amos Buettell's S1-S Pitts Special. A clear panel in the undercarriage is used for taking quick glimpses of the box markings during competition flight. Above: On the judging line. The box marker is flipped from red to white, signaling that the box is clear for the next plane to enter. Below: The cockpit of a Christen Eagle II, the workplace of the aerobatic pilot.

Gymnastics Take Wing And Apples Keep Score

BY ANDREW CHRISTIE

Borrego Springs is a little town in the high desert country of southern California. A long, flat stretch of two-lane blacktop connects the town to an airfield, with a lot of desert and blue yucca in between. When night falls, the dim outline of the Chocolate Mountains glows purple. You can see every star in the sky, you can hear for a thousand miles, and the only thing making a sound is a single cricket.

It's the kind of country where private pilots come to fly airplanes. On this night, dozens of gleaming sport planes line the length of the airstrip, wingtip to wingtip. The Decathlon, a monoplane, is a popular item. The stubby Pitts Special biplane, smaller and more powerful, is also much in evidence. There are a few one-of-a-kind custom-built planes here, like El Gato and Super Fli—owned by one man, piloted by another, and maintained by someone else, like thoroughbred horses.

They are all here for the Akrofest '82; three days of fancy flying with rules like an Olympiad and—this time—an Apple computer to keep score.

Daring Young Men. Much of the history of aerobatic flying was written in places like this. All of southern California once looked like Borrego Springs, and it was a hotbed of technical and cultural innovation even when the century was young. Air-devils and inventors were drawn to the area, and the combination proved fruitful.

Lincoln Beachey, a man whom a lot of aviators died trying to imitate, made his home base Dominguez Field and went there to work out with his Curtiss pusher, a mass of struts and wheels with no cockpit, controlled mainly by leaning to the left or right while perched between the wings. Orville Wright called him "the most wonderful flier I ever saw." Whenever a stunt pilot was killed, Beachey would attempt to re-create the reported conditions of the fatality and find a solution. After Eugene Ely was killed when his plane went into a flat spin, Beachey took his plane up and kicked it into a flat spin. The controls became unresponsive as the plane screamed earthward, and he finally shifted his weight and slammed the joystick and rudder against the direction of the spin. The plane leveled off, and Beachey had discovered how to beat the flat spin.

Aerobatics thrived with people like Beachey investigating what could be done in an airplane and gradually persuading the government and the infant commercial aviation establishment that there were more benefits inherent in the development of pilot confidence and skill than there were in the cheers of a carnival crowd. World War I produced a lot of new maneuvers, inspired by the desire of one pilot to avoid being shot down by another. The most successful of those pilots made livings in the twenties as barnstormers or members of air circuses.

The air show almost died in 1945, when a spectator flying to a show at Flagler, Colorado, did a slow roll over the field and crashed into the grandstand, killing twenty-two people. The Federal Aviation Administration clamped down hard, and there was little organized activity in American private aviation for the next twenty years, a time when the rest of the world was building on and surpassing American achievement in the air. No American flier could successfully compete against the highly trained, government subsidized Russian and European teams at the international meets until Frank Price, barnstormer extraordinaire, won the international meet in Czechoslovakia in 1960, hiring a local to carry the stars and stripes onto the field.

Price brought back something more than new hope and pride for American aviators. At the Czechoslovakia meet, he had noticed European pilots taping up diagrams of lines, squares, triangles, circles, and arrows inside their cockpits. This was the aerocryptographic system of Count Jose Aresti, providing a symbolic code to tell a pilot at a glance which of the known eighty-seven maneuvers or their three hundred and fifty variations he was to perform next in his precisely choreographed routine. It beat trying to memorize them, or reading a script and flying at the same time. Price brought back to America the Aresti system and began the next chapter in American aerobatics.

A Joystick Is Also a Joystick. The International Aerobatic Club defines four categories of competition. Sportsman is the general category, simply requiring a pilot's license and a will to learn; Intermediate requires an FAA waiver to fly below an altitude of 1,800 feet; Advanced requires an additional waiver to fly under 1,500 feet. Each category has a Known division, in which the pilot flies a set routine that has been rehearsed as much as possible before the meet, and an Unknown, in which Aresti charts for a series of maneuvers are distributed minutes before takeoff.

For those hopeful of breaking into the fourth and final IAC category, Harold Krier, winner of the 1965 U.S. Aerobatics Championship, provides a how-to in his handbook, *Modern Aerobatics*, describing the beginning of his routine:

> As the controls come to life, we hold the airplane on the runway a bit longer than usual . . . we're going to need that extra speed! Now, slight right-stick . . . the left wheel leaves the runway, and then the right one. Back slightly on the stick, with a little rudder—and watch the right lower wingtip as it clears the runway . . . As we roll past vertical, feed in slight forward stick and, at the twenty-foot level, we are in inverted flight. You may be fairly sure everyone is watching now!

Flying upside down twenty feet off the ground on takeoff is a maneuver that requires a special FAA waiver allowing one to fly in aerobatic meets in what is known as the Unlimited category—which is just what it sounds like.

The pilots at Akrofest '82 are a mix. Half of them work in nonaviation professions that allow them the economic wherewithal to purchase and maintain their own aircraft, and half of them are crop dusters and full-time commercial pilots. Jim Neal is a dentist. Bridget St. Thalle is a computer programmer. Chief contest judge Bob Porter flew AV-10A Broncos as a cargo-drop specialist in Vietnam and more recently took a team of four Pitts Specials to South Africa for an air show. (En route, flying cross country, the map flew out of the cockpit and Porter had to buzz down to check the road signs.) All of them are here for one reason. They agree to act as judges for each other's flights; they mark off "the box," a 300-square-yard area within which pilots must perform all maneuvers; they arise at dawn for the briefing on each day's flight, and, in fine, give up days of work for minutes in the air.

And, before the Apple blew in, it took a lot longer.

A Happy Confluence. Bill Cornick's red windbreaker is laterally bisected by a row of patches designating chronologically his category standings, from Sportsman to Unlimited. Cornick is a pilot with United Airlines and is active in most Western division meets. His co-worker, Chuck Rigsby, would like to be but hasn't found the time.

Sitting around with a few beers one night, Cornick mentions that the most fatiguing part of the meets used to come after all the flights had been flown, all the judging had been done, and all the scores were in. Then two or three people got to sit around in a motel room with hand calculators until three o'clock the following morning, figuring out how everyone did. Wouldn't it be neat if they had a calculator that could do this stuff faster than the hand calculators?

Well, yes. Rigsby had just bought an Apple and had several degrees in statistical analysis and related fields. He got into Apple's little books and went as far as they would take him. He bought some more detailed texts and learned about sorting and error routines. He read programs, figured out what they did, and adapted them. After three months of Apple-soft self-taught, he wrote a simple high-low program—throwing out the high and low in each group of scores and giving the average.

The program grew in complexity as Rigsby decided to give it the ability to print out; then Bill Cornick came around again and mentioned the Bauer-Terrasov model, using averages and standard deviation to normalize scores. It preserves the scoring of the judge who may score higher or lower than the average but does so consistently, while detecting judges who pick the wrong order of finish but stay in a midrange. The math required to normalize scores, with the end result of the same score assigned to the same maneuver by each of five judges, would take two days on a hand calculator. When Rigsby had his program completed, it could keep track of all judges' raw scores, reenter all data in the event of a crash (computer variety), and have all results printed out fifteen minutes after the last pilot landed.

Making It Fly. Enter Larry Lowe, the man who provided the Apple/videodisc interface for the America's Electric Energy exhibit at the Knoxville World's Fair. When he's not exploring the Apple's visual potential, Larry Lowe is flying airplanes in the Unlimited category, or thinking about flying airplanes, or hanging out a lot with people who love to fly airplanes, or tinkering with the disassembled carcass of his customized Piper Club so he can spend more time flying airplanes. It's been three years since he flew his Cub from Arizona to Southern California and took it apart. Putting it back together has proven to be another matter entirely . . . "But hey, the longer you wait, the more sexy it's going to be."

As a member of the San Diego chapter of the International Aerobatic Club, he was the obvious choice to customize Rigsby's high-low scoring system to the club's needs. Rigsby mailed in his disk and Lowe tweaked it, renumbered it, and mailed back a copy of his version. (Rigsby donated the Bauer program to the Aerobatic Foundation.) All was in readiness.

Gentlemen, Start Your Engines. An early morning overcast will give good silhouettes to the planes, an aid to judging. The judges are lined up on the yucca flats behind the airfield terminal in a line on the center of the box. They are in five groups of three; caller, judge, and recorder. Jim Neal is calling for number 5 judge Bill McIntyre, with Larry Lowe kibitzing until enough scores can be collected to start entering data. Waiting for the first flight of the second day, McIntyre is in a reminiscing mood. He and Larry Lowe flew in air circuses together in Arizona, once flying to Texas just to hear aviation legend Charlie Hillard talk about flying.

Left to right: Akrofest '82 chief scoreperson Susan Porter, contest chairman Bob Porter, and programmer-at-large Larry Lowe.

McIntyre declaims to the desert air: "The first time I ever flew upside-down was with Larry. I'm in the back seat; the plane has these skinny little stock belts. He says, 'Are you ready?' I say 'Yeah,' and he rolls it to the inverted . . . and there I am, looking out the cockpit bubble at the Beeline Highway running from Mesa, Arizona, up to Payson. I'm hanging there, looking up at the ground and looking back down, and thinking, 'This seat belt is *not* gonna help. There is no way that a thousandth of an inch of plexiglass is going to keep me in here if this little thing doesn't stay latched. . . .' Larry's saying, 'You like it? You see the attitude?' and I'm going, 'Uh. . . .' "

"Yeah," drawls Jim Neal, "Larry scares me, and I'm fearless."

There is a consensus that aerobatic sequences are getting tougher, as in any professional sport. A copy of the original Aresti manual shows the Unlimited Freestyle competitions from the world meets of the sixties; they look like current Advanced Free. "I looked at the 1977 Unlimited Known, and I could fly that in my airplane," says Neal. "It wouldn't be easy but I could do it. I couldn't touch this year's."

The planes start coming into the box. McIntyre holds up a draftsman's triangle and a pencil to check their angle. A pilot "draws a line," establishing a direction straight up and pulling back on the stick until the nose is up, pointing the airplane, and holding the position. It should be a 45 degree line. If it is 40 degrees, the pilot receives one point off his score; two points off if it is 35 degrees, and so on.

Jim Neal calls out the intended maneuvers from an Aresti diagram. "Center-box slow roll to inverted . . . other end, reverse half-Cuban . . . slow roll center-box . . . hammerhead . . . center-box four-point . . . 180 degree turn . . . double slow roll center-box; half roll to upright. . . ."

McIntyre punctuates the calls with his commentary for transcription on each maneuver.

"Roll was late going up, early coming down: eight. Slight barrel in the roll: nine. Tucking. . . ."

Everyone admires the moxie of an Intermediate pilot named Brown who executes an outside loop to a hammerhead roll in a Decathlon—a maneuver usually attempted in the more powerful Pitts Special or a customized plane. It is not done that well, but the judges are impressed that it should be done at all. "We're seeing some wild sequences out here in Intermediate today," Neal comments approvingly.

Meanwhile, the runners are taking the score sheets of the five judges to Larry Lowe in the scoring room. Aided by a Centronics printer, numeric keypad, and a Practical Peripherals Microbuffer II allowing for data entry and printing simultaneously, Chuck Rigsby's high-low program acts as final arbiter. Sue Porter, having just flown a sequence herself, enters the numbers into the Apple as Lowe calls them off.

The score sheets for her own sequence come up, and she sees a lot of disappointing sixes and sevens. Was it her climbing snap? She looks closer. No, it was her positioning.

APPLE FLEDGLINGS

Rubye Bushey became involved in video productions in 1977, assisting a Newhall, California, cable firm in the production of a half-hour weekly show on local events. At the same time, Rubye's husband Bob, a former Air Force pilot and flight ground instructor, currently a commercial jet captain, was sent to school by his airline to learn about inertial navigation, a revolutionary computer-controlled navigation system. Soon thereafter, he became involved with microcomputers, and his contagious interest soon spread through his family. Bob purchased an Apple II Plus, father and son attended computer classes, and all soon decided to pool their respective abilities in a single enterprise.

Avion Video Computers of Newhall was founded in February of 1982 by Bob and Rubye Bushey with the idea of providing instruction in the operation of the Apple II Plus and pilot ground school training. Bob Jr. became primary instructor in the computer classes for ten-year-olds to fourteen-year-olds. It was quickly perceived that adults, though interested in the courses, could not easily schedule the time to attend, and the Busheys put their classroom courses on videotape.

Designed to allow students to work at home or at the local flight training center at their own pace, the AVC series of instructional videotapes on "Preparing for the Private Pilot Written and Oral Examination" uses the *E-Z Draw* program and shape tables for text, pictures, and graphs, combined with live action scenes. The course covers navigation, meteorology, basic federal air regulations, flight theory, and other items in the airman's information manual.

The series will be followed by additional tapes on commercial, instrument, and airline transport pilot written and oral examinations.

Bob Bushey and Apple assist a future aviator.

One judge has given her all nines. "Now there's an honest fellow," she smiles.

The screen prompts with "pilot maneuver?" "# maneuvers?" and "K-factor each?" (the point value of the maneuver, depending on difficulty). The degree of difficulty is multiplied by coefficient X; in the Known sequences, the number of maneuvers and K-factor are entered automatically and the corresponding scores are read off.

"With three people running the system, you've got it made," says Lowe. "One reads off, one types, and one watches the screen and listens to the reader. With two people, you always have to be checking your input."

Hour of Decision. The pilots land, park their planes, and make their way to the scoring room, outside of which the printed evaluations of their efforts are tacked to the wall.

They can now argue about the scores almost a full day sooner than usual.

"I got okay scores; some nines, some nine-fives; maybe one eight-five. . . ."

"They're accepting the entry as a fact of life and just looking for recovery. . . ."

"Not only is that hard to find in the nineties, what I'm doing is breaking that down to thirty-degree intervals. The first was pretty good, the second slowed down a bit. . . ."

"I watched the entry into the box. You were one point off and then you capped it. Good scores on the tail slide; you held it right straight up and then you cheated a little. . . ."

"I just wanted the longest hammer I could get out of the thing. . . ."

"How was the pitching? Was there a lot of pitching? I feel I can't do a lot about it. If all you were looking for was a deviation in roll or turn. . . ."

"You see the new kid coming up?"

"He's shaggy."

"What was the story with McCorkle's spinner coming off?"

Ernest K. Gann, author of *The High and the Mighty* and an old mail pilot, has written a few thoughtful words on the subject.

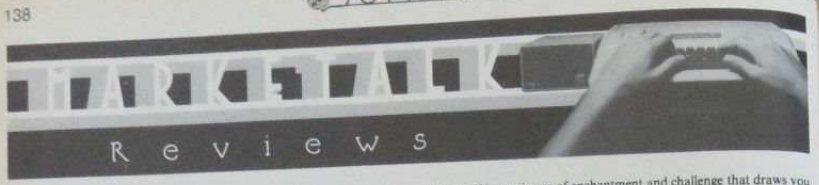
> Once the sensual pleasure of controlling an aircraft is mastered, exhilaration diminishes and incurable habit takes command. Once the secretly nursed conviction that something dangerous and dramatic might happen—with yourself as hero or victim—is forgotten, then the heart joins the mind in submission to the necromancy of flight. Pilots who have flown for fifty years are unable to shake it—whatever it is.

The next morning, the judges are back in a line on the field for the final sequences. Jerry Boughner is the first pilot up, the snarling roar of his red, white, and blue Skybolt rising in pitch and booming across the desert floor as his ship dives down through the overcast.

Jim Neal, whose flight time is logged in thousands of hours, leans back in his folding chair and grins.

"Hey, doesn't that just make the neatest sound?"

A TEACHER FOR THE APPLE



Unless otherwise noted, all products can be assumed to run on either Apple II, with 48K, ROM Applesoft, and one disk drive. The requirement for ROM Applesoft can be met by RAM Applesoft in a language card. Many Apple II programs will run on the Apple III in the emulator mode.

Ultima II. By Lord British. Hooray! Hooray! *Ultima II* is actually, finally out! And the wait was worthwhile. *Ultima II* is light-years ahead of *Ultima* in graphics and plot. It also is a lot more fun to play.

Lord British has mastered machine language at long last, and this program sparkles with speed and superb graphics. The waters of the oceans move in wavelike motion. The towns, villages, and castles have been expanded to many screenfuls of hi-res detail.

The game occupies three disk sides this time instead of two, but very little disk switching is required. One disk is the program disk. Another is the master player disk, from which you generate the disk on which your character is generated and stored. The third is the galactic disk; like the original *Ultima*, part of this game takes place in the far-flung future in space. This time, though, space travel is much more interesting than the quasi-arcade shoot-'em-ups of the first one. Now there is the entire solar system to explore, planet by planet. Each planet is a separate hi-res world just waiting for an intrepid explorer to set foot on it. Most planets contain mysterious places and inhabitants to tell you dark secrets; and you'll find some of the most bizarre dungeons you've ever tried to map.

Not that things aren't wondrous on Earth. Lord British uses the fabulous shimmering time doors a la *Time Bandits*. Five eras are available through shifting travel routes. Among these is the mythical time of legends. Each era has unique locations and inhabitants to aid or hinder you. You must take the time to talk to everyone in the game; you may miss important clues and aids otherwise. Besides, Lord British added a fun feature to this game by scattering among its population well-known people from the microcomputer industry. Each of these characters has something relevant to say about themselves or the game. It's a challenge to see if you can find and identify these people.

The story, subtitled "The Revenge of the Enchantress," revolves around your search for Mondain's evil apprentice Minax. Mondain was the evil wizard you destroyed long ago if you won *Ultima I*. His apprentice has surpassed him, having succeeded in destroying the world of the future. Your task is to gather together all the special weapons and magic items needed, travel to Minax's heavily guarded castle, and there find the way to destroy her.

Once again, the choice of what type of character (race, class, and attributes) you'll be is up to you. It has been whispered that clerics have more fun, but no one seems to be able to figure out just what kind of fun that might be. It's your responsibility to help your character grow and develop and, of course, to survive.

A very nice feature is that the game automatically saves at various points in your wanderings. So, if you suddenly do something that turns out to be fatal, you don't have to begin all over again. Upon rebooting, you start off at the last saved game position. Incidentally, the save takes place on the player disk. Since each person playing must have a copy of the master player disk for storing a character, members of a family can be saved at different points.

Sierra On-Line's publishing of this product has certainly enhanced it. The merchandising alone is worthy of an award. The box is a beautiful rendition of the advertisement they have been using for the game. The manual is fun to read. The most delightful detail, though, is a large, full-color cloth map that depicts the earth and shows the placement and paths of each of the shifting time gates. The map is beautifully done in elfin runes and temporal symbols.

Ultima II is a universe of enchantment and challenge that draws you deeper into its unique spell with each playing. It will provide many hours, possibly months, of delight and enjoyment.

The winds of time whistle softly . . . *Ultima III*? Perhaps. Who knows what the multitalented Lord British will choose to do next. RRA

Ultima II, by Lord British, Sierra On-Line (36575 Mudge Ranch Road, Coarsegold, CA 93614; 800-344-7448). $59.95.

Data*Trans. By Mingche M. Li. With the wealth of modem communications packages available today, it is very difficult for a new package to be innovative. Rarely do any of the programs address the more active uses of telecommunications, such as bulletin board or Source applications. Instead, they appear to be aimed at the passive user who wants to download a fixed file, upload a prepared text, or transfer some *VisiCalc* data.

*Data*Trans*, rising like a phoenix after many years of development by the giant government think tank, ABT Associates, is just such an innovative program. This is certainly one of the most versatile telecommunications packages to date. It allows the Apple to interface with mainframes, minis, and micros.

*Data*Trans* handles all aspects of electronic mail, such as unlimited autodialing and autolog-on files and remote control of your Apple. It lets you run, load, list, write, save, and, of course, transfer any Applesoft programs to another unattended Apple. Here is really intelligent terminal power. You have the ability to print and copy data simultaneously as it is downloaded.

One of *Data*Trans*'s most useful features is the ability to work with and edit text files while on-line. This is very important to those involved with serious frontier modem communications, especially those working in a conference tree. Text files can be created, printed, and transferred effortlessly via this system from either the 20K buffer or disk. Any disk can be catalogued while on-line also, in case you have forgotten what data was on which disk. A very nice touch is that the program automatically varies the speed of transmission to mesh with the fluctuating response time of the receiving database.

*Data*Trans* also handles the transfer of *VisiCalc* DIF files (likewise *DB Master* DIF files). The program goes one more giant step forward with its ability to create the DIF files itself out of raw data. Until now, only one other program on the market could accomplish this, but it has very limited functions and only works at 300 baud.

All in all, this is a very impressive package that's easy to use and thoroughly documented. If you're shopping for software to use with the new modem you got for Christmas, *Data*Trans* looks like a good place to start. RRA

*Data*Trans*, by Mingche M. Li, ABT Microcomputer Software, a division of ABT Associates (55 Wheeler Street, Cambridge, MA 02138; 617-492-7100). $100.

Jawbreaker II. By Chuckles. Who are they trying to kid? This may be the official *Jawbreaker*, but it doesn't look like the *Jawbreaker* that's rattled around the Apple market for the past year. As a matter of fact, it's not. It's an entirely new game under an old name.

Sierra On-Line seems to be having name difficulties these days. They changed *Gobbler* to *Jawbreaker*. They changed the name of their company. They changed the name of their word processor two times. At that point they must have figured that all the good names were taken.

They put out a program named *Frogger* that should have carried the legend "Any resemblance to other programs, living or dead, is purely coincidental."

Now comes a new version of *Jawbreaker* that deserves the same disclaimer. As noted above, *Jawbreaker* stemmed from *Gobbler*. And even the unwashed could see that *Gobbler* had *Pac-Man* as an inspiration.

But *Jawbreaker II* is as new, as original, and as enjoyable as any arcade product to come down the pike in some time. It's an eat-'em-up, but not in a maze. The playing field more resembles *Apple Panic* or *Cannonball Blitz* than it does *Pac-Man*.

Chuckles's first program for Sierra On-Line was *Laf Pak*, which was greeted by massive indifference in the marketplace. But *Jawbreaker II*, with its smooth animation, vibrant colors, and responsive controls, should establish him as a programmer to follow in his future endeavors.

Be not confused by the name. *Jawbreaker II* is new, fun, and different. In the morass of arcade games, it should rise to the top. AH

Jawbreaker II, by Chuckles. Sierra On-Line (36575 Mudge Ranch Road, Coarsegold, CA 93614; 209-683-6858). $34.95.

Chess 7.0. By Larry Atkin. Odesta claims this to be the definitive microcomputer version. Such lack of modesty immediately alerts potential buyers to be especially critical to see what all the bravado is about.

What it's about in *Chess 7.0* is by far the most thorough implementation of the ancient strategy game on the microcomputer. It's not the definitive one—all chess buffs will find something to gripe about—but it's hard to believe that anyone else will care enough to do their very best.

This package exudes love of the game and care in its implementation. From the nifty manual that all too briefly summarizes playing strategies to a collection of saved games and problems that highlight the delights of chess, this program is meant for the chess aficionado.

In fact, if it's still possible for strategy games to sell computers, *Chess 7.0* will sell Apples.

The best thing about *Chess 7.0* is that it plays a good game without a long delay to think things through. That combines the best of its predecessors. *Microchess* played a lousy game, but played quickly. *Sargon* played a good game, but took all night to do it.

Even at its simple levels, *Chess 7.0* is a respectable opponent. It recognizes all relatively standard openings and plays a solid, and sometimes spectacular, middle game.

But what sets this program apart is its combativeness in the end game. Right down to the mainframe chess programs that vie in highly publicized chess tournaments for the title of electronic king of the chess board, computer programs have been notorious for their poor performance in the end game.

Chess 7.0 seems to break new ground in that regard. Even when suffering great material disadvantage, the program will exploit any misstep its human opponent might make. No chance here to take a breather when you're a rook ahead in the end game.

The package includes almost everything any chess player might have dreamed about. Among its features is keyboard cursor control. You don't have to know whether the computer understands European or American chess notation to play; just use the cursor to indicate which piece to move and where to put it.

As the microcomputer infiltrates new areas of society, it's not clear that the newer owners are settling back with it for contemplative games. Mostly, they seem to be buying the games that provide instant gratification from a quick decision.

But, for those who like to confront the silicon-implanted machine in thoughtful battle, *Chess 7.0* will be a godsend. AH

Chess 7.0, by Larry Atkin, Odesta (930 Pitner, Evanston, IL 60202; 312-328-7101). $49.95.

Bumble Plot. By Leslie Grimm. *Bumble Plot* is a series of five number games designed to teach children of ages eight through twelve the fundamentals of number lines, negative numbers, graph plotting, and computer graphics.

This program is a good example of what educational researchers call a *learning hierarchy*. Its five games are presented in order of increasing complexity. *Trap and Guess*, the first game on the menu listing, asks the child to trap a number that's being kept secret by the computer, using first a horizontal and then a vertical number line to limit the range of possible guesses. (The total number of possible choices on both horizontal and vertical number lines is seven, from −3 to 3.)

Bumblebug, the next game in the listing, combines the horizontal and vertical lines from *Trap and Guess* to form a grid and asks the child to surround the Bumblebug with traps located on the grid points.

Hidden Treasure adds the notion of X and Y coordinate axes. The child must describe the hidden treasure's position by specifying its X,Y coordinates. Word and arrow clues are used to promote the understanding of coordinate systems while helping to locate the treasure.

Bumble Art reinforces the concepts learned in the previous three games. The child is asked to invent and draw pictures and to carry out drawing instructions supplied by the program.

Finally, *Roadblock* puts the child in charge of an entire police force in hot pursuit of an escaped bank robber. This game makes full use of the concepts learned in the previous four games; the child must erect roadblocks across each of the robber's potential routes of escape.

In a day and age when many twelve-year-olds are doing complex computer graphics, some children may find *Bumble Plot* too simple. But children who don't have much experience with computers or with plotting graphs may find *Bumble Plot* just the thing to get them started. It can also give younger children a head start in learning number concepts.

Bumble Plot offers a choice of sound or silence and comes with instructions and a backup-disk certificate. K

Bumble Plot, by Leslie Grimm, The Learning Company (4370 Alpine Road, Portola Valley, CA 94025; 415-851-3160). $60.

Miner 2049er. By Mike Livesay and Bill Hogue. If a modern day Nicely Nicely had wandered through the microcomputer community recently wanting to bet on Micro Lab's Micro Fun division coming up with a bestseller-caliber game within the next few months, few in the industry wouldn't have given him high odds.

The laugh would have been on all of us as we coughed up the cash to make Nicely Nicely a rich man. Because he would have won. Micro Fun's new game, *Miner 2049er*, is every bit bound to be a hit.

You'll think it's from Sirius. No, On-Line. Maybe even Broderbund. That's how it looks and that's how it plays—super and well. From Micro Lab? Sure as hell.

Miner 2049er has every element of a Top Ten home arcade, including that of cohesion; all the elements integrate smoothly and right. Everything works.

There are bits of business that might recall other games; but no rip-off here—the pieces come together in a way that's new and fresh, challenging and addictive, and delightful. To say *Miner 2049er* is based on another game would be like saying an apple pie is based on a stick of cinnamon.

To begin with, there are not one or three or even five variations on the first screen, all to be repeated at a faster pace when you've mastered the lot. *Miner 2049er* has ten very different screens, each with its own unique challenges; and, when you've mastered them all (whew!), the challenge is to do it all more efficiently, for higher points. *You* learn to go faster: it doesn't.

Bright, clean, sassy hi-res animated graphics provide the milieu of obstacles a happy little miner must overcome in his resolute determination to solidify the mine floor and, wherever necessary, to rid the mine of lazy, dangerous mutants.

Only when he's in possession of a tiny rainbow-striped apple are the mutants vulnerable; otherwise, running into them is fatal. Mutants mindlessly patrol out-of-the-way, difficult-to-reach places; they don't attack or give chase, but they don't give in either. The effect of the apples lasts a very short time, and the apples often seem to be located in spots that aren't the least helpful. The little miner must grab an apple and do in a mutant or two; how to do it offers several tactical challenges per screen—delightfully fun and very difficult to figure out. On some higher levels, no apples appear; then it's time for leapfrog—uh, leapmutant.

In the beginning there are ramps and ladders; then chutes join in—and elevators and hydraulic lifts and many surprises. Deeper in the mine, you wish there were ladders: jumping among isolated ledges can be very dangerous. There are even moving ledges you must time your jumps to.

The flavor is reminiscent of *Apple Panic*, with much more variety and much more strategy. There is a touch of *Cannonball Blitz*, but, where *Cannonball* became frustrating, *Miner* succumbs to logical solution as well as practice. And, in the triggering of vulnerability through touching a particular item, there's a tip of the hat to the multitude of Pac-Men; but, rather than being an end in itself, here this M.O. serves a larger purpose.

When things go right, they go very right. *Miner 2049er* requires a joystick, and Livesay has found a way to make the joystick palatable and even friendly to those who hate them animals. You get to calibrate the game to your joystick—taming the instrument to work perfectly and easily, even for joystick-klutzes like this reviewer.

If you ever enjoy arcade games, you're bound to be hooked on *Miner 2049er*. Forget who published it until you've played it; then help welcome Micro Fun into Top Ten territory.

Micro Lab prez Stan Goldberg doesn't look much like Nicely Nicely; but he's been willing to bet on Micro Fun's product for a long time. On this outing, he'll collect. He's on a sure winner. MC

Miner 2049er, by Mike Livesay and Bill Hogue, Micro Fun/Micro Lab (2310 Skokie Valley Road, Highland Park, IL 60035; 312-433-7550). Joystick or Atari joystick. $39.95.

Aztec. By Paul Stephenson. After the Aztec empire crumbled, legend had it that a golden idol was somewhere in the lost temple. Only a few people believed the legend and archaeologist Professor Von Foerster was one of them. Foerster located the temple and went in search of the golden idol. He was never seen again.

Through extensive research, you have tracked the route to the temple. To quell your fear on entering the pyramid, you keep in mind the millions of dollars the golden idol will bring you if you find it. Your main concern is staying alive while you do it.

Aztec is a fascinating game by Paul Stephenson. Anyone who has seen Stephenson's previous game, *Swashbuckler*, will note similarities; but *Swashbuckler* is like a rough draft of *Aztec*.

In *Aztec*, you must find a golden idol and escape from the temple. Easier said than done. The temple is swarming with monsters willing to kill you without remorse. Some of the monsters are poisonous scorpions, deadly black widows, man-eating alligators, slithering snakes, and a ravenous tyrannosaur or two. Many other monsters inhabit the temple and they intend to protect their territory.

Scattered through the maze are chests and rubble piles. Both contain surprises, some helpful, others not so helpful. For example, in some chests and piles you'll find weapons such as pistols or machetes that are essential for protection against the evil monsters. In others, you'll find sticks of dynamite. These too are useful—unless they're lit when you find them.

Equally hazardous in *Aztec* are numerous traps, some of which are amusing. But when you stop laughing, you'd better start running; you probably have limited time to find an escape from death. Some of the rooms have water faucets in them; when they start leaking, you're in deep water. Other rooms close in on you, all exits sealed. Before you hurl your disk drive through the monitor, try anything you can think of to escape. For all of these situations, there *is* an escape. Not always a logical one, but a way.

Fighting with a machete is done in the same fashion as in *Swashbuckler*, but your machete is not effective against all monsters. A pistol, however, is—if you have an adequate supply of bullets.

The graphics in *Aztec* are very detailed. The animated monsters move in a startlingly realistic manner. The sound is somewhat less than expected but the general playability renders this flaw irrelevant.

With seven levels of difficulty and eight floors per temple, there's lots of variety for a wide range of player abilities.

In *Aztec*, DataMost has done a good job of combining a fantasy adventure and an arcade-style game into one package. Until now, only *Castle Wolfenstein* had crossed the line between fantasy and arcade games. *Aztec* builds on this concept with more of everything. And if it's a little rough around the edges—well, so was Rembrandt. AH

Aztec, by Paul Stephenson, DataMost (9748 Cozycroft Avenue, Chatsworth, CA 91311; 213-709-1202). $39.95.

Lunar Leeper. By Chuckles. While many arcade games are visually stunning, gripping, or lightning fast and exciting, only a select few are, well, delightful. *Lunar Leeper*, from Charles "Chuckles" (*Laf Pak*) Beuche, fits the latter category perfectly. It's a lot of fun.

Your job in this game is to rescue little folk from certain doom at the hands (or beaks, if you will) of the dreaded Leepers. You do so by swooping down upon them in your trusty spacecraft, picking them up with the bottom-mounted grappler, and rushing them to safety on a nearby cliff. Sounds easy, huh?

Meet the Leeper. He's a cute fellow with very looooonnnggg legs, a huge eye, and a menacing beak. But you don't notice his long legs at first, because the Leeper squats before he leaps. When he leaps, however, his legs show him to be a distant cousin of the ostrich, without the long neck and with a penchant for propelling himself straight up instead of horizontally. Should one of his prodigious leaps bring him within reach of you and your trusty spacecraft, well, adieu! Rather like a terrier, the Leeper shakes you a bit before you vanish down the hatch.

By staying at the very top of the screen, you can avoid the suckers, but woe to the fellow you just rescued. He hangs below your ship and is quite vulnerable.

You could, of course, just shoot the Leepers on sight and avoid the whole mess, but there are some complications. To begin with, each Leeper left alive after all the men are rescued is worth 1,000 points. A dead Leeper is worth a mere 20. Each man you rescue is worth 300 points, while each one consumed by a Leeper costs you 1,000 points. So it pays to let the misbegotten spawn of pogo sticks survive.

Furthermore, your ship is a little tricky in tight spots. You see, the budget was short and they opted not to install brakes on your trusty spacecraft. Instead, there's an inertia meter that lets you know in which direction momentum currently holds sway. The only thing that slows you down is yourself, so to speak. It is simplicity of the log-falling variety to coast into the maws of a Leeper.

Oh, yes, the Leepers are capable of horizontal travel as well, although hardly in such a manner as to excite admiration. They just sort of sidle along, doing a lovely vaudeville soft-shoe, until they find somebody to eat. Therefore, it is incumbent upon you, as rescuer, to rescue the rescuees before they become fodder for the shuffling Leepers.

Then there are the Trabants. Bearing in mind that your beat covers the planet Opthamolia, prepare to meet the underground Trabants and later their surface-dwelling brethren. They are giant eyeballs, larger than your ship. You must fly down a tunnel, avoiding the occasional laser base, until you encounter these rather large eyes, bouncing up and down in the tunnel. You must dispense with these refugees from CBS as rapidly as possible to get to what must be the Queen Trabant, a huge eyeball that fills the entire tunnel. Shoot it and you get back to the serious business of rescuing little folk. Just don't run out of fuel in the tunnel; there are no gas stations nearby.

Fuel?

The budget was kind of short, remember? Your trusty spacecraft does not have a heck of a lot of room for trinkets or frills such as fuel. So you have to refuel quite frequently at the base of the cliffs, while avoiding the Leepers. Fortunately, a refueling station appears at the base of either cliff when your fuel gauge is at the halfway point. All you have to do is hook up to it, inertia notwithstanding.

At each succeeding level, the game gets harder. In the rescue scenario, more Leepers and, after a couple of times, flying Trabants appear. The Trabants must be totally avoided; they're indestructible. The Leepers, as always, are to be shunned. During the cave scenario, the Trabants shoot back, and the laser bases multiply.

Lunar Leeper is eminently playable. The spacecraft control is both frustrating and challenging. The Leepers are swift and unpredictable. The game is both silly and enjoyable. The people at Sierra On-Line have thoughtfully provided the option of choosing your starting level and the ability to use an Atari-style joystick, providing your hardware permits such, making a very good game even better. DA

Lunar Leeper, by Chuckles, Sierra On-Line (36575 Mudge Ranch Road, Coarsegold, CA 93614; 209-683-6858). $29.95.

Merlin. By Glen Bredon. Packed with features, *Merlin* may be the best assembler we will ever see for the Apple II.

Although *Merlin* won't teach you assembly language, it won't stand in your way either. It's set up to make assembly programming as interactive as possible—almost as interactive as Applesoft itself, in fact.

This means that you can type in your source code, assemble it, save the source and object codes to disk, quit *Merlin*, execute the routine (or load and run the Basic program that the routine is designed to be used from), note if there is something wrong, and, with the single command *assem*, reenter the assembler.

Yes, the assembler is still in memory. Furthermore, your source code is still intact when you go back to the editor—unless your program did something to cream it, as machine language routines, especially faulty ones, are sometimes wont to do.

Many of *Merlin*'s features are well conceived and executed. The editor incorporates some of the features of a word processor, including single key commands to delete a character, add to the middle of a line without overwriting the end, skip to the end (or back to the beginning) of a line, and even move the cursor to the next occurrence of a specified character. *PLE* users will find the commands very familiar.

Furthermore, there are a number of global editing commands. You can find or replace all instances of a given string within the listing. Other commands allow a section of code to be copied or moved to another location. The source listing can be printed with or without page headers and breaks. In addition to these options, of course, are the standard commands for adding, deleting, inserting, replacing, and editing lines.

The editor has a classy bonus feature. It can translate between decimal and hexadecimal. This is so convenient, in fact, that you may find yourself jumping back into *Merlin* from Basic just to translate numbers.

The editor has several other features that are more specialized. It even allows for such peripherals as lower-case chips and eighty-column cards, but none of this sophistication should make the beginner apprehensive, as *Merlin*'s basic functions aren't difficult to master and the others can be learned one at a time as they are needed.

The code must be assembled before you can run it, of course, but *Merlin*'s assembler is as painless and quick as the Basic list command. It understands the decimal, hex, and binary number systems as well as a wide variety of pseudo op-codes designed to bend it to your will in numerous ways.

Merlin's exec mode acts as a steppingstone between the editor and Applesoft and handles all disk activity. Saving files is easy; *Merlin* remembers the starting address and length of the most recently assembled file. And, because all commands in exec mode are done from a menu and *Merlin* remembers the last file name used, most disk access can be done

with a mere two keystrokes.

Besides these small but significant departures from the ways of other assemblers, *Merlin* has two aces to play: macros and the *Sourceror*.

Macros eliminate a lot of the repetitive work involved in using assembly language. They allow you to substitute a single command for a routine, with variables taking the place of as many as eight addresses. You can even assign macros to call other macros. Macros can give assembly some of the simplicity of Basic.

Sourceror is a program that disassembles machine language to create a new source file that can be edited with *Merlin*. This can come in handy in situations where the source file is lost or otherwise unavailable. The really impressive feature of *Sourceror* is the way it assigns labels. It can recognize the locations of a large number of routines in the Apple's ROM, so if you used any of Applesoft's or the Monitor's resident routines *Sourceror* will label them correctly. It assigns labels for other addresses by the hex number of the address, so you can use *Merlin*'s global search and replace commands to substitute more meaningful words.

Assembly language is a great way to make your Apple live up to its full potential, but too many programmers, while excited about its speed and efficiency, are turned off by the difficulty of learning or using it. A superb assembler like *Merlin* can turn that situation around. DD

Merlin, by Glen Bredon, Southwestern Data Systems (10761-E Woodside Avenue, Santee, CA 92071; 714-562-3670). 16K RAM card required. $64.95.

Super Taxman 2. By Brian Fitzgerald. What's small and circular, runs around a maze eating little white dots, and boots up on an Apple? That's right, it's *Taxman*. Now Brian Fitzgerald, mad scientist of H.A.L. Labs, brings us *Super Taxman 2*.

That little round guy with the uncanny resemblance to that other world-renowned dot eater is now a civil servant, a tax collector, and his mission is: "Get all the money dots and collect government bonuses, avoiding violent citizens."

Super Taxman 2 has the same multiplayer selection, smooth, flicker-free graphic animation, and arcadelike sound effects found in Fitzgerald's original game, along with the color/black-and-white select mode that makes it easier to tell if the taxman is chasing the violent citizens or being chased no matter what type monitor the game is played on.

There's no Apple-type joystick mode in *Super Taxman 2* (aww), but take heart, America, for H.A.L. Labs now offers the Gizmo, a device that adapts Atari-style switching joysticks to Apple computers, allowing faster playing on the upper levels. For those who don't care to use Atari-type joysticks, the game has a choose-your-own keyboard control.

Super Taxman 2 earns its name with some rather super improvements over the original *Taxman*. Most notable are larger, more interesting mazes (four of them), each with six power dots instead of the traditional four. Ninety-nine levels of play progress in speed and difficulty, with every third level an ultra-fast "challenging stage" (preceded by a musical interlude—the Beatles' "When I'm Sixty-Four").

Another nice improvement in *Super 2* is a level-select option that provides the player with a choice of any of the ninety-nine levels of play at the start. There are also a pause control and five new cartoon intermissions that may be viewed independently of play. MM

Super Taxman 2, by Brian Fitzgerald, H.A.L. Labs (4074 Midland Road, Suite 23, Riverside, CA 92505; 714-359-8480). $25.

Bellhop. By Garry Kitchen and John Van Ryzin. Many figures have been romanticized in our popular culture, but this may well be the first time—possibly excepting Philip Morris's Johnny—for the bellhop as hero. It's a cute idea, and the game itself isn't bad, but it's likely to sink without a trace in a market that's becoming saturated with top-notch software.

You're a bellhop in a hotel, scavenging for tips by hustling seven pieces of luggage to seven suites on the sixth floor. As you cart the stuff to the elevators and to the various suites on the top floor, a counter that keeps track of your tips is rapidly running down in the lower left corner of the screen. Ergo, the faster you get the luggage to the rooms, the more you collect in tips.

You begin with $60 in tips at the first level, $50 in tips at the second level, and so on. At the third level the hotel ghost appears at the left of the screen and slowly works his way up to the top floor. Should he encounter a piece of luggage, he steals it and returns it to the lobby.

To get the luggage up to the top floor, you must call an elevator and then ride it up to the floor you desire. Getting back to the lobby works the same way, with one notable exception: You can jump down the elevator shaft below an elevator and get there twice as fast. No injuries allowed.

There are some problems with *Bellhop*. To begin with, the game is somewhat slow and unexciting. The only speed increase in the game occurs in the amount of time it takes for your tips to run out. Furthermore, there is no cumulative score, which means that no matter how well you do early in the game you still start out with the same time (amount of tips) for additional levels. There would be a great deal more incentive to play if one could earn extra time in the upper levels by succeeding in the earlier ones.

All in all, it's not a bad game, but it's not a great one either. DA

Bellhop, by Garry Kitchen and John Van Ryzin, Hayden Software (600 Suffolk Street, Lowell, MA 01854; 617-937-0200). $34.95.

Apple Assembly Line. For the hardcore Apple hacker, there may be no better publication than *Apple Assembly Line*. Mostly written by Bob Sander-Cederlof as a monthly newsletter to support owners of his *S-C Assembler*, the digest-size magazine is much more.

In recent months there have been such articles as the one that enumerated the differences and equivalencies between pseudo op-codes in the various popular Apple assemblers. Obviously, the material was intended to aid *S-C* owners in adjusting from prior assemblers or in translating programs written for another assembler into *S-C* code. Just as obviously, the article had import far beyond just those assembly language folks who own the *S-C Assembler*.

Sander-Cederlof is a man of far-ranging interests, and that shows in the kinds of projects he tackles within the framework of the magazine. From speech synthesis to time keeping, *Apple Assembly Line* hunts down the most efficient code to do a given job. Belying the adage that programmers can't write, Sander-Cederlof and his correspondents manage to communicate in a language that looks suspiciously like fluent and understandable English, even if they do throw a JSR or a BNE into the fray occasionally.

All you need to get a lot out of this publication is an abiding interest in assembly language programming. If you have that, the $15 yearly subscription rate is a real bargain. ART

Apple Assembly Line, S-C Software Corporation (Box 280300, Dallas, TX 75228; 214-324-2050). $15.

Repton. By Dan Thompson and Andy Kaluzniacki. How long has it been since you played an arcade game that actually had something to do with its melodramatic documentation/scenario? One in which the feel of the story came across in the play of the game, along with the pleasant alpha wave reflex-testing activity of blasting thine enemies? When was the last time you sat and played for hours just to get the feel of the thing and a bare outline of scoring/survival strategy—never mind higher levels—until your vision blurred, your throat cracked, and your spine atrophied?

That's how good *Repton* is. You are the Reptonian commander of the good ship *Armageddon*, racing to the belated aid of your defeated planet. The Quarriors are consolidating their ill-gotten gains and building their base on your planet, taking chunks out of the Repton business district and making energy siphon withdrawals from the Reptonian power bank. They are also swarming about underground. They are, in a word, parasitic. And they feel like it. And there are a lot of them. Yuck.

Play is as fast as you want it to be; you can have your hands full just standing still. You can play offensive or defensive. In the latter, you ignore the Drayne ships depleting Repton's energy supply—let them do so, in fact—to concentrate on blasting and dodging everything else. This failure to save your planet, believe it or not, gets you to the next section of the game, where you make your last-ditch stand underground and get a shot at the Quarrior nerve center (a bit of a turkey shoot). Losing the battle, in other words, is the way to win the war.

This is not an infinite loop shoot-'em-up where waves of invaders keep coming until you are overwhelmed or you get tired of shooting back. The enemy is finite, though they will send in reinforcements just to make it tough to get to the next level. The first time you get there, you're not likely to stay long, nor will you wish to return until someday when you're older.

You cannot, however, remain indefinitely on the first level. You can try, of course. Flying back and forth over a Quarrior SAM missile base (dense pack, no less!) will trigger waves of warheads launched in your wake. You can wheel around and blast away at them for some fast point totals, but, if you try this stunt for long, the Quarriors' deadly single-ships will come warping in planetside like a cloud of mosquitos, making your radar screen look like the inside of a snowflake paperweight and dictating a rapid strategic retreat.

A word about the radar: nice. It's the best use of this device in an arcade game since *Wayout*. In most of these games, a little 'scope full of blips is either decorative or distracting. In *Repton*, it pays to keep one eye on the tactical while dealing with the main action. It's the only way to keep up with what the little buggers are up to, or to brace yourself to meet a bunch of them when they start heading your way.

A word also about the documentation: cool. It's all on the screen and constitutes the demo. Brief, direct, and to the point, it's user-selectable (no *Castle Wolfenstein* three-minute bootup here).

A word about the programming: magic. You can't begin to count the number of objects all doing their individual thing on screen at the same time, but it never slows down.

Use of a joystick here is a matter of taste: it depends on the joystick. The keyboard mode allows the precision and control you will need, and you will need a lot—shields and nuclear devices are available for the faint of heart.

One of the most tired and certainly most treacherous terms in this field is *state of the art*. So we won't use it here. Let's just say that *Repton* makes everything else currently available in this particular genre look very thin and small. JC

Repton, by Dan Thompson and Andy Kaluzniacki, Sirius (10364 Rockingham Drive, Sacramento, CA 95827; 916-366-1195). $39.95.

Battle for Normandy. By David Landry and Charles Kroegel, Jr. SSI's latest release, *Battle for Normandy*, simulates that longest day, June 6, 1944, and the aftermath of the Allied invasion of Hitler's Fortress Europe.

Years after that day, Dwight Eisenhower, the supreme allied commander for Operation Overlord, the invasion of Europe, revealed that he had written two drafts for the invasion's press release. One acknowledged the success of the landings; the other accepted full responsibility for the failure of Overlord and the withdrawal of the Allied troops. After a few hours with *Battle for Normandy* you'll understand Ike's feelings on that June morning.

The game set comes traditionally boxed. Included are one double-sided disk (one side for Apple II or III, the other for Atari), a seven-page game manual, and two game crib cards. Absent is the extended map board and cardboard counters used in SSI's "advanced simulations" such as *Napoleon's Campaigns* and the *Gettysburg Campaign*. In their place SSI has used some interesting graphic techniques to assist game play.

You begin *Battle for Normandy* in your game manual, where you'll find an Overlord briefing to you, the commander of the operation, written in the style of a secret military directive and delivered to your headquarters in London. The briefing effectively presents the basic outline and strategic overview of the game; it also helps re-create the mood of that historic time.

Battle for Normandy strikes a good balance between complexity and playability. On-screen menus and user-selectable time delays enhance game flow.

Play may be either one or two player. In single-player mode the computer always takes the role of the German opponent; in two-player mode the computer serves as referee and game administrator. The player may choose a historically accurate scenario or alter the game parameters to reflect what-ifs. Ever wonder what would have happened if Rommel hadn't been absent from the front to attend his wife's birthday? Or if General Patton rather than Montgomery had been in charge of the landings?

Level of player experience is selectable to accommodate everyone from novice to veteran.

The simulation is divisional/brigade level in scope. Terrain and the unpredictable channel weather, as well as the historical combat strengths and leadership of the participating units, are modeled. An important addition to the game is the necessity to allocate shipping and air and naval support. Resources are finite and their allocation is a major factor in the game.

The game's graphics use two technologies to provide a strategic overview of the rapidly developing tactical situation. An aggregated hi-res map of the entire battlefield shows gross terrain and troop locations. The strategic map may be toggled on and off during game play, and the graphic hex screen may be scrolled under the direction of a movable cursor. This same cursor is used to designate units, move troops, and select combat. The standard hex grid is used to indicate direction and movement. The quality of the graphics is excellent, especially the terrain and unit counters. SSI has gone a long way in improving information flow and using the graphic interface in a logical design format.

The five principle invasion beaches, as well as the strategic areas of St. Lo and Cherbourg, are reproduced on the graphic map. From assigned disembarkation areas your troops hit the beaches, racing against time, weather, and the Whermacht. In all, eighteen offensive and defensive strategies are available to assign to particular units during combat. The computer opponent is an aggressive and capable player who will capitalize on any obvious blunders or hesitation on your part. Whatever you do, get those troops off the beaches and moving inland. General, or else. Remember what happened at Anzio!

SSI lists *Battle for Normandy* as an intermediate level simulation, but all war gamers are likely to enjoy its challenge and playability. It's the best intermediate level game entry yet from SSI. Wilk

Battle for Normandy, by David Landry and Charles Kroegel, Jr., Strategic Simulations (465 Fairchild Drive, Suite 108, Mountain View, CA 94043; 415-964-1353). $39.95.

Zargs. By Max McKee. It has been learned that there are armadas of alien fighters on their way to earth to take over the planet. (Where have we heard that one before?) The only way that the earth can be saved from this hideous fate is to complete and arm the Zarg superstructures. Zarg is a multipart military weapon designed to ward off extraterrestrial attacks. Two goals must be accomplished: create the Zarg and then defend the earth.

The first goal is accomplished in three phases. Unfortunately, the first two phases entail an activity that is much like hopping spaceships through bidirectional traffic—first vertically, then horizontally. The third phase is an interesting docking maneuver around the main Zarg structure. Once four ships have been docked, the superstructure is complete and undergoes a very nice graphic transformation into the Zarg fighting machine.

The battle is joined with the alien vessels attacking from all four sides. Four keys control which direction a shot will be fired. Every alien destroyed adds to the power level, while every strike on the Zarg drains power. The battle continues until the aliens surrender (not likely for a while) or until the Zarg is destroyed.

Zargs is written in GraForth. The graphics are clean and flicker-free and the sounds are superb. The game is a delight to play, and the final battle will be tough to win. It is a good intermediate level arcade game for all ages. Ω

Zargs, by Max McKee, Insoft (10175 S.W. Barbur Boulevard, Suite 202B, Portland, OR 97219; 503-244-4181). $34.95.

THE Spreadsheet. By Randy Wigginton, Guil Banks, and Steve Wozniak. Just when you thought it wasn't safe to enter your local computer store lest you be struck by one of a proliferating breed of electronic spreadsheet packages, the Apple PugetSound Program Library Exchange (A.P.P.L.E.) comes to the rescue.

Back in the days when the world was young and Mother Nature ruled the earth, there was but one—*VisiCalc*. Now there's *VisiCalc* and *SuperCalc* and *MagiCalc* and *CalcStar* and *Multiplan*, and heaven only knows what's lurking just beyond the horizon. It's enough to give your RAM chips the electronic shivers trying to figure out which package is the one for you.

Oddly enough, none of the Calc-alikes have managed to add one whit to the accumulated body of technology as it pertains to spreadsheets. Even more oddly, the result of all this competition has had a decidedly uncompetitive result—prices keep rising.

Remember when *VisiCalc* was $99 and everyone was agog at Per-

sonal Software's affrontery in assuming that any software could be worth that much? Now it costs you two or three times that much to get into a spreadsheet package that's only bells and whistles better than the original. So much for Adam Smith and the virtues of competition.

But where Adam Smith's theory may have failed, A.P.P.L.E. found a gaping hole in the market and moved to fill it for its club members. *THE Spreadsheet* is that program, and it does more than the original *VisiCalc* at *File Cabinet* prices—$22.50. At that price nobody, repeat nobody, should be without the power of a spreadsheet in their home or office. And, while it may not be as much fun as *Choplifter*, anyone new to spreadsheets will be suitably awed by the power and performance of the package.

It serves little purpose to enumerate here the virtues of spreadsheets, nor is it useful to itemize those slight differences that distinguish this version from others.

The salient point is the price. You no longer have to be Daddy Warbucks to play "what if" on your Apple. If you are not an A.P.P.L.E. member, the cost rises to $67.50. That includes a $25 initiation fee into the club and $20 for annual dues. But for that you get other club benefits, including the outstanding *Call −A.P.P.L.E.* magazine.

The competitive market in spreadsheets is changing rapidly and breakthroughs may soon occur. But at the moment Apple owners generally need consider only three: *THE Spreadsheet* for economy, *Multiplan* for certain three-dimensional applications, and *VisiCalc: Advanced Version* for power extraordinaire. Implementing a spreadsheet program into an overall business environment may dictate choosing another; but those who have been laboring over a hot computer can now get 80 percent of the power of a spreadsheet for less than 10 percent of the price. ART

THE Spreadsheet, by Randy Wigginton, Guil Banks, and Steve Wozniak, A.P.P.L.E. (available to members only) (304 Main Avenue South, Suite 300, Renton, WA 98055; 206-271-4514). $22.50.

Beagle Bag. By Bert Kersey and the Beagle Bros Staff. When asked about *Beagle Bag*, the great guru gamemeister replied, "Games in Applesoft Basic? Bah! You can't write games in Applesoft; they're no good, and that's the truth."

Well, the truth is false.

Beagle Bag is the latest in the Beagle Bros indoor sports line—a collection of twelve games and several miscellaneous programs, all in that inimitable Beagle Bros style—fun for fun's sake. Everything is written in sluggish Basic; speed, however, is not the crucial element that makes this disk work.

The games require quick thinking, but they're not mere tests of quick reflexes. They're more the brain-twister variety, forcing you to think before making your move, and forcing you to do it quickly. *Texttrain*, for example, puts you in the engineer's seat of a choo-choo. The object is to manipulate railroad switches, car coupling, and forward/reverse controls, to hook up cars in a sequence so they spell out a preselected word, all in the smallest amount of time. *Triple Digits* lets you play against human competition as you both try to outscore one another by placing numbers strategically in sequences that give you the most possible points.

All the games on the disk are educational, but not in the scholastic sense. Youngsters will enjoy playing *Hang Person* (the guessing game on the gallows), *Sub Search* (a real-time version of *Battleship*), *Pick-A-Pair* (just like *Concentration*), and the rest of these mini-games because they are challenging and fun ways to develop mental agility. For mature audiences, *Beagle Bag* offers a little more.

In keeping with the Beagle tradition, Kersey and company have left the disk unprotected for the user's benefit. You can catalog it, load, modify, delete, rename, and save each program. And, yes, that means you can look at each listing and see why the program does what it does and how you can do the same thing in your own programs. Be careful, though. Gazing through the listing of some games may ruin the challenge of future play.

Of particular interest is a group of four "tricks" called *Magic Pack*. With this program in memory, you are the magician, and those around you are your victims—er, audience. To the audience, the Apple will appear to be performing impossible tricks. Actually, it's just Kersey up to his old shenanigans. A few pokes here, a few string variables there, and a whole bunch of gosubs make this an entertaining way to answer those smarty-pants friends who ask, "So, what can you *do* with this thing, anyway?"

Then there's the manual.

The disk is not expensive. But, at any price, the twenty-page manual that comes with it is worth a few bucks by itself. Even if you don't have the *Beagle Bag* disk, find someone who has the manual and read it. Why? Because it's funnier than *Mad Magazine* and shorter than *National Lampoon*. The prankster prose ("Secretly load *Naked City* into a friend's Apple and stand back and watch . . .") is good for a few guffaws, the Monty Python-style art will keep you giggling, and the section on $200 free cash will have you thinking that maybe Reaganomics isn't so bad.

If someone asks you if you've seen *Beagle Bag*, you'll have to answer, "Which part?" For the price, *Beagle Bag* gives you twenty programs at $1.50 each. Try finding a decent hamburger and a soda for that price.

The folks at Beagle Bros took a simple, unsophisticated programming language and turned out an innovative package—one that's educational, interesting, and entertaining. By technical standards, *Beagle Bag* is where the film industry was in the thirties.

Seen the line for Marx Brothers movies at revival theaters lately? HTY

Beagle Bag, by Bert Kersey and the Beagle Bros Staff, Beagle Bros (4315 Sierra Vista, San Diego, CA 92103; 619-269-6400). $29.50.

Sea Dragon. By John Anderson. And suddenly there were submarine games . . . *Neptune*, *Jellyfish*, *Seafox*, and now *Sea Dragon*. *Sea Dragon* is definitely another thread-your-way-through-the-tunnels game, complete with mines, electric eels, lasers, sea fleas (!), and gun turrets—and, of course, a dragon.

When you boot the disk it talks to you, or at least the computer does. It says "*Sea Dragon*" in what seems like a Japanese accent heard over a terrible intercom. Then it asks you to wait while it "initializes systems." Cute.

Once you get underway, you are the pilot of a submarine that looks

When these great companies
bring you sound and speech programs,
they do it with

mockingboard™

Budgeco
Earthware
Gebelli Software
G.Y.S.T.
Hayden Software
Legend Industries
Lightning Software
M & R Enterprises
Penguin Software
Sierra On-Line
Sir-Tech Software
Sunnyside Soft
Synergistic Software

Sweet Micro Systems
150 Chestnut Street, Providence, Rhode Island 02903 (401) 273-5333

mockingboard
mockingboard
mockingboard™

Compatible with Apple II and Apple II Plus microcomputers, these single board advances bring excitement and realism to any type of programming. From now on, every Apple II user can add sound effects and speech to programming and gaming by simply plugging in Mockingboard. With the low cost addition of the Text to Speech Algorithm, this new capability is expanded further and any typed word(s) or phrase(s) are clearly spoken. These added dimensions bring vitality to every type of computer use. Your menus speak, your error messages supply advice or new directions — user friendliness takes on new meaning. Games explode to life and game characters develop personalities. Designed to answer specific requirements and priced to fit any budget, Mockingboard products allow you to select just the capability you need.

SOUND/SPEECH I — speech synthesizer which offers unlimited, programmable vocabulary (speech) plus simultaneous capability of hundreds of sound effects. $299

SOUND II — sound effects generator produces two distinct, simultaneous sounds via separate audio channels. $199

SPEECH I — speech synthesizer producing unlimited, programmable vocabulary (speech). $149

SOUND I — sound effects generator capable of producing a wide ranging repertoire of sounds. $99

Sweet Micro Systems
150 Chestnut Street, Providence, Rhode Island 02903 (401) 273-5333

not unlike a shark. You must guide your craft, via keyboard, joystick, or Atari-style joystick, through a treacherous tunnel some forty-four screens wide, avoiding all sorts of perils along the way. You are armed with limitless torpedoes and a sonic deflector. The aforementioned deflector would more aptly be called an annihilator, for that's what it does: obliterates anything on the screen. In the interest of fairness, the folks at Adventure International don't award any points for hazards destroyed via sonic deflector; it's too easy. Instead, using the sonic deflector consumes five hundred units of air, of which you only have six thousand units to begin with. It does not pay to run out of air, and thus it does not pay to use the aforementioned deflector with any frequency.

Should you survive the mines, lasers, turrets, eels, seaweed, sea fleas, and sundry others, you reach the dragon. Depending on how much air you have left, there will be a certain number of scrolling bricks imprisoning the dragon. Your task is to shoot away all the bricks and free the serpent, but don't hit him, for god's sake! There's a thousand point penalty for each hit on the dragon.

Freeing the dragon replenishes your air supply and allows you to continue your suicide trek through the underwater passageways.

Sea Dragon seems slow at first, but the hazards are so thick as to make any gamer forget about the speed of the graphics. The graphics are good, while not finely done. The tunnel itself runs horizontally for the most part, with occasional vertical dips and rises of a somewhat precipitous nature. Instead of having several ships as second chances, you are permitted to take 100 percent damage before starting anew. Running into a wall causes 10 percent damage and running into a hazard causes 15 percent. Thus, you get between seven and ten tries per game.

The game itself is quite playable. Although the speed makes the sub's responsiveness seem sluggish, perseverance proves the ship to be up to the task. With practice, you can learn how to clear the mines and avoid the lasers. Fortunately, the folks in Florida put a practice mode into the game. This mode plays at the same speed as the normal game but starts you off with a good deal more air, permitting you to use the sonic deflector more frequently and thus see more of the tunnel and game.

If running a gauntlet of jagged scrolling tunnels filled with innumerable bad guys is your cup of tea, you should find *Sea Dragon* most enjoyable. It plays well and is by no means an eyesore. It should provide a good deal of entertainment. DA

Sea Dragon, by John Anderson, Adventure International (Box 3435, Longwood, FL 32750; 800-327-7172). $34.95.

Computer Fundamentals and **Programming in Basic.** By Bob Bushey. The ultimate self-teaching aid for learning microcomputer principles and operation is one that tells you and shows you what you are supposed to do and then lets you take your time to do it. This, of course, is the definition of videotape instruction, increasingly popular in schools and now catching on in homes that have television, computer, and videotape recorder. If you have the technology, you could do worse than Avion Video Computers's *Computer Fundamentals* and *Programming in Basic* (Beta or VHS).

The fundamentals tape gets down to brass chips, going into detail on the structure and function of the micro and how it's used. You may well find that there's more of this information than you want to know. Fortunately, each section of instruction is keyed with an index number that can be used for fast-forwarding past or rewinding back to a given section.

In the how-to section, each lesson signals when it's completed and gives you the opportunity to try out what you've learned—hit the pause button, switch from VCR to computer, exercise your new abilities on your keyboard, then on to the next lesson. "Live" demonstrations are interspersed with diagrams and illustrations. The graphics are bright and colorful.

For the climax of the fundamentals tape—writing a Basic program—watching a program take shape and seeing it run on the tape as it is supposed to run on your computer is helpful indeed and a good early lesson in neat programming habits.

Both tapes are based on a course of classroom instruction offered by Avion Video Computers. Often as not, it really does feel like classroom instruction—perhaps with one of your high school civics teachers on a June afternoon. The narration is informal and digresses freely to related topics. For those recently graduated from the American educational system, the temptation to turn off the teacher and run outside to play may prove too strong too frequently—the obvious drawback to all learn-at-home courses. This course was put together by teachers rather than marketing specialists trained in the art of getting your attention. As such, it is strictly an adult education course for those motivated to learn.

But, if motivated you be, rewards you shall have. Expect to take from five to ten hours to complete a course of instruction, including the performing of all drill, practice, and review. AC

Computer Fundamentals and *Programming in Basic*, by Bob Bushey, Avion Video Computers (22916 Lyons Avenue, Suite 2A, Newhall, CA 91321; 805-259-2910) $69.95 each.

Tubeway. By David Arthur Van Brink. Deep in space, you're patrolling the galaxy on your usual route when suddenly something strange occurs. Your ship begins to shake uncontrollably and an unearthly force pulls you to a galactic battlefield, the Tubeway. A force field traps you at the rim as a storm of aliens begin to travel up the Tubeway in pursuit.

That may sound like a bad dream or a clip from *Star Wars*, but it's neither. It is the setting for *Tubeway*, a fast-action home-arcade game from DataMost.

In *Tubeway*, your ship has two defense mechanisms—your laser-firing ability and your super zapper. The zapper, which is time-released, appears as a white line rotating around the outer rim. When detonated, the super zapper will destroy all aliens that have reached the outer rim.

On the first four levels, the aliens you battle are called Homers and Seekers. These levels are introductory in comparison to the challenges that await you on the upper levels. On levels five and six, the Homers and Seekers call upon the S.U.s, another breed of alien, to make your mission even more difficult. When destroyed, S.U.s transform into Homers and Seekers.

Tubeway has thirty-two levels of difficulty; not many people will be able to say they've seen the final level. From level seven up, the aliens waste no time attempting to blow you out of the sky. They come racing up the Tubeway after you, some leaving deadly spikes in their trail. When you destroy all the aliens on your level, these spikes are released and shoot up the tube. They must be avoided or blasted away.

Of all the aliens in *Tubeway*, the deadliest is the Destroyer. The Destroyer lives in a cage; when it's released it will track you relentlessly. The Destroyer cannot be destroyed, only returned to its cage when hit by your super zapper. On these levels of extreme difficulty, your ship's versatility is most important. Your ship can travel all the way around the Tubeway in a split second—and may have to.

Tubeway is highly addictive and very challenging. As a warning, you may want to remove any breakable objects within arm's reach. AH

Tubeway, by David Arthur Van Brink, DataMost (9748 Cozycroft Avenue, Chatsworth, CA 91311; 213-709-1202). $34.95.

Executive Briefing System. By Mitch Kapor. Graphing and plotting programs can be frustrating to use. The visual output they provide is often limited, and a slide show function isn't effective if your presentations aren't self-explanatory.

Executive Briefing System solves these problems. Its range of graphics is large and it permits explanatory—or any other—text on the slides along with graphics.

With *EBS*, you can use your Apple II, or III, to generate a slide show eminently suitable for use in decision support presentations. The formatting flexibility of *EBS* is wonderful; its use of color graphics makes it one of the few decision-support products that successfully take advantage of the Apple's graphics capabilities.

EBS can produce multiple text formats, screen graphics, normal and inverse video, multiple colors, and numerous font types. Tutorials in the manual carefully walk you through the different data and text entry formats to produce these wonders logically following normal format.

EBS may be used alone or in conjunction with several other business programs, including *VisiTrend/VisiPlot*, *ApplePlot*, the Apple Graphics Tablet, and any program that stores hi-res pictures normally.

EBS is extremely easy to use, and it's fun to play with—a plus for a program aimed toward business applications. Novice computer users should have no real problems setting up and getting started with this program. *EBS* is recommended for two disk drive systems; but it will work with one if you're willing to spend plenty of time doing a push-pull routine with your disk. *EBS* copies or transfers one file at a time.

Designing your slides is simple. Text entry is done by typing in the text from the keyboard. The main constraint here is the lack of word wrap; this forces you to be more aware of spacing and the number of characters you want to produce on one line. Numerous text sizes and fonts are available. Letters may be entered in normal, boldface, inverse, and upper and lower case, and it is possible to use mixed formats. Several letter fonts come standard, and Lotus has a full line of other font types available. Accessory disks may be purchased for use on standard or color monitors, and some contain both color and black-and-white fonts. In addition to alphabets, some of the disks contain special characters and shapes to enhance your slides.

EBS produces black-and-white printouts of slides through compatible graphics printers.

You can select among several formats for your slide shows. Choose the amount of time you want each slide to appear on the screen; then select the kind of screen transition you'd like between slides. Transitions can be spiral, curtain up, curtain down, or all three interspersed randomly.

Transfer of graphs and charts from other programs is simple. For example, a line chart generated on *VisiTrend/VisiPlot* was saved to a data disk using *VisiPlot*'s pixsave format. From *EBS*, the chart was called onto the screen via the get option and saved to an *EBS* slide disk. The whole procedure took less than ten minutes.

EBS would enhance any business software library. There are a few problems, like the time required for certain functions to be performed and the need to switch back and forth between command lines when designing complex slides, but these are not significant.

But you don't have to be in business to enjoy *EBS*. It can also be used to enhance school reports or in designing simple flyers for meetings. You probably have a project going right now for which you could use *Executive Briefing System*. AH

Executive Briefing System, by Mitch Kapor, Lotus Development (55 Wheeler Street, Cambridge, MA 02138; 617-492-7171). Language card or 16K RAM card required. $199.

from page 43

must be moved to the end of a program line before the return key is pressed becomes cumbersome. If programming in Basic is a goal for you, the *Global Program Line Editor* by Neil Konzen from Synergistic is salvation in software. The price is reasonable for the screen-editing capabilities you gain.

Join a user group even if you are not a joiner. A local computer user group never fails to be a powerful resource. The group may appear to baffle you with new information, but most members are friendly and willing to shed some light on topics or programs you want to master. We teach a course on selecting a home computer at the Rochester Museum and Science Center. One of the major points is to know what you want the computer to do before picking a system, then fit the available system to your needs. For Mrs. Heckler that's water under the bridge. We hope the above suggestions will cast a little hope on new user troubles.

There is no such thing as the perfect computer. The limitations of the Apple II Plus have been compensated by a huge range of utility software created by active and talented Apple owners. If a new user can set priorities, request information from experienced help, and be patient, the Apple will become an effective tool to enhance your creativity within the realistic limitations of the system. Good luck!
David and Cynthia Boyer, Rochester, NY

Contest Clarification

I didn't think the rules for the Shapes contest in the September '82 issue were clear enough. Would you have accepted multiple entries for the same shape, and what about specificity? If one entry is "car" and another is "Thunderbird," which would be correct?

I think readers should be limited to one entry per contest. A winner should be judged on the basis of quality, not quantity. If I were to send in 40,000 entries to the Oracle contest I'm sure I'd win. In fact, I would win any contest that uses the random number generator to pick the winner just because of the quantity of entries I'd sent. What do you think?
Paul Zerner, Weston, MA

Because the rules for the Shapes contest were open-ended, we were pretty flexible in the judging (see Contest Winners, December 1982). Whenever we run a contest that is judged on creativity, such as the contest this month, we encourage readers to enter as many times as they please. Shoot, if you have more than one idea for an entry, send them all! We love reading them.

On the other hand, for contests in which ties are possible we allow one entry per person per answer. This means if you send in several entries, each with a different answer, you're in the running since only one will be right. But if you bombard us with 40,000 correct entries only one goes into the dreaded random number generator with your name on it. This keeps things fair for those who can't afford $8,000 in postage stamps.

(Actually, because you're allowed as many as ten entries per envelope, the cost would be only $800—easily within the budget of every contest entrant. . . .)

Great Expectations

I would like to voice my disappointment with the Applefest held in Minneapolis. This was not an opportunity to meet the top ten software publishers; it was not an opportunity to meet representatives from Apple Computer. It was merely a local display of dealers from the Twin Cities with every computer off the showroom floor, except an Apple.
Bruce Steinhagen, Algoma, WI

Up to Something

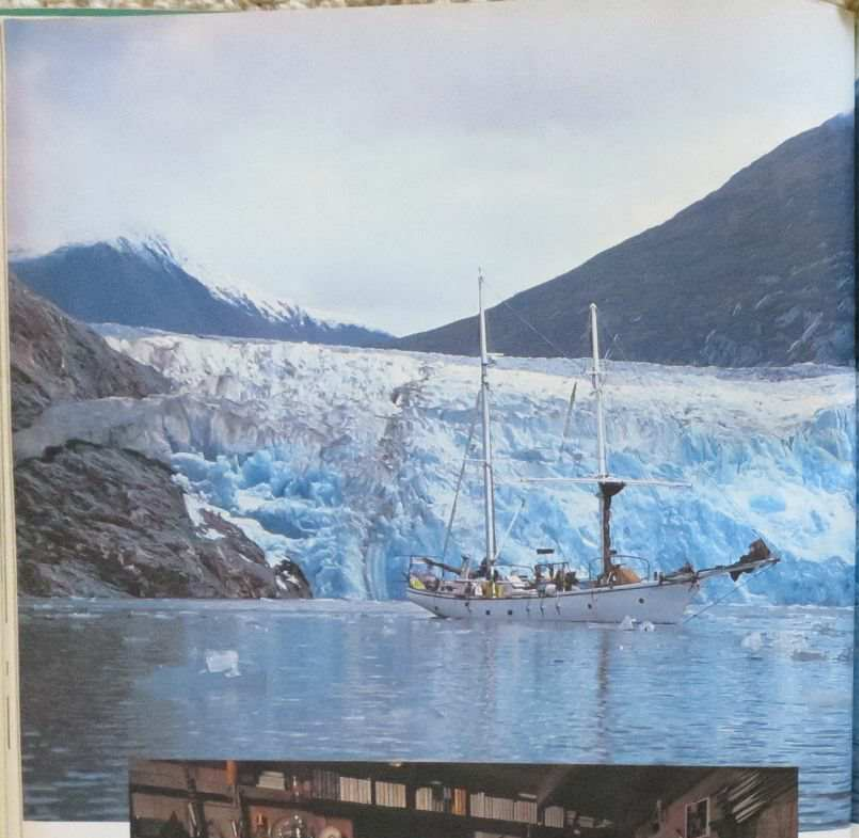
I have had an Apple II Plus for nearly a year and I was wondering if any readers of *Softalk* have a software conversion for TRS-80 or Atari cassettes to the Apple, allowing the programs to be saved to Apple disks and run later.

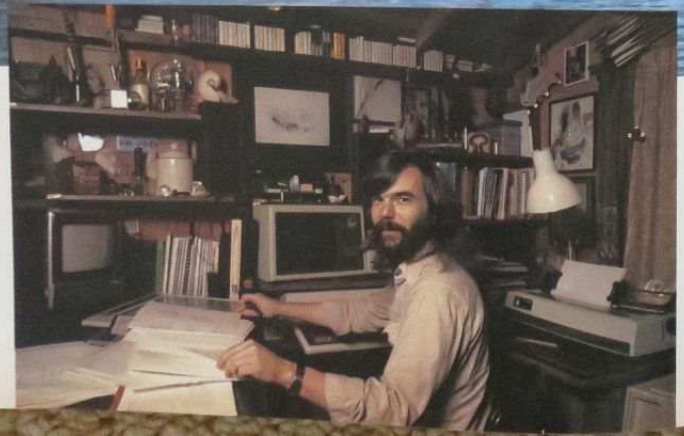
Also, I've seen a hardware modification that converts a raster-scan screen to vector style, but it's not for the Apple. If anyone knows if a vector graphics system could be developed for the Apple II Plus I would appreciate hearing about it through Open Discussion.

Finally, I'm interested in making a direct-memory video digitizer that shows, in the same color and in the same amount of pixels across the monitor, an image that could be stored on a hi-res page, then on a disk. Any ideas?
Gregg Johnson, Westerville, OH

New Year's Resolution

I write in response to Bob Wiseman's metaletter in Open Discussion (November 1982), despite his complaint about printing letters that refer to previous letters. Apparently Mr. Wiseman would like each issue that is discussed to be settled in one pass. If this were done, we would not have the opportunity to question the ideas of other writers or provide solutions to questions posed. Mr. Wiseman predicts (erroneously) that "in six months no new material will appear." This is not a reasonable statement of a truly wise man. The flow of ideas in continued exchange, challenge, and suggestion is what leads thinkers to modification, new conclusions, and creative thought. Examined ideas are the basis of much of our rational knowledge, indeed the core of reasoned thought. Unexamined ideas are only to be expected in totalitarianism. Please keep Open Discussion a place for free, spirited quest—our Apple public forum.
Mike Carlson, Minneapolis, MN





Programmer For Hire:

Have Apple, Will Travel

BY ANDREW CHRISTIE

Outward bound, Seattle. The sailing ship *Varua*, out of Hawaii via Glacier Bay, is headed for Monterey before making the last leg of her long voyage home. Built in 1945 by one of the world's master shipbuilders to be the finest sailing vessel in the world, it's caught in a full Pacific Northwest gale and it's tossing like a cork. The time is four bells, two in the morning. The man at the helm is holding on to it to keep himself from being blown or washed off the deck as much as to keep the *Varua* on course.

The skipper's head appears at the door of his cabin. He looks around the fo'c's'le briefly, blinks, and nods, satisfied.

"Okay," he says to the man at the helm; "yell if anything goes wrong."

And he disappears back into his cabin.

Yell if anything goes wrong?

Tracy Valleau is an Apple programmer/consultant.

He's an independent.

Start Up Time. Five years ago, Tracy Valleau had entered into a partnership in a Berkeley camera store and built it from a marginal concern to a profit-making enterprise. Having more than tripled the store's income, he was now paying out a couple of hundred bucks a month to an accounting firm to put the business' books on a computer. Watching the program being periodically updated, looking over the programmer's shoulder as the code scrolled by, a thought eventually occurred to him: "Hey, I understand some of this."

He sold his share of the store, bought himself half a suit, an Apple, and a couple of disk drives, and told himself, "Well, you just spent your last dime on this machine and you don't know anything about it; better read a few books and see if you can do something with it."

After writing several programs for his own entertainment and that of the small, exotic tribe that owned Apples at the time, he realized it was possible to make a living at it. He decided to specialize in the one-shot deal: something that somebody needs done that there isn't anything commercially available for; something that's a challenge. He called his one-man company DataSmith.

"I got my machine right when Apple was moving to the Bandley Drive address," he recalls. "One day, I got an anonymous phone call:

" 'You . . . deal with Apples?'

" 'Yeah.'

" 'What do you think of . . . the *Controller*?'

" 'I think it's a fine product for 99 percent of the people who use it. I write specialized programs.'

" 'Oh. Okay, thanks.'

"Two months later I found out I was on Apple's recommended programmers list. I got a kick out of that."

Since then, he has spent a lot of time testing and reviewing products for Apple Computer and commenting on documentation.

Tracy Valleau lives and works in the Monterey Peninsula community of Pacific Grove in a bustling one-room abode about fifty yards off the main house belonging to his landlord and attorney, Steve Slatkow. He shares his quarters with a small black cat that answers to "Hey, you," and a library of four hundred Apple programs that includes numerous early lo-res games and Paul Lutus's first editor. Rush-hour traffic becomes a problem in Pacific Grove once a year in autumn when the Monarch butterflies return—"You have to wade through them with a baseball bat." Noise pollution is a problem whenever a high wind blows through the branches of the Monterey pines that densely populate the area.

Hey, it's a living.

You'll Never Get Rich by Digging a Ditch. Valleau did not discover the Monterey Peninsula intentionally, but as a side effect of a personal invitation from the United States Army to try a complimentary two-year stay at their luxurious Fort Ord resort—an offer he couldn't refuse. Valleau whiled away his two years as company clerk, fending off the ennui of the bureaucratic military life by periodically reporting himself AWOL and ordering his records flagged. When he got out of the service, he stayed in Monterey. Today, his one-room abode in its pine grove is the only area in his rather restless thirty-five years that he would characterize as "the kind of place you can come back to and look at it and say 'home.' "

The army experience paid off in other ways as well. Shortly after he had established DataSmith, and right after the Pascal language had come out, the army contacted him again. They needed a program that could keep track of up to a thousand individuals on a base with up to three hundred fifty rooms, configured by any number of beds up to five, each bed identified as single and full or free. They wanted a reservation system set up, with each man identified by rank, why he was there, and how long he was going to stay. They'd called IBM and had been told it couldn't be done on a small computer. "I'll be right over," said Tracy.

"I told them sixty days and wound up taking almost a year. And the only thing that made it possible was bit arrays; great, massive quantities of Boolean arrays. And we did it: eighty-six pages of code with no repetitions. The real challenge was that the system would have to be used by, shall we say, anybody. It had to be bulletproof; real simple.

"Got it done; got it running on an Apple with two disk drives and you never have to take a disk out. The army uses it on five bases."

Sail Away. Another unexpected opportunity came through his connection with Apple. One day he got a call from Apple to say that the University of Hawaii was looking for a programmer for a scientific proj-

ect. How would Mr. Valleau like to sign on as a crew member on a brigantine sailing vessel and spend a few months chasing whales around Alaska with an Apple on board?

"I said, 'I'll have to think about it . . . yes.'"

It seemed that a drastic drop in the humpback whale population had been recorded in 1977, the year the Princess cruise ships started sailing into Glacier Bay, the whales' traditional spawning grounds. The bay was now closed to all major nautical traffic and the *Varua* had to obtain a special entry permit to observe the whales for possible changes in their migratory patterns.

Said observation was accomplished in the main with a theodolite, a surveying device used to take a reading on the point where a whale surfaced and then track it for changes in course and speed. Feeding the information to the Apple would produce two lines on the hi-res screen, dumped to a Silentype, indicating the path of the whale relative to the course of the vessel.

All hands were required to do an equal share of work; cooking, standing watch, deck swabbing, steering the ship through gales, and the like.

"It was not the kind of thing I would have been doing if I had been working eight to five for Apple Computer," muses Valleau. "In fact, after the five months, I had plans just to take my Apple II, stash it on board, and stay with the boat . . . but pragmatism took over at the expense of romanticism."

Success Is What You Make It. Tracy Valleau has made his own lifestyle. He enjoys his work, he gives himself frequent vacations, and he lives in one of the most beautiful corners of the planet. There are a lot of romantics who would approve of Valleau's brand of pragmatism. But he started out as a businessman and he remains one.

What advice does he have for anyone on the trail of the Good Life through programming?

"There are a lot of opportunities open for people who want to get serious with this kind of thing. The standard advice is 'Write a game,' and there's nothing wrong with that. I don't want to develop anything that's really mainstream commercial because of all the headaches it involves—and the packaging, marketing, and distribution. I've always believed it's possible to lead a comfortable, happy life where nobody ever hears of you except six or seven clients."

Watching for whales on the deck of the Varua.

Aside from a little packet of real wood keyboard inserts with which to beautify your Apple, DataSmith's only other commercial product has been an ill-fated capital assets depreciation program.

"Somehow, the ad turned up in a magazine for Commodore Pet owners, which didn't really get the word out in the Apple market; then they changed the depreciation laws. . . .

"I prefer to be in a position where I can help somebody; say, 'Yeah, I can write a program that will do that for you,' and take six months to do it. It's a professional relationship. You have to be there for the client not only during the writing of the program, but afterward. Clients are

generally just like anyone who first goes into computers. The first thing they say is, 'Ask it what the capital of Armenia is.' And you say, 'Well, it doesn't quite work that way. . . .' And, when you're halfway through writing the program, they'll invariably say, 'Oh, you know what would be another nice thing to have it do?' and you smile and say, 'Yes, that *would* be nice.' And you go back and write a bunch of code and start over again.

"If someone asks me, 'Hey, can you make the Apple do X, Y, and Z?' and I know of a piece of canned software that will, I'll tell them; I feel obligated to. Aside from it being unethical not to, it would be bad business. If clients were to discover that I'd hit them up for $6,000 for something they could have gotten at a store for $150, they'd be a little upset. My business depends on word of mouth, and when bad word of mouth starts, there's no way to stop it.

"It's satisfying to take an assignment and make the program do exactly what the client wants it to do. You can't walk into a store and buy a program that will track Hobie Cats for the Monterey Regatta, certainly not with the speed and accuracy of the one David Kempton wrote for it (*DASH*).

"But, when I can see that they can get close to what they want with *VisiCalc* or a commercial database, I'll tell them, 'Sure, I could make it do exactly what you want and it would cost you buckets of money, but if you're willing to cut this little corner right here, you could get it down the street for a couple of hundred bucks.' That is, in my opinion, the best business technique, because those are the people who later say, 'Hey, here's a guy who didn't try to oversell me.'

"The secret of success is to honest-to-God care about the customer."

You Shall Enjoy the Fruits of His Labor. Another secret of success is to be good at what you do. Valleau's latest project, slated for imminent release by a major software house, is a 3-D graphics package for the Apple III done in a combination of assembly language and Pascal routines that interface like blocks, with Dr. John Jeppson's disassembler handling the color fill with dispatch. The program allows the user to work with three hundred planes and three thousand vertices in the construction of objects and display twenty objects per scene with up to thirty-two surfaces each.

The program's potential for computer-aided design is obvious. Its potential as a game programming utility reels the mind. ("You can fly around and into three-dimensional objects, using a joystick. This is gonna make some people's eyeteeth drop out.") In his spare time, Valleau is, in fact, working on a game: an interactive role-playing space adventure with "legitimated visuals" and a moral imperative. On such matters, of course, one must say little.

What It's All About. "Everybody specializes in different things. I chose to specialize in unusual or difficult programs. There are two reasons for that: I like a challenge; and, if the requirements are that difficult, I can charge a lot of money for the programs—which is fortunate, because such projects are few and far between.

"Also, it's nice to be in a business in which you can do something you believe in. There are a lot of good machines on the market, but the best is still the Apple. I can look anybody in the eye and say that because it's true. It was the best in the beginning because of the hardware—there was nothing around that was as dependable or was built as well—and it's still the best today because of the software.

"The small independent consultant is rather rare, I think. You can't just leap into the business. I was able to because I was single and young. I'm sure there are a lot of former Honeywell employees doing consulting work for Hilton Hotels and whatever, but full-time one-man programming . . . that's another thing.

"When I decided to get into this, it seemed to me that what was really important was the quality of life; not just what kind of gee-whiz thing you can do but how you can do it and be happy."

For the future: "Retirement. Computers are great, but I'd rather be sailing." ■

VENTURES WITH VISICALC

BY JOE SHELTON

The Challenge! Last month you were left with a challenge—a problem to solve. That challenge is repeated here so those of you who don't have the last issue handy can review it. The only stipulation for solving this problem is that you have a version of *VisiCalc* that contains logic functions.

The Problem. The problem concerns a sales analysis for the sales office of a company that manufactures photocopiers. The sales office is responsible for three geographic areas and has hired sales representatives to cover two of the three areas. Now the sales manager must decide whether it would be wise for the company to hire more sales reps to cover the third area. All sales in Area 3 are responses to national advertising by the parent company. The sales manager must decide whether sales reps actually contribute enough to warrant adding more of them.

Figure 1 shows the template the sales manager has developed. The values in columns B and C are values he entered. The formula in column D is column B divided by column C (for example, in cell D4: +B4/C4). Since there are no sales reps in Area 3 (cell C6), *VisiCalc* gives a division by zero error and displays ERROR in cell D6. When the sales manager tries to average column D in D7, it displays another ERROR.

The sales manager could ignore the ERROR in D6 and average Area 1 and Area 2 sales reps (+D4+D5) /2) in D7, but he would like to be able to use and expand the template when he becomes national sales manager in a few months. When he adds sales representatives or moves them to different territories, he doesn't want to have to reenter the averaging formula.

Can you fix the template so that any ERRORs that show in column D don't affect the formula in D7?

The Solution? In solving anything, the first thing to ask is "What is the problem?" Functions in *VisiCalc* allow the ERROR display to override whatever the function is trying to accomplish. For example, if you try to sum three cells, one of which displays ERROR, the sum will also be ERROR. *VisiCalc* works this way to ensure that you know about any errors that might corrupt your answer.

So the problem is, how do we insulate our answer from the ERROR condition? A look at the reference card or the user manual reveals a function known as @ISERROR(v). This function is designed to return a TRUE if v is an ERROR. There's a companion function, @ISNA(v), that does the same thing for cells that display NA.

So, it looks as though we have a solution. All we have to do is apply @ISERROR to each cell in column D. But there's one glitch. The @ISERROR function returns TRUE if an ERROR is indicated or FALSE if no ERROR is displayed. Great—we know now that we can isolate the ERROR, but we get TRUE or FALSE instead of values.

Another logic function listed in the user manual is the @AND function. The @AND function displays TRUE if *all* the values in the list are TRUE; otherwise it displays FALSE. That doesn't seem to resolve the need for a value to be displayed either. Likewise, the functions @NOT and @OR return only TRUE or FALSE. That leaves @IF. Conveniently, @IF allows values to be returned. Before we make a move toward solving our challenge, let's look a little closer at @IF.

	A	B	C	D	E	F
1						
2		1982 Sales	Assigned Salesrps	Sales/ Slsrp		
3						
4	Area 1	100	2	50		
5	Area 2	125	5	25		
6	Area 3	55	0	ERROR		
7			Average/sales rep	ERROR		

Figure 1

If, Then, Else. @IF is one of the most useful, and possibly one of the least understood, *VisiCalc* functions. A good way to clarify how it works is to phrase the function in English. You might ask *VisiCalc* to do the following: *If* my expenses this month are less than my pay, *then* add the difference (profit) to my savings account and tell me the total balance; *else* tell me how much I still owe.

The function would be written as follows:

@IF(*I*,*v1*,*v2*)

where *I* is any logical value and *v1* and *v2* are legal *VisiCalc* values.

Using a combination of *VisiCalc* and English, our example would be stated this way:

@IF (EXPENSES < PAY, THEN SAVINGS BALANCE + (PAY – EXPENSE), ELSE PAY – EXPENSE)

In *VisiCalc* alone this might be written:

@IF(C50<B50,G11+(B50–C50),B50–C50)

At the beginning, the *VisiCalc* version of our example reads much like our original English. If expenses (C50) are less than pay (B50), take the savings account balance (G11) and add the remainder of pay minus expenses (B50–C50). If expenses aren't less than pay (that is, they are greater), then tell me how much I still owe (B50–C50).

The initial statement in the @IF formula is called a *logical value*. This statement is the comparison of any values using the logical operators, the @TRUE and @FALSE functions, or one of the logic functions that returns TRUE or FALSE. Effectively, what happens in our example is that *VisiCalc* evaluates the logical value C50<B50 and returns TRUE if C50 is smaller than B50. Otherwise it returns FALSE. If TRUE is returned, then the first value (G11+(B50–C50)) is calculated and displayed. Otherwise the second value (B50–C50) is calculated and displayed. Simple and elegant (you'll hear this phrase often here).

What if pay and expenses are exactly equal? Look at the logical value C50<B50 again. If pay and expenses are equal, then expenses (C50) are not less than pay (B50), and FALSE is returned.

The @IF function gives us a very powerful way to make *VisiCalc* choose or compute a specified value depending on criteria we stipulate. And, remember, the two values don't actually have to be part of the equation. They can be references to other cells or they can even be other functions.

Back to Our Challenge. It would seem that @IF could be used to isolate @ERROR and then return the correct value. How? If an ERROR is generated by the original formula, we should be able to use @ISERROR to change the ERROR to TRUE and then return the desired number from the first value in the @IF statement.

Looks like we are on the road to our answer—but not quite. Look again at the @IF statement in our example:

@IF(C50<B50,G11+(B50–C50),B50–C50)

Assuming that B50 is an ERROR, look now at our restated formula. This time we'll use @ISERROR and replace B50 with ERROR.

@IF(@ISERROR(C50<ERROR),G11+(ERROR–C50),ERROR–C50)

This formula can be resolved to read:

@IF(TRUE,G11+(ERROR–C50),ERROR–C50)

We've learned something important here. Even though we've man-

aged to isolate the initial ERROR, the two value alternatives still return ERROR.

If at this point you feel like throwing up your hands and running off into the distance screaming and yelling, take heart. That's exactly what the author of this column did at first.

Knowing what we know now, let's think about our original problem and see if we can apply some of the things we've just learned.

The Challenge—Solved. We know that we won't be able to isolate the ERROR indications easily from the final @AVERAGE and still be able to display the desired values. Another way to attack the problem is to back up one level and try to isolate the ERROR indication at the place where it is first displayed.

Returning to our original template, we'll have to change the formulas in cells D4, D5, and D6. The formula in cell D4 should be @IF(@ISERROR(B4/C4) ,0,E25). If there's an ERROR calculated by the statement B4/C4, @ISERROR will return TRUE and the first value will be displayed. The first value is a 0. That is why using this combination of functions works in our challenge. The value returned by an ERROR calculation can't be an ERROR indication. If both values in the @IF statement referenced all possible places that an ERROR might be displayed, then an ERROR indication can't be isolated.

Replicate the formula in D4 into D5 and D6, using relative references, and our challenge is solved. See figure 2.

As we said in the beginning, the solution is simple. But we didn't really solve the problem we originally posed. Our task was to fix the template so that any ERRORs that showed in column D didn't affect the formula in D7. We didn't actually do that. We just ensured that there would be no ERRORS displayed in D7.

	A	B	C	D	E	F
1		1982 Sales	Assigned Salesreps	Sales/ Slsrp		
2						
3	1	100	2	50		
4	2	125	5	25		
5	3	55	0	0		
6		Average/sales rep		25		
7						

Figure 2.

Logical Operators	Logic Functions
<>	@IF(t,v1,v2)
<=	@AND(list)
>=	@ISERROR(v)
=	@ISNA(v)
	@NOT(t)
	@OR(list)

Figure 3.

One thing worth learning from the challenge is that you can use combinations of functions to solve many problems. In fact, you could probably solve our challenge a couple of other ways, using multiple columns to help with the isolation. Try it; if you come up with an interesting and compact answer, send it in. Remember, though, the trick with *VisiCalc* is to keep the templates as small as possible, which means using as few cells as necessary.

More Logic? Is your appetite sharpened for logic functions? They are the least understood of the *VisiCalc* functions and, as a result, are rarely used. That is in part because they aren't exactly intuitive and in part because many users haven't learned to think in *VisiCalc*'s logical terms. Logic functions can turn *VisiCalc* into a much more effective tool and can help make reports more useful. When you can automatically choose which of a number of options are displayed or printed, much of the ambiguity of many reports disappears.

Since we've spent most of our time so far discussing logic functions, let's look at some of the other logic functions and how they interrelate.

The logical operators and logic functions shown in your reference card are listed in figure 3.

Logical operators are the foundation of all logic operations. The premise is that whenever you compare two values (they can be numbers, formulas, cell references, or functions) the comparison will be either TRUE or FALSE. The comparisons are *less than*, *equal to*, and *less than or equal to*.

The logic functions use the TRUE or FALSE values that logical operators return. Logic functions are divided into two types. As we found earlier, @IF is the only logic function that returns a value. The remaining logic functions return either TRUE or FALSE.

The functions @AND and @OR are used similarly. If *all* values in a list are TRUE, @AND returns TRUE; otherwise it returns FALSE. If *any* value in a list is TRUE, @OR returns TRUE; otherwise it returns FALSE. Remember, a list can be composed of values, cell references, formulas, or ranges.

The functions @ISERROR and @ISNA turn ERROR or NA displays into TRUE or FALSE. @NOT turns TRUE into FALSE and vice versa.

Now that all of that is clear, you might reasonably wonder, "So what?" How can we use all this? The @IF function we discussed is, at least for *VisiCalc*, fairly straightforward. We've already seen how @ISERROR and @ISNA can make ERROR and NA displays useful to the logic functions; now we'll look at the other functions.

Let's start with another example. Coming back to our sales manager, we could hypothesize that he might want to compare the sales figures on different machines he sells against the national average. His template for this purpose would look like figure 4.

	A	B	C	D	E	F
1						
2			Regional Sales	National Sales		
3	Copier A		203	180		TRUE
4	Copier B		85	94		FALSE
5	Copier C		116	114		TRUE
6	Copier D		12	22		FALSE
7						
8				Comparison		FALSE
9						

Figure 4.

Columns C and D contain the sales figures for the sales manager's region and the national sales average. Column F tells the sales manager at a glance which copiers are selling at a rate that exceeds the national average. The formula in F3 is +C3>D3,F4 is+C4>D4, and so on. Because the sales manager expects his area to outsell the national average, the formula in F8 is @AND(F3. . .F6). The fact that there is a FALSE in the list (actually, there are two) causes F8 to display FALSE.

Suppose the formula in F8 had been @OR(F3. . .F6). What would the cell have displayed? TRUE. @OR only requires that one TRUE be included in the list. The sales managers for slower regions might want to know if they have exceeded the national average for sales of any particular machine. Of course it is easy to look at our two columns and pick out the differences, but imagine if there were fifty or sixty choices in each column. In such a case, @OR would be helpful.

Let's return to our example and see how we can modify the template to tell us something else. Assuming that our example has fifty comparisons, how can we total the number of comparisons that are TRUE?

In cell G3 enter @IF(F3,1,0) and replicate it through G6 (using relative reference). For this example, imagine that we have fifty cells in this column. In F8 you have only to enter @SUM(G3. . .G6) and you'll have the total number of copiers whose sales exceeded the national average. If you want to compare those that sold at a rate equal to or greater than the national average, you have only to change the comparisons in column F to be >= and G8 will tell you any copiers that are selling at a rate that's equal to or better than the national average.

If you were very aggressive and only wanted to know the copiers that were selling at a rate less than the national average, you could enter @IF(@NOT(F3),1,0) and replicate that. It would take the TRUE displays in column F and turn them into FALSE, and all FALSE displays into TRUE. Your answer in G8 would now display the number of copiers that sold at a rate less than the average. Of course you could have changed the logical operators in column F to compare for less than (for example, in F3: +C3<D3).

Functions without Arguments

@ERROR
@FALSE
@NA
@TRUE

Figure 5

That leaves us with one more type of logic function to discuss. Figure 5 shows a list of functions without arguments.

While the functions shown in figure 5 aren't really logic functions, they are modifiers for logic functions. You might notice that the @PI function isn't included in our list. That's because pi isn't a logic function, it's a mathematical function.

As we discussed last month, @ERROR and @NA can be used in debugging formulas, but they can also be used to key specific answers, especially in connection with the @CHOOSE function.

Nonlogic Logic? @LOOKUP and @CHOOSE are search functions, not logic functions. While it is true that these functions don't use logical operators to find a value, like @IF they return a choice of different values.

The look-up function is written @LOOKUP(v,range). *Choose* is written @CHOOSE(v,list). The two functions differ in two ways. First, *lookup* requires a contiguous range. That means a continuous series of cells in either a row or column. There should also be a parallel series of values either in the cells directly to the right of the column or directly below the row. *Choose*, on the other hand, requires a list. A list can be composed of ranges, values, and cell references mixed together. It doesn't require a second series of values.

The second and most important difference between the look-up and choose functions is the location and manner in which each determines the value it will return. *Lookup* takes the value represented by v in the look-up statement and looks through the range stated in the second part of the look-up statement until it finds a value greater than *v* (the look-up value). It then returns the value in the cell to the right of the previous value if the range is a column, or below the previous cell if the range is a row. *Choose* takes the value represented by *v* (which can be a value, a cell reference, or another function) and counts that many values through the list and returns that value. For example, if *v* is 5, *choose* will return the fifth value in the list. If the function is stated:

@CHOOSE(5,A15. . .A17,237,21,@AVERAGE(L@23. . .29))

the choose statement will return 21.

Remember that *lookup* requires two continuous and parallel columns or rows of information. One is for look-up and one is for data. *Choose* requires only a single series of values, and the values can be in any number of locations.

You can use @IF as the value in either function to provide true logic capability. Either *choose* or *lookup* can be the *v* in an @IF statement. In addition, *choose* or *lookup* can be either, or both, of the *then* and *else* parts of an @IF formula. Sounds good, but how does it work?

Let's look at a couple of examples that use *lookup* and *choose*. Assume we own a mail order business where people order different combinations of products. We've decided that we'll ship any package weighing up to ten pounds by air freight and anything weighing between ten and twenty pounds by normal shipping carrier. We've also decided not to ship any boxes that weigh more than twenty pounds.

Our template should take the weight of a box, decide whether the amount is less than or greater than ten pounds, and then apply the per pound shipping rate from the specified rate table. The sample template is shown in figure 6.

Note that the rates shown in this template are only examples and have no relation to actual shipping rates.

The function in D5 calculates shipping cost. Before reading any further, why don't you stop and take a shot at solving our problem? It's fairly simple, and you have enough information.

Okay, let's see how we can solve this problem. Each time we get a box to ship we have to decide whether it weighs more or less than ten pounds. If it weighs ten pounds or less we should look in the air shipping rate schedule to compute the shipping charge. If the box weighs more than ten pounds, we should look in the regular shipping schedule. Given the

	A	B	C	D	E	F
1						
2		Weight of Package:		19 lbs		
3						
4		Cost to Ship:		$ 187.73		
5						
6		Air			Regular	
7						
8		Lbs	Cost/lb		Lbs	Cost/lb
9		1	1.87		11	3.23
10		2	2.06		12	3.71
11		3	2.26		13	4.27
12		4	2.49		14	4.91
13		5	2.74		15	5.65
14		6	3.01		16	6.50
15		7	3.31		17	7.47
16		8	3.64		18	8.59
17		9	4.01		19	9.86
18		10	4.41		20	11.36
19						

Figure 6.

information, the answer is easy to figure. The function in D5 that calculates shipping cost is:

@IF(D3<11,@LOOKUP(D3,B10. . .B19)
*D3,@LOOKUP(D3,E10. . .E19)*D3)

The @IF function decides which rate table should be used, and the appropriate *lookup* takes the weight, finds the appropriate rate, and multiplies the rate times the weight (D3). Simple and elegant.

What about *choose*? Now that you have seen how *lookup* works, can you take the same information and use the choose function to determine the shipping cost? You can enter the choose function in D5 or in an equivalent cell for comparison.

The function is:

@CHOOSE(D3,C10. . .C19,F10. . .F19)*D3

Choose takes the weight and counts through the first range, and then the second, and so on, until it counts to the correct value. In our example, *choose* counts through the values until it comes to the nineteenth value. You'll notice that the list is in different columns than in the look-up statement. *Choose* uses the columns that contain the actual shipping rates.

In our example, both methods calculate the same answer. Normally, each function would be better used for solving different types of problems.

Both of these methods are subject to the same rounding problems that we discussed in this column two months ago. To see an example, enter ten pounds in D3 and see how that compares to ten times the ten pound rate. You can use the rounding solution in your manual if you wish.

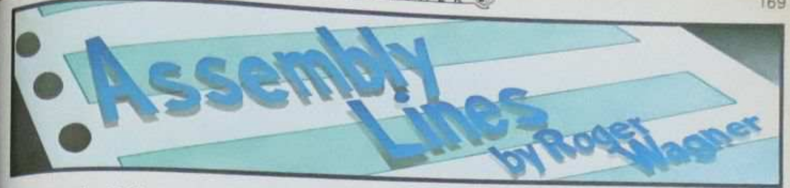
The look-up example could have been accomplished with a single look-up statement rather than with an @IF statement if we hadn't wanted to keep the charts separate. To do this, you would put all of the rates in a continuous column (or row) and just use *lookup* to calculate the answer.

If an amount greater than twenty pounds is entered, the choose function will return NA. This is so because the choose function won't be able to count past 20. *Lookup*, in contrast, will continue to use the last shipping rate. Using *lookup*, a hundred-pound box will be charged at the twenty-pound rate.

The End of Logic. When trying to solve logical problems (in *VisiCalc* and sometimes in life), the best way to start is to write your problem and solution in English and then convert it to *VisiCalc*. Usually, doing the conversion is much simpler than trying to solve the problem mathematically first.

Now you have seen some of the more powerful capabilities of *VisiCalc*. All the functions we've discussed, as well as most of the other functions, can be used in conjunction with one another. In fact, *VisiCalc* can accommodate up to nine nested functions.

The best way of learning to use logic functions is to try them out. In fact, that is the best way to learn about *VisiCalc*. Experiment—you'll be amazed at what it can do! ■



Everyone's Guide to Assembly Language, Part 28

This month's discussion centers on a little-mentioned operational mode of the 6502 microprocessor known as BCD, which stands for binary coded decimal.

In previous issues we've looked at arithmetic operations that use binary and hexadecimal representations of the numbers involved. Such operations often require a certain degree of mental translation to produce a decimal equivalent. In terms of printing a number in ASCII form, even more difficulty is to be expected if you're using your own conversion routines, rather than the built-in functions of DOS, Applesoft, and Integer Basic.

The BCD mode greatly simplifies this process by storing numbers in one or more byte registers (either memory, X, Y, or the accumulator) in a decimal-oriented manner. It does this by using two four-bit groups in each byte to represent a digit in base ten. In this way two digits per byte can be stored, thus giving a total value range of 0 to 99, versus 0 to 255 using binary.

Table 1 provides an example of how the BCD counting scheme goes.

Decimal	Hex	Binary	"Real Value"
0	$00	0000 0000	0
1	$01	0000 0001	1
2	$02	0000 0010	2
3	$03	0000 0011	3
.			
9	$09	0000 1001	9
10	$10	0001 0000	16
11	$11	0001 0001	17
.			
14	$14	0001 0100	20
15	$15	0001 0101	21
16	$16	0001 0110	22
17	$17	0001 0111	23
18	$18	0001 1000	24
19	$19	0001 1001	25
20	$20	0010 0000	32

Table 1

One of the nice things about hexadecimal notation is that each digit of the hex number represents one-half (four bits) of the binary number. This is a great help when you must mentally convert from hex to binary and back again. BCD is a variation on this theme in which the hex number really can be said to equal the decimal value (that is, the decimal and hex columns will *always* match).

About this time you may be thinking, "Well, that's all very nice, but where does the 6502 come into the picture?"

So far, all we have here is a possible system for storing decimal numbers via our usual hex bytes. The good news is that the 6502 actually supports this mode in the addition and subtraction operations.

That's right. The secret to making it work is to tell the 6502 that you wish to operate in this mode. This is done by means of the instruction SED, which stands for set decimal mode. Once this instruction has been executed, all future add and subtract operations will be done in the BCD mode. When you're done, be sure to clear everything back to normal with the CLD, for clear decimal mode, instruction.

Special note: Inadvertent setting of the decimal mode can cause the Apple to behave rather strangely and can be most puzzling when you're trying to debug programs. Reset does not clear the decimal flag (bit 3 of the status register). When in doubt do a call −155 or FF65G from the Monitor to clear the decimal mode.

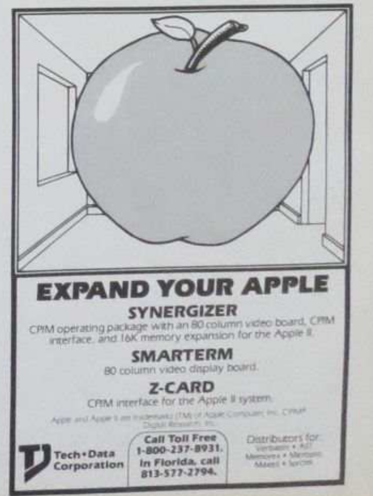
Let's verify that this mode actually works with a sample program.

```
              ****************************************
              *                                      *
              *          BCD DEMO ROUTINE #1         *
              *                                      *
              ****************************************
              *
8000: F8      START   SED                ;SET BCD MODE
8001: 18              CLC
8002: A9 12           LDA  #$12
8004: 69 34           ADC  #$34
8006: D8              CLD
8007: 00      DONE    BRK                ;BRK TO DISPLAY
```

Using the BRK command is an easy way both to end the program and display the result of the addition in the accumulator. When this routine is called with either an '8000G' or a call 32768 from Basic, you should get the Monitor break response with a display something like this:

```
8009-    A=46 X=90 Y=00 P=34 S=DE
```

Ignoring the rest of the line, when we see the A=46 we know that the accumulator holds '46', the correct result of the addition operation. You can substitute other numbers for '12' and '34' to verify that it works cor-



rectly with all legal values.

A similar experiment works with subtraction:

```
                        1  *****************************************
                        2  *                                       *
                        3  *        BCD DEMO ROUTINE #2            *
                        4  *                                       *
                        5  *****************************************
                        6  *
8000: F8                7  START  SED                ; SET BCD MODE
8001: 38                8         SEC
8002: A9 34             9         LDA  #$34
8004: E9 12            10         SBC  #$12
8006: D8               11         CLD
8007: 00               12  DONE   BRK               ; BRK TO DISPLAY
```

In this case, the result should be '22'. Again, you may wish to substitute different values to verify its operation.

For both addition and subtraction, results of the operations "wrap around" in a manner similar to the way hexadecimal calculations do. That is to say that 99 + 1 will give a result of '00' (100 less the leading '1') and 0 − 1 will give '99'.

Limitations. Like everything else in life, BCD has its tradeoffs and warnings. The first involves that vague reference made earlier to everything working with "legal values." "What's legal?" you may ask. You'll note that in table 1 certain hex values, such as $0A, never appear. This is because in the BCD mode such a value is "illegal" because it uses a digit out of the range of 0 to 9. If you attempt to use such a value in the BCD mode, you'll get inaccurate results.

To add to the fun, note also that the BEQ, BNE and INC, DEC families of instructions don't work as expected either. The N (sign) and Z (zero) flags are all linked to binary operations and not BCD. Thus 01 + 99 will yield 00, but N and Z remain unaffected, since the "true" binary result should have been $9A. Also, no provision is made for negative numbers (signed arithmetic). How, then, do we test for special conditions?

The Carry Flag! The carry flag is the only direct indication of arithmetic results in BCD. In addition operations, the carry will be set if the result exceeds 99 (overflow). In subtraction, the carry will be cleared if the result is less than 0 (underflow).

In multiple byte operations the carry is used in the same way as it is in "normal" hexadecimal arithmetic.

Common Operations. Since INC and DEC don't perform properly, their functions must be implemented by using the ADC and SBC instructions:

```
                        1  *****************************************
                        2  *                                       *
                        3  *      BCD DEMO 'INC' ROUTINE           *
                        4  *                                       *
                        5  *****************************************
                        6  *
                        7  MEM    EQU  $06
                        8  BEEP   EQU  $FBDD
                        9  *
8000: F8               10  START  SED                ; SET BCD MODE
8001: 18               11         CLC
8002: A5 06            12         LDA  MEM
8004: 69 01            13         ADC  #$01
8006: B0 04            14         BCS  ERR          ; OVERFLOW
8008: 85 06            15         STA  MEM          ; MEM=MEM+1
800A: D8               16         CLD               ; CLR BCD MODE
800B: 60               17  DONE   RTS
800C: 4C DD FB         18  ERR    JMP  BEEP
```

```
                        1  *****************************************
                        2  *                                       *
                        3  *      BCD DEMO 'DEC' ROUTINE           *
                        4  *                                       *
                        5  *****************************************
                        6  *
                        7  MEM    EQU  $06
                        8  BEEP   EQU  $FBDD
                        9  *
```

```
8000: F8          10 START  SED              ; SET BCD MODE
8001: 38          11        SEC
8002: A5 06       12        LDA  MEM
8004: E9 01       13        SBC  #$01
8006: 90 04       14        BCC  ERR         ; UNDERFLOW
8008: 85 06       15        STA  MEM         ; MEM=MEM-1
800A: D8          16        CLD              ; CLR BCD MODE
800B: 60          17 DONE   RTS
800C: 4C DD FB    18 ERR    JMP  BEEP
```

Notice how the carry status is checked to detect overflow (result > 99) or underflow (result < 0) in the addition and subtraction routines, respectively. MEM is a memory location presumed to hold a legal BCD value.

Multiple byte operations are done in a manner similar to the way their hexadecimal equivalents are handled:

```
                   1  *****************************************
                   2  *                                       *
                   3  *      BCD ADDITION ROUTINE             *
                   4  *                                       *
                   5  *****************************************
                   6  *
                   7  *
                   8  MEM1   EQU  $06        ; 6,7
                   9  MEM2   EQU  $08        ; 8,9
                  10  RSLT   EQU  $0A        ; A,B
                  11  BEEP   EQU  $FBDD
                  12  *
8000: F8          13 ENTRY  SED
8001: 18          14        CLC
8002: A5 06       15        LDA  MEM1
8004: 65 08       16        ADC  MEM2
8006: 85 0A       17        STA  RSLT
8008: A5 07       18        LDA  MEM1+1
800A: 65 09       19        ADC  MEM2+1
800C: 85 0B       20        STA  RSLT+1      ; OVERFLOW RSLT
                                               =MEM1+MEM2
800E: B0 02       21        BCS  ERR         ; OVERFLOW
8010: D8          22 DONE   CLD
8011: 60          23        RTS
8012: 4C DD FB    24 ERR    JMP  BEEP

                   1  *****************************************
                   2  *                                       *
                   3  *      BCD SUBTRACT ROUTINE             *
                   4  *                                       *
                   5  *****************************************
                   6  *
                   7  *
                   8  MEM1   EQU  $06        ; 6,7
                   9  MEM2   EQU  $08        ; 8,9
                  10  RSLT   EQU  $0A        ; A,B
                  11  BEEP   EQU  $FBDD
                  12  *
8000: F8          13 ENTRY  SED
8001: 38          14        SEC
8002: A5 06       15        LDA  MEM1
8004: E5 08       16        SBC  MEM2
8006: 85 0A       17        STA  RSLT
8008: A5 07       18        LDA  MEM1+1
800A: E5 09       19        SBC  MEM2+1
800C: 85 0B       20        STA  RSLT+1      ; RSLT=MEM1-MEM2
800E: 90 02       21        BCC  ERR         ; OVERFLOW
8010: D8          22 DONE   CLD
8011: 60          23        RTS
8012: 4C DD FB    24 ERR    JMP  BEEP
```

Printing BCD Values. One of the biggest advantages of BCD is that the values are easily printed to the screen or disk. When using hexadecimal math, some sort of hex to ASCII string decimal conversion routine is required. This is then followed by the printing of the digits via some string print routine. In BCD, only a minimal conversion is needed, and the printing is done fairly easily.

The easiest way to print a number is to use one of the Monitor routines. PRBYTE ($FDDA), for example, prints the contents of the accumulator as a hex byte. Here's a routine that takes two BCD values from memory and prints the sum:

```
                   1  *****************************************
                   2  *                                       *
                   3  *      BCD PRINT ROUTINE #1             *
                   4  *                                       *
                   5  *****************************************
                   6  *
                   7  *
                   8  MEM1   EQU  $06
                   9  MEM2   EQU  $07
                  10  PRBYTE EQU  $FDDA
                  11  BEEP   EQU  $FDBB
                  12  *
8000: F8          13 ENTRY  SED
8001: 18          14        CLC
8002: A5 06       15        LDA  MEM1
8004: 65 07       16        ADC  MEM2        ; ACC = MEM1 + M
8006: B0 05       17        BCS  ERR         ; OVERFLOW
8008: D8          18        CLD
8009: 20 DA FD    19        JSR  PRBYTE
800C: 60          20 DONE   RTS
800D: 4C BB FD    21 ERR    JMP  BEEP
```

You can experiment by putting different values in $06 and $07 and calling the routine. For two-byte values (0 to 9999) one can use PRNTAX ($F941), which expects the accumulator and X register to be loaded with the bytes to be printed prior to the call:

```
                   1  *****************************************
                   2  *                                       *
                   3  *      BCD PRINT ROUTINE #2             *
                   4  *                                       *
                   5  *****************************************
                   6  *
                   7  *
                   8  MEM1   EQU  $06        ; 6,7
                   9  MEM2   EQU  $08        ; 8,9
                  10  PRNTAX EQU  $F941
                  11  BEEP   EQU  $FBDD
                  12  *
8000: F8          13 ENTRY  SED
8001: 18          14        CLC
8002: A5 06       15        LDA  MEM1
8004: 65 08       16        ADC  MEM2
8006: AA          17        TAX              ; STORE RSLT IN X
8007: A5 07       18        LDA  MEM1+1
8009: 65 09       19        ADC  MEM2+1      ; RSLT+1 IN ACC
800B: B0 05       20        BCS  ERR         ; OVERFLOW
800D: D8          21        CLD              ; CLR FOR PRNTAX
800E: 20 41 F9    22        JSR  PRNTAX
8011: 60          23 DONE   RTS
8012: 4C DD FB    24 ERR    JMP  BEEP
```

It is important to notice that in each routine the CLD is used to clear the decimal mode *before* calling PRBYTE or PRNTAX. This is so because the Monitor needs the normal binary mode to calculate screen addresses and positions properly. If you call the Monitor with the BCD mode set, strange things will happen when the text reaches the end of the line or the screen needs to be scrolled and the Monitor routines attempt to calculate where to put the next line of text.

If you don't want to use the Monitor byte print routines or, for whatever reason, just want to create the ASCII characters yourself, the con-

Letter	ASCII Value (*)	BCD Value
0	$B0	$00
1	$B1	$01
2	$B2	$02
3	$B3	$03
4	$B4	$04
5	$B5	$05
6	$B6	$06
7	$B7	$07
8	$B8	$08
9	$B9	$09

(*) high bit set

Table 2

versions are straightforward and COUT ($FDED for character *output*—usually pronounced "C-out") can be used directly.

The only real obstacle is how to convert the BCD digits to their ASCII equivalents. As it happens, this is even easier to do than you might at first suppose. Consider table 2.

From looking at table 2, we can see that the lower digit of the ASCII value corresponds to the digit encoded in the BCD format and, coincidentally enough, to the number itself to be printed. If there was a way of adding $B0 to the value for the digit to be printed, we'd have just the value we needed to send to COUT to print the appropriate character.

To add #$B0 to the BCD values shown would normally require the usual CLC, ADC instructions. There is a more elegant (that is, shorter) way, however. You may remember that the 'ORA' (for logical OR function) can be used as a mask to perform an overlaylike operation. Here's how a possible ORA operation would appear:

```
Accumulator:    0000 0110    ($06 BCD)
ORA #$B0:       1011 0000
---------------------------
Result:         1011 0110    ($B6 = ASCII "6")
```

What if the upper BCD digit is involved? The procedure then is first to shift the upper four bits "down" to the lower nibble position:

```
BCD value:      0110 0000    ($60 BCD)
LSR             0011 0000
LSR             0001 1000
LSR             0000 1100
LSR             0000 0110
---------------------------
Result:         0000 0110    ($06 BCD)
```

Ah, you ask, what if *both* digits possible are indicated by the BCD value? The answer here is first to shift out the lower nibble, as was just shown, and to print the ASCII character arrived at. Then the original value is reloaded into the accumulator and the upper nibble is masked out. This can be done using the 'AND' instruction, which has the ability to clear a designated portion of a byte to zeros. For example:

```
Accumulator:    0101 0110    ($56 BCD)
AND #$F0:       0000 1111
---------------------------
Result #1:      0000 0110    ($06)
ORA #$B0:       1011 0000
---------------------------
Result #2:      1011 0110    ($B6 = ASCII "6")
```

Here, then, is the complete routine:

```
                   1  *****************************************
                   2  *                                       *
                   3  *      BCD PRINT ROUTINE #3             *
                   4  *                                       *
                   5  *****************************************
                   6  *
                   7  *
                   8  MEM    EQU  $06
                   9  COUT   EQU  $FDED
                  10  *
8000: D8          11 ENTRY  CLD              ; BCD MODE NOT NECC
8001: A5 06       12        LDA  MEM         ; GET BCD NUMBER
8003: 4A          13        LSR              ; SHIFT UPPER NIBBLE
8004: 4A          14        LSR              ; TO BOTTOM POSN
8005: 4A          15        LSR
8006: 4A          16        LSR
8007: 09 B0       17        ORA  #$B0        ; %1011 0000
8009: 20 ED FD    18        JSR  COUT        ; PRINT DIGIT
800C: A5 06       19        LDA  MEM         ; RETRIEVE ORIG BCD
800E: 29 0F       20        AND  #$0F        ; %0000 1111
8010: 09 B0       21        ORA  #$B0        ; %1011 0000
8012: 20 ED FD    22        JSR  COUT
8015: 60          23 DONE   RTS
```

The CLD is done at the beginning just to emphasize that the BCD mode is not required here since the digit is presumed to exist already in MEM and no arithmetic operations are anticipated. Remember that the BCD mode is only required during the actual addition or subtraction operations. Although the BCD mode would have no harmful effect on the AND or ORA operations, COUT would certainly take offense at being called while the BCD mode was still in effect.

Lines 12 through 16 get the original BCD value from memory and then shift it left four times to move the upper nibble to the lower position. At this point the ORA #$B0 is done to convert the value in the accumulator to the proper ASCII value, at which point the JSR COUT on line 18 prints the first digit. Line 19 retrieves the original value again, after which the AND #$0F clears the upper digit to '0' and the ASCII conversion is completed and printed as before.

The remainder of the routine is identical to the previous example program.

When run, this program will provide visual feedback as to what happens each time a key is pressed. It will show the buffer being updated each time as well as the new POSN value for each digit entered.

Conclusions. The Binary Coded Decimal mode of the 6502 can be convenient for a variety of reasons. The most frequent use is to facilitate input and output, particularly for scientific instrumentation.

A number of points should be kept in mind when using the BCD mode:

1. The mode should be set only for arithmetic processes that use BCD values, such as addition and subtraction.
2. Only legal values are allowed. Values outside the expected range will generate inaccurate results.
3. The BCD mode should be cleared as soon as possible when arithmetic operations are completed so as to avoid possible complications with other software in the Apple that neither expects, nor checks for, the BCD mode.
4. Reset does not clear the decimal mode of the 6502. Only the CLD instruction does. You can also clear the mode by means of a call −155 or an FF95G from the Monitor.
5. The N and Z flags are unreliable as a means of detecting the results of comparisons or of increment/decrement operations. Only the carry should be used to detect the results of such operations.
6. The carry will be set for results greater than 99 (overflow) and cleared for results less than 0 (underflow).
7. BCD operations *do* "wrap around." That is, 99 + 01 = 00 and 00 − 01 = 99.

Special Note: These are some general rules to help in programs using the decimal mode of the 6502. There is only one notable exception that may on occasion prove useful. The test for zero (BNE, BEQ) can be used when counting *down* in the BCD mode. For example,

```
SED
SEC
LDA #$01
SBC #$01
BEQ DONE
```

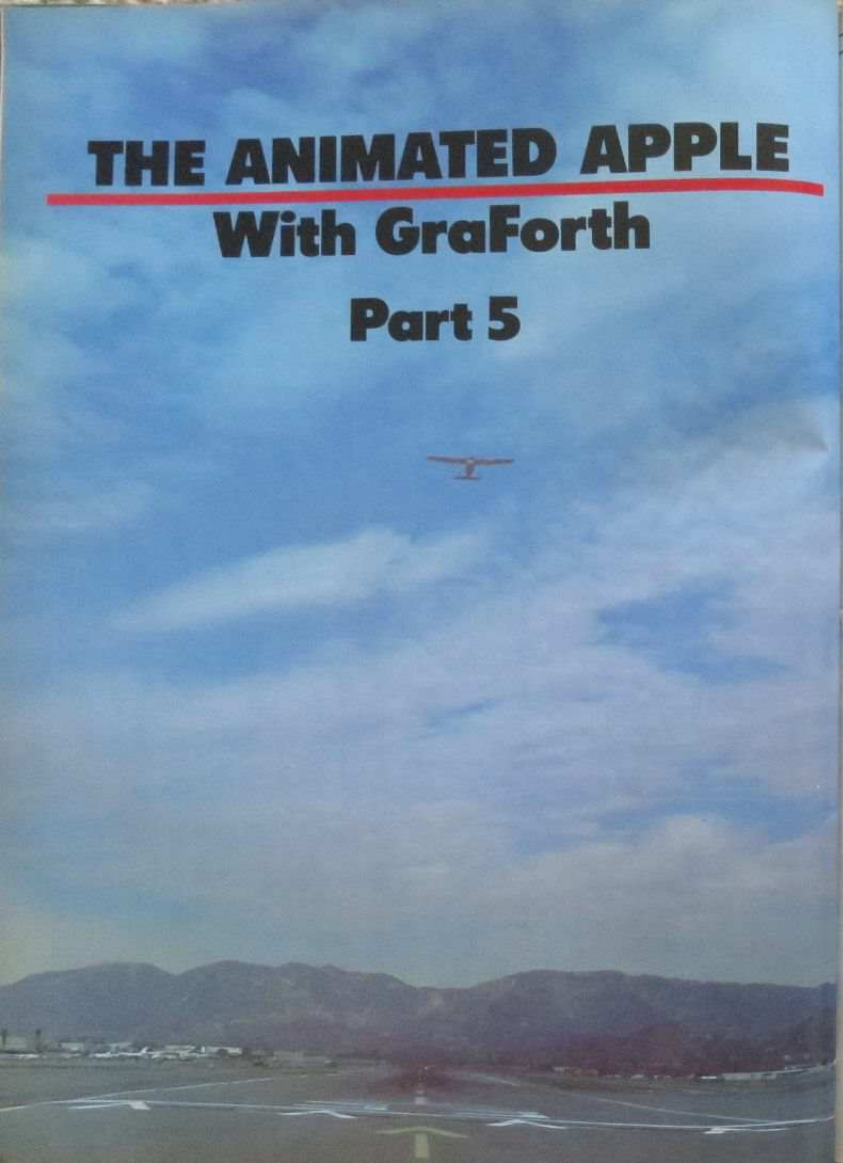
would work, whereas,

```
SED
CLC
LDA #$99
ADC #$01
BEQ DONE
```

would not.

It might be an interesting challenge for you to use the information given in this column and previous issues to try to write a routine that would add two Applesoft strings together using the BCD mode and return the result in a third string. This would provide a way of extending the normal precision of Applesoft for mathematical operations requiring more than nine digits, a problem that unfortunately does not hinder my personal checkbook program.

See you next month!



THE ANIMATED APPLE
With GraForth
Part 5

BY PAUL LUTUS AND PHIL THOMPSON

Last month's introduction to GraForth's three-dimensional graphics demonstrated some straightforward methods for manipulating 3-D images. With a better understanding of how GraForth handles things internally, we can create some interesting new effects.

Remember that using 3-D graphics is a two-step process: First the *Image Editor* is used to create a 3-D *image* in memory; then GraForth's 3-D commands are used to display the image as an *object* on the screen, with a given position, size, and orientation. The object commands do not affect the image in memory; they only change the way it is displayed.

For some 3-D animations, you may want to have several images in memory at one time. In order to keep track of what free areas of memory are available, it's handy to know how the images are stored.

A 3-D image is made up of a number of line entries, one for each move or draw in the image. Each line entry uses four bytes of memory. The format for the entry is described fully on page B-6 of the GraForth manual. Briefly, the first byte determines color and whether the entry is a move or a draw, and the next three bytes specify the X, Y, and Z positions for the point in space. The end of a 3-D image is marked with a 255 (hex $FF) in memory immediately after the last entry. The length in bytes of a 3-D image is four bytes multiplied by the number of entries plus one more byte for the end-of-image marker.

From the *Image Editor*, the List and Enter options show the address in memory for the beginning of each line entry. An image starting at address 2816 will have line entries at 2816, 2820, 2824, and so on. The last address shown in the listing is for the beginning of the last line entry. The end-of-image marker will be four bytes after this. The byte immediately after the end-of-image marker (five bytes after the last listed address) is the next byte free for use.

Loading the *Image Editor* every time you want to find the length of a 3-D image can be cumbersome. The following short word definitions will also do the job. With a 3-D image already in memory, *end.3D* can be used to find the next free location after the image, and *length.3D* will return the length of the image in bytes:

```
: END.3D
BEGIN
    DUP
    PEEK 255 <>
WHILE
    4 +
REPEAT
1 + ;
: LENGTH.3D
DUP END.3D
SWAP - ;
```

To use end.3D, simply place the starting address of the image on the stack, then call end.3D. The routine scans through the list of four-byte line entries until it finds the 255 end-of-image marker. It then adds 1 to this address, leaving the address of the next free byte on the stack.

Length.3D is called the same way. It uses end.3D to find the next free address, then subtracts the original starting address from this value to find the length of the image.

Here is an example that uses end.3D and length.3D to find the length of the XYZ image stored on the GraForth system disk. First, load the XYZ image into memory starting at location 2816:

```
CR 132 PUTC PRINT " BLOAD XYZ.A2816 " CR
```

To find the next free address after the XYZ image, enter:

```
2816 END.3D .
```

End.3D returns 2877, the address of the next free area of memory. This means that the XYZ end-of-image (255) marker must be one location before 2877:

```
2876 PEEK .
```

This should display a 255.

The length of the XYZ image can be found by typing:

```
2816 LENGTH.3D
```

In this case, you should get a length of 61.

The Keep command in the *Image Editor* allows you to save 3-D images to disk, one image per binary file. If you're working with a graphics program that uses several images, it's often easier to combine all of the images into one disk file. By keeping track of image-length information, you can bload several images into memory one immediately after another, then bsave the images back to disk as a single binary file. This can save a lot of space on a crowded disk and make it easier to load several images at one time. The 3-D bat used in *Die Fledermaus* in the demonstration program is an example of this. Three separate images making up the bat are stored together in the binary file Bat.

Another technique is to combine two 3-D images together into one larger image. We can create a "double" image by concatenating the XYZ image with the Cube image, also on the GraForth disk. First, let's get the XYZ shape displayed on the screen:

```
OBJERASE
0 OBJECT 2816 OBJADR
0 40 20 24 WINDOW ERASE
12 SCALE 4 SCALZ 80 YPOS
10 XROT 20 YROT DRAW
```

We can bload the Cube image right over the end of XYZ, so that the first line entry in the cube overwrites the end-of-image marker for XYZ. This leaves a single image with a set of line entries for XYZ and a set for the cube, with one end-of-image marker left at the end.

We know that the XYZ end-of-image marker is at location 2876, so this is where we load the cube:

```
CR 132 PUTC PRINT " BLOAD CUBE.A2876 " CR
0 OBJECT DRAW
```

Other unusual effects are possible by changing line entries from a running program, moving end-of-image markers, and so on. Keep in mind, however, that the image in memory is also used by GraForth when erasing old objects from the screen before redrawing. If you modify images while displaying them, you might affect what lines from old objects are erased and what lines aren't.

Another internal aspect of 3-D graphics worth looking at is the Image Data Map (described on page B-4 of the GraForth manual), which stores the parameters used to orient and draw objects on the screen. (A more accurate name would be Object Data Map, but we'll stick to the name used in the manual.) The map keeps track of all of the 3-D parameters for sixteen objects for three different functions: erasing the old picture, keeping track of the picture on the other graphics screen, and

Function	Relative Byte
Flag (draw, nodraw)	0
XROT	1
YROT	2
ZROT	3
XTRAN	4
YTRAN	5
ZTRAN	6
XPOS	7
YPOS	8
SCALX	9
SCALY	10
SCALZ	11
OBJCOLOR	12
Image Address	13 and 14
(unused)	15

The starting addresses for the data sets are:

Decimal	Hex	Data Set
5888	$1700	Undraw
6144	$1800	Interim
6400	$1900	Draw

Table 1. Object table format and addresses.

drawing the new picture. The manual calls these data sets Undraw, Interim, and Draw.

Last month's column included a program that used variables to keep track of 3-D parameters. However, the Image Data Map already stores this same information and can be used the same way. The technique is to determine the address for the desired parameter of a given object, then take a peek at that address to find the current value of the parameter.

The Image Data Map is first broken into the three data sets, with sixteen object tables for each data set and sixteen bytes for each object table. Table 1 shows the format for object tables.

Here's an example to show how it all works. Suppose you want to find the current X position (xpos) for object 5. Start with the base address for the Draw data set at 6400. Multiply the object number by 16 and add this to the base address:

```
6400 + (5 * 16) = 6480
```

Now add the relative byte for the xpos command, which is 7:

```
6480 + 7 = 6487
```

By this procedure you can determine that 6487 is the address that stores the X position for object 5. You can find the current value by peeking at this address.

Using this technique, you can define words to update a 3-D parameter by peeking its value, adding some offset, then resetting the parameter. Here is a word that will increase the X position for object 5 by two pixels:

```
: INCX
  5 OBJECT
  6487 PEEK 2 + XPOS ;
```

The following is another version of the *Roll.tetra* program shown last month. It uses the Image Data Map to update parameters so separate variables are not needed:

```
0 VARIABLE OBJ   ( Object number )
  VARIABLE DIR   ( Scale direction larger or smaller?)

: FIND                  ( Find address of parameter given relative byte: )
  OBJ 16 * +            ( Object number * 16 + relative byte )
  6400 + PEEK ;         ( Add base address and peek current value )

: UPDATE.TETRA
  1 FIND 3 + XROT       ( Increase X rotation by 3 )
  2 FIND 5 + YROT       ( Increase Y rotation by 5 )
  7 FIND 6 + XPOS       ( Increase X position by 6 )
  8 FIND 3 + YPOS       ( Increase Y position by 3 )
  DIR IF                        ( If scale is increasing: )
    9 FIND 1 + SCALE            ( Increase scale by 1 )
    9 FIND 18 = IF 0 -> DIR THEN        ( Change direction? )
  ELSE
    9 FIND 1 - SCALE    ( Decrease scale by 1 )
  THEN ;

: ROLL.TETRA2
  OBJ OBJECT            ( Set object number )
  0 XROT 0 YROT         ( Initialize parameters )
  25 XPOS 35 YPOS
  0 SCALE
  1 -> DIR              ( Set scale direction )
  36 0 DO               ( Start loop )
    UPDATE.TETRA        ( Set new parameters )
    DRAW                ( Draw object )
  LOOP ;                ( Loop back )
```

Special provision is made to work with any object, regardless of what object number is used. The object number is kept in the variable OBJ. Note the new word *find*. This takes the relative byte for a 3-D parameter from the stack, finds the address for that parameter, and peeks the current value.

To run the program, first bload the tetrahedron image into memory:

```
CR 132 PUTC PRINT " BLOAD TETRA.A2816 "
OBJERASE
0 OBJECT 2816 OBJADR
```

Now store the object number into OBJ and call *Roll.tetra*:

```
0 -> OBJ
ROLL.TETRA
```

If you want to use other images with *Roll.tetra*, you may need to adjust the scale so that the object fits on the screen.

Another aspect of 3-D graphics to consider is the way in which images are transformed into objects on the screen. The three-dimensional mathematical method on page B-5 of the manual describes some of the technical wizardry performed on the points using matrices. Understanding the details of the math isn't really important. What is important is the order in which the operations are performed.

The 3-D transformations are performed in this order:

1. X, Y, and Z translation
2. X, Y, and Z scaling
3. X rotation
4. Y rotation
5. Z rotation

This means that the 3-D points in an object are first translated, and then the translated points are scaled. These new points are rotated around the X, Y, and Z axes in order. Actually, GraForth does steps two through five simultaneously, though the formula used treats the objects as if the transformations were done one at a time.

The order of the steps affects rotations most noticeably. Consider an object that has been set to *30 xrot 20 yrot*. The X rotation is performed first. Remember that the X axis passes through the object from left to right. A rotation around this axis tips the top of the object forward and the bottom back. The Y rotation is done next. But since the X rotation occurred first, the Y axis has been tipped forward, along with the points in the object. The Y rotation will actually be done on this new tipped axis. Similarly, any Z rotation that follows will rotate around an axis that has already been tipped by both X and Y rotations.

Suppose you want to create an animation of a 3-D spaceship that continually banks or rolls as it flies across the screen. This rolling should be the same regardless of which way the ship is facing. Before designing the ship with the *Image Editor*, you need to know which axis (X, Y, or Z) to use for the rotation. The Z axis is the proper choice for this kind of situation. The rotations around the X and Y axes will first set the ship (and the Z axis) facing the desired direction, then the Z axis rotation will perform the rolling in that direction.

This is best demonstrated using the program *Play*. Compile *Play* into memory and execute it:

```
READ " PLAY "
PLAY
```

Select the 3-D shape XYZ from the disk. When the object appears, rotate it a little so that you can see all three arrows: Press 2, right arrow, F, 1, right arrow, F. The white arrow pointing from back to front corresponds to the Z axis. Start the object rotating around this axis at a good rate by pressing 3 and then pressing the right arrow key about six times. Notice that the rotation does occur around the white arrow.

Now press either 1 or 2 and an arrow key, wait a bit, and then press F. This changes the rotation around the X or Y axis. Note that, even though the white arrow is now facing a new direction, the Z rotation is still around the white arrow. This is so because the X and Y rotations tip the Z axis before the Z rotation is done.

Because of these rotation gymnastics, it is handy occasionally to have the ability to transpose a 3-D image in memory, changing X coordinates into Y coordinates, Y to Z, Z to X, or whatever. This has the effect of turning an image around so that it lies in a new direction along the three axes.

The following word definition does just that. It reads each line entry of an image in memory, plucks out the values for the (X,Y,Z) point, and places them back into the image in a new order:

```
VARIABLE ADDR
VARIABLE FIRST
VARIABLE SECOND
VARIABLE THIRD
VARIABLE X
VARIABLE Y
VARIABLE Z
```

```
TRANSPOSE
  – >THIRD – >SECOND – >FIRST      ( Save new order )
  – > ADDR                         ( Save starting address )
BEGIN
    ADDR PEEK 255 <>               ( While not end-of-image: )
WHILE
    ADDR 1 + PEEK – > X            ( Peek X, Y, and Z values
                                     from image )
    ADDR 2 + PEEK – > Y
    ADDR 3 + PEEK – > Z
    X  ADDR FIRST + POKE           ( Poke values back into
                                     image in new order )
    Y  ADDR SECOND + POKE
    Z  ADDR THIRD + POKE
    ADDR 4 + – > ADDR              ( Increment address to next
                                     line entry )
REPEAT ;
```

Before calling *Transpose*, four values should be on the stack. They are the image address, the new X place, the new Y place, and the new Z place. The numbers for the places should be 1, 2, and 3, in the desired order: 1 for X, 2 for Y, and 3 for Z. For example, to trade the X and Y coordinates in an image at location 2816, you would enter:

```
2816 2 1 3 TRANSPOSE
```

To run *Transpose* without exchanging coordinates (a do-nothing operation), you would type:

```
2816 1 2 3 TRANSPOSE
```

The numbers 1, 2, and 3 must be on the stack in some order or the image will be destroyed. For example, typing

```
2816 2 2 2 TRANSPOSE
```

copies all three coordinates in turn into the Y coordinate position, with the Z coordinate copied last, and loses the old Y value.

The concepts we've been dealing with are more than isolated ideas. They can be combined to produce interesting new animations. Now let's look at a space shuttle simulation program that makes extensive use of 3-D graphics. The program shows the shuttle flying into view over the earth, performing some rotations, releasing a satellite, and flying away. The graphics dumps in figure 1 comprise a storyboard of the animation. The simulation makes use of a number of concepts, including multiple 3-D objects, undraw, GraForth's Image Data Table, and Z axis rotations.

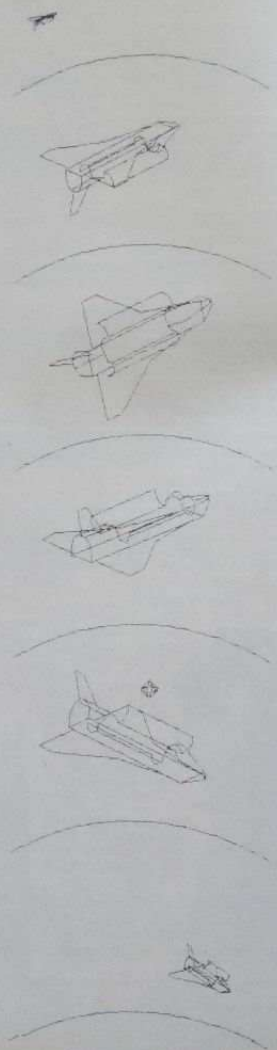


Figure 1

The first step is to create the shuttle using the *Image Editor*. Below are the line entries that make up the shuttle. Read the *Image Editor* onto the word library (*read " imageditor "*), run the program, and type Z to zero any image (or garbage) that may be in memory. You can set a viewing angle before entering points. Enter a scale of 14, an X rotation of 20, and a Y rotation of 32. Now press E to enter the points. The general shape of the shuttle should become apparent after the first fifteen or twenty line entries. When asked for color at each line entry, just press return.

Note that the shape is somewhat complicated. This gives the shuttle greater detail, but it requires more time to draw. To speed the program up, you may want to design a simpler shuttle later.

	Body		X	Y	Z
1.		Move	−12	−20	64
2.		Draw	−12	−20	80
3.		Draw	−12	−10	90
4.		Draw	−4	0	127
5.		Draw	4	8	127
6.		Draw	20	16	100
7.		Draw	20	16	−120
8.		Draw	−20	16	−120
9.		Draw	−20	16	100
10.		Draw	−4	8	127
11.		Draw	4	0	127
12.		Draw	12	−10	90
13.		Draw	12	−20	80
14.		Draw	12	−20	64
15.		Move	20	16	−120
16.		Draw	20	0	−120
17.		Draw	17	−12	−120
18.		Draw	8	−20	−120
19.		Draw	−8	−20	−120
20.		Draw	−17	−12	−120
21.		Draw	−20	0	−120
22.		Draw	−20	16	−120
23.		Move	8	−20	−120
24.		Draw	8	−20	−64
25.		Move	−8	−20	−120
26.		Draw	−8	−20	−64
27.		Move	20	16	100
28.		Draw	−20	16	100
	Wings				
29.		Move	−20	16	−112
30.		Draw	−88	16	−112
31.		Draw	−88	16	−84
32.		Draw	−36	16	−10
33.		Draw	−20	16	80
34.		Move	20	16	−112
35.		Draw	88	16	−112
36.		Draw	88	16	−84
37.		Draw	36	16	−10
38.		Draw	20	16	80
	Cargo opening				
39.		Move	20	0	−64
40.		Draw	17	−12	−64
41.		Draw	8	−20	−64
42.		Draw	−8	−20	−64
43.		Draw	−17	−12	−64
44.		Draw	−20	−12	−64
45.		Draw	−20	0	−64
46.		Draw	−20	0	64
47.		Draw	−17	−12	64
48.		Draw	−8	−20	64
49.		Draw	8	−20	64
50.		Draw	17	−12	64
51.		Draw	20	0	64
52.		Draw	20	0	−64
	Fin				
53.		Move	0	−20	−120
54.		Draw	0	−60	−128
55.		Draw	0	−60	−112
56.		Draw	0	−20	−84
	Open doors				
57.		Move	20	0	−64
58.		Draw	33	−3	−64
59.		Draw	40	−12	−64
60.		Draw	40	−20	−64
61.		Draw	40	−20	64
62.		Draw	40	−12	64
63.		Draw	33	−3	64
64.		Draw	20	0	64
65.		Move	−20	0	−64
66.		Draw	−33	−3	−64
67.		Draw	−40	−12	−64
68.		Draw	−40	−20	−64
69.		Draw	−40	−20	64
70.		Draw	−40	−12	64
71.		Draw	−33	−3	64
72.		Draw	−20	0	64

After creating the shuttle, press K to save the image to disk. Use the file name Shuttle. At this point, you might want to use the *Play* program to check the shuttle image more closely. Remember to forget the *Image Editor* program (*forget X*) before typing *read " Play "*. From *Play*, you can easily see the shuttle from all angles.

The next step is to create the satellite. This is a simple shape created with the *Profile* program. (*Profile* is described in greater detail on pages 8-18 to 8-21 in the GraForth manual.) Use forget to remove any other programs from the word library, then type:

```
READ " PROFILE "
RUN
```

Answer the questions as follows:

```
Enter number of polygon sides : 6
Enter Object File Address : 3500
Data from (K)eyboard or (D)isk ? K
Enter X,Y pair (end = "E") : 0,20
Enter X,Y pair (end = "E") : 20,0
Enter X,Y pair (end = "E") : 0,−20
Enter X,Y pair (end = "E") : E
```

The program will generate a small twelve-sided, diamondlike shape. This is the satellite. As prompted, save the image to disk with the name Satellite.

We now have the 3-D images needed for the simulation. The program must be entered next. Clear the word library, then type:

TEXT EDIT

to enter the text editor. Calling text switches GraForth out of graphics mode so that scrolling in the editor will go much faster. Type E to erase the editor memory, then enter the following word definitions. As usual, the comments are optional:

```
SETUP
CR .132 PUTC .PRINT " BLOAD SHUTTLE.A2816 "
CR .132 PUTC .PRINT " BLOAD SATELLITE.A3500 " CR
OBJERASE
0 OBJECT 2816 OBJADR !
1 OBJECT 3500 OBJADR 0 SCALE
39 19 BLKSIZE !

PARAM
SWAP 16 * 6400 + +                ( Retrieve address of parameter )
PEEK ;                            ( And peek current value there )

FLY.IN
ERASE 0 VTAB 0 HTAB               ( Set position for UNDRAW )
PENUP
0 191 MOVETO 60 TURNTO            ( Initialize turtle and )
PENDOWN
36 0 DO 4 MOVE 1 TURN LOOP        ( Draw outline of Earth )
0 OBJECT
-20 XROT 32 YROT 128 ZROT         ( Set initial position of shuttle )
18 XPOS 132 YPOS 0 SCALE
21 0 DO                           ( Fly shuttle into view: )
   0 7 PARAM 5 + XPOS               ( Move to the right )
   0 8 PARAM 3 - YPOS               ( Move upward )
   0 9 PARAM 1 + SCALE              ( Increase in size )
   UNDRAW DRAW
LOOP ;

ROTATE1
-1 124 DO
   1 ZROT                         ( Roll to upright position )
   UNDRAW DRAW
-4 +LOOP
21 -18 DO
   1 XROT                         ( Tip down for better view )
   UNDRAW DRAW
2 +LOOP ;

RELEASE
1 OBJECT 128 XPOS 66 YPOS         ( Select and position satellite )
16 SCALE 20 XROT 32 YROT
10 0 DO
   0 OBJECT
   0 8 PARAM 2 + YPOS             ( Move shuttle down )
   1 OBJECT
   1 8 PARAM 2 - YPOS             ( Simultaneously move satellite up )
   DRAW
LOOP ;

DRIFT.AWAY
0 OBJECT                          ( Select to redraw shuttle )
1 OBJECT                          ( Select satellite )
-1 15 DO                          ( Quickly drift away: )
   1 SCALE                          ( Decrease in size )
   1 7 PARAM 3 + XPOS               ( Move to the right )
   1 8 PARAM 3 - YPOS               ( Move up )
   DRAW
-1 +LOOP ;

ROTATE2
0 OBJECT                          ( Reselect shuttle )
105 32 DO
   1 YROT                         ( Pivot around )
   UNDRAW DRAW
4 +LOOP
-21 18 DO
   1 XROT                         ( Tip down )
   UNDRAW DRAW
-4 +LOOP ;

FLY.OUT
21 0 DO                           ( Fly away: )
   0 7 PARAM 5 + XPOS               ( Move to the right )
   0 8 PARAM 3 + YPOS               ( Move down )
   0 9 PARAM 1 - SCALE              ( Decrease in size )
   UNDRAW DRAW
LOOP ;

FLY.SHUTTLE                       ( Call each part one at a time )
FLY.IN ROTATE1
RELEASE DRIFT.AWAY
ROTATE2 FLY.OUT ;
```

Save the program to disk using the file name Fly.shuttle. This is also the name of the last word in the program.

All of the tools needed for the simulation are now on disk. Since the turtle graphics capabilities are used to draw the outline of the earth, the turtle file must be read into memory. Type:

```
READ " TURTLE "
READ " FLY.SHUTTLE "
SETUP
FLY.SHUTTLE
```

The shuttle flies! To run the program again, you only need to type *Fly.shuttle*.

Let's take a closer look at how each of the words works:

Setup simply loads the images into memory and selects them as 3-D objects. Notice that the satellite object is also set to 0 scale. Since the satellite has been selected, this will prevent it from being drawn at the first frame of the animation.

Param is called by later words. It reads a parameter value from the GraForth Image Data Map much like the word *find* (which we discussed earlier) did. However, param removes two numbers from the stack, one to select the object number and the other to select the relative byte for the desired parameter.

Fly.in begins the simulation. It first clears the screen, then uses turtle graphics to draw an arc along the bottom of the screen. This is an outline of the earth. It sets the initial orientation for the shuttle, then uses a do-loop to draw the shuttle repeatedly while changing its scale and position. Each parameter is updated by using param to read its current value from GraForth's Image Data Map, adding an offset, and then resetting the parameter.

Note that undraw is being used to speed up drawing. An interesting tradeoff occurs here: The shuttle is a complicated shape and is being drawn to a fairly large size. Both of these contribute to slower drawing times. Undraw can be used to speed up the animation, but a large block size is needed to cover the object. Erasing a large area also takes time. Even though the block size covers nearly the entire screen (39 19 blksize), undraw is still faster.

Rotate1 performs two rotations on the shuttle, one after another. The first rotation rolls the ship around the Z axis to an upright position. The shuttle was designed with its body along the Z axis so that this rotation could be done regardless of which way the shuttle was facing. The second rotation simply tips the shuttle down for a better view.

Release causes the satellite, which is 1 object, to appear (magically!) in the shuttle's cargo hold. The satellite then moves up twenty pixels while the shuttle moves down, lifting the satellite away from the shuttle. The drawing time is a little slower here because undraw is not used. If it were used, it would have to be called for both 0 object and 1 object, since calling undraw prevents the automatic erasing of only the currently selected object.

Drift.away causes the satellite to drift quickly into the distance and disappear. Note that the shuttle (0 object) is referenced once before the satellite is moved. This puts the same view of the shuttle on both graphics screens, preventing residual motion. Undraw is not used here, because the satellite shape is simple, small, and therefore fast.

Rotate2 rotates the shuttle into position for fly.out.

Fly.out again adjusts scale, xpos, and ypos to fly the shuttle away and out of sight.

Fly.shuttle performs the entire animation by simply calling each of the segments in turn.

This shuttle simulation program was designed to give you a better idea of how more detailed three-dimensional animations are implemented. This should help make the creation of 3-D graphics displays less mysterious.

Apple on the Phone

What Is and What's To Come in TELECOMMUNICATIONS

BY DALE ARCHIBALD

A writer in Minneapolis double checks facts in a United Press International (UPI) file at 1:00 in the morning. A university professor in Miami sends the text of her latest article to a magazine in New York on Saturday.

A boy in Anchorage chats with a girl from New Orleans. A Seattle businessman sends a copy of a bid to his Chicago office in a minute or two.

A San Francisco investor updates her portfolio information and charts without even touching her keyboard. A Nebraska farmer ponders commodity prices to decide whether it's a good time to sell his crop. A hobbyist in Salt Lake City sends the latest version of a program she's writing to a friend five miles away.

If a Person Answers, Hang Up. The advent of the personal computer is bringing the information explosion closer and closer to our homes and businesses. Changes in the way we live and work reflect this influence.

The central instrument in the communication of information is the United States telephone system. In some countries, it may take months

or years to get a telephone installed. In this country, we take for granted what people overseas are awestruck by.

This nationwide web of communications gives us almost instantaneous access to information, regardless of where that information may have been generated or stored. A wide variety of information goods and services can be used by individuals, small businesses, and Fortune 500 firms.

There are databases to draw information from; electronic mail for sending digital memos and letters around the nation in seconds; terminal programs to transmit and receive text files and programs, business data, and mail; and even means to use computers and computer languages not normally available in the home.

There are community bulletin board services (BBSs) through which people can exchange insights and information electronically and exciting new briefcase-sized gadgets that can store messages and send information to any computer that can be reached over the telephone.

In this series of articles, we'll look at the world of telecommunications. We'll examine how modulator/demodulators (modems) change electronic impulses into sound and back again, and we'll consider how data transmission speed, or baud rate, is important. We'll also discuss specific brands of modems and the software that makes them work to their utmost.

We'll look at national networks and at various terminals that connect directly to the telephone and can correspond directly with your Apple. We'll also get acquainted with community bulletin boards you can talk to through your computer. (You may even want to consider establishing and operating one yourself.)

We'll examine word processing programs that have electronic mail capability, see how the many brands of terminal software compare, and discuss on-line databases that allow you to use your Apple to gather information from around the world.

So let's get started.

Hello Central. The telephone uses practically all the same principles now that it did when Alexander Graham Bell invented it. There have been many improvements, certainly, but the technology of a vibrating diaphragm sending electricity through a wire to vibrate another diaphragm remains the same.

Whether the means be copper wire, glass fibers, microwave dish, or satellite, under the seas or over them, the end is still to carry that vibration-generated electricity through space at the speed of light.

The fact that small computers have become so widespread means that they will soon be accommodated in ways only dreamed of today. Hotels, for example, are already investigating feasible methods of providing computers in their guests' rooms. And various companies are working to perfect voice and data transmission from telephones aboard aircraft, at a projected cost of around $7 per minute.

In the wake of recent Supreme Court and government rulings, a number of telecommunications networks now offer telephone communications at a lower cost than the Bell System. Such firms as MCI, Sprint, and Western Union offer some 15 percent to 50 percent lower rates than Bell companies, although usually only on calls to and from major metropolitan areas. The major savings are on evenings and weekends.

MCI, for example, charges $10 per month for service to a residential user. To use the system, you telephone a local number via Touch-Tone. On connection, another distinct tone sounds, and you key in your code number followed by the number you want to reach. If the area you're telephoning is one that's served, your call goes through.

Transmission quality is sometimes too poor for sending data, however; and to make a call through a modem that doesn't offer Touch-Tone capability you need to dial the number separately. If you're located outside normal dialing areas, you may have to pay a premium to telephone the microwave number. So it's a good idea to try to determine ahead of time whether using a telephone communications network will actually save you money once you add on the extra monthly fee.

Modems. Once the computer existed, it wasn't long before someone asked why it wouldn't be possible to send computer data across the millions of miles of telephone lines. Since both telephones and computers communicate through electricity, this was a logical possibility to consider.

If binary data were to be transmitted through a telephone wire, however, the transmission medium would have to be within the frequencies of the human voice. After all, the voice is what the Bell System was originally developed for.

A *modulator* was invented to take the binary computer codes of 0 and 1 and change them into something that could be transmitted over the telephone lines. A *demodulator* was designed to read the modulated code back into binary form upon its arrival at the receiving computer. The resultant device was called a *modem*.

An on-line modem has to be wired into the telephone lines. In many cases, on-line modems dial and answer automatically, as programmed. These devices are the ones that answer with a high-pitched tone if you happen to telephone a friend or business acquaintance whose computer is programmed to answer.

The *acoustic coupler* or *acoustic modem* uses the handset of the telephone. That is, the computer talks into the mouthpiece and listens to the sound from the earpiece. The user must dial the telephone and place the handset into the cradle.

Noise is a problem for these acoustic devices. The carbon granules in telephone mouthpieces tend to "pack" and lose their transmission capability. This can be corrected by shaking or tapping the mouthpiece or by replacing the carbon mike with another type of microphone (such as Novation's Super Mike) that doesn't pack.

Most modems are connected to the computer by means of an RS-232-C serial interface. This often requires that a serial card be connected to the Apple and thus means an additional investment of $100 or more. (Both the Novation Apple-Cat II and the Micromodem II from Hayes Microcomputer Products incorporate serial boards, so there's no extra expense.)

The recently released Dual-Comm Plus from Bit 3 adds two serial I/O interfaces to the Apple on one circuit board. Each one can be set to respond to a different slot number, or both can be used from the same slot location. Baud rates range from 50 to 19,200. You can hook a

modem to one port and connect your printer to the other.

Baud. *Baud rate* is the speed at which data is transmitted. Dividing the baud rate figure by ten gives you a rough idea of how many characters per second are being transferred. Thus, 110 baud sends eleven characters per second (cps), 300 baud is equivalent to thirty characters per second, and 1200 baud corresponds to 120 characters per second. Some expensive machines operate at 4800, 9600, or even higher baud rates.

Some special equipment for the deaf (Telecommunications Devices for the Deaf, or TDD) operates at 45.5 baud, or Baudot. Telecommunications offers deaf people a whole new world of communication. Controllers such as the BSR, for example, blink the lights in a room to alert deaf users whenever there's a call for them on the terminal.

On personal computers, the 0 to 300 baud modem, known as a Bell 103 compatible, is probably the most common. Modems of this type are available with provision for originate only, answer only, and originate and answer. Be wary of cut-rate modems that may only originate or answer, unless you're certain you'll never need the corresponding feature.

If you plan to send a great deal of data or textual material through the telephone line, the extra speed of a 1200 baud device can save you money in the long run. Even though that's four times as fast as the common 300 baud, the price to connect up to various services is often only a little more for 1200 baud than for 300.

Conversely, if you just want to check into your local community bulletin board service occasionally, or you just like to chat on the Source or CompuServe's CB simulators, you have little need for a 1200 baud modem. After all, you can only type so fast.

A Duplex Built for Two. *Duplex*, or *full duplex*, describes a situation in which there can be simultaneous two-way communication between individuals or computers. In *half duplex* only one person can communicate at a time, such as when a ham radio operator must say "over" to signal a correspondent that it's her turn to talk.

Older rotary dial telephones rely on pulse dialing. *Pulse dialing* sends a pulse over the wire to select a number. That's the click you hear when you turn the dial. Newer telephones use tone dialing, such as Touch-Tone, which generates a tone at a certain frequency depending on which button is pressed. Tone dialing is required for using long-distance microwave networks, for example, MCI.

If you plan to have your Apple use such microwave networks automatically, you may want to purchase a modem that will tone dial. The Novation Apple-Cat II and the Bell 212A-compatible Hayes Stack Smartmodem will. Many other modems won't.

A few terminal programs (they're called this because they enable you to use your Apple as a communications terminal) allow you to turn off the carrier tone. (*Data Capture 4.0* for the Hayes Micromodem II is one such program.) You can then use the Touch-Tone buttons for microwave. If you don't have a telephone that will tone dial, you might want to consider buying an inexpensive tone generator that screws into the mouthpiece of a rotary handset, assuming that your local telephone system can accommodate such a device.

More Baud Rates. The Bell 103-compatible modem is any modem that will accept baud rates of between 0 and 300. A Bell 202A-compatible device is a modem that's capable of sending or receiving 1200 baud, but it is only able to use half-duplex and doesn't offer 300 baud. In addition, 202A devices can't take advantage of the vast majority of networks for the average user and thus aren't heavily supported by software.

The 212A-compatible modem allows full-duplex and is compatible with practically all services. This type of device has both 0 and 300 baud and 1200 baud capability. Apparently the 1200 baud segment is the expensive part of the device: the 0 to 300 part hardly adds to the cost at all and helps make the 212A attractive to buyers. The higher 1200 baud rate is really the minimum baud capability you would want for business purposes.

The 1200 baud speed is the clearest high-speed data transmission rate available using regular telephone lines. It's easy to lose information at this speed, though, so if you're operating a terminal program at 1200 baud it's a good idea to be sure it has error-checking capability.

The maximum data transmission speed for most telephone lines is 4800 baud. This is accomplished by means of special equipment that will send four 1200 baud signals at one time. Because of noise on the line and other factors, the transmission quality varies.

Beyond this, a variety of very high speed rates of transmission are possible. For example, the capability to transmit from 1.544 megabits per second (mps) to 60 megabits per second is available now or soon will be for large businesses, governments, and other major users.

Now that we've discussed some of the basics of data transmission and looked in general at various types of modems, let's take a closer look at two of the more popular Apple-compatible modems.

Novation Apple-Cat II. The Apple-Cat II is a multifunction modem that can be upgraded to interface with any 1200 baud modem. The basic unit from Novation consists of a 0 to 300 baud modem, a 1200 baud Bell 202 modem, a built-in serial card, and communications software. Remember, the Bell 202 format is not yet well supported for the microcomputer environment. For an additional sum, you can buy the Bell 212A connector. This is the 1200 baud format that is well supported. The 212A connector may be plugged into a slot or, if all slots are filled, laid atop the power supply.

The basic Cat II unit consists of a telephone line connector, Touch-Tone or pulse dialing, a handset connector, a tape recorder control, and dial tone/busy detection. The circuit board plugs into a controller slot the same way that any board does. The telephone line is plugged into a connector that dangles outside the case of the Apple. The optional handset snaps into a second connector. (This second connector will accept any standard handset jack.)

Boot the communications disk that's included in the package and you're ready to use 50 or 75 baud; 110 or 150 baud for TWX or Telex; 300, 600, and 1200 baud in Bell 202 configuration; nineteen selectable functions from the keyboard with a separate keypress, including auto tone and pulse dial, auto answer, and auto redial; and a terminal program with sending and receiving capability.

The Apple-Cat II is not meant to replace your normal telephone, but it can certainly supplement it. For run-of-the-mill incoming calls or when the Apple is being used for something else, you should have an inexpensive telephone plugged into a Y-jack.

The Apple-Cat II is fully programmable for those who are capable assembly language programmers. It can be programmed to do such things as automatic dialing for the handset operation. If a number is busy, the unit can pass on to the next number in a sequence or redial, depending on how it's programmed.

In addition, the Apple-Cat II has options: an expansion interface for RS-232C, a telephone handset, a BSR AC line modulator, a Touch-Tone LSI receiver, firmware for the deaf user, and a cassette recorder interface cable. If you have the Touch-Tone option, you can program in such things as remote controls through the BSR power modules or operation of an attached cassette recorder. And the Baudot firmware option sets baud at 45.5 for Telecommunications Devices for the Deaf.

Novation operates a free bulletin board twenty-four hours a day at the company's home office.

Hayes Stack Smartmodem 1200. The Bell 212A-compatible Smartmodem 1200 allows 0 to 300 baud or 1200 baud capability for less than seven hundred dollars. It has built-in auto-dial and auto-answer capability. In addition, it can use Touch-Tone dialing, pulse dialing, or a combination of the two. Busy signals and wrong numbers can be brought to your attention, and, if you like, the Smartmodem 1200 can automatically redial the last number called.

An earlier version of the Smartmodem has the same adaptability as the Smartmodem 1200 but offers only 300 baud.

A companion unit to the Smartmodem is the Stack Chronograph, RS-232C-compatible with small computers. It is a quartz-crystal clock/calendar that can tie time to your computer. The Chronograph incorporates a backup battery that will keep it on time for as much as a year's worth of power failure or disconnection.

Hayes released its first communications software last summer, intended for use with the Micromodem II. The Micromodem II was designed specifically for the Apple, and as such it has a built-in serial card. A 110 to 300 baud device that can only be used on pulse circuits, the Micromodem II has a very good reputation for high quality. Because of the Micromodem II's dependability, many of the terminal programs that have automatic features are designed to operate with it.

Hayes and Novation are perhaps the best known modem manufacturers in the Apple world. Another company, Universal Data Systems, offers a 212LP model, powered by the telephone line, 1200 baud only, for about five hundred dollars. They also have a 212A model with both 300 and 1200 baud for about seven hundred dollars. And a 300/1200 baud unit is available from Cermetek for around six hundred dollars.

There are numerous other suppliers, so look around.

Eighty-Column Cards. Communications consultant Dr. David A. Lingwood of Action Research Northwest (distributor of *Zardax*, a word processor that contains its own communications system in eighty columns) says that any terminal intended for serious use needs eighty columns. But the Apple user hotline at one state university strongly advises against using an eighty-column card with communications software on the Apple.

The Micromodem II, for example, apparently has some problems with eighty-column cards. The control codes it uses are, in many cases, the same ones that the cards use. Videx even offers a special EPROM chip of protocols for using the Micromodem II with its eighty-column card.

The Full-View Board from Bit 3 can be operated with Southwestern Data's *ASCII Express: The Professional*. This board opens up the possibility of programming a blank EPROM chip to allow a user-designed character set. This means that a programmer could incorporate symbols such as those used in APL or in a foreign language into communications.

Apple and Telidon. Apple Computer has just released its Telidon Graphics System for the Apple II. This is a *videotex protocol*, or a way to transmit graphics and text through the telephone. Most videotex transmissions send only text that can be shown on a television or monitor.

The Canadian Department of Communication developed the system and it is supported by AT & T, Norpak, Limited, of Canada developed the interface card and software, which Apple now distributes, having recently received Federal Communications Commission approval.

The Telidon PDI protocol (short for pictures, description, instructions) creates screens, then sends them over the telephone to any Telidon-compatible system, which may range from another Apple II to a mainframe computer.

Primary users at first are expected to be databases, multinational corporations, and other large organizations that need to transmit graphics and text. Eventually, however, this capability will reach the general market.

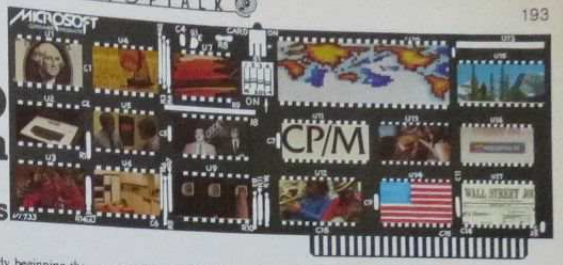
The Telidon software has built-in page editor functions. There is, however, some variation in color resolution when graphics are transmitted between different makes of machines. Color graphics between Apples have a resolution of 128 by 96 pixels; noncolor Apple graphics are 256 by 192.

Pricing on the combined Telidon circuit board, with RS-232C interface (to connect the parallel computer to the serial telephone line), software, and manual, is around six hundred dollars. No special type of modem is needed.

Coming Up. When we reconnect next time, we'll take a look at telecommunications networks, such as Telenet and Tymnet, and at the information services they facilitate. ■

Action Research Northwest, 11442 Marine Drive S.W., Seattle, WA 98146; (206) 241-1645. Apple Computer, 20525 Mariani Avenue, Cupertino, CA 95014; (408) 996-1010. Bit 3 Computer Corporation, 8120 Penn Avenue South, Suite 548, Minneapolis, MN 55431; (612) 881-6995. Cermetek, 4016 Youngfield, Wheat Ridge, CO 80033; (303) 423-8229. CompuServe, 5000 Arlington Centre Boulevard, Box 20212, Columbus, OH 43220; (614) 457-8650. Hayes Microcomputer Products, 5835 Peachtree Corners East, Norcross, GA 30092; (404) 449-8791. Novation, 18664 Oxnard Street, Tarzana, CA 91356; (213) 996-5060. Southwestern Data Systems, Box 582, Santee, CA 92071; (714) 562-3221. Source Telecomputing, 1616 Anderson Road, McLean, VA 22102; (703) 734-7500. Universal Data Systems, 5700 Bradford Drive, Huntsville, AL 35805; (205) 837-8100. Videx, 897 N.W. Grant, Corvallis, OR 97330; (503) 758-0521.

SOFTCARD Symposium



by Greg Tibbetts

It's hard to believe that we're already beginning the new year. This installment of the column marks one and a half years of Symposium, and in that time the SoftCard has grown from a relatively minor component of the Apple market to a substantial segment all its own. The number of CP/M Apple users today is counted in the tens of thousands and is growing even larger, with users in countries all over the world.

Apple CP/M's position would not have been possible without the efforts of many people, but one person in particular, Neil Konzen, had a direct effect on what we examine every month in Symposium. As the author of the SoftCard software, Konzen was a prime contributor to SoftCard's success. His work in establishing the dual processor BIOS is an accomplishment he has the right to be proud of. With that lead-in, we'll begin this month to examine Konzen's creation. By the time we finish, we should have a much better picture of the scope of that effort.

Explaining the BIOS. The CP/M BIOS module is the catalyst for every successful CP/M implementation. Different computer systems require different CP/M BIOS modules. When software designers set out to create such modules, Digital Research supplies them with a list of criteria that contains the necessary hardware functions required by BIOS in its operations and a sample BIOS source listing. It's then up to the individual software designer to produce the smallest possible CP/M BIOS module that meets these criteria and maximizes the special attributes of a particular system's hardware.

This task is not a trivial one by any standards, even for system software types who are employed by the computer manufacturer and have access to all pertinent information on the hardware and related software. When it came to designing a CP/M BIOS module for the Apple, however, the level of difficulty was considerably higher. Even with the help of Apple Computer, no one was available to provide the intense familiarity with the system that comes from having participated in its design. In addition, the amount and diversity of peripheral hardware available for the Apple, all of which had to be taken into account in the design of the BIOS, made the task enormously more complex.

This is not to say that there weren't mistakes, that there weren't things that would be done differently if the project were being initiated today. But, considering that the effort began more than three years ago with the information available at that time, the job that was done was excellent.

In recognition of the fact that there are some areas that could be improved, a new version of SoftCard CP/M is currently being developed. This development effort began in mid-1982. As of this writing, the new version, called 60K CP/M, has not yet been released, although it has been in beta testing for some months. Radically different internally from the 44/56K variety, this new version provides an additional 4K of user TPA space and makes maximum use of the switchable 4K bank of the RAM boards in existence, as well as allowing an additional 12K of space per disk in certain cases. Although our present discussions won't address the new version of SoftCard, we will cover it in this column once it has been released.

System Calls and BIOS Functions. As for the version of the SoftCard CP/M BIOS in use today, it in many ways performs the same functions as its non-Apple counterparts do. In other ways, however, it is radically different. We'll explore these areas of similarity and difference as we discuss the BIOS. Our approach will be to cover what a standard BIOS contains and does, then to contrast that with the SoftCard BIOS.

To do that, we must look first at the Digital Research criteria for what is required in a working BIOS. Since the BIOS is used mainly by BDOS to communicate with the hardware, the primary criteria for a working BIOS are the functions BDOS requires in order to perform that communication.

During its operation, the BDOS needs to perform a set of sixteen such functions. It uses these sixteen low-level functions, called *primitives*, individually and occasionally in combination with each other to perform the more than thirty-seven higher-level functions listed in your manual (page 3-76) as system functions or system calls. (In addition to functions 0 through 36 shown on that page, there are two more: function 37, which resets a single system drive, and function 40, which, like function 34, performs a random disk file-write but fills the record with zeros before writing the data.)

These system calls are what an operating system is really all about: providing a network of functions that applications and utility programmers can use to build even higher-level functions into their products. For example, a single command in a disk utility program may make use of up to ten or more of the CP/M system calls during its operation.

The same principle holds true for languages such as Basic, which are collections of high-level functions called *commands*. The commands dealing with I/O are usually implemented using one or more of the OS system calls. We can see, therefore, that, no matter how complex the functions required by the user, all I/O results from some combination of the sixteen functions required by BDOS.

The functions BDOS requires are implemented by means of a jump table that's placed at the memory addresses immediately following BDOS. This table actually forms the beginning of BIOS and looks similar to table 1.

To function, the jump table must always be located immediately following BDOS. This structure was established so that the requirements for the construction of the BIOS would be as flexible as possible.

During design of the CP/M BIOS, the system programmer can make each of the sixteen routines as large or as small as necessary, locate them wherever seems best, and order them however he wishes. This is so be-

JMP	BOOT	N/A (used once during cold start)
JMP	WBOOT	disk I/O
JMP	CONST	character I/O
JMP	CONIN	character I/O
JMP	CONOUT	character I/O
JMP	LIST	character I/O
JMP	PUNCH	character I/O
JMP	READER	character I/O
JMP	HOME	disk I/O
JMP	SELDSK	disk I/O
JMP	SETTRK	disk I/O
JMP	SETSEC	disk I/O
JMP	SETDMA	disk I/O
JMP	READ	disk I/O
JMP	WRITE	disk I/O
JMP	LISTST	character I/O
JMP	SECTRAN	disk I/O

Table 1

cause, as a result of the arrangement just mentioned, the address of the appropriate routine will be placed in the jump table at assembly time and BDOS will always know where each entry in the jump table is located. BDOS can then simply call the address of the appropriate entry in the jump table and control will be vectored to the correct input or output routine. A simple RET instruction at the end of the routine returns control to BDOS for further processing.

As shown, the sixteen routines (excluding BOOT) are broken down into two groups—character I/O and disk I/O. There are nine disk-related functions and seven character functions. Let's deal with the character functions first.

Character I/O. Character I/O can best be defined as everything that is input to or output from the microprocessor that does not interact with the disk. While this may seem a case of stating the obvious, it is not presented with the intent of being confusing. It's important to realize, though, that in a technical sense the term *character I/O* is itself confusing. For example, many instances of disk I/O involve the transfer of character data. Does that, then, make disk I/O character I/O? Further, since characters are nothing more than eight-bit values (usually with the high-order bit ignored or treated as an attribute indicator), then every possible eight-bit value can be said to represent some character. Does this mean, then, that any eight-bit value transmitted is character I/O? It would seem to be so, since many manufacturers of peripherals require the transmission of certain character sequences to printers and other devices for purposes of control and not display.

The answer, of course, is that everything is character I/O or nothing is character I/O, depending on your point of view. And, in fact, what we are transmitting is simply eight-bit values that have no significance except that which comes from how they are used.

Since the real need is to distinguish between disk and nondisk I/O, the term *character I/O* has come to mean those types of data transfer where single-byte values, usually characters, are being manipulated, printed to the screen, sent to a printer, or perhaps received from the keyboard. Therefore, as long as we don't look too closely at the word and simply accept it, it is very descriptive for the purposes of categorization. It is important to remember, though, that all nondisk I/O is labeled as character I/O, regardless of its purpose. With that in mind, BDOS requires the seven types of character I/O outlined here.

1. Determine console status, that is, whether a character is available from the console device (CONST)
2. Read a character from the console device (CONIN)
3. Write a character to the console device (CONOUT)
4. Write a character to the punch device (PUNCH)
5. Read a character from the reader device (READER)
6. Determine list status, that is, if the list device is ready to receive a character (LISTST)
7. Write a character to the list device (LIST)

Devices and Names. As you can see from the list just given, BDOS is aware of only four different devices—console, punch, reader, and list. The representation of these names in CP/M is CON:, PUN:, RDR:, and LST: Of these, the only device actually used by BDOS without an outside request from a utility or application program is the console, which the BIOS needs in order to get user input from the keyboard and to print messages to the screen. With the exception of the console device, all of these devices are distinctly unidirectional; that is, they can only pass data in one direction. The console device is capable both of giving characters to BDOS and of accepting them for processing (reading and writing respectively).

In retrospect, for purposes of clarity only, it would have been much more universal to have had separate devices for the two types of console I/O (perhaps DISP and KYBD). But the reasoning behind the present choice is that with most of the early CP/M systems the keyboard and display were a single hardware interface to a single physical device, a Teletype, perhaps, or in later times a video terminal. The fact that many of today's CP/M machines would have separate CRT and keyboard interfaces was not foreseen during CP/M's design phase. But, as we will see, little is lost by not having these functions separated.

This device structure is one of the traits that sets CP/M apart from many less flexible operating systems. Comparatively speaking, CP/M's structure makes it simpler to conceptualize than other operating systems. Treating everything in the system as though it is one of the four devices allows for a smaller and more efficient BDOS module.

How the four devices are actually connected to the outside world has absolutely no effect on the operation of BDOS. So long as the seven functions BDOS requires are available and produce the expected numeric values when called, the four devices may literally be communicating with any type of hardware device, or with a strictly software device for that matter. It is the choice of the BIOS designer.

For the most part, of course, the functions BDOS requires are designed to be analogous to certain types of hardware; so, in fact, the console should be a keyboard/display device to be truly useful, but keep in mind that it does not have to be.

Assuming for a moment that the four devices are connected to their suggested real world counterparts, having only four devices may seem rather limiting given the range of available peripheral hardware. This too was anticipated by CP/M's designers. Their response was to implement the concept of the IOBYTE. Given that BDOS will only talk to the four devices, there needed to be some method of assigning and reassigning these devices to additional devices in the system.

The IOBYTE. The IOBYTE is the means of making those device assignments. The IOBYTE itself is not actually used by BDOS specifically, although BDOS tolerates the presence of the IOBYTE at location 03H and contains system calls to determine and alter the IOBYTE value at user-program request. Essentially, the IOBYTE is a single eight-bit byte, broken down into four two-bit fields, one corresponding to each of the four devices. Before doing its I/O function, each routine in the BIOS representing one of these devices examines the corresponding field of the IOBYTE to see where to direct or get the transmitted data.

Since four distinct numbers can be represented in two bits, each of the four devices known to BDOS can assume one of four separate identities. For purposes of distinction, the original four devices were called *logical devices*, while the devices they could potentially assume were called *physical devices*. The names *logical* and *physical* fit well, since the four devices known to BDOS have no real connection to the physical hardware and are, consequently, logical constructs (constructs of the mind) only. On the other hand, the physical devices that can be assigned to those four logical devices actually do represent the driver routines in the BIOS that service real physical devices.

In theory, there could be sixteen such physical devices, four for each of the four logical devices. There are, however, only eleven discrete device names. Table 2 shows the IOBYTE structure. It includes the logical device names, the bits comprising the two-bit field in the IOBYTE for each device, the four possible physical device assignments, and the values for the two-bit field of each device.

Note that the IOBYTE is set up in standard form with the most significant bit (leftmost bit) as number 7 and the least significant bit (rightmost bit) as number 0.

When the IOBYTE is implemented in a particular system, the process goes something like this. BDOS receives, through system call number 3, a request to get one character of input from the logical reader device. BDOS calls the jump vector corresponding to the reader routine in the BIOS (JMP READER in table 1), which immediately transfers control to the BIOS reader subroutine. This subroutine then examines the IOBYTE. Based on the value it finds in its assigned field (bits 2 and 3), it branches to one of four subroutines for handling various reader type devices. The routine branched to obtains a character of input from the device and, via an RET instruction, returns to BDOS with the character in the [A] register. BDOS then passes the character and control back to the user program that called for the input function.

The foregoing description assumes a fully implemented BIOS, with all four of the physical devices corresponding to actual hardware. TTY:, for example, might be the keyboard, RDR: might be a paper tape read-

Logical Device	Bits	Possible Physical Devices
(Bit Values)→		00, 01, 02, 03
1. CON:	0,1	TTY:,CRT:,BAT:,UC1:
2. RDR:	2,3	TTY:,RDR:,UR1:,UR2:
3. PUN:	4,5	TTY:,PUN:,UP1:,UP2:
4. LST:	6,7	TTY:,CRT:,LPT:,UL1:

Table 2. IOBYTE structure.

er, UR1: could be a card reader, and UR2: might be a cassette tape interface. Of course, for a particular system, it may be that none of this hardware is actually hooked up; consequently, the reader routine in the BIOS wouldn't contain any drivers except a vector to the keyboard device for TTY: and a single routine to return $1A, the ASCII end-of-file character, for the other three devices.

In another system, the three reader devices (excluding TTY:) may be a single driver routine, perhaps the cassette interface, in which case the IOBYTE would be examined only to determine if it indicated the TTY: device, and, if it did not, cassette I/O would be performed. Given that each system is different, few systems have identical device mapping.

The flexibility of this overall device structure has benefits for the author of a BIOS module. In the early days of CP/M, many people who had purchased it were writing their own BIOS modules for their own systems. The standard console and list devices were implemented during this time and the question of further interfacing was a relatively simple one. To add a new device, you wrote (or purchased) a new driver and installed it into the BIOS source code. The entire source was then assembled, tested, and placed on the boot tracks.

As CP/M has become more and more a commercial product, received when a particular computer system is purchased, the idea of assembling one's own BIOS and then over time adding drivers and reassembling has lost popularity. As a result, the inherent flexibility of this system has been lost to a degree.

Most of today's users are purchasing their computers to run applications software, not because they want to become system programmers. As a consequence, most CP/Ms sold today are already implemented with functional BIOS modules that contain any drivers necessary to run the manufacturers' peripheral products (whether they are installed or not) at the time of purchase. Generally, such modules are only marginally alterable through manufacturer-supplied utility programs that enable or disable existing driver routines.

Another option, used mainly by peripheral manufacturers, is to request the type of computer in use before purchase and to provide a utility that replaces all or part of the computer manufacturer's BIOS with that of the peripheral manufacturer. The latter BIOS or partial BIOS is, of course, able to communicate with the new peripheral hardware. And, in some cases, even today, the BIOS source code is shipped with the original computer, allowing the user to alter and reassemble it at will.

The job of the BIOS implementer, then, is to create a collection of driver routines for character I/O that correspond to the physical devices contained in this system. There can be up to eleven devices in all. Why only eleven, though, when there are sixteen possible IOBYTE field values? Well, three discrete device names are lost because, as evidenced by table 2, the TTY: device may be assigned to all four logical devices.

Some question has been raised as to why anyone would want to do this, but suffice it to say it was a perceived need on the part of CP/M's designers to have the capability of one universal device assignment. While no specific examples are recommended by Digital Research in their documentation, the system is somewhat more flexible because of the potential to have a single physical device that any logical device can be assigned to, and the flexibility is realized at little cost. In fact, while it may appear that three potential physical devices are lost, this really makes little difference since the TTY: device accessed by the PUNCH routine need not be the same TTY: device accessed by the LIST routine if there is a crushing need to implement more than eleven devices. Again, it is up to individual BIOS designers to make use of the physical device names as they see fit, based on the number of actual interfaces they need to service with routines. From a practical standpoint, such tricks are seldom needed, however. It's unlikely that so many physical devices would need to be interfaced that the capacity for eleven would not be sufficient.

Subtracting the three, though, still leaves us with thirteen possible devices. Where are the other two? The first is the CRT device, which is repeated in the CON: and LST: fields, again because of a perceived need to be able to assign the logical printer output to the same physical device as the logical console output.

While it's easier to see here than in the case of the TTY: assignment, the rationale behind this is somewhat archaic. It goes back to a time when the console display was often a hard copy device such as a Teletype. In such instances, there was little need for a system printer since the console display could do double duty. However, the list device was required to be directed there separately, since its output format is considerably more flexible for purposes of printed material than the output format obtainable through the console device.

The final potential device name is lost as a result of the inclusion of the BAT: device in the possible assignments of CON:. BAT: is not a physical device; rather, it is a means of providing so-called batch processing by reassigning the console input and output functions. When STAT (or some other means) is used to assign CON:=BAT:, any further calls to the CONIN subroutine are automatically vectored to the READER subroutine; likewise, further calls to CONOUT are vectored to the LIST subroutine. It's important to realize that this is done in the BIOS in the CONIN and CONOUT routines themselves and that BDOS is unaware of the change.

Whatever physical devices the logical RDR: and LST: are assigned to will be the actual physical devices providing input and output. This is so because the READER and LIST subroutines will not be able to detect that they have been called by CONIN and CONOUT, rather than by BDOS directly. The effect here is to cause the keyboard to go dead and eliminate all output to the screen in favor of getting all input from whatever physical device is assigned to RDR:. What this might be used for is up to the ingenuity of the individual user; however, if the keyboard and screen are physical device TTY:, then, by assigning RDR:=TTY: and LST:=TTY: and then assigning CON:=BAT:, the effect is one of having done nothing whatsoever. Well, at least no harm is done.

SoftCard Singularities. It's time now to examine the SoftCard character I/O to see what similarities and differences exist with regard to what we have just been looking at. First of all, the IOBYTE structure, which is an option for CP/M systems, was included in SoftCard CP/M. This means that the logical-to-physical device mapping we've just been discussing is applicable. In the case of the Apple, however, some changes and even some compromises were made because of the method of Apple interfacing. This method—using seven peripheral slots for the installation of interface boards and memory mapping the seven slots—posed some difficulties for Konzen to work around. The fact that each slot could be just about anything when it came time for I/O made things a little too flexible for the system to deal with efficiently. For that reason, it was decided early on to make CP/M slot-dependent for certain devices. Slot 1 would, for example, always contain the system printer interface, slot 2 the reader and punch communications interface, slot 3 the alternate console device (if an eighty-column display was desired). Slots 4 through 7 are left to be occupied by disks or user-installed peripherals. In this way, some of the ambiguity was removed from the device structure.

If the BIOS had not been set up to follow this slot-dependent architecture, there would be no way of knowing, for example, which of two serial cards installed in the system was the printer interface and which was the reader/punch interface or even the terminal interface. The only solution would have been to force users to identify their systems, either at purchase or to some type of utility program, and then to create a special version of the BIOS for each user. Such a version would have to be re-created each time some trivial change was made in the system. The slot-dependent structure, on the contrary, allows the BIOS to reorganize the system during every cold boot by making a one-pass identification of slot contents and later rearranging vector addresses based on this information.

Obviously, the system just described had some limitations. For one thing, not all of the many peripherals now available could be included. The reason here, of course, is that, because many of them have special driver subroutines and data handling protocols, it would mean separate routines for most of them. Including even half would mean that the BIOS would be huge and half unused in any given case.

It was not possible, therefore, for the BIOS to be all things to all peripherals. So the various eighty-column cards, the more common serial and parallel communications interfaces, and the common printer cards were included. The boot code was "educated" to recognize these common cards and, based on the slot they were located in, to identify them as the TTY: device or the LPT: device and so on. Small multipurpose subroutines to transmit and receive single characters from each of these devices were written and placed in the BIOS to be called by different device routines like READER and LIST, thereby avoiding duplication where possible.

While this system worked well, it was inherently not very expandable for the user. With that in mind, Konzen established a secondary set of vectors in a known place in the BIOS (the I/O configuration block) that users could manipulate to vector output to their own special-purpose drivers for any additional hardware they wished to add. In most cases, the original device routine jumps to the appropriate vector (based on the IOBYTE field value), which in turn jumps right back (normally to within a few bytes of the place it came from) to the "standard" driver for that type of device.

This may not seem particularly efficient, but it does provide a very good way for the user to patch the various vectors to point to his own routines. And, since the I/O configuration block won't change size or location, the user's patches are unaffected by changes in the construction of the BIOS. There are eleven of these special vectors, one for console status (CONST) and two for each of the other five character I/O routines (that is, CONIN, CONOUT, READER, PUNCH, and LIST). The list status (LISTST) routine does not have a vector because the BIOS for SoftCard always returns a ready status to BDOS.

We now have eleven physical devices and eleven vectors. Unfortunately for clarity's sake, they do not match up. Really, it was obvious they couldn't anyway, since console status is not a device. The map of these devices is shown in table 3.

	Vector	Assigned Physical Device
1.	Console Input Vector #1	TTY:,CRT:
2.	Console Input Vector #2	UC1:
3.	Console Output Vector #1	TTY:,CRT:
4.	Console Output Vector #2	UC1:
5.	Reader Input Vector #1	PTR:
6.	Reader Input Vector #2	UR1:,UR2:
7.	Punch Output Vector #1	PTP:
8.	Punch Output Vector #2	UP1:,UP2:
9.	List Output Vector #1	LPT:
10.	List Output Vector #2	UL1:

Table 3.

From table 3, we can see that, for the purposes of the SoftCard version, only eight possible CP/M physical devices are covered by the ten vectors. In every instance in which two devices are served by the same vector, as is the case with the TTY: and CRT: devices, the UR1: and UR2: devices, and the UP1: and UP2: devices, the two devices are thought of as only a single physical device. Changing the IOBYTE, therefore, from UR1: to UR2: can have no effect. This is the second compromise made with SoftCard CP/M for the purpose of simplifying the I/O structure. It seems a trivial compromise, however, since a total of eight devices to utilize seven slots does not seem too limiting, especially since some of the slots will be taken up by disk controllers and the SoftCard itself.

We have covered a lot of ground in this column, and yet we still have not examined the character I/O routines and vectors in detail. The ground we have covered, however, should give you a better understanding of how these functions work in an overall sense.

As a final recap, let's take one such system call and see what happens to the flow of control. (1) BDOS receives a request for a character from the reader device through system call number 3. (2) BDOS calls the READER routine via the BIOS jump table. (3) The READER routine examines the IOBYTE and finds the field set to the value corresponding to UR1:. (4) The READER routine branches to READER input vector number two. (5) This vector branches back into the BIOS several bytes past the READER start location to a routine called RDRIN1. (6) RDRIN1, which was created to deal with a serial card or communications card in slot 2, loads the slot number into register [A] and jumps to one of several input routines, depending on whether the boot routine discovered a serial card, comm card, or parallel interface in slot 2 during boot. (7) That routine uses the slot value in [A] to get a character from the interface in slot 2. (8) The character gotten, the routine branches back to BDOS with the character now in [A]. (9) BDOS returns to the caller program with the character.

Whew! Here's hoping this followed your conception of what would happen. With that, we'll bring this column to a close. Next time, we'll go into further detail on character I/O and make a start as well on the discussion of disk IO. Until next month. . . ■

THOUSANDS OF CP/M SOFTWARE PACKAGES ARE GATHERING IN SAN FRANCISCO

CP/M '83®

Sponsored by
DIGITAL RESEARCH™
The Creators of CP/M

Moscone Center, San Francisco
Friday-Sunday, January 21-23, 1983
Show hours: **11 AM to 6 PM Daily**
Seminars (for the trade): 9 AM to 11 AM
User Conferences and Workshops: 11 AM to 5 PM

Beginning January 21 your CP/M computer will never be the same.

On January 21, you are going to discover an incredible world of software, services and support for your CP/M computer. January 21 marks the beginning of CP/M '83 . . . a 3-day-long international extravaganza for CP/M users. For the first time ever, you'll be able to learn about, try out and compare literally thousands of products for your system. All of the CP/M support companies . . . large and small . . . will be there, demonstrating the newest, most innovative CP/M products. Leading authorities from the software industry will run dozens of in-depth seminars and workshops to help you explore the full potential of your CP/M computer.

CP/M '83 will actually be the *largest first-year computer show in history*! More than 25,000 CP/M users from around the world will view nearly 700 exhibits and attend more than 100 workshops.

The World Under One Roof

At CP/M '83, you will find everything new for your CP/M computer under one roof. In a couple of days, you can sample software, accessories and services for every conceivable application you may have. You can talk to experts and learn about both the features *and* limitations of these products. You can 'test drive' software and purchase it right at the show . . . at special show prices. You can attend seminars, workshops and panel discussions, presented by the most renowned authorities in the software field, and learn how to make your computer more productive and more powerful. You'd have to spend months visiting computer stores and reading trade journals before you'd find even a fraction of the information you'll get at CP/M '83. CP/M '83 will bring it all together . . . under one roof.

The World's Leading CP/M Experts

CP/M '83's Conference and Seminar program will include noted leaders from the industry including Gary Kildall, President, Digital Research Inc.; Sol Libes, Editor, Micro Systems Magazine; Christopher Morgan, Editor-in-Chief, Byte and Popular Computing Magazines; Adam Osborne, President, Osborne Computer Corporation; Tony Gold, Founder CP/M Users Group and Lifeboat Associates; Ben Rosen, President, Rosen Research; Portia Isaacson, President, Future Computing; Maggie Cannon, Editor in Chief, InfoWorld; David Crockett, Senior Vice-President, Dataquest; and Gordon Eubanks, Vice-President Language Div., Digital Research. CP/M '83's conference, seminar and workshop program is designed for computer tradespeople and users. These individuals, plus dozens of others, will conduct informative discussions exploring CP/M applications, technical information, development aids, venture capital programs and software distribution. End user workshops will show users how to get the most from their CP/M computer.

Low Hotel Rates

CP/M '83's Show Management has made special group discount arrangements with several of the hotels closest to Moscone Center. The unbelievably low discount rates are available at such luxurious facilities as the Hyatt on Union Square, and at modest priced accommodations, including the Pickwick Hotel and Holiday Inns.

The Hyatt on Union Square—415-398-1234—345 Stockton St.—new, modern, and four blocks from Moscone. CP/M '83's discount rates are $75 per night single, $85 per night double, versus their regular rates of $95 to $130 single and $130 to $165 double.

The Sheraton Palace Hotel—415-392-8600—639 Market St.—elegantly refurbished fine older hotel, and the closest hotel to Moscone—only one block. CP/M '83's special rates are $70 single and $80 double, versus their regular rates of approximately $90 single and $100 double.

The San Francisco Hilton & Tower—415-771-1400—333 O'Farrell St.—a modern facility located three blocks from the Show. CP/M '83's special rates are $75 per night single and $95 double, versus this Hilton's regular rates of $66 to $127 single, and $86 to $147 double.

The Holiday Inn on Union Square—415-398-8900—480 Sutter St.—a modern facility with all the conveniences, and located in the heart of San Francisco. This facility offers CP/M '83 attendees accommodations for $70 per night single and $80 per night double.

The Holiday Inn Financial District/China Town—415-433-6600—750 Kearny St.—this high rise Inn is six easy blocks to Moscone and offers CP/M '83's attendees single or double accommodation at $55 per night versus their regular rates of $78 to $92.

The Pickwick Hotel—415-421-7500—85 5th St.—Just two blocks from the Show, this fine hotel has clean modest accommodations, and is offering CP/M '83's attendees rates which are lower than motel rates. The rates are $30 per night single and $35 per night double, and can't be beat anywhere in town.

Make your reservations today. Over 20,000 are expected to attend CP/M '83.

Hotel Reservation Request

How to Make Your Hotel Reservations

1. For you to receive CP/M '83's special discount convention rates, all reservations must be made on this form, or a facsimile. The form must be completed in detail, including date and hour of arrival, date of departure, and names and addresses of all persons who will occupy the room. Reservations can not be processed without this information.
2. Indicate at least three choices of hotels and rates. Requested rates cannot be guaranteed, but the Housing Bureau will make every attempt to assign rooms as near as possible to the requested rate.
3. The Housing Bureau requires written reservations. Only late requests, after Friday, January 7, will be accepted by telephone. After Friday, January 14 try to make your reservations directly with the hotel, but be advised that the hotels are expected to be full by that time. The Housing Bureau telephone number is 415-626-5500.
4. Mail this hotel reservation request directly to CP/M '83 Housing Bureau, P.O. Box 5612, San Francisco, CA 94101, and *not* to National Computer Shows.
5. Confirmations will be sent from the Housing Bureau up to two weeks prior to the event. Allow up to two weeks for processing.
6. Cancellations. Notify the CP/M '83 Housing Bureau of all cancellations up to Friday, January 7. After January 7, make cancellations directly with the hotel. Changes. All other changes, such as arrival or departure times or changes in type of accommodations required, should be made directly with the hotels at all times.
7. Hotels will hold reservations only until 6 pm unless otherwise requested. If you are delayed in transit, phone ahead and advise the hotel of your arrival time. Reservations can be guaranteed to assure a room regardless of arrival time. However, if you do not pick up or cancel the reservation, you will be billed for one night's room rate. If you make a reservation, even a guaranteed reservation, it will be held only for that night. Thus, if you designate a Monday arrival and do not arrive until Tuesday, you will not have a room unless you notify the hotel beforehand.

Show & Conference Preregistration Request

1. Complete this form (or a facsimile) and mail it with a check to CP/M '83 to The National Computer Shows, 822 Boylston Street, Chestnut Hill, MA 02167.
2. All preregistration requests must be received by Friday, January 14, 1983. No telephone or credit card orders can be processed.
3. Use a separate form for each person preregistering for a three day badge.

Name _____

Company (if any) _____

City _____ State _____ Zip _____

Telephone (Area Code) _____

4. Badges and tickets will be mailed back to preregistrants, providing the order is received by Friday, January 7. For orders received after that date, badges and tickets will be held for pick-up at the preregistration desk at the Show.
5. It is recommended that attendees preregister. However, it is not necessary, as badges and tickets can be purchased at the Show.
6. Persons preregistered by January 7 receive an exhibits Conference & Seminar Schedule by return mail.

Check Applicable Box

☐ Enclosed is my payment for _____ (quantity) one day, exhibits-only tickets at $10 each.

☐ Enclosed is my payment of $20 for a three day exhibits and conference ticket/badge. (Use duplicate copy to order more than one)

Hotel Reservation Request

A. Please make the following hotel reservations
Hotel Choice
1st _____
2nd _____
3rd _____

B. Please enter my reservation at the hotel for
_____ single room(s) at $_____/ day
_____ double room(s) at $_____/ day
_____ twin room(s) at $_____/ day
_____ one bedroom suite(s) at $_____/ day
_____ two bedroom suite(s) at $_____/ day

C. Arrival date _____ time _____ ☐ am ☐ pm
Departure date _____ time _____ ☐ am ☐ pm
Occupant _____
Share With _____

D. Mail my confirmation to
Name _____
Firm (if any) _____
Address _____
City _____ State _____ Zip _____
Telephone (Area Code) _____

E. Complete this form and mail to CP/M '83 Housing Bureau, P.O. Box 5612, San Francisco, CA 94101

NEWSPEAK

The fall '82 COMDEX was a resounding success. Next year's show will be even bigger.

VEGAS HOSTS DEALER BASH

COMDEX is the flea market of the microcomputer industry, where dealer and supplier can attempt to do business face to face.

This time around, observers found evidence for any trend they were particularly biased for. In a nutshell, here's what was there:

- The eight-bit world may be dead, but the Japanese apparently haven't seen the corpus delicti. From Seiko to Sanyo, they displayed smaller, more powerful Z80-based machines.
- Apple remains the dominant eight-bit machine. Atari yielded the field without a fight by not showing up. This was not Tandy's kind of show, and Commodore still has fences to mend with dealers. Primary proof: the existence of at least four, and possibly more, Apple emulators.
- The number of IBM pc emulators or MS-DOS machines went from a trickle to a flood in one year. Right now IBM owns the sixteen-bit market just as surely as Apple owns the eight-bit market.
- Jaded software users were disappointed. Other than *TK Solver!* and *1-2-3*, both announced previously, there was little true sixteen-bit software at the show. It's all eight-bit software warmed over. *The Incredible Jack* was the most revolutionary of the eight-bit packages. Has all the good software been written, or are we on the brink of a new wave of software innovation?
- The unexpected source of the first breakthrough in Xerox Star user interfaces was VisiCorp. They introduced *Visi-On*, a user interface shell for all VisiCorp products, to a suitably awed audience. Microsoft and Digital Research were evidently expecting a plunge up the middle and VisiCorp successfully pulled off the end run.
- Even dealers with access to all three major microcomputers—IBM, Osborne, and Apple—were looking for a quick fix to heal sagging margins. There was none to be found.
- There were lots of new products at COMDEX, and the exhibitors would have you believe that the developments were major. In fact, nothing introduced at COMDEX this fall will ever stand next to *VisiCalc* or the Apple II as an influencer on our life and times.

Computerized Center Combines Post Office and Indy Carriers

The country's first private post office has opened computerized branches in Ventura and Century City, California.

World Mail Centers, the brainchild of electrical engineer Scott Adler, premiered last year in Camarillo, California. This pilot center combined the full services of the United States Post Office with Federal Express, United Parcel Service, Trans-Pak, Burlington, Purolator, DHL International Overnight Service, and Emery. The couriers regarded the center as an extension of their own systems with no overhead cost to them and made regular pickups.

The Camarillo center generated a quarter of a million dollars in postal revenues in its first six months of operation, netting Adler a good profit on his $60,000 investment and paving the way for a projected three thousand franchised centers over the next five years.

Software engineer Forrest Kendall, who has worked with Adler on missile systems, designed a computer system for the World Mail Center and wrote programs for accounting and courier rate tables. Computers made possible the use of a debit card, replacing cash, identifying the customer, and keeping an ongoing record of all account activity as well as allowing after-hours transactions.

The center works like this: the employee, using a terminal at the counter, prompts the computer to make a generic selection based on where a customer wants a package to go, how fast, and for how much money. It then prints out a mailing label and a receipt, completing the transaction in thirty seconds or less. Profits are made from a 5 percent markup on stamps and a one dollar handling charge for transactions that require paperwork, like registered and certified mail. The system uses Hewlett-Packard's Image database for the management of records and reports.

World Mail Centers will eventually have electronic mail capability, either as a tree structure or from the center's Century City mainframe. If all goes as planned, the centers will become a free-enterprise alternative to the modernization of existing postal services with taxpayers' money and will generate three billion dollars in revenues for the United States Postal Service over the next five years.

LUCASFILM TAKES PLASTIC OUT OF GRAPHICS

New Technique First Seen in *Star Trek II*

Making metal look like metal with the correctly computed specular component.

In the field of computer graphics, perhaps the ultimate test of technological sophistication is the ability of the human eye to discern that a computer-generated image was, in fact, generated by a computer.

It usually isn't too difficult to tell. A computer-generated image, though simulated to look exactly like the real thing, will invariably have the appearance of a plastic imitation of the real thing. That problem has now been effectively licked by Rob Cook of Lucasfilm's computer graphics laboratory in San Rafael, California. Speaking at the Boston Siggraph '82 convention, Cook revealed that the plastic problem is caused by an error in the calculation of an object's reflective properties; in the computer graphics industry to date, "The model that has been used is a really good model for plastic and a really poor model for anything else."

Diffuse reflection—light from a specific source but reflected off a surface equally in all directions—has always been correctly computed as being the same color as the object, while specular reflection—the component representing concentrated highlights, light reflected directly to the eye—has been calculated as being the color of the light source, usually white. In fact, Cook found, "The specular component is usually the color of the material, not the color of the light source."

Computing white spectral reflection for plastic or waxy objects is okay, since such objects, in the real world, always have a transparent or white molecular surface layer. Computing the same reflective properties for a computer-generated, kandy-kolored, tangerine-flaked, streamlined spaceship is, however, a big mistake and would result in the made-by-Mattel effect.

Cook explains that, as all reflection "essentially occurs at the surface," the correct specular components for metals, painted surfaces, and other "homogeneous" objects should be calculated from reference tables of thermophysical spectral data.

The new technique is currently under development at Lucasfilm and was first employed on a limited basis in the creation of the "Genesis effect" for *Star Trek II*.

Computers Pursue Perfect Pasta

Food scientists at Foremost-McKesson ampany are using a new noodle to perfect the old noodle at their million-dollar pasta research facility in Dublin, California. The new noodle is their time-sharing piece of a mainframe computer, taking the experimental drudgery out of the never-ending search for the perfect pasta.

Pasta may be little more than wheat and water, but there are thousands of ways that scientists can manipulate those ingredients' interaction during the production process. They can alter proportions, for starters, and vary time and temperature setting for mixing, drying, and pressing routines. In the pasta past, researchers had to proceed step by laborious step in their unsung quest—altering a single variable, recording the result, then moving on to yet another experiment. Now they use a terminal's keyboard and let a General Electric Information Services computer back in Ohio do most of the work.

The brains of the system is a sophisticated Basic program known as *RSM*, for *Response Surface Methodology*. Using sample lab test data for its parameters, *RSM* is able to shuffle one or more production-process variables as it runs scores of experiments in the time it would take Mama Celeste to say, "Abondanza." A subprogram with an end-user orientation offers a further enhancement—a model by which to create a people-pleasing pasta. It mixes variables for flavor, color, thickness, viscosity, and that all-important factor, "mouth feel."

"These *RSM* studies won't give us the final answer," cautions Gaylord Palmer, a senior scientist at Foremost-McKesson and *RSM* coauthor. "When we get back into the production facility, we may find different numbers, but the direction, trends, and instructions the program gives us are in the ballpark."

Foremost-McKesson's investment in pasta research is predicated on two market factors—the public's pasta preferences and anticipated industry growth. When it comes to pasta, says company spokesman Bruce A. Lewis, the public knows its own noodle. It covets a pasta that is firm, not mushy. It yearns for golden threads that cheerfully hold their own when convenience-minded consumers cook them to death.

Foremost-McKesson is hoping its Dublin research facility will give it the competitive edge in a billion-dollar industry that is expected to triple in size by 1990. Long seen as the principal factor in the girth of Italian opera singers, pasta is now promoted as a fadcum-health food—no sodium, no salt, no sugar, practically no fat, and, yes, relatively few calories.

"Noodles aren't fattening; it's what you put on them that is," says Lewis.

Wall Street Adopts Computer Technology To Beat the Crunch

On November 3, 1982, the Dow Jones soared 43.41 points, its biggest gain ever, to a new closing high of 1,065.49. Aside from the emotional boost such a rise gives to stock market participants, the figures are misleading. Cold statistics show that this total would nearly have to triple, soaring to 3,000, to keep up with the cost of living over the last ten years. Market analysts call the recent climbs more psychological events than accurate reflections of the market's health.

Economically accurate or not, there has been plenty of action on Wall Street recently. And all that action requires paper, like last October 7 when 147 million shares were traded, causing proportionate accompanying paperwork.

Avalanches of order tickets have been falling from the Big Board lately in dizzying flurries. Wall Street is able to avoid a crunch of paperwork with the help of computers.

According to the *Wall Street Journal*, there are three computer systems now on the floor of the New York Stock Exchange that are "crucial for managing the current volume. They are:

"The Designated Order Turnaround, or DOT, which routes small orders to the specialist posts on the floor, eliminating the need for floor brokers to represent such orders in the trading crowd.

"The Opening Automated Report Service, or OARS, which matches buy and sell orders before the market opens each day and spits out the imbalance; rather than do the matching themselves, specialists handle only the imbalance.

"A batching system for the five hundred most active stocks, which lumps together orders in these active issues throughout the day, thereby cutting down on the number of transactions."

When NYSE president John Phelan warned of coming volumes of one hundred million shares a day, a level most people thought unrealistic as recently as the late seventies, more and more Wall Street firms began investing in computer systems to handle the predicted onslaught—a wise investment, as it turned out. The Dow Jones may not be as reliable as it once was, but the new trading levels have Wall Street firms seriously consulting their local computer dealers.

One such firm is Pershing & Company, who spent $25 million on a new system to handle order processing and clearing for the hundred-odd correspondent firms it serves. Today, its vast processing rooms are lined with banks of computers and terminals.

Another firm, Prudential Bache, has improved its services with a GTE system that routes branch office orders to the various exchanges for execution. Until a few years ago, these orders were relayed to New York and routed by clerks. ADP Brokerage Services, which handles accounting for about ninety securities firms, moved into larger quarters to make room for their new computerized accounting system. E. F. Hutton, the investment broker, is installing a central computer system right now. Apparently, when computers talk, even E. F. Hutton listens.

Meanwhile, Big Board officials contemplate further improvements in floor operations. Phelan is now talking volumes of 250 million shares, and nobody is doubting him. The one tiny snag in all this computerization is the transaction tape that monitors the trading action on the floor. It's running as much as an hour behind.

It's not a mechanical problem; it's just that people can't read any faster than nine hundred characters a minute. Ironically, the lateness of the ticker tape may now be looked at as a market barometer, a good indicator of activity.

Too Expensive?

Impact of E-Beams On Semiconductor Industry Uncertain

In the short and booming history of the semiconductor industry, technology has paced a succession of quantum leaps. Within the year, semiconductor manufacturers will be offered another state-of-the-art advance.

The technology in question is the E-beam—a very expensive machine that uses beams of electrons aimed through negative-like masks to imprint integrated circuits on wafers of silicon. Heretofore unavailable on a wide scale commercially, these impressive machines will enable semiconductor manufacturers to pack more layers of circuitry (and hence more memory) onto a silicon wafer.

The process of printing integrated circuits using optical techniques, the industry standard to date, is impeded by an effect known as defraction, which causes microscopic lines in the chip's power grid to grow fuzzier the more tightly they're packed. E-beams, which print dot by dot like a television image, trace circuits directly onto a wafer and are not as subject to defraction.

There are two reasons why the semiconductor industry is debating whether to embrace this technological marvel—money and speed. The new machines will cost anywhere from one million to three million dollars per unit compared with today's equipment—optical devices known as steppers and projection aligners that run around $500,000 and $25,000, respectively. In addition, E-beams sacrifice speed for greater precision; they're just plain slower than steppers and aligners.

E-beam manufacturers believe the immediate future of their product lies in sophisticated mainframe circuitry for military and other high-performance systems. It's a growing market and, according to some observers, E-beams could figure in half the semiconductor industry's production by 1990.

"The E-beam's impact on personal computers is further down the road," says Ron Felker, vice president of Electron Beam Corporation of San Diego, California, an affiliate of General Sigma Corporation. His company, one of three manufacturers offering the advanced technology commercially, sold its first unit to the navy for a San Diego based computer fabrication facility late in 1982.

"When the industry (as a whole) gets ready to take advantage of the new machine's ability to shrink circuit lines—to pack more and more logic on a chip, more and more cells of memory—then our machines become economical, because you can no longer do it optically."

Other companies banking on commercial acceptance of E-beam technology are GCA Corporation and Varian Associates, which plan to offer their own systems within the year. Their faith in the technology's prospects may hinge in part on the actions of three industry pacesetters—Texas Instruments, IBM, and Hewlett-Packard, each of which has developed in-house E-beam writers.

☐ **Take-Out Computer Boom.** Cupertino based market research firm InfoCorp defines the market for "mobile" computers in a recent report and projects likely market trends toward the end of the decade. The three kinds of mobile computers are *portables* (Epson HX-20 and GRiD Compass), *handheld computers* (HP-75C, Panasonic LINC, and Sharp PC1500), and *transportable computers* (Osborne 1). According to InfoCorp, the entire market will reach 5.1 million units in 1987 from a projected 527,000 units in 1982.

The fastest growing segment of the mobile computer market will be portables, which merge the computing power of desktop machines with the portability of handhelds. The portable market is expected to reach sales of $2.3 billion by 1987. IBM, Apple, and a number of Japanese suppliers are looking seriously at portables. Handheld computers will enjoy a fast growth period, selling 400,000 units in 1982 and climbing to 4.2 million in 1987. Transportable, suitcase-size computers will peak at 260,000 units in 1983 according to InfoCorp, dropping to 150,000 units by 1987.

☐ **Decline of Business Personal Computers Forecasted.** A new report generated by International Resource Development, a Connecticut-based market research company, predicts the virtual demise of the business personal computer, as we now know it, by 1992. According to IRD's 163-page report, by 1987 the personal computer market will begin to be absorbed by the multifunction workstation. In the coming years, office automation tools will converge on a common point of applications capability. For instance, today (if you're lucky) you can get a program that runs on a personal computer that is as good as a dedicated word processor. Presumably, a multifunction workstation will include everything you need in one cost-effective package. This is not to say that personal computers in business will disappear overnight (in 1992 there will still be around 9.9 million installed, according to IRD), but the multifunction workstation will account for 90 percent of the new business in office automation equipment.

☐ **Plugging in the Atlas Workstation.** Dolch Logic Instruments has announced its new integrated digital workstation, called Atlas, that combines logic analysis, in-circuit emulation, word generation, and other measurement techniques in one system. Atlas integrates these various digital test methods yet maintains the ability to adapt to individual test environments through the use of plug-ins. Atlas (for *adaptive test and logic analyzer system*) combines a plug-in instrument front end with a CP/M-driven universal computer. Each plug-in module contains the instrument hardware and dedicated software required to perform a specific simulation or measurement task and all plug-ins use the same computer bus control.

When a plug-in is inserted into the Atlas mainframe, its functional software is automatically loaded into the controller. Special function soft keys and flip keys are defined automatically to support the specified test function. Atlas also includes communications software that allows it to emulate a VT-100 terminal on a Digital PDP-11; the computation and memory capacity of a PDP-11 can be integrated into an Atlas workstation, allowing software compiled on the PDP-11, for example, to be downloaded and debugged locally. Atlas comes standard with two 5¼-inch disk drives used in combination with a Z80A cpu to support the CP/M operating system. The Atlas mainframe is priced around $12,000 with plug-in modules ranging from $2,000 to $10,000, depending on the application.

☐ **Apple-Compatible Portable.** M.P. Computer Services of Sunnyvale, California, has just released the Rover I, a standalone computer that fits into a regular size attache case. The portable computer features dual eight-bit microprocessors, 128K of RAM, a full/travel keyboard with ten function keys, a tape drive capable of storing up to 125K of data, a serial communications interface, parallel printer interface, and a real-time clock. The Rover Display Unit provides high-resolution graphics on a 5½-inch diagonal screen. Multifunction expansion units allow the use of a modem, two disk drives, or a five megabyte Winchester hard disk. Software supplied with the Rover I includes word processing (*Rextex*), spreadsheet forecasting (*AbaCalc*), and an advanced calculator (*Abacus*). With a disk drive and the Rover Operating System (Rosy), the Rover is Apple software compatible, according to M.P. Computer Systems. ■

NEWSPEAK STAFF

Editor David Hunter

Contributors Al Tommervik, Andrew Christie, Michael Ferris, Jonathan Miller

HARDTALK

BY JEFFREY MAZUR

Less than ten years ago, the most common device for interactive communication with a computer was a printing terminal (such as the Teletype Model ASR-33 or one of the Selectric terminals) rather than the now ubiquitous CRT. At first glance, one of these machines typing along at ten to fifteen characters per second (cps) was a rather impressive sight. Waiting several seconds for a simple response or minutes for a small program listing, however, left much to be desired. At the time, a CRT or video terminal was considered a luxury; its writing speed of a thousand or more characters per second made it an excellent means of communication. But once you'd finished working with the computer you had no written record of what had transpired.

Today the situation is totally reversed. Advances in microelectronics have made video display much more economical, and almost all personal computers use a video terminal as the primary output device. This leaves many computerists without any means of generating hard copy. For a while, in fact, many people considered a printer a luxury item. But in the last two years we've seen a number of well-built, low-cost printers arrive, mostly from Japan. These printers have enjoyed great popularity.

While some printers cost less than a disk drive, a good letter-quality printer can still set you back as much as or more than the computer itself. Therefore, it's important to know the pros and cons of the different kinds of printers. Then you can choose a machine that meets your needs.

Printer Basics. All printers can be classified as *impact* or *nonimpact*, with either fully formed characters or matrix character formation. Impact printers operate much the way an ordinary typewriter does; that is, they rely upon a hard object striking against a ribbon and thereby transferring ink to paper. Nonimpact printers use various other technologies to create readable copy.

You will also have to choose between fully formed and dot-matrix characters. Fully formed characters are printed by means of a single stroke of an appropriately shaped die. Matrix printing, on the other hand, divides each character field into a finite number of points (dots) in the same manner as most CRT video displays do.

The tradeoffs involved in choosing impact or nonimpact printing and fully formed or dot-matrix characters can help you narrow down your selection of printers quite rapidly. For example, nonimpact printers offer quiet operation, good printing speed, and low cost. However, they usually require special paper and can't print on multiple-copy forms or make carbon copies. Thus, if you have mailing labels or special forms to print, you need an impact printer.

For the highest quality printing, fully formed characters are desirable; but a matrix printer offers higher speed and lower cost as well as allowing for special characters and graphics to be printed along with ordinary text. Categorizing printers by whether they are impact or nonimpact and by their character formation method gives us four basic printer types. We'll discuss each of them in turn.

Type I: Fully Formed Impact

Printers in this category range from modified typewriters to high-speed drum and band printers. If the printer offers alternate typefaces, the font is changed by replacing the print element (often a wheel or sphere). Paper motion can be either tractor feed or friction feed. Some of the very slowest and very fastest printers fall into this group.

Magic Fingers. One of the least expensive approaches to making a printer is to add a bank of solenoids to an ordinary, preferably electric, typewriter. The solenoids are positioned over each key and connected to the computer via some interface. Thus the computer can activate these the electric "fingers" in a manner similar to the way someone types at the keyboard. Because of mechanical limitations, printing is very slow (less than ten characters per second) and, of course, fairly noisy. Print quality can be very good, and the ability to use almost any type of paper is a real plus. If your primary need is to be able to print letters on your own stationery, then a printer of this sort might be an economical solution (assuming you already own a typewriter). One word of caution, however—most typewriters aren't designed to withstand heavy use as a printer, so take it easy!

Selectric-Based Printers. Probably the highest print quality can be obtained using printers that are built around the IBM Selectric mechanism. With their familiar "golf ball" shaped elements, these machines can reach a top speed of about fifteen characters per second. Many different fonts are available, and changing elements takes only a few seconds.

There are basically two kinds of Selectric printers—those that are converted typewriters and those that were originally designed for use as terminals. The latter usually have a heavy-duty mechanism that's more suitable for use as a printer. Some of the older terminals require a different electrical code than microcomputers use. Thus making them work with the standard ASCII code may call for additional circuitry. Because of this, most Selectrics can be used for printing only (that is, the keyboard cannot be used for input into the Apple). Many suppliers sell used or refurbished units at a reasonable price; some of these units, but not all, include the ASCII conversion. Aside from making letter-quality printouts, such a unit will double as an excellent typewriter.

Electronic Typewriters. Along these same lines, several electronic daisy wheel typewriter/printer combinations have recently become available. Although slightly inferior to a Selectric in terms of print quality, the new electronic typewriters do offer many additional features that you might find useful. Among these are multiple pitch (including proportional spacing), automatic underlining, centering, justification, and reverse print. These features also work when the unit is used as a manual typewriter. Speeds of up to thirty characters per second are typical.

Cylinder/Drum Printers. The heart of these printers is a character-studded, rotating cylinder or drum (see figure 1). A series of hammers strikes the paper against an inked ribbon and the drum, thereby transferring an image of the character onto the paper. Each hammer is fired under electromechanical control such that it strikes the paper at the precise time when the desired character on the drum is opposite the hammer.

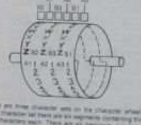


Figure 1. Cylinder-based printing is used in many terminals. The perimeter of the drum is surrounded by the complete repertoire of characters. Across the length of the drum, the characters are repeated.

Drum printers have a complete character set for each column to be printed. Thus an entire line is printed at once without any horizontal carriage movement (see figure 2). This category includes a wide variety of printers. The price can range from less than $100 to more than $10,000. Some print at ten characters per second, others are as fast as two thousand lines a minute. Most of the "standard paper" (that is, nonthermal) printing calculators use this type of mechanism.

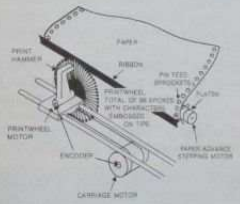


Figure 3. A typical daisy wheel printer. Many printers use a microprocessor to control the printwheel and the print hammer's impact force.

Daisy Wheel Printers. Probably the most popular type of printer in the fully formed impact category, the daisy wheel printer (see figure 3) gets its name from the petal-like construction of its printhead. Imagine taking all of the keys from a standard typewriter and forming them into a circle similar to the arrangement of the spokes on a wheel. An electronic servomechanism controls the position of the character wheel to bring the desired characters into place. When the correct "petal" has been spun into position, a hammer forces the petal into contact with the ribbon and paper.

Their relatively low cost (less than $800 dollars now), moderately high speed (thirty-five to fifty-five cps), and interchangeable fonts have made daisy wheel printers very popular. There's quite a lot of competition among manufacturers who create this type of printer. Each newer model represents someone's effort to offer increased features, reduced noise, and/or lower price.

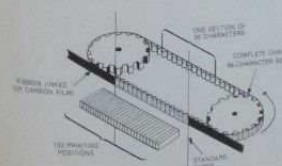


Figure 4. Band or chain printers are among the fastest units available.

Band Printers. At the high end of the impact printer category are the band printers (see figure 4). Used with many mainframe computer systems, they offer extremely high throughput. In this type of printer, the character elements are mounted on a continuous band or chain. The band constantly rotates in front of a bank of hammers, one hammer for each column. Again, precise timing is incorporated so that each hammer strikes when the appropriate character is in place. Because of the carefully designed sequence of characters along the chain, several hammers often fire simultaneously.

Type II: Matrix Impact

This is by far the fastest growing sector of the printer market. All the printers in this category use a technique similar to that shown in figure 5. Because each character is formed by several strokes, more sophisticated electronics is needed to control the printhead. The head itself consists of several tiny hammers or "wires," each of which is activated separately to print a small dot on the paper. Customarily, seven to nine wires are mounted vertically on the head, which moves horizontally across the page. As the head steps from left to right, the electronics determines which wires to "fire." After several such steps, a complete character has been printed.

How many print wires and horizontal steps per character there are determines the size of the dot matrix used (it's typically seven by nine). Some printers allow multiple strikes or multiple passes of each character as a means of producing bold, enhanced, or very high-resolution characters. An advantage of this approach is that the same printer can be used for high-speed draft-quality printing as well as for lower-speed near letter-quality output.

Character fonts are software-controlled and are usually stored in ROM. Some dot-matrix printers don't have enough print wires to include descenders on the lower-case characters. Instead, they push the character up onto the baseline, making text a little harder to read. Most newer machines have at least one or two dot descenders. Because the Apple's hi-res screens are already in dot-matrix form, graphics are easily printed. Friction and tractor feed are available, with many printers offering both as standard features.

Dot-matrix printers are known for their relatively high speed (80 to 350 cps) and low cost ($300 to $2,000). Compared to most other impact printers, matrix types usually operate more quietly. Most of the popular printers likely to be found as part of an Apple system fall into this category, including those from Epson, Okidata, NEC, C. Itoh, and Centronics.

Type III: Matrix Nonimpact

The increased need for low cost, quiet, and fast printers has spawned the growth of several new printer technologies. Fundamentally, these nonimpact techniques prohibit multiple copying and may require special paper. Because of this, friction feed is almost always used and the lack of fan-fold paper makes long printouts difficult to handle. Ink-jet printers can use tractor feed and normal paper but still can't make multiple copies.

Thermal Printers. Thermal printing requires the use of a specially treated paper. This paper is heat sensitive; it will change color when

Figures 1, 2, 3, 4, 5, and 8 are based on illustrations that appeared in *EDN Magazine*. Courtesy *EDN Magazine*.

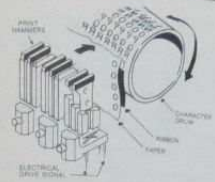


Figure 2. Drum impact line printers work by comparing the contents of a buffer containing the data to be printed with the positions of the characters that are needed from each column. When the time is right, the necessary hammers are activated and the line is printed.

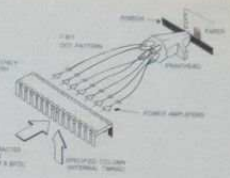


Figure 5. A typical serial dot-matrix mechanism. In this one from Centronics, the character codes that command the printhead's seven print wires are contained in a ROM. The ROM's output is fed through power amplifiers and then activates solenoids that cause the print wires to impact printer and paper.

heated to a temperature of about 200 degrees.

A thermal printhead is made up of tiny heating elements arranged in a dot-matrix pattern (see figure 6). Speed limitations are imposed by the fact that the elements must cool down sufficiently before the printhead can move on to the next position. Some of the lower-cost thermal printers print "on the fly," that is, while the head moves at a constant horizontal rate. Although this causes the dots to smear a little, it can actually improve the print quality.

Thermal printers boast high reliability, low cost, and almost silent operation. Because of paper restrictions, the maximum width for these printers is usually eighty columns. Another problem with thermal paper is that the printouts can fade with age. Worse yet, if left in a hot car or placed near any hot object, the paper can turn completely dark. A good example of a compact thermal printer for the Apple is the Silentype printer.

Electrosensitive Printers. Like thermal printers, electrosensitive printers require special paper and create characters via a dot matrix. In this case, a dark paper is coated with a thin conductive layer, giving it a shiny, metallic appearance. This paper is passed in front of the printhead, which consists of several small electrodes that glide along the surface (see figure 7). These electrodes are connected to a voltage source (about fifty volts) through electronic switches. A pressure roller contacts the entire width of the paper below the printhead to complete the circuit. When an electrode is switched on, a small spark jumps between the electrode and the paper. This spark burns away the metalized coating at that point, leaving the dark paper showing through.

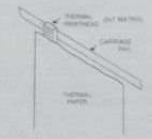


Figure 6. The printhead of a thermal printer may contain as many as thirty-five resistive heating elements.

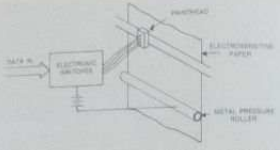


Figure 7. An electrosensitive printhead works by creating a spark between the head and the paper. A metal roller completes the circuit by contacting the paper surface across its full width.

The main advantage that electrosensitive printers offer is their speed; their biggest disadvantage is the silver paper. Words printed on this kind of paper can be difficult to read and the paper is sometimes hard to get. Fortunately, excellent photocopies can be made from this paper, so that's one way to improve the final printed result.

Type IV: Nonimpact Fully Formed

Ink-Jet Printers. The latest advances in nonimpact printing have been the ink-jet type devices (see figure 8). Several different schemes, which operate something like the one shown in figure 8, have been developed. Basically, ink-jet printers try to imitate the operation of a CRT screen. A stream of ink is broken up into many tiny droplets, and the droplets are given an electrical charge. These drops are then deflected electrostatically so that they land on the paper in the shapes of the desired characters.

Ink-jet printing offers many advantages, including very high speed, excellent print quality, and silent operation (except for the sound caused by the motion of the paper). Although they're still relatively expensive, ink-jet printers could begin to dominate the market in the near future.

Xerographic Printers. While not exactly a personal computer item, xerographic printers (costing from $20,000 to $200,000) represent the top of the line. These machines usually print an entire page at one time; many types can print as fast as two pages per second! As you might guess, these devices use an electrostatic/toner process similar to that used in ordinary photocopy machines. Other designs even incorporate lasers to achieve even finer print quality.

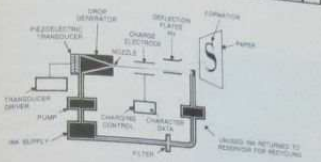


Figure 8. One of many ink-jet printer designs. In IBM's Model 6640 printer, a quiet ink jet generates hard copy. Average throughput measures 54 ipm for continuous printing and a six-inch writing line.

Conclusion. Paper feed mechanisms, carriage size, and special functions are some other aspects worth looking at when you're choosing a printer. For many applications, friction feed is sufficient (you can use continuous feed paper in a friction feed printer). Of course, pin feed is nice for helping to keep the paper straight. Full tractors are needed only when precise alignment of the paper is required or special forms are used. Above all, it's important to make sure that the printer will handle the maximum size of paper that you wish to use.

If your work requires precise horizontal or vertical tabbing, check out each printer's capabilities in this area. Other features—such as multiple fonts, underlining, variable line pitch, and graphics—may or may not be important to you.

Be careful when comparing overall printer speeds. Don't confuse a printer's maximum or instantaneous printing speed with its sustained throughput capabilities. Unfortunately there's no standard definition for determining throughput, so you must interpret the manufacturer's specifications carefully.

Furthermore, serial printers may communicate at a baud rate that is much higher than the speed at which they can print. If there's any type of input buffer, then the printer can accept data in a rapid burst and then print it out at its own pace. Depending on the size of this buffer and the speed at which data is being sent, this can sometimes reduce the overall time needed to perform a given task. With long, continuous printouts, however, the buffer may reach its full capacity and no longer offer any speed improvement.

Bidirectional printing is another proven technique for increasing throughput; with logic seeking, bidirectional printing can make even more of a difference.

Logic seeking allows the printer to compare the printhead position at the end of each line with the length of the following line. When the lines being printed are of equal length, the printhead should alternate printing directions. However, when one line is substantially longer than the previous one, it may be quicker if the printer returns the carriage and prints the second line in the same direction. This decision can be made electronically by the logic-seeking circuitry.

Once you've narrowed the choice down to a few printers, it's time to visit your local computer store for a firsthand look. Examine sample printouts to compare print quality. It's also important to take mechanical construction into consideration, especially if the printer will see heavy use. The specified mean time between failures (MTBF) and mean time to repair (MTTR) figures for some printers are only estimates. When you're considering a printer, don't be afraid to remove the cover and look inside. You can usually tell where short cuts have been taken to reduce a printer's price.

Finally, it's imperative that you have the proper interface between your computer and your printer. This will be either a serial (RS-232) or a parallel (Centronics) interface with proper handshaking. See the March 1982 and November 1982 installments of this column for information on these interface boards. ■

GLOSSARY

Baud rate—The speed of serial data transmission in bits per second. The most common format uses eight data bits plus one start and one stop bit, for a total of ten bits per character. Thus, dividing the baud rate by ten usually yields the transfer rate in characters per second.

Bidirectional—The capability to print characters from right to left as well as vice versa. This increases a printer's throughput.

Buffer—The capability to store incoming characters so that, to a limited extent, the printer can accept data faster than it prints it.

Carriage return—The act of returning the printhead or printwheel to the starting position for the next line.

Centronics interface—See *parallel interface*.

Continuous feed—See *fan fold*.

CPI—Characters per inch. This will determine the maximum number of characters across each line on the page. The standard pica font (similar to that found on most typewriters) allows ten cpi.

CPS—Characters per second. This specifies the maximum speed (usually under ideal conditions) at which characters can be printed. See *throughput*.

Current loop—An old-fashioned standard for serial communication between peripherals.

Descenders—The small "tails" of certain lower-case letters (j, p, q) that extend below the baseline.

Dot-matrix—A type of printing in which characters are formed by using a number of strategically placed dots.

Fan fold—A type of paper or multipart form that can continuously feed into the printer (usually via tractor feed). The name comes from the fanlike appearance of the paper when it is folded along its perforations.

Font—A particular style of printing defining the size and shape of each character. For example, pica, gothic, script, and so on.

Friction feed—A simple mechanism for moving paper through the printer by pressing it between two rotatable rubber rollers. Similar to the mechanism used by most typewriters, this arrangement is required when using paper or forms without tractor holes.

Fully formed characters—Characters that are printed all at once with complete, unbroken lines.

Handshaking—Signals used to coordinate the transfer of data from one device to another.

Hard copy—Computerese for any form of printed output.

Instantaneous print speed—The maximum rate at which the printhead can actually print characters. This does not include the time needed to perform carriage returns, line feeds, and so forth. See *throughput*.

Justification—Alignment of text along the left and/or right edge so that each line begins (or ends) in the same column.

Letter-quality—See *fully formed characters*.

Line feed—The action of moving the paper to the next line.

Logic seeking—The ability of bidirectional printers to determine the fastest way to print the next line.

LPI—Lines per inch. The vertical spacing of printed lines on a page.

LPM—Lines per minute. This is another speed rating used mainly to quantify line printers.

MTBF—Mean time between failures. A good indication of how well a machine is built.

MTTR—Mean time to repair. An indication of how easily a machine can be fixed (not too important to the average consumer).

NLQ—Near letter quality. A dense printing using multiple passes with a dot-matrix head. Can approach the quality of fully formed characters.

Parallel interface—A connection to the computer whereby character information is transferred along seven or eight data lines. The most common standard for this type of connection has been set forth by the Centronics company and bears its name.

Pin feed—A simpler form of tractor feed. Instead of full tractors, only one pin wheel is used on each side. Sometimes the pin wheels are mounted at the ends of the platen, giving them a fixed spacing.

Pitch—The character spacing in characters per inch.

Platen—The rubber drum behind the paper. Usually acts as a cushion for the striking elements of an impact printer.

Proportional spacing—A font that assigns variable width, or pitch, to each character according to its actual size.

Ready/busy—A hardware handshaking signal used for serial data communication. Data flow is controlled by the use of a dedicated wire to carry the control signal.

RS-232—A standard for serial data communication specifying signal voltages and connector pin assignments.

Serial interface—A connection between two devices where data flows over a single pair of wires, one bit at a time. See *RS-232* and *current loop*.

Slew rate—The maximum speed at which the printhead can be moved (horizontal) or the paper can be advanced (vertical). This figure determines the time it takes a printer to perform a carriage return and is also important when using horizontal and/or vertical tabs.

Tab—The act of rapidly moving the printhead to a specific position. Vertical tabbing involves moving the paper to a desired vertical position.

Throughput—A more accurate measure of a printer's overall speed. By printing actual text and taking into account carriage returns, line feeds, and so on, this figure gives a more accurate representation of a printer's performance than instantaneous speed.

Tractor feed—A mechanism for moving paper through the printer in a very precise manner without slippage. This is especially desirable when using preprinted forms where the printer's output must be aligned with the paper. Requires special paper or forms with guide holes on each side.

X-on/X-off—One type of handshaking used for serial communication. This system uses bidirectional communication to control the flow of data. ■

Word Processing:

The Power of Consultation

BY JONATHAN MILLER

Alan Tompkins had been thinking the unthinkable for a number of years, imagining a scene as if it were in a movie. He'd stride purposefully into the boss's office, smile good-naturedly, then drop his verbal bomb: I'm leaving IBM. The imaginary camera would pan to show the boss wearing a stunned expression and making strange wordless noises. Then a reaction shot. Tompkins basking in the exhilaration of his bold decision, yet wondering, even at the eleventh hour, if maybe he hadn't dived off the shallow end of the pool.

"The notion of leaving IBM," he says, "was a very scary one."

Thinking the Unthinkable. There were the kids, of course. His three oldest would soon be entering college and the bill, joke as Tompkins might, loomed as large as the defense budget. Gone would be the regular paychecks, the security blanket of fringes, not to mention an eleven-year association with a great organization that had always done well by him.

He wouldn't be quitting just a job, he'd be leaving home and mother church. This hardly sounded like the considered decision of an IBM systems engineer, yet here he was, at age forty-two, deciding to do just that. Risking everything, or so it seemed, in order to strike out as an independent computer consultant in, of all places, rural Waitsfield (population 2,000), Vermont.

It was a scary proposition, all right, but that was the point. Alan Tompkins was loving every born-again minute of it.

"The certain knowledge that you're to hang the day after tomorrow brings a marvelous clarity to the mind," jokes Tompkins, paraphrasing Samuel Johnson. "I find that for me at least that's very true. If I wake up on a Tuesday morning and I know that at the end of this week there are some bills I have to pay and I don't have all the money in hand, it really focuses my attention."

Tompkins is sounding very focused this Tuesday morning. Snug in his study on a cool autumn day, he reflects on the bold step he took only two years ago. He's sitting in an old college chair under the beneficent gaze of a geodesic dome made from wooden dowels.

"To keep in touch with Bucky Fuller," Tompkins explains. The sculpture is gently turning in the updraft from an oil and wood burning Danish furnace that's been left half-enclosed so its cheery orange paint can brighten the large room.

It's a rough cut office, a mix of sheet rock and bare plywood walls and filled with the eclectic tools of a computer squire. An Apple II Plus (he was attracted by its open system and a cornucopia of hardware and software), two disk drives, and NEC Spinwriter printer wait obediently on an oversized, makeshift desk. A library, primarily of computer literature, lines an entire wall.

And scattered about, in various crannies, are reminders of rural Vermont—a battery charger, a five-gallon can of weed killer, sacks of rye and corn seed, and a spooky door that blows open at the whim of the wind.

Tompkins has been spending an interesting day, dividing his time between his organic enthusiasms, which he loosely refers to as subsistence farming and data processing. Farming can mean a little gardening in the meadow, but this morning it embraced a little recreational wood cutting. Where others play tennis or go sailing on nearby Lake Champlain, Alan Tompkins chops, stacks, and delivers cords of wood. Not for the money, but for the focus of strenuous exercise.

Call of the Wood. As with programming, what fascinates is the process and its human dimensions. There's the multifaceted business of managing a forest resource in a state where 90 percent of the homes rely in part on wood for heating, and there's his boundless admiration for those hardy souls who earn a living at the risky maneuver of felling mighty trees. A hundred years ago when Vermont was farm country, 75 percent of the land was cleared and only 25 percent forested. Today, those percentages are reversed, a neat reconfiguration of the system engineered by economics and the state's commitment to conservation.

"And the esthetics of a pile of wood stacked up are quite pleasing," he adds, "if you're into it."

At the moment, Tompkins is intrigued by another esthetic. He's at his Apple using Logo graphics to draw pictures of gravestone monuments for a local granite company, one of a diverse collection of several clients. What he's trying to do, he explains, is write procedures for an IBM minicomputer to ensure that order forms sent to the cutting shop bear a dot-matrix representation of the gravestone design ordered by a customer. The first step in the implementation came last year when he and another programmer defined the turtle graphics primitive.

"So now," he enthuses, "I'm sitting here with Logo, saying, 'What do I have to say in turtle graphics to draw a gravestone with a nice curvy top?' "

If Tompkins sounds delighted, it's because he's gained greater control over his life and his livelihood. He could have plodded on at Big Blue in his ten-pound wing tips, but he finally realized the shoes didn't fit.

"I found the things I was interested in—microcomputers—were something I got no chance to work on at IBM," he explains. "What IBM really needed were people willing to understand the guts of a very few packages expertly. I found that emotionally I was a lot happier if I was able to amass new materials, sometimes under pressure. I find a lot more satisfaction in doing the whole job for someone. When it's done, I can look them in the eye and say, 'That was really a good thing we did. It really helped your business.' "

Power to the Text! His own business is ideas, as he says—programs, system management advice, and in-service computer training for local schoolteachers. These ideas usually take written form, and written form (correspondence, reports, class notes, program documentation, and proposals) means word processing in the age of the moving cursor. For Tompkins it has also meant *PowerText*, the word processing system from Beaman Porter that is form-oriented and Pascal-based. To hear Tompkins tell it, *PowerText* is the independent businessman's supersecretary—it not only takes dictation, it's smart enough to get all the formats straight.

Like many word processing pilgrims, Tompkins arrived at his ultimate choice by accident. He can still recall the misadventure, the day the magic deserted his first word processor, *Magic Wand*.

"The crisis point came when I was formatting a twenty-five page report for a client and suddenly the system locked up and the report wasn't there anymore—either in memory or on disk. I remember walking around and around this office making funny noises."

It wasn't the first time Tompkins had power problems, just the last. Quickly recalling an ad he'd seen for *PowerText*, Tompkins got on the phone to Beaman Porter's David Guest and within two days was back processing the King's English.

The superb support made a distinct impression, but it was the system itself that sealed the sale. *PowerText* wasn't perfect, mind you, just what the customer ordered.

"You can find things *PowerText* doesn't have," Tompkins allows. "It doesn't have footnotes at the bottom of the page. It doesn't do sorted indices. But, because all the format information is stored on a format file, I find I can crank out the several kinds of documents I do at a tremendous speed."

Banks and Bibliographies. Tompkins is able to do that because of *PowerText*'s format-oriented approach to word processing. Since it's the contents of letters that change, not their basic formats, why not spare users the trouble of having to style every letter individually. Why not create a bank of commonly used forms so they can concentrate on the business at hand—what is being said rather than how it's going to look on the page. Leave that thankless chore to a very efficient word processing secretary that takes dictation, automatically eliminates awkward page breaks and widows, tracks footnote or bibliography numbers, and knows exactly how to generate a polished piece of correspondence, because the form has been defined by the user.

PowerText comes with five standard format styles—business letter, personal letter, memo, document, and a landscape file (allowing the creation of up to twelve printed columns on a paper when turned on its long axis). It also includes a *PowerStyle* file generator that enables users to design their own formats.

What is striking about the program is its kinship with dictation, a feature designed to make it particularly attractive to businesspeople who are accustomed to dictating letters or who write as they speak. Text appearing on the CRT resembles shorthand; it's usually entered as a full screen of unindented copy consisting of the actual letter content run together with embedded keystroke commands (like "/PAR" for paragraph)—again, as if someone were dictating a letter. What you see on the screen, therefore, is not what you get. What you get is professional-looking correspondence, an impression that is furthered by the thoughtful inclusion of the "typist's" initials at the bottom of your printed page.

The program gives users control over a range of secretarial functions, such as underlining, justification (right and left), subscripting and superscripting, space blocks for illustrations, and pitch modification. In addition, it supports two unusual features—table of contents generation (from headings and subheads in the document format) and variable column creation. The latter function is a major timesaver, according to Tompkins. "Often in a letter or document, I'll have to insert something like 'Here's the equipment you ought to acquire,' say, and I don't have to count spaces. Just use the column command and it all happens."

PowerText's principal plus in Tompkins's view is what might be termed a well-intentioned accident of birth—its language, Pascal. Pascal is portable, says Tompkins, which is why he promotes it among his clients. If they elect to get a new generation of micro two years hence, their Pascal-based packages stand a better chance of interfacing with what's around then. But it's marvelously compatible even now, he adds.

"The fact that *PowerText* is in Pascal allows me to take output from a simple database like *VersaForm*, which I like very much, and feed it directly into *PowerText*."

Tompkins does have a few reservations about the program, but they range from the piddling to the middling to more elusive questions of style. Of the trifling order, we have his original program that arrived with a binder too small for its documentation. Then there's the actual manual, a more serious shortfall. It reads all right, Tompkins concedes, but it seems to lack organization, the deft hand of a professional manual writer. And, last, there's its image—if you like, its commercial sex appeal.

"I think a difficulty that *PowerText* has is that the documentation and packaging isn't slick enough for today's microcomputer market," Tompkins says. But that, he allows, is more a matter of long-term market positioning than it is of program performance.

"While I recognize that there are slicker, more widely used packages like *WordStar* that run in the CP/M environment," he adds, "*PowerText* seems to me to meet the needs for the large bulk of office typing that people do."

Two Roads Diverged. Meeting people's needs, of course, is the bottom line in programming, but it's going to take more than slick packaging to go on fulfilling them, says Tomp-

kins. The computer industry is at a critical juncture, he argues. It's in a classic economic bind. It has to keep profits up at a time when unit costs are coming down, and the only way to do that is by building an appliance that's as simple to operate as a toaster.

"Unless that problem's solved, the market will dry up," he says. "There are only so many people who are fascinated by computers, use them professionally, or are wealthy enough to buy them as status symbols. These groups don't constitute a mass market. Computers have to be simpler—simpler and more reliable."

This challenge to the industry, he says, is reflected in the formidable transformation of IBM. It's going from being a corporation that sold a relatively small quantity of very expensive items to one that will sell huge quantities of very cheap items.

"I think what's going to happen is what happened with the automobile and telephone. It'd be interesting to do research to see whether, when the automobile first came out, there was any push for people to become internal-combustion-engine-literate so they could drive cars and whether there were any predictions that said sales of automobiles were going to be limited by the availability of chauffeurs."

The auto industry solved that problem by building cars any klutz could drive, and the computer industry must follow suit, according to Tompkins, by building smarter machines programmable in everyday language.

"That's why there's so much excitement about artificial intelligence," he notes. "It may be overrated, but these guys may come up with the answers." The major problem is less technical than hominoid. What electrical engineers can't do today, Tompkins says admiringly, they'll do tomorrow at half the price.

"The real challenge is in the human interface."

That's also Tompkins's specialty—writing custom programs with a personal touch.

"When I look at what I offer in the marketplace, some of it's programming ability, but more of it is just being able to listen," he observes. "You listen to the client describe what he perceives as a problem in his business and then you help him clarify it."

Tompkins tries to determine if the businessperson's problems can be solved with information and, if they can, whether he can do that mechanically by exploiting existing resources or by recommending a new system. Finding the right answers pays the bills and fills a need, but the intellectual payoff for Tompkins is the joy of the search.

"I seem to be happiest when I'm doing something that I don't quite understand how to do."

Taking on challenges appeals to Tompkins's pioneering nature. When he was an undergraduate at Yale, he was moving right along to a degree in Russian studies when he caught a glimpse of his future in some diplomatic outback. "I have all my eggs in one basket," he remembers thinking, and the next thing he knew he was beating a path to the placement office at Yale for an interview with a local IBM recruiter.

"The guy from IBM looked at my resume and said, 'Why are you here?' because there was nothing about computers or math on it. 'I think computers are here to stay, although I've never seen one,' I said, and then I told him that I heard IBM was a good company."

Big, Blue, and Shrewd. Something tells you a vital piece of dialogue is missing from this historical narrative, but IBM liked what it saw and so did Tompkins—for a while anyway. He discovered he was working for a good company, one that was genuinely committed to the credo that to elicit and maintain the best efforts of people you have to treat them with respect and concern. Moreover, he was being introduced to what he terms IBM's crisp, egoless approach to problem solving.

"If you were in a meeting and some guy came up with a doubtful proposal, you didn't have to spend twenty minutes telling him he's a wonderful person before saying his idea was no good."

In the process, of course, he learned the business, picking up the master's degree in computer science from Rensselaer Polytechnic Institute along the way.

To outward appearances he had it made, but by February of 1971, after six years (in harness) at IBM facilities in Connecticut, Tompkins was eager for greater challenge. When his boss could offer nothing better in that department, Tompkins went elsewhere, working three years as director of information services for a fledgling health maintenance organization. He designed its records system, managed computer personnel, and generally administered the place, but after three years he wanted to go back to IBM. The fun was in putting a system together, not in administering it. It was time to return to his main business—computers.

His return, as it happened, coincided with a nearly forgotten skirmish in the early years of the computer age—IBM's minicomputer counterattack. Pressure was building within large organizations for a mid-size computer to handle business deemed too low in priority for the company mainframe. IBM, which didn't make minis at the time, responded to the threat from down below by hatching a remote terminal system based on the interactive programming language APL. And by launching Alan Tompkins.

"I was sitting in a meeting one day and I suddenly realized that there wasn't anyone there who could go much beyond spelling APL, which is a language that uses funny symbols and looks unintelligible," Tompkins recalls. "So I sort of disappeared for a couple of weeks and learned APL and then my whole mission in life became to go around fomenting APL-based guerilla movements."

Tompkins discovered once again that IBM was a great organization, but the working environment was still more suited to people who delighted in becoming the world's foremost au-

thority on the minutiae of some operating system.

"I remember I went to a two-week class in Poughkeepsie, New York, to study MVS, a big IBM operating system. It was a marvel and it was really wonderful that it worked at all, but I had this sinking feeling in my heart that I couldn't get my emotions to the point of taking joy in knowing more about that operating system than anyone else."

Toe to Toe with the Russkis. What had drawn Tompkins to Russian studies—to the economics, politics, and language—was this strange, interactive system known as communism. How did you manipulate incentives in a planned economy and were there correlations between it and the organization of a large company?

"It made me tend to think in terms of systems of inputs and outputs. If you set up a charge-back system on a big mainframe incorrectly, people won't use it properly. I'm a lot more interested in human aspects of technology. How do you get a machine, a computer, to be more productive considering that it's surrounded by human beings?"

Tompkins's partial answer was to get more in touch with himself, his friends, and his surroundings. Like many others in the early seventies, he was caught up in the back-to-the-land movement. He'd spend his evenings curled up with *The Whole Earth Catalog*, thinking of ways to fulfill his long-time desire to move to northern New England.

When a friend stumbled upon an available thirty-acre tract in Waitsfield, Tompkins, the friend, and three other families bought in—the Tompkinses relocating in 1976 after squire Tompkins engineered a transfer to IBM's Vermont office.

Two years ago, Tompkins engineered his biggest move yet, bidding Big Blue a final adieu as he embarked on a bold adventure as an independent programmer/consultant. It took him awhile to adjust to scheduling his time and getting his outgo in phase with his cash flow.

"If you measure income just in dollars, my income is lower," he says, "but high income isn't one of my goals. If I spend all afternoon reading a journal or a book, I won't make any money, but it may enhance my ability to do another job." The big difference, as he saw it then and sees it now, is that he's doing a variety of challenging things—writing code for a trucking company minicomputer network, doing some consulting work via IBM for a Florida industrial group, interfacing an IBM minicomputer with a Japanese telephone exchange in rural Nevada.

"The telephone exchange normally writes its toll billing data onto a magnetic tape," he explains, "so there was an effort mounted to see whether, instead of having real magnetic tape, we could program a minicomputer to look like a tape drive."

Pastoral Symphony. Challenging jobs like these were what Tompkins was after when he left IBM, and so far the choice assignments have been finding him. Combine this serendipity with the earthly delights of country living, a four-level town house built into the side of a hill, fields of hay, and a tiny golden pond, and it's easy to see why Tompkins might think he's reliving a childhood fantasy from the summer of '44.

"When I was little, there used to be a big golf course surrounded by woods next to our house," he says, recalling his early kinship with open spaces and rural areas. "All during the war, the place was mostly empty, and I used to spend hours and hours imagining various kinds of adventures happening to me as I ran around and around."

Today, on a course of his own design, Tompkins is letting his imagination do the running. After all, he's got more serious business to attend to, that yin and yang of Vermont living, circa 1983—chopping and computing.

Olivieri's Outline of Word Processors

by Peter Olivieri

It's time to look at two more word processing packages, *ScreenWriter II* and the original *Magic Window*. It's time also to reemphasize the importance of doing your homework. Taking the time to define your needs and preferences will help you determine which program is the right one for you and will reduce the number of packages you need to look at.

In the end, only a handful of word processing packages will turn out to be really excellent. We've already looked at some of these (*Zardax* and *Super-Text* come to mind) and will examine others in the course of this series. Some require that a board be added to your Apple if you want eighty-column display; others give you eighty columns from within the software. And don't forget the hardware aspects of a word processing system. The monitor, the printer, a modification that gives you upper and lower case, an extra RAM card—all can be important factors in determining how useful your system will be.

ScreenWriter II. Sierra On-Line, 36575 Mudge Ranch Road, Coarsegold, CA 93614; (209) 683-6858. $129.95.

Equipment required: 48K, one or two disk drives, monitor, printer.

At one time this package was called *SuperScribe II*. More has changed than the name—the program has undergone some enhancements as well.

ScreenWriter comes close to being a complete word processing system, offering many features that are normally found only on stand-alone systems. It's easy to move, insert, or delete text, and global search and replace and proportional spacing are also provided.

This system allows you to display on the screen exactly what will be printed on paper, including upper and lower case letters, without a hardware modification. It has a seventy-column display mode as well as forty columns for those who prefer it.

A feature that's often found on standalone word processing systems (and not often included in microcomputer word processing programs) is hyphenation. The proper placement of hyphens in text can be a real chore. *ScreenWriter* has a manual mode that allows you to indicate where words should be hyphenated or an automatic mode that hyphenates only when needed, according to guidelines established by you.

If you're an author, you're likely to appreciate the option that allows you to create an index for your document. As many as four separate indices can be created for a single document. You simply surround every word you want to include in the index with a character of your choice. The words you indicate are then stored, along with their page numbers, and when you are ready you can direct the program to print an index. This feature should be considered a definite plus.

When selecting a word processor, many people look for the capability to create and access a large document. Some word processing programs only allow you to work with small documents (ten pages or less). For many applications, of course, this is just fine. By contrast, *ScreenWriter II* uses the disk drive as a sort of "extension" of the Apple's memory. This setup enables the system to handle documents of up to fifty pages (which is about all that can fit on a single disk). Of course, it is still possible to link documents together to create final copy that's even longer.

Special Features. This package has some unique aspects. One is the capability to edit one document while another document is printing (this is possible only if you have two disk drives in your system). The process is called *spooling*. Basically, a document is "put on a spool" (in this case, one of your disk drives) and unraveled as needed.

Some other aspects of *ScreenWriter* also deserve mention. For starters, text files, binary files, and even Basic programs can be edited. In addition, the program offers user-definable keys (you might decide to have the sequence control-R stand for "word processing," for example). There is a form letters option and page headers and footers can easily be incorporated into a document.

Documentation. The *ScreenWriter* documentation includes a user manual, a tutorial on disk, and a reference card. The user manual is nicely put together but doesn't include enough screen images to suit some users. While comprehensive (it runs almost two hundred pages), the manual is not always clear and easy to follow. It's much better than some of the manuals that accompany competing programs and better than On-Line's own earlier efforts, but it could still be improved.

Essentially, the manual is divided into three sections. The first of these discusses the editor, something called runoff (for printing documents), and the embedded commands needed to achieve certain kinds of formatting (a feature not all users like). The intent of this first section is to get the user up and running.

Section 2 is designed for the serious writer. It discusses some of the more advanced editing commands (such things as search and replace, merging, and tabs), the creation of mailing labels and letters, preparation of professional manuscripts, and special formatting commands. Section 3 of the manual is a summary of the commands that have been described throughout the text.

The appendices are quite useful. They cover getting started, caring for disks, testing your printer, doing the shift key modification, and ways of converting *SuperText II* and *Apple Writer* files for use with *ScreenWriter II*. An error-message list is also included here.

User Views. Users talking about *ScreenWriter*'s strengths cite not having to buy additional hardware, being able to enter some printer commands (so as to be able to take advantage of a printer's special features), and the ability to display from ten to seventy columns on the screen. (You should really have a monitor rather than a television set for viewing these characters, although they look fine on some television sets.)

The cursor movement commands are not as convenient in *ScreenWriter II* as they are in

some other systems. Furthermore, some of the embedded commands don't work as specified. This can be frustrating. It's not necessarily easy to get one's printer to do everything it's supposed to be able to.

As with most systems, the keys that must be pressed to accomplish certain maneuvers are sometimes cumbersome and difficult to remember. (Where, oh where, are those sticky labels?) To advance a page in *ScreenWriter II*, for example, you press shift-control-P. Unfortunately, until a new keyboard is designed or an attachment to the game port is developed, this will always be the case to some degree with word processing programs on microcomputers. In any case, once you learn your word processor's keystrokes, you won't notice the inconvenience.

Conclusion. All in all, *ScreenWriter* is a thorough and professional word processing system. It includes all the features that even the most discerning user might want. It should certainly be among the packages on your final list of candidates.

Magic Window. Artsci, 5547 Satsuma Avenue, North Hollywood, CA 91601; (213) 985-2922. Equipment required: 48K, one disk drive, monitor, printer. $99.95.

There are two versions of this program—the original forty-column *Magic Window* and *Magic Window II*, an enhanced version that offers forty, fifty-six, and eighty columns as well as other features. We'll look this month at the original *Magic Window* and consider *Magic Window II* later on in this series.

Magic Window is one of the easier word processing systems to learn to use. In designing their system, the program's authors have tried to maintain the operating simplicity of a typewriter. As a result, the program operates in a somewhat different fashion than many of the word processors we've discussed to date. For one thing, it doesn't require the user to place any formatting commands within the text itself. Instead, a format subsystem is used to describe the page length, the size of the top margin, and the line-spacing requirements. (This is basically equivalent to placing the paper in a typewriter, setting the margins and the line spacing, and so on.)

Once the form is designed, you can start typing your document. The cursor doesn't move; rather, it remains in the center of the screen as text passes by it, a process that resembles the way the carriage and paper function together when you are using a typewriter.

In this version of the program, only forty characters can be displayed on a line at one time (twenty on either side of the cursor). You can look at the entire line; you just can't see it all at once. While some people find this aspect of the program hard to get used to, others are not bothered by it at all.

From the main menu, you can select the format subsystem just described or any of the options that follow.

The Editor. When you enter the editor, you are in a position to enter text. Simply start typing. Escape can be used to signify an upper-case letter, although an inexpensive hardware modification is recommended so that upper and lower case can be displayed.

All editing commands are entered by using the control key in combination with other keys. For example, control-Q moves the cursor up one line; control-Z moves it down one line. The cursor can be moved to the first line of text in a file, twelve lines ahead, twelve lines back, one line up or down, one character forward or backward, and to the beginning or end of the file.

Other edit commands allow you to search for a particular string, move to the top of a particular page, delete a character, insert a character, delete a line, recover a deleted line, insert a line, and justify lines or center them. In addition, you can split and glue lines (this is analogous to the "cut and paste" features for moving text that are available in many word processing packages).

The Flier. This subsystem performs file maintenance. A menu is presented that allows you to load, save, and delete files. You can also save unformatted files so that information you've entered can be recalled later and used in a variety of ways.

The Printer Subsystem. This subsystem presents you with a list of choices that will affect the final printing of your document. It is here that you can identify which pages of a document are to be printed.

The Configuration Subsystem. The configuration subsystem is ordinarily the first selection you'll make when you're beginning to use this program. This system defines for *Magic Window* what components make up your particular system. You're asked whether or not you have a lower-case video adapter, what slot your printer is in, and whether or not you want an audible sound when you press a key. You're also asked for some information about your printer. It's possible to put printer control characters in the body of the text you create so that you can make use of the special features of your particular printer.

The User Guide. The *Magic Window* manual is very clearly written. Because of the ease of operation of the package, a lengthy guidebook is not really necessary. With the appendices, which contain information on installing RAM printer drivers and on using the Apple parallel interface, the manual contains less than sixty pages. Included in the manual is a very handy reference card that displays (in color) what the keyboard looks like when the keys are "labeled" with the appropriate *Magic Window* commands.

Summary. *Magic Window* is a very nicely done package, and it is certainly one of the easiest to learn to use. It may not have all the "bells and whistles" of some of the other systems, but clearly it contains most of the essential features of a good word processor. If ease of use is one of your primary considerations, then *Magic Window* should be a major contender on your selection list. ■

THE PASCAL PATH

By Jim Merritt

Tools of the Craft, Part 19

The Complete Cable. As you read this, most parts of the United States are firmly in the grip of crotchety Old Man Winter. If you're living through a cold spell yourself, you've probably already discovered that the versatile Apple makes a dandy hand warmer. With any luck, the following listing, which presents the *Cable* data manipulation program in its entirety, will give you plenty of reason to huddle near your silicon stove. Before proceeding with this month's material, look over the program carefully, noting changes and additions from last month. In particular, examine the global tools SCapitalize, Alphanumeric, SNToString, DisplayHome, GoodAddr, GoodStreet, and GoodHNum first. Familiarity with these routines will enable you to understand ChangeTown and DisplayTown more easily.

```
PROGRAM
 Cable;
(* DESCRIPTION: Permit the interactive establishment and maintenance of
   records concerning a Cable Television franchise's subscribers. *)
 CONST
  Header=        'CABLE DATA BASE (V1.3 01-Nov-82)';
  Blank=         ' ';
  EMPTY=         '';
  (* Maximum house number—unrealistic *)
     MaxHNum= 999;
  (* Customer account numbers range from 1 to MaxAcctNum; 0 as an
     account number signifies that the home in question contains no
     subscribers. *)
   NoSubscriber=
                 0;
   MaxAcctNum=
                 MaxInt;
 TYPE
  AcctNumType=
   NoSubscriber .. MaxAcctNum;
  StreetName=
   (Redwood, Tanglewood, Sandalwood, Driftwood);
  HouseNumber=
   1 .. MaxHNum;
  (* How our model is structured:
       A Town is composed of named Streets.
       A  Street is composed of numbered Homes.
       A    Home is modeled by the information we wish
              to record about it and its residents. *)
 Home=
  AcctNumType;
 Street=
  ARRAY[HouseNumber] OF Home;
 Town=
  ARRAY[StreetName] OF Street;
  CComType=          (* Cable program commands *)
   (Change, Display, Quit);
VAR
 Smallville
   :Town;
 UserQuits
   :Boolean;
FUNCTION
 Capital(Ch
          :Char)
   :Char;
(* Return Ch, converted to upper case (capital), if Ch is lower case. *)
BEGIN (* Capital *)
 Capital := Ch; (* No change unless lower case *)
 IF ((Ch >= 'a') AND (Ch <= 'z'))
  THEN (* it's a lower-case letter—transform it! *)
   Capital := Chr(Ord(Ch) - Ord('a') + Ord('A'));
  (* Otherwise, it's not a lower-case letter, so leave it alone. *)
 END (* Capital *);
 PROCEDURE
  SCapitalize(VAR S: String);
 (* Capitalize all lower-case letters in S *)
  VAR
   I
    :Integer;
BEGIN (* SCapitalize *)
 FOR I := 1 TO Length(S) DO
  S[I] := Capital(S[I]);
END (* SCapitalize *);
FUNCTION
 Alphanumeric(Ch: Char)
  :Boolean;
(* Returns True if Ch is a letter or a digit *)
BEGIN (* Alphanumeric *)
 Ch := Capital(Ch);
 Alphanumeric := (((Ch>= 'A') AND (Ch <= 'Z')) OR ((Ch >= '0') AND
                  (Ch <= '9')));
END (* Alphanumeric *);
FUNCTION
 SkipBlanks(VAR S: String; VAR SP: Integer)
  :Integer;
 VAR
  OriginalSP
   :Integer;
(* On entry, assume Length(S)>= SP >= 1. On exit, SP points to the first
   nonblank character at or after the original SP position. If all characters
   from original SP onward are blank, SP contains Length(S) + 1 on exit. In
   all cases, return as function value the number of blanks actually skipped
   (possibly 0). *)
BEGIN (* SkipBlanks *)
 OriginalSP := SP;
 (* Skip any blanks: *)
 (* NOTE: In the following loop, the expression Copy(S, SP, 1) is used
    instead of S[SP] to avoid value-range errors in extreme cases. Don't
    alter this expression unless and until you know why it is "safe." *)
 WHILE (Copy(S,SP,1) = Blank) DO
  SP := SP + 1;
 (* Compute and return number of blanks skipped: *)
 SkipBlanks := SP - OriginalSP;
END (* SkipBlanks *);
PROCEDURE
 IntToString(Source: Integer; VAR Dest: String; MinFW: Integer; LPad:
             String);
(* Build the character-string representation of decimal Source, such that it
   contains at least MinFW characters. Pad on the LEFT using Pad string, if
   necessary, to achieve the Minimum Field Width, MinFW. Concatenate the
   final, padded representation of Source onto the right of Dest. *)
 CONST
  Radix= 10;
 VAR
  Sign
   :String[1];
  TDest
   :String;
BEGIN (* IntToString *)
 TDest := Empty;
 IF (Source < 0)
  THEN
   BEGIN
    Sign := '-';
    Source := -Source; (* make positive *)
   END
  ELSE
   Sign := Empty;
 REPEAT
  (* These next two lines made necessary by the fact that String and Char
     are two separate and distinct data types. In particular, you cannot use
     a value of type Char as an argument to Concat, which deals
     exclusively with Strings. This common method of "Concatenating"
     a Char to a String involves concatenating a "dummy" one-character
     string (such as a Blank) to the victim string, then accessing that
     character position directly, using Char ARRAY syntax. *)
  TDest := Concat(Blank,TDest); (* blank char at front *)
  (* Now, replace the blank with appropriate digit char *)
  TDest[1] := Chr(Ord('0') + (Source MOD Radix));
  (* Conceptually, "lop off" least significant digit and slide each survivor
     over one to the right. *)
  Source := Source DIV Radix;
 UNTIL (Source = 0);
 TDest := Concat(Sign, TDest);
 IF ((Length(TDest) < MinFW) AND (LPad <> Empty))
  THEN
   BEGIN (* Apply leftward padding *)
    WHILE (Length(TDest) < MinFW) DO
     TDest := Concat(LPad,TDest);
    (* In case Length(LPad) > 1, make sure padding process doesn't
       overshoot the mark. *)
    TDest := Copy(TDest,Length(TDest) - MinFW + 1,MinFW);
   END;
 (* Now, we have final string representation; shove it on the end of Dest as
    specified. *)
 Dest := Concat(Dest, TDest);
END (* IntToString *);
FUNCTION
 GoodInteger(VAR S: String;        VAR SP: Integer;
             VAR Dest: Integer)
  :Boolean;
(* Return True if character sequence in S, starting at position SP, represents
   a valid Integer (ignoring leading blanks). If so, SP becomes SP +
   <length of sequence>, and Dest acquires the value of the
   corresponding Integer. On False return, SP and Dest remain
   untouched. *)
 CONST
  Radix=          10;
 VAR
  (* Again, we work with temporary string pointers, dest variables, until we
     know we have a winner; then, everything is made permanent. *)
  TSP,
  TDest
   :Integer;
  SyntaxOK,   (* True if Good Integer so far. *)
  Sign        (* True if Dest should be neg *)
   :Boolean;
  SignChar     (*"Holding Tank" that lets us        *)
   :String[1]; (*   convert easily between String   *)
               (*   and Char. Made small to         *)
               (*   conserve memory space.          *)
BEGIN (* GoodInteger *)
 SyntaxOK := False;
 TSP := SP;
 IF (SkipBlanks(S, TSP) = 0)
  THEN
   (* No problem—either 0 or nonzero is okay. *);
 (* Just gimme some kinda sign, yeah! *)
 (* OneChar is used for convenience only *)
  SignChar := Copy(S, TSP, 1); (* Get possible sign *)
  Sign := (SignChar = '-');
  IF ((SignChar = '+') OR (SignChar = '-'))
  THEN (* Look beyond it *)
   TSP := TSP + 1;
 (* At this point we had better be looking at a digit or else say no go, no-oo
    oh, I can't go for that, no can do ... *)
 TDest := 0;
 WHILE ((Copy(S, TSP, 1) >= '0') AND (Copy(S, TSP, 1) <= '9')) DO
  BEGIN (* Looks a lot like IntegerInput *)
   SyntaxOK := True;
   TDest := (TDest * Radix) + (Ord(S[TSP]) - Ord('0'));
   TSP := TSP + 1;
  END;
 GoodInteger := SyntaxOK;
 IF SyntaxOK
  THEN
   BEGIN (* make everything permanent *)
    Dest := TDest;
    IF Sign
     THEN
      Dest := -Dest;
    SP := TSP;
   END;
END (* GoodInteger *);
PROCEDURE
 SNToString(StName: StreetName; VAR Dest: String; MinFW: Integer;
            RPad: String);
 VAR
  S
   :String;
  Padded
   :Boolean;
(* Build the character-string representation of StName, such that it contains
   at least MinFW characters. Pad on the RIGHT using Pad string, if
   necessary, to achieve the Minimum Field Width, MinFW. Concatenate the
   final, padded representation of Source onto the right of Dest. *)
BEGIN (* SNToString *)
 CASE StName OF
  Redwood:
   S := 'Redwood';
  Tanglewood:
   S := 'Tanglewood';
  Sandalwood:
   S := 'Sandalwood';
  Driftwood:
   S := 'Driftwood';
 END (* CASE StName *);
 IF ((Length(S) < MinFW) AND (RPad <> Empty))
  THEN
   BEGIN (* pad to the right *)
    WHILE (Length(S) < MinFW) DO
     S := Concat(S,RPad);
    (* In case Length(Rpad) > 1, make sure padding process doesn't
       overshoot the mark. *)
      S := Copy(S,1,MinFW);
   END;
 Dest := Concat(Dest, S);
END
PROCEDURE
 DisplayHome(VAR T: Town; StName: StreetName; HNum:
             HouseNumber);
(* Displays Home Information in one line, thusly:
   NNN SSSSSSSSSS            Acct: AAAAA
  where
   NNN is three digit HouseNumber, left padded with blanks;
   SSSSSSSSSS is ten-character street name, right-padded with blanks;
   AAAAA is five-digit account number, left-padded with zeros, except that
   "NoSubscriber" account number 0 is displayed as the word "NONE."
  Examples:
    23 Redwood                Acct: 01234
   567 Sandalwood             Acct: 98765
     1 Driftwood              Acct: 00009
   123 Tanglewood             Acct: NONE *)
 VAR
  DisplayLine  (* Strategy is to build a single string  *)
   :String;    (*   display line, then show it all at   *)
               (*   once, using a single output call.   *)
BEGIN (* DisplayHome *)
 (* Start building DisplayLine *)
 DisplayLine := Empty;
 IntToString(HNum, DisplayLine, 3, Blank);
 DisplayLine := Concat(DisplayLine, Blank);
 SNToString(StName, DisplayLine, 10, Blank);
 DisplayLine := Concat(DisplayLine, '   Acct: ');
 IF (T[StName, HNum] <> 0)
  THEN
   IntToString(T[StName,HNum], DisplayLine, 5, '0')
  ELSE
   DisplayLine := Concat(DisplayLine, 'NONE');
 WriteLn(Output, DisplayLine);
END    (* DisplayHome *);
FUNCTION
 GoodHNum(VAR S: String; VAR SP: Integer;
              VAR Dest: HouseNumber)
   :Boolean;
 VAR
  TSP,
  TDest
   :Integer;
(* Return True if character sequence in S, starting at position SP, represents
   a valid HouseNumber. If so, SP becomes SP + <length of sequence>
   and Dest acquires the value of the corresponding HouseNumber. On
   False return, SP and Dest remain untouched. *)
```

```
BEGIN (* GoodHNum *)
  (* Let GoodInteger do most of the work; just check to see that Integer it
     gets is in HouseNumber range. *)
  GoodHNum := False;
  TSP := SP;
  IF GoodInteger(S, TSP, TDest)
    THEN
      IF ((TDest >= 1) AND (TDest <= MaxHNum))
        THEN
          BEGIN (* make everything permanent *)
            GoodHNum := True;
            SP := TSP;
            Dest := TDest;
          END;
  END     (*GoodHNum *);
FUNCTION
  GoodStreet(VAR S: String; VAR SP: Integer;
             VAR Dest: StreetName)
    :Boolean;
(* Return True if valid StreetName found in S, beginning at position SP;
   False otherwise. On True return ONLY, SP will point to the character
   position just past the end of the StreetName (possibly Length(S) + 1),
   and Dest will contain the value of the StreetName found. On False return,
   SP and Dest remain unchanged.
   StreetNames are assumed to be rendered in capital letters; it's therefore a
   wise precaution to use SCapitalize on S before calling this procedure.
   Leading blanks are skipped. StreetName may be terminated with any
   punctuation character (NOT alpha, NOT numeric). *)
  VAR
    Candidate
      :StreetName;
    TSP
      :Integer;
    Foundit     (* True if we have a winner *)
      :Boolean;
    CName       (* String rep of "current"      *)
      :String;  (*    StreetName.               *)
    NextChar    (* Another Char/String          *)
      :String[1];  (*    hermaphrodite. Holds    *)
                   (*    value of character      *)
                   (*    immediately following   *)
                   (*    what we think may be    *)
                   (*    a valid StreetName.     *)
  BEGIN (* GoodStreet *)
    (* Nothing is made permanent until very end *)
    TSP := SP;
    IF (SkipBlanks(S, TSP) = 0)
      THEN
        (* No problem—otherwise, we skipped some! *);
    Foundit := False;
    Candidate := Redwood; (* First street name, in order *)
    REPEAT
      CName := Empty;
      SNToString(Candidate, CName, 0, Blank);
      SCapitalize(CName);
      IF (Copy(S, TSP, Length(CName)) = CName)
        THEN
          BEGIN (* Make sure that acceptable name is not merely part of an
                   unacceptable name; that is, do not accept 'Redwoodland' as
                   'Redwood'. End of String or some punctuation must
                   terminate the StreetName. *)
            (* Again, the String versus Char incompatibility rears its ugly head—
               NextChar exists to handle single characters as either strings
               (NextChar) or Char values (NextChar[1]). *)
            NextChar := Copy(S,TSP+Length(CName),1);
            IF (NextChar = Empty)
              THEN
                NextChar := Blank;
              (* NOTE: IF (NextChar = Empty), expression NextChar[1] would
                 cause a "VALUE RANGE ERROR." *)
            IF (NOT Alphanumeric(NextChar[1]))
              THEN
                BEGIN
                  Foundit := True;
                  TSP := TSP + Length(CName);
                END;
          END;
      IF (NOT Foundit)
        THEN
          Candidate := Succ(Candidate);
          (* Go on to the next one *)
    UNTIL (Foundit OR (Candidate > Driftwood));
    GoodStreet := Foundit;
    IF Foundit
      THEN
        BEGIN (* all is well; make everything permanent *)
          Dest := Candidate;
          SP := TSP;
        END;
  END     (* GoodStreet *);
FUNCTION
  GoodAddr(S: String;      VAR SP: Integer;
           VAR StN: StreetName;
           VAR HNum: HouseNumber)
    :Boolean;
(* Return True if character sequence in S, starting at position SP,
   represents a valid Home Address. If so, SP becomes SP + < length of
   sequence >, and StN and HNum are set to the corresponding values.
   On False return, SP, StN, and HNum remain untouched. Note two
   special "abbreviated addresses:" 'F' and 'L', provided for the
   convenience of the user. 'F' stands for the first address in the
   neighborhood, 'L' for the last. *)
  VAR
    TSP         (* Temporary SP for string scan *)
      :Integer;
    (* Temporary StN and HNum, will be made permanent once values
       found for both prove valid. *)
    TStN
      :StreetName;
    THNum
      :HouseNumber;
  BEGIN (* GoodAddr *)
    GoodAddr := False;
    SCapitalize(S);
    S := Concat(S, Blank); (* Sneaky *)
    TSP := SP;
    IF (SkipBlanks(S, TSP) = 0)
      THEN
        (* Say, Bud, let's paaaaarty, y'know? *);
    IF ((Copy(S, TSP, 2) = 'F ') OR (Copy(S, TSP, 2) = 'L '))
      (* Sneaky blank used  ^              and              ^ *)
      THEN
        BEGIN (* Handle these special cases *)
          CASE S[TSP] OF
            'F':
              BEGIN
                HNum := 1;
                StN := Redwood;
              END;
            'L':
              BEGIN
                HNum := MaxHNum;
                StN := Driftwood;
              END;
          END (* CASE S[TSP] *);
          GoodAddr := True;
          SP := TSP + 1;
        END
      ELSE (* Handle usual case (HouseNumber-StreetName) *)
        IF GoodHNum(S, TSP, THNum)
          THEN
            IF GoodStreet(S, TSP, TStN)
              THEN
                BEGIN (* Everything becomes permanent *)
                  GoodAddr := True;
                  SP := TSP;
                  HNum := THNum;
                  StN := TStN;
                END;
  END     (* GoodAddr *);
PROCEDURE
  NewTown(VAR T: Town);
(* DESCRIPTION: Ready the model of a new town by "emptying" all its
   houses. This routine optimizes speed of initialization—clears first street in
   Town house by house, then uses "whole array" assignment to clear each
   succeeding street in a single step. *)
  VAR
    HNow
      :HouseNumber;
    SNow
      :StreetName;
  BEGIN (* NewTown *)
```



```
    FOR HNow := 1 TO MaxHNum DO
      T[Redwood][HNow] := No Subscriber;
    FOR SNow := Succ(Redwood) TO Driftwood DO
      T[SNow] := T[Redwood];
    T[SNow]     (* NewTown *);
  END
  PROCEDURE
  ChangeTown(VAR T: Town);
(* DESCRIPTION: Permit the interactive selection and modification of one
   (or more) Home(s) in a Town, T. *)
  VAR
    Answer      (* Storage for responses typed by user *)
      :String;
    ASP         (* Answer string pointer—used to isolate portions of user
                   response for close examination. *)
      :Integer;
(* Change Homes, from StartHNum of StartStreet, until user specifies
   QUIT or until the last Home in the neighborhood has been changed.
   After changing last Home on a particular Street, skip to the first Home
   on the next Street. *)
    StartStreet,
    StreetNow   (* Street of Home being changed. *)
      :StreetName;
    StartHNum,
    HNumNow     (* Number of Home being changed. *)
      :HouseNumber;
    Good        (* Set True when user specifies good address.    *)
      :Boolean; (* An address is the string representation       *)
                (*    of an Integer in the HouseNumber range,    *)
                (*    followed by one or more blanks, then the   *)
                (*    capital or lower-case representation of    *)
                (*    a valid StreetName such as "45 Redwood."   *)
    GoodReply
      :Boolean; (* Set True when we're happy with what the       *)
                (*    user tells us. Used here to keep track     *)
                (*    of whether or not the user replies sensibly *)
                (*    to our request for an acct number.         *)
    Quit
      :Boolean; (* Set True when, in the midst of Change,        *)
                (*    user wants to quit and return to main      *)
                (*    menu.                                      *)
    ANum        (* Temporary receptacle for any acct numbers     *)
      :Integer; (*    specified by the user.                     *)
  BEGIN (* ChangeTown *)
    REPEAT
      Write(Output, 'Change starting at what home ... ? ');
      ReadLn(Input, Answer);
      ASP := 1;
      Good := (GoodAddr(Answer, ASP, StartStreet, StartHNum) OR
               (Answer = Empty));
      IF (NOT Good)
        THEN
          BEGIN
            WriteLn(Output, '*** Not a legal address');
            WriteLn;
          END;
    UNTIL ((Answer = Empty) OR Good);
    IF (Answer <> Empty)
      THEN
        BEGIN
          Quit := False;
          StreetNow := StartStreet;
          WHILE ((StreetNow <= Driftwood) AND (NOT Quit)) DO
            BEGIN (* Show all selected houses on this street *)
              (* On StartStreet, selected houses begin at StartHNum, otherwise,
                 at 1. *)
              IF (StreetNow = StartStreet)
                THEN
                  HNumNow := StartHNum
                ELSE
                  HNumNow := 1;
              (* First, show the Home, then permit change! *)
              WHILE ((HNumNow <= MaxHNum) AND (NOT Quit)) DO
                BEGIN
                  DisplayHome(T, StreetNow, HNumNow);
                  WriteLn(Output,
                          '  Press RETURN key for same, ',
                          'type QUIT to quit');
                  REPEAT (* Keep asking for acct # until a good reply. *)
                    Write(Output,' New account #:');
                    ReadLn(Input, Answer);
                    ASP := 1;
                    GoodReply := GoodInteger(Answer, ASP, ANum);
                    IF (GoodReply AND (ANum < 0))
                      THEN (* acct number is out of range *)
                        BEGIN
                          WriteLn(Output,
                                  '*** Account # can't be negative');
                          GoodReply := False;
                        END;
                    ELSE
                      IF GoodReply
                        THEN (* acct number in range *)
                          T[StreetNow, HNumNow] := ANum
                        ELSE (* no acct number—is reply sensible? *)
                          BEGIN
                            (* Here, we decree that a string containing only blanks is
                               the same as an empty one. Empty reply means "don't
                               change this particular acct number." *)
                            GoodReply := (SkipBlanks(Answer, ASP)
                                          = Length(Answer));
                            IF (NOT GoodReply)
                              THEN (* it better be QUIT *)
                                (* Be kind—just check first char *)
                                IF (Capital(Answer[ASP]) = 'Q')
                                  THEN
                                    BEGIN
                                      GoodReply := True;
                                      Quit := True;
                                    END
                                  ELSE
                                    WriteLn(Output, '*** Not a valid reply.');
                          END;
                  UNTIL GoodReply OR Quit;
                  WriteLn(Output);
                  (*$R-*) HNumNow := Succ(HNumNow); (*$R+*)
                  (* The $R- compiler directive disables range-checking code,
                     while $R+ enables it. This is done here to avoid spurious
                     "value-range errors" during the transition between Streets
                     (when HNumNow must necessarily and temporarily become
                     one greater than MaxHNum). This precaution is not necessary
                     when incrementing variables of enumerated types, only when
                     dealing with numeric values. See text for further clarification. *)
                END;
              StreetNow := Succ(StreetNow);
            END;
        END;
  END     (* ChangeTown *);
```

```
      END;
     StreetNow := Succ(StreetNow);
     (* Note: no "range-checking" trickiness needed here. *)
    END  (* WHILE *)
   END;
  WriteLn(Output); (* Avoid screen clutter *)
 END  (* ChangeTown *);
 PROCEDURE
  DisplayTown(VAR T: Town);
 (* DESCRIPTION: Permit the interactively controlled display of information
   recorded for one or more Home(s) in a Town, T. *)
  VAR
   Answer              (* Storage for responses typed by user *)
    String;
   ASP                 (* Answer string pointer—used to isolate portions of
                          user response for close examination. *)
    Integer;
   (* Display proceeds from StartHNum of StartStreet through EndHNum of
     EndStreet and includes all Homes in between. *)
   StartStreet,
   EndStreet,
   StreetNow           (* Street of Home being displayed *)
    StreetName;
   StartHNum,
   EndHNum,
   HNumNow,            (* Number of Home being displayed.  *)
   FHNum,              (* Numbers of first and last Homes to *)
   LHNum               (*    display on StreetNow.          *)
    House Number;
   Good                (* Set True when user specifies good address.  *)
    Boolean;           (* See ChangeTown for address definition.   >  *)
BEGIN (* DisplayTown *)
 REPEAT
  Write(Output, 'Display from what home ... ? ');
  ReadLn(Input, Answer);
  ASP := 1;
  Good := (GoodAddr(Answer, ASP, StartStreet, StartHNum) OR (Answer
          = Empty));
  IF (NOT Good)
   THEN
    BEGIN
     WriteLn(Output, '*** Not a legal address');
     WriteLn;
    END;
 UNTIL ((Answer = Empty) OR Good);
 IF (Answer <> Empty)
  THEN
   BEGIN
    REPEAT
     Write(Output, ' ... to what home?          ');
     ReadLn(Input, Answer);
     ASP := 1;
     Good := (GoodAddr(Answer, ASP, EndStreet, EndHNum) OR
             (Answer = Empty));
     IF (NOT Good)
      THEN
       BEGIN
        WriteLn(Output, '*** Not a legal address');
        WriteLn;
       END;
    UNTIL ((Answer = Empty) OR Good);
    IF (Answer <> Empty)
     THEN (* show 'em what you've got! *)
      FOR StreetNow := StartStreet TO EndStreet DO
       BEGIN (* Show all selected houses on this street *)
        (* On StartStreet, selected houses begin at StartHNum,
          otherwise, at 1. *)
        IF (StreetNow = StartStreet)
         THEN
          FHNum := StartHNum
         ELSE
          FHNum := 1;
        (* On EndStreet, selected houses begin at EndHNum, otherwise,
          at MaxHNum. *)
        IF (StreetNow = EndStreet)
         THEN
          LHNum := EndHNum
         ELSE
           LHNum := MaxHNum;
        (* So, show 'em already! *)
        FOR HNumNow := FHNum TO LHNum DO
              DisplayHome(T, StreetNow, HNumNow);
           END;
        END;
   WriteLn(Output); (* Avoid screen clutter *)
  END  (* DisplayTown *);
  FUNCTION
   CableCommand
     CComType;
  (* DESCRIPTION: Prompts for and accepts user input characters until one
    corresponds to a CComType command, then returns the matching value.
    C, D, and Q map onto Change, Display, and Quit. Treats capitals and
    lower case as identical. Echoes blank for blank, command name for
    command characters, and the input character itself, along with the
    message '—NOT A COMMAND', for all others. Pressing the return key is
    equivalent to pressing space bar. *)
   CONST
    Prompt=     'Command: Change, Display, Quit >> ';
   VAR
    ComCh
     :Char;
    Valid
     :Boolean;
  BEGIN (* CableCommand *)
   Write(Output, Prompt);
   Valid := False;
  REPEAT
   Read(Keyboard, ComCh);
   CASE Capital(ComCh) OF
    'C':
     BEGIN
      Valid := True;
      Write(Output, 'Change');
      CableCommand := Change;
     END;
    'D':
     BEGIN
      Valid := True;
      Write(Output, 'Display');
      CableCommand := Display;
     END;
    'Q':
     BEGIN
      Valid := True;
      Write(Output, 'Quit');
      CableCommand := Quit;
     END;
    Blank:
     Write(Output, ComCh);
   END (* CASE Capital(ComCh) *);
   IF ((NOT Valid) AND (ComCh <> Blank))
    THEN
     BEGIN
      WriteLn(Output, ComCh, '—NOT A COMMAND');
      Write(Output, Prompt);
     END;
  UNTIL Valid;
  WriteLn(Output);
  END    (* CableCommand *);
 BEGIN (* Cable *)
  WriteLn(Output, Header);
  WriteLn(Output);
  NewTown(Smallville);
  UserQuits := False;
  REPEAT
   CASE CableCommand OF
    Change:
     ChangeTown(Smallville);
    Display:
     DisplayTown(Smallville);
    Quit:
     UserQuits := True;
   END (* CASE CCom *);
  UNTIL UserQuits;
 END     (* Cable *)
```

Points of Interest. Because the listing is designed to be self-explanatory, and because you should be able to determine gross program structure and behavior on your own, our discussion will touch primarily upon *Cable*'s subtler aspects. Don't make the mistake of trying to explore *Cable*'s innards without first having browsed the listing from top to bottom at least once. You need to get in the habit of learning about a program from reading only the listing; in many cases, the listing is the only "program manual" you'll receive!

SCapitalize and Alphanumeric. The listing alone is sufficient to clarify the two minor tools SCapitalize and Alphanumeric, so we will not discuss them here except to note that both are so handy that they are bound to be needed by future programs. By coding them as independent subroutines, we have made it easier to transplant them when the time comes. You might find it worthwhile to collect the Pascal source code for such tools as Capital, SCapitalize, Alphanumeric, SkipBlanks, IntToString, and GoodInteger into a single file named "TOOLS.TEXT". In the future, whenever you devise a new tool, add its source code to the "TOOLS" file as well. Keeping all your "utility routines" in one place will prove more and more convenient the longer you follow the Path.

SNToString. The job of SNToString is to produce the String representation of a StreetName. As a data conversion procedure, then, it is similar in purpose and design to IntToString, which we developed last month. SNToString is somewhat less complicated, of course, needing nothing more than a CASE statement at its heart. It also differs from its predecessor in its "padding" behavior.

Remember, a field width is nothing more than the minimum number of character places that a data representation must fill. If the "natural" String representation for a given datum occupies less than the required amount of space, a data conversion routine must fill, or "pad," the unused space with neutral characters (usually blanks). A conversion routine determines, according to its design, where to append these extra "pad" characters.

Tacking pad characters onto the left side shoves a data representation over to the right side of the field. Adding them on the right side keeps the datum on the left side of the field. Putting equal amounts of padding on both sides has the effect of positioning the datum in the center of the field.

As implemented here, IntToString positions convert data on the right side of the field, while SNToString puts its data on the left. The two routines differ in their padding behaviors because of the conceptual differences that exist between StreetNames and Integers. Numbers are frequently displayed in columns, such that their rightmost digits are flush with the right column margin. Data that look like English words, such as StreetNames, are usually displayed flush with the left margin. So, IntToString and SNToString are designed to pad properly in the majority of cases. Compare the set of statements used for String padding in IntToString with the set in SNToString; note that each is, in some sense, the mirror image of the other.

DisplayHome. DisplayHome's brevity is due to the existence of IntToString and SNToString, which perform most of the work. The routine is so small, in fact, that commentary—not working code—dominates its listing. Look at the output examples as given in the comments; then try to see how DisplayHome might produce each display. In particular, notice that the HouseNumber in each example is padded with blanks, while the account number is padded with zeros. The padding mechanism treats HouseNumbers and account numbers equally, yet the visual impacts of the Strings produced differ greatly. The difference is clearly due to the programmer's choice of "padding pattern."

DisplayHome also demonstrates that it is not always appropriate to place numbers on the right sides of output fields. If a number is the leftmost item on a display line, as it is in the format chosen for DisplayHome, aesthetic sensibilities almost demand that the leftmost digit be aligned flush with the left-hand margin. To be fair, IntToString permits this, after a fashion. If you specify a field width of 0 or 1, you disable the padding process, and the datum in question will then be flush with the left side of the field. However, in this instance, such a choice would most likely force the street names and account numbers out of their own neat alignment. Do you see why?

With a little thought, you can also devise examples in which left-hand placement is clearly inappropriate for certain nonnumeric data, thus invalidating our design choice for SNToString. The best solution is to permit the caller of a data conversion routine to specify the padding behavior that is most suited to the datum in question. Given the data type Orientation, defined as

```
Orientation = (Left, Center, Right);
```

you could introduce a new parameter to both SNToString and IntToString, as follows:

```
PROCEDURE
 IntToString(Source: Integer;
            VAR Dest: String;
            MinFW: Integer;
            Pad: String;
            Placement: Orientation);
PROCEDURE
 SNToString(StName: StreetName;
            VAR Dest: String;
            MinFW: Integer;
            Pad: String;
            Placement: Orientation);
```

Placement would tell each routine where to place the data representation within the specified field. Thus,

```
IntToString(23,DestString,5,'0',Right)
```

would tell IntToString to produce '00023', while

```
IntToString(587,DestString,10,' ',Left)
```

would cause the routine to generate '587_____', and so on. SNToString would act similarly. In future, all routines presented in this column for the purpose of converting data to string representations will allow the caller to specify left, right, or center placement. When next you see IntToString and SNToString, they will also have been modified to respect this standard. If you feel ambitious, you might try updating your own copies of these routines to reflect the new design.

GoodHNum and GoodStreet. Of the "fraternal twins," GoodHNum and GoodStreet, GoodStreet is the more interesting. GoodHNum merely adds a "range check" to the action of GoodInteger, while GoodStreet demonstrates some useful string-handling techniques. In particular, the expression

```
(Copy(S, TSP, Length(CName)) = CName)
```

is True when the String CName exists within S beginning at position TSP. In GoodStreet, this expression is used to determine whether or not the String under examination, S, matches a given candidate StreetName

GoodStreet tries to match S with each StreetName, in succession, until every StreetName has been tried or a match occurs.

As we agreed in a previous column, the user should be able to abbreviate the first address in the town to 'F' and the last to 'L'. If it weren't for these "abbreviated addresses," GoodAddr would involve little more than calling GoodHNum and GoodStreet. However, special cases always complicate matters, and so more than half of GoodAddr is dedicated to handling 'F' and 'L'.

We've seen that, when accepting human input, the polite computer ignores leading and trailing blanks that are obviously unrelated to the data. When given a regular address, GoodAddr relies on GoodHNum and GoodStreet to ignore any extra blanks. In the case of the abbreviated addresses, we use a trick to accomplish the task. We concatenate a blank to the end of the input String, S, skip any leading blanks using SkipBlanks, and check to see whether or not we find 'F ' or 'L ' at the current scan position in S. Think carefully about why this guarantees that GoodAddr will always accept the single character 'F' or 'L' regardless of the number of blanks that come before or after it but will always reject multiple-letter input such as 'FIRST', 'LAST', 'F&' or 'L*'. Do you think that GoodAddr should accept 'FIRST' and 'LAST' as well as 'F' and 'L'? If so, modify the routine to do this, but make sure that it still rejects input such as 'FARKLE', 'LUSTY', 'F123', and 'L&M'.

DisplayTown and ChangeTown. DisplayTown requires the user to specify a range of Town addresses, then displays the Home information (so far, only the account number) for every address in the given range. Nested FOR-loops—the outer one stepping from Street to Street and the inner one moving from Home to Home—are used to drive the display scan. The FOR-loop is appropriate in this case, because of the deterministic nature of the scan; the starting and ending points are well known even before loop execution begins.

ChangeTown is not so deterministic. It only asks the user for a starting address, then scans the Town, Home by Home, until either the user decides to quit or the Town's last Home is scanned. As every Home is scanned, the user has the option of quitting the Change process, skipping the current Home, or changing that Home's account number. As in DisplayTown, two nested loops are used in scanning the Town, but neither can be a FOR-loop because of the complexity of the termination conditions involved.

Remember, FOR-loops are appropriate only for pure "counting" applications, such as the display scan in DisplayTown. When a loop may terminate on any one of several conditions, you must program it using either a WHILE-DO or a REPEAT-UNTIL construction. REPEAT-UNTIL loops are employed in ChangeTown because each loop body must execute at least once before the termination condition can be checked. The computer must actually ask the user whether or not to quit, or it must increment the loop counter past the limit value, before it can decide if another iteration is required. Since both these actions are properly part of the loop bodies, REPEAT-UNTIL loops are most appropriate here. If you think that WHILE-DO loops would be clearer or more efficient, try writing your own versions and send them to this magazine for verification.

There is a problem inherent in abandoning FOR-loops, however. The inner REPEAT-loop in ChangeTown relies partially on a "counter," the variable HNumNow, in order to determine the status of its termination condition. According to the UNTIL clause, inner-loop execution may cease whenever HNumNow becomes greater than MaxHNum. As we might expect, the loop body contains a statement that ensures that this part of the termination condition will eventually hold true. Specifically, at the end of the loop body, we assign Succ(HNumNow) to HNumNow, thus incrementing that variable's value by one step.

So long as the user doesn't specifically tell the computer to quit changing account numbers, the inner loop will iterate until HNumNow eventually equals MaxHNum. At the end of this crucial iteration, the last statement in the loop body will attempt to increment HNumNow so that its value equals HNumNow + 1. At this point, the Pascal system will abort the *Cable* program, complaining of a *value range error*. The system is well within its rights to issue this error message. Our program does, in fact, try to assign HNumNow a value that lies outside the HouseNumber range.

We have created a paradox. The value of HNumNow that signals loop termination is not a valid HouseNumber. Whenever our program attempts to assign that value to HNumNow, the Pascal system steps in and aborts execution. Certainly, loop execution does cease at the desired point, but not in a friendly manner! How do we resolve our problem? Actually, there are several steps we might take; each, however, involves compromise, and one entails considerable frustration, on the way to a dead end! Let's examine each option, along with its consequences:

Enlarge the true HouseNumber range, leave loop untouched: We would like to define HouseNumber as "1 .. MaxHNum+1" in order to leave room for the "spillover" that is a natural consequence of our REPEAT-UNTIL loop. Unfortunately, we can't define a subrange in terms of expressions like "MaxHNum + 1", so we'd be forced to define another constant, TrueMaxH, say, to be equal to a value one greater than MaxHNum. Thus, in keeping with the MaxHNum value that we have used so far, TrueMaxH would be set equal to 1000. Then, the HouseNumber range may be redefined as "1 .. TrueMaxH" while we remain careful never to assign numbers greater than MaxHNum to any HouseNumber variables. The only permissible exception to our rule is the occasional "off-by-one case," such as the needy REPEAT-UNTIL loop. Of course, if ever we decided to change MaxHNum, we would also have to remember to change TrueMaxH; the compiler has no way of knowing that the value of one constant depends upon that of the other.

This method of compensating for the spillover—by extending the HouseNumber type but restricting the bulk of computation to a slightly smaller subrange—is like asking a chronically tardy friend to arrive five or ten minutes before you really want to meet with him. You know you can't keep him from being late, so you arrange matters such that his tardiness is always irrelevant. In the same way, we can't avoid the spillover without changing the loop, so we redefine the limits of the HouseNumber type so that, in the eyes of the Pascal system, no spillover occurs.

There is one noteworthy drawback to this otherwise comfortable

solution: since the array Street is indexed by HouseNumber values, adding any numbers to the HouseNumber range increases the size of a Street. This increase is superfluous, since we intend never to use any HouseNumbers above MaxHNum. We can either choose to ignore the waste, or we might redefine a Street as "ARRAY [1 .. MaxHNum] OF Home." Most programmers choose the latter adaptation.

Rewrite the loop, leave the HouseNumber and Street definitions alone. This is what many beginning programmers would choose to do. The first attempt would probably be to change the termination condition to include "HNumNow = MaxHNum" instead of "HNumNow > MaxHNum". It quickly becomes clear that such a modification makes it impossible ever to Change any Home that is numbered above 998! This is because MaxHNum is incremented at the bottom of the loop! When HNumNow = (MaxHNum − 1), the loop body's last official act would be to increment HNumNow to the value of MaxHNum. The new termination condition would hold, loop iteration would cease, and the Change process would never be able to affect the last Home on the current Street.

If you think we can eliminate our problems by moving the incrementing of HNumNow to the top of the loop body, think again! True, the loop would be able to process the last Home on any Street, but it would also always skip the starting address, as specified by the user, because the program, as currently written, sets HNumNow equal to the HouseNumber of the starting address initially, and the first thing the loop body would do would be to increment the value of HNumNow!

At this point, the last gasp of the thoroughly confused novice programmer is "Aha! Just initialize HNumNow to one value less than the desired starting HouseNumber and let the loop body do its thing!" But what if the starting HouseNumber is 1? You can't assign 0 to a variable of type HouseNumber, because 0 is not within the permissible range. Any attempt at such an assignment causes a value range error.

And so, we've come full circle. If there is an elegant way to solve the spillover problem by rearranging the statements in our program, it is not readily apparent. This approach to the problem fails to provide a solution.

Cheat—leave HouseNumber, Street, and loop body alone; tell Pascal system to ignore any spillover. This is the more dangerous of the two workable tactics we can use to eliminate our difficulty. As you might expect, it is the one we have chosen to use in *Cable*, in a blatant attempt to live dangerously and prime ourselves to study advanced topics that we'll be covering very soon.

Under normal circumstances, the Apple Pascal compiler habitually sneaks special P-codes into any object program it produces. These special codes implement the Pascal system's range-checking facility. For instance, after generating the P-codes that compute an array (or string) index, the compiler will produce a code sequence that, when executed, determines whether or not the index value produced by the first P-code sequence is proper for the array in question. If so, program activity is permitted to continue. If not, control is passed to the operating system's error-handling routine.

Suppose the compiler is processing an assignment statement. First, it generates the P-codes that compute the value to be assigned. Then, it produces a sequence that checks to see if the computed value falls within the acceptable range of values for the destination variable. If it does, the assignment is completed, but, if it doesn't, the system error-handler is called, as in the array-indexing example.

The extra range-checking code emitted by the compiler is said to be "invisible." Obviously, the P-machine sees and executes this code, but you, as the programmer or user, are not usually aware that it is there, protecting the program from going too far astray in its computations. Range-checking code might therefore be called a "freebie," except that it does cost you memory and execution speed. After all, the range-checking P-codes must take up some memory space, and they certainly require a finite amount of time in which to execute. On the other hand, you usually want to know if your fledgling program contains bugs. The slight increase in program size and execution time is a small price to pay to know that the system is watching out for—and guarding itself against—your mistakes.

In the case of ChangeTown, however, we have written a reasonable loop, which, in certain situations, attempts a "suspicious"—but safe—assignment. Because very few assignments involving out-of-range values will ever be truly "safe," the system is certainly correct in warning us, one way or another, of the danger that lies along our chosen path. Granted, to use the run-time error-handling mechanism as a notification channel is a bit extreme. Even so, the system is simply looking out for our best interests. On the other hand, after we have acknowledged the possibility of our error and have checked our code thoroughly to be sure that its "illegitimate actions" are made only for legitimate purposes, we are justified in informing the compiler that our hands need no holding; that we are capable of writing well-behaved code. In other words, we are free to disable range-checking. How?

A Digression: The R−Compiler Directive. The compiler usually ignores all comments. However, comments in which a dollar sign lies immediately to the right of the opening delimiter (with no blanks in between) are taken as instructions to the compiler itself. More precisely, the text that follows the dollar sign, up to the end of the comment, is taken as containing *compiler directives*, which are used to control the compiler's behavior. That is, with compiler directives, you—the programmer—can tell the compiler how to do its job.

A detailed study of compiler directives is scheduled for a future column. Now, we are only concerned with telling the compiler to quit emitting range-checking code. Here is the directive that accomplishes this:

```
(*$R-*)
```

As you might expect, when range-checking is off, and you wish to enable it again, you use the following:

```
(*$R+*)
```

The compiler will produce no range-checking code for statements that occur after an R− directive until it encounters an R+ directive. Pascal's range-checking features are intended to help you discover subtle but potentially catastrophic errors in your programs. If you defeat this protection, you make it possible for erroneous program behavior to escape undetected. Thus, you should disable range-checking only after careful study of your program's construction. Only when you are convinced that your program is sound and that there's no better method of achieving your ends should you issue the R− directive. By refusing the Pascal system's help, you volunteer to be entirely responsible for any errors in your programs, especially those that the unrestrained system might conceivably find through range-checking. In general R− should be applied only to those statements that would directly benefit by it and not to one statement more! In *Cable*, only one statement in the entire program is compiled without range-checking. To have extended the effect of R− into other areas of the program would have been sloppy and negligent on the part of the programmer.

The compiler always elects to produce range-checking code for assignments to variables of scalar subrange types but not for assignments involving variables of unabridged scalar types. The HouseNumber data type is a subrange of Integer. Thus, range-checking code is emitted for all assignments to HNumNow because it is a HouseNumber variable. StreetNow, on the other hand, is declared as being a type StreetName. This type is not a subrange of any other type; it is a complete, unabridged scalar enumeration. Therefore, no range-checking code is ever emitted after assignments to StreetNow. This is why we must use the R− compiler directive to disable range-checking after the incrementing of HNumNow but need not worry about the incrementing of StreetNow.

Next Month: Beyond Cable. Finally, you have the complete *Cable* program in hand. Take the opportunity to play around with it; study it; get to know it so well that you can't help thinking of ways to improve and enhance it. When you sit down to read next month's column—the conclusion of the series on *Cable*—keep your ideas (and a program listing) handy, because we'll explore motivations and methods for tailoring this program to fit more closely the industry it purports to serve. In the process, we'll examine some of the considerations involved in modifying a lengthy program.

Beginning in March, we'll delve into advanced, sometimes esoteric aspects of the Apple Pascal language and operating system. If there are specific topics you would like to discuss, now is the time to suggest them! Your mail is not ignored; indeed, it served as the inspiration for the March column, so don't be shy!

The Schoolhouse Apple

by Jean Varven

There's a little of the artist in each of us, so they say. It just needs a little encouragement.

With that in mind, the current installment of Schoolhouse Apple is devoted to a colorful and longer than usual Logo tutorial. Join Jim Muller as he explores the world of recursion and the intricate patterns this procedure can be used to create.

Next month, educational software takes center stage. Some exciting and innovative programs have been released lately; we'll take a good look at a whole bunch of them.

A Schoolhouse Apple Tutorial

LOGO

JIM MULLER

Have you taught your turtle any good tessellations lately?

You know what tessellations are. They show up everywhere—in wallpaper, linoleum, tile, brick houses, and even in nature. When quilters painstakingly piece together cotton squares, triangles, and other geometric shapes to form intricate patterns, they are making tessellations. Take away the honey and bees from a honeycomb and the result is a natural tessellation of repeating hexagons. One of the deans at a Southwest Conference University who doodles during conferences and luncheons creates tessellations that evolve into elaborate works of art.

As you've probably determined by now, tessellations are patterns made by repeating the same geometric shape or combination of shapes again and again. Children can learn a great deal about the way geometric shapes fit together by carefully examining tessellations and by designing their own tessellations on graph paper or with cut-out shapes that can be manipulated.

Imaginative young Logo users will have fun designing their own tessellations. They'll enjoy seeing, by means of off-computer activities, how shapes fit together to make beautiful repeating designs. But tessellations using Logo are not exactly child's play. It takes a more experienced Logoite to translate these kinds of designs into turtle graphics.

The challenge of teaching the turtle to draw an elaborate tessellation requires imagination and an understanding of both conditional statements and recursion. In fact, creating tessellations provides a great way to experiment with the power of recursion, an important aspect of Logo.

Recursion is the ability of a procedure to "call itself." In its simplest usage, recursion can cause a procedure to call itself "forever" or until it is stopped manually. Here's an example of a simple recursive procedure.

```
TO SPIRAL :S :A
FD :S
RT :A
SPIRAL :S+3 :A
END
```

This procedure will continue to run until the programmer stops it.

Adding a conditional IF-THEN statement is one way of building a stop into the procedure.

MIT Logo	Apple Logo
TO SPIRAL :S :A	TO SPIRAL :S :A
IF :S > 150 STOP	IF :S > 150 [STOP]
FD :S	FD :S
RT :A	RT :A
SPIRAL :S+3 :A	SPIRAL :S+3 :A
END	END

The rest of the examples will be given in Apple Logo. If you're using MIT Logo, remove the brackets in the conditional statements.

Conditional statements enable us to stop a procedure in a variety of ways. For example, when the turtle reaches a certain X or Y coordinate, we can use IF :XCOR = 100 [STOP] to stop a procedure; when a procedure has been repeated a certain number of times, we can use IF :TIMES = 10 [STOP] to stop it; and when the turtle faces a certain direction, (IF :HEADING = 180 [STOP]) does the trick. Using conditional statements allows us to repeat a shape or design, to stop, to move the turtle, or to continue drawing. In other words, using conditional statements gives us the power to draw tessellations.

The inspiration for a couple of the designs included in this article came from *Trianglepoint*, a needlework book by Sherlee Lantz. Lantz has studied the use of triangles and hexagons in artwork over the centuries. What we're about to do is participate in an ancient art form using a very modern tool! The result of our efforts will look like figure 1.

As usual, we'll break down what we're going to do into its simplest form. Thus, our first task is simply to define a triangle with a variable.

```
TO TRI :N
REPEAT 3 [FD :N RT 120]
END
```

By studying the result, we discover that a triangular design or unit is

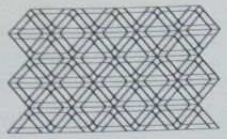


Figure 1: LATTICE

being repeated over and over, with one triangle pointing up, the next pointing down, and so on in series. Our next task is to define a procedure for that unit. See figure 2.

```
TO TRIS
TRI 50
TRI 40
FD 10 TRI 40
BK 10 RT 60 FD 10 LT 60
TRI 40

RT 60 BK 10 LT 60
END
```

Figure 2: TRIS

The next step is to define a procedure for putting several TRIS in a row. Since they have to alternate between pointing up and pointing down, we'll have to define two moves for the turtle to make between TRIS. See figure 3.

```
TO ROW :T
IF :T = 0 [STOP]
TRIS
MOVE1
TRIS
MOVE2
ROW :T - 1
END

TO MOVE1
LT 60
END

TO MOVE2
FD 50 RT 60
END
```

Figure 3: ROW

ROW contains a conditional statement that tells the turtle to stop after repeating the procedure a prescribed number of times. For example, if the command ROW 4 is given, the turtle carries out the procedure four times. The recursive line, ROW :T − 1, calls the procedure ROW 4 − 1 or ROW 3. The next time it becomes ROW 2, then ROW 1, then ROW 0. The test line, IF :T = 0 [STOP], causes the procedure to stop. (The turtle reads that line every time it repeats the procedure, but as long as T is not 0 the turtle keeps going.)

Here's the procedure we'll use so that the row can run horizontally on the screen and so that an odd number of TRIS can be included. See figure 4.

```
TO TRIROW :T
RT 30
TRIS
MOVE1
ROW :T
END
```

Figure 4: TRIROW

Now all that remains is to figure out a way to repeat TRIROW enough times to fill up the screen. A START procedure like the one that follows will position the turtle in the upper right-hand corner of the screen.

```
TO START
PU SETX 80
SETY 70 PD
END
```

Two more move procedures can be defined to position the turtle to draw one TRIROW from left to right and one from right to left. Experimentation reveals that five TRIROWs seem to make a more esthetically pleasing pattern than four, so a TRIROW has been included in the conditional statement. Otherwise, because of the need for two different move procedures, the LATTICE procedure will always draw an even number of TRIROWs.

```
TO LATTICE :N
IF :N = 0 [TRIROW 4 STOP]
TRIROW 4
MOVE3
TRIROW 4
MOVE4
LATTICE :N - 1
END

TO MOVE3
RT 120 FD 50 RT 90
END

TO MOVE4
PU RT 60 FD 100 RT 150 PD
END
```

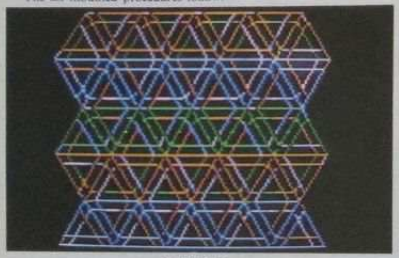
One last procedure positions the turtle and calls the tessellation:

```
TO CREATE
HT
START
LATTICE 2
END
```

As with everything else in Logo, there's something more you can do using the procedures we've been working with. As you look at how LATTICE covers the screen, you may begin to wonder if there's any way of adding color in a more dramatic way. Instead of just changing the background and drawing with a different color pen, wouldn't it be great to be able to change the color throughout the design, making it resemble a rug of multicolored yarn?

Because the turtle retraces its steps so many times, this is no easy task. One way of approaching things would be to use a random command with conditionals, but we want more control so we can be sure that color changes throughout the procedure. If we use a random command, it's possible that we will end up with blocks of color (look at QUILT). By changing the colors on all of the different levels of the procedure and by using the conditional controls, we can create a "controlled form of randomization" so that the colors will keep changing.

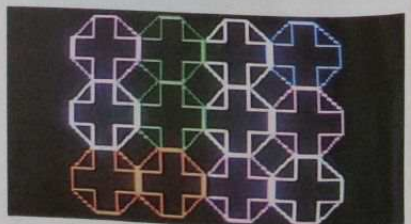
The six modified procedures follow.



LATTICE


```
TO CREATE :C
HT
START
LATTICE 2 :C
END

TO LATTICE :N :C
FULLSCREEN
IF :N = 0 [TRIROW 4 :C STOP]
TRIROW 4 :C
MOVE3
TRIROW 4 :C + 2
IF :C > 5 [MAKE "C :C - 4]
MOVE4
LATTICE :N - 1 :C + 3
IF :C > 5 [MAKE "C :C - 3]
END

TO ROW :T :C
IF :T = 0 [STOP]
TRIS :C
MOVE1
TRIS :C
MOVE2
ROW :T - 1 :C
END

TO TRIS :C
IF :C > 5 [MAKE "C :C - 5]
TRI 50 :C
TRI 40 :C + 1
FD 10 TRI 40 :C + 2
BK 10 RT 60 FD 10 LT 60
TRI 40 :C + 3
RT 60 BK 10 LT 60
END

TO TRIROW :T :C
RT 30
TRIS :C
MAKE "C :C + 1
IF :C > 5 [MAKE "C :C - 1]
MOVE1
ROW :T :C
END

TO TRI :N :C
IF :C > 5 [MAKE "C :C - 4]
SETPC :C
REPEAT 3 [FD :N RT 120]
END
```

The listings for three other programs—QUILT, WHIRLS, and TOWERS—are included here as well. Notice that QUILT uses a random method for adding color.



QUILT

```
TO QUILT :X
IF :X = 0 [STOP]
CROSSTOWER 3
DOWN
CROSSTOWER 3
UP
QUILT :X - 1
END
```

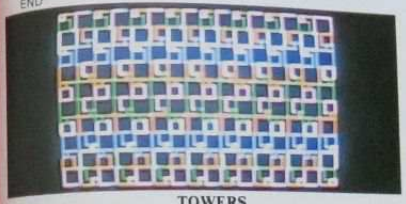


```
TO BEGIN
FULLSCREEN
START
QUILT 2
END

TO CROSSTOWER :T
IF :T = 0 [STOP]
COLOR
CROSS
AROUND
PU FD 40 PD
CROSSTOWER :T - 1
END

TO DOWN
PU BK 40 RT 90 FD 80 RT 90 FD 40 PD
END

TO UP
PU BK 40 LT 90 FD 40 LT 90 FD 40 PD
END

TO CROSS
REPEAT 4 [SIDE LT 90]
END

TO MOVE2
RT 48 FD 20
END

TO MOVE1
RT 42 FD 29
END

TO AROUND
FD 20 RT 90 FD 20
REPEAT 3 [MOVE1 MOVE2]
MOVE1
LT 42
END

TO SIDE
REPEAT 2 [FD 20 RT 90] FD 20
END

TO COLOR
SETPC 1 + RANDOM 5
END

TO START
PU
SETX -100
SETY -30
PD
END
```

WHIRLS

```
TO START
PU
SETX 100
SETY -70
PD
END

TO TRI :S
```

```
REPEAT 3 [FD :S RT 120]
END
TO WHIRL :S
REPEAT 6 [TRI :S RT 60 FD 12]
END
TO WHIRLS :S :T :C
FULLSCREEN
SETPC :C
IF :T = 0 [STOP]
RT 30
WHIRL :S
MAKE "C :C + 1
IF :C > 5 [MAKE "C :C - 5]
PU LT 30 FD :S * 2.3 LT 90 FD :S / 2 RT 90 PD
WHIRLS :S :T - 1 :C
END
```



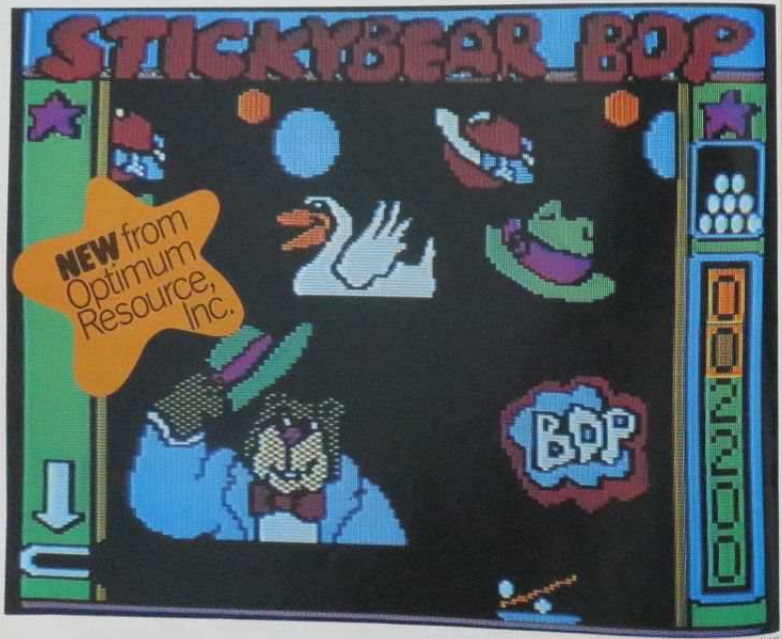
TOWERS

```
TO TESSELLATION
CLEARTEXT
PRINT [ENTER 'GETGOING AND A COLOR]
PRINT [ ]
PRINT [NUMBER TO SEE THE PROCEDURE.]
END

TO GETGOING :C
PU LT 90 FD 130 RT 90 BK 50 PD
COVER 8 :C
END

TO COVER :X :C
IF :X = 0 [STOP]
TOWER 15.5 :C
MOVE1
TOWER 15.5 :C
MOVE2
COVER :X - 1 :C
END

TO TOWER :S :T :C
IF :T = 0 [STOP]
SQUARES :S :C
MAKE "C :C + 2
IF :C > 5 [MAKE "C :C - 5]
TOWER :S :T - 1 :C
END

TO FRAME
PU LT 90 FD 100 RT 90 BK 60 PD
REPEAT 4 [TOWER 15.5 1 RT 90]
HT
END

TO SQUARES :S :C
IF :S < 0 [STOP]
SETPC :C
REPEAT 4 [FD :S RT 90]
FD :S
MAKE "C :C + 1
IF :C > 5 [MAKE "C :C - 5]
SQUARES :S - 5 :C
END

TO MOVE2
RT 90 FD 15 BK 15 RT 90
END

TO MOVE1
RT 90 FD 30 RT 90
END
```

The Young Peoples' Logo Association has started a collection of tessellations. Send in a tessellation on a disk and you'll get your disk back completely tesselated—that is, full of tessellation procedures. Send your tessellation to Softalk Tess, Box 60, North Hollywood, CA 91603.

WHAT YOU SEE IS WHAT YOU GET.

Introducing the most advanced computer animation ever created for the Apple®!



Actual Reproduction

"They told us that graphic animation this sophisticated can't be done."

– Richard Hefter, creator of STICKYBEAR and Old Ironsides and world-famous author and illustrator.

It took 2½ years in testing and development, but we've finally done it! Announcing a breakthrough in color graphics that will put the most advanced computer animation at your fingertips!

SEE FOR YOURSELF.

What you see on this page and on the lively eye-catching packaging, is exactly what you get in these four bright, bouncy new programs developed by Optimum Resource, Inc. and distributed by Xerox Education Publications/Weekly Reader.

Unlike other programs, where the pictures on the packaging bear no resemblance to the screen images, these programs deliver precisely what's promised!

HERE'S WHAT YOU GET...

- Bright, interest-grabbing packaging made of durable, wipe-clean vinyl. Easy to store... attractive to display!
- STICKYBEAR™ – the first animated character created exclusively for your Apple® by a popular children's illustrator!
- Hi-resolution graphics never before seen on an Apple®! And, for the first time, shapes move freely over color backgrounds!
- And each program is packed with full-screen, full-color pictures!

You won't be disappointed when you see these programs or the packaging. We guarantee it.

Look for STICKYBEAR BOP, STICKYBEAR ABC, STICKYBEAR Numbers and Old Ironsides programs in finer computer stores everywhere. Dealers are invited to inquire by calling toll-free 1-800-852-5000.

If there is no store near you, Visa and MasterCard holders may order by calling toll-free 1-800-852-5000. Or, send a check or money order for $39.95 for each program, plus $2.00 for shipping and handling (and please add state sales tax). Mail to Xerox Education Publications/Weekly Reader, Dept. 1-A, 245 Long Hill Road, Middletown, CT 06457.

Apple II and Apple II Plus 48K 3.3 DOS

Apple and Apple ® are registered trademarks of Apple Computer, Inc.

Stickybear Bop
Three animated shooting galleries of STICKYBEARS, ducks, planets, balloons, more...to knock off the screen. Each round requires more skill. FREE game, poster and stickers. For ages 3 to 99.
By Richard Hefter. Program by Jack Rice.

Stickybear ABC
Big, full-screen full-color moving pictures, with sound, represent each letter in the alphabet. FREE poster, book and stickers. For ages 3 to 6.
Pictures by Richard Hefter. Program by Jamie and Steve Worthington and Spencer Howe.

Stickybear Numbers
Colorful groups of big moving objects – trucks, ducks, planes, more – teach numbers, counting and simple arithmetic. FREE poster, book and stickers. For ages 3 to 6. Pictures by Richard Hefter. Program by Jamie and Steve Worthington.

PLUS, FOR THE WHOLE FAMILY...

Old Ironsides
This colorful, new 2-player naval battle features animated sailing ships firing broadsides at each other. Tactics include ramming, shooting out the opponent's ammunition and hiding in fog... while contending with wind direction, clouds obscuring your vision, more! FREE poster. For ages 8 to 99.
By Richard Hefter and Jack Rice.

Distributed by
Xerox Education Publications
Weekly Reader
Computer Software Division

San Francisco Applefest

Apple's Party Comes Home

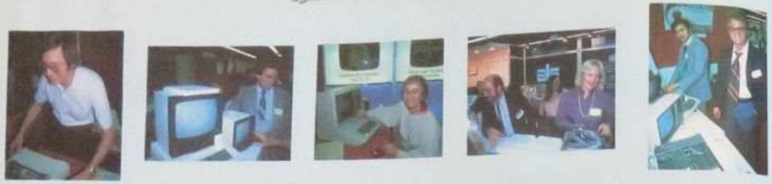
First row, left to right: Of course, Apple's booth was the center of attention, which must have pleased Mike Markkula, who toured the show Sunday. Many of the most exciting new products were hardware: Andrew Najda and Stanley Bialek of Number Nine unveiled an ultra hi-res board and a faster 6502 on a memory expansion board; attendee Richard McCarnan was attracted by the detached keyboard offered by Jim Cook of Executive Peripheral Systems; Gary Barret showed Touch Technology's system for touch control of the Apple; Eric Moody of Brem Enterprises also had a 6502 on a board, designed for background communications processing or for print buffering; Davong Systems unwrapped a five-megabyte "universal" hard disk; Jay Weiss had all the software publishers interested in his Mockingboard speech peripheral at the Sweet Micro Systems booth; Carl Baker of Micron Technology stunned the hackers with a real-time digitizer; Diana Kehrig of Northwest Instrument Systems had a line of specialized peripherals including the aScope that turns the Apple into an oscilloscope.

Second row: Eden Recor and Tim Gill of Quark, here visited by Carole Ransome of Quest Computers, enthralled Apple III owners with the Catalyst; Bob Vessas, George Otis, and Audrey Warner displayed Goci Software's line of accounting packages; Sonya Sanchez showed a line of surge suppressors from Panamax; Wanda Volpe and David Guest of Beaman Porter demonstrated *PowerText*; Carrie Levine of Software Publishing plumped the company's PFS line; Ken Emsl had a new *VisiCalc* utility from Formula 1 Computers; Harvey Friedman of Paypac introduced a mixed bag of business software; Mike Fails of MBA Software entranced real estate investors with a creative financing package; Dave Wertzberger of SSM boosted the *Transend* communications package.

Third row: Ron Allord plays a Symtaun at the Mountain booth; Teddi Cook repped Novation's modems; Joanne Edwards oversaw the Ditech booth, where plotters were the center of attention; Roy Hicks of RH Electronics had fans galore; Cameron Rose of Advanced Business Technology had a point-of-sale inventory system to display; Bruce Janiszewski touted Saturn's line of memory expansion boards; Frank McNeil of Chatsworth Data attracted attention with an optical card reading system; Arthur Roberts of Selectone offered a new solution to letter-quality printing; Cybele and Peter Benkowsky of Olive Branch also addressed the letter-quality print market; Erik Gregersen of Strobe showed a graphics plotter; Don Schuman of Synetix wowed them with a disk emulator; Tom Porett and Bill Wolff of Interactive Structures showed PKASO and tantalized guests with news of Pipeline.

Fourth row: Tom Koves displayed a line of Apple furniture from Universal Industries; Dan Hartman of PageMate proffered a nifty appurtenance for word processing; Apple Center furniture from Doss Industries; an installed Disk Lock, obviously a data security device, from Orange County Technology; the Savvy booth attracted strong traffic; the Voice Box system from Multi-Tech; memory expansion cards to the sixth power from Legend Industries; Lord British finally had an Ultima II to show; programmers Randy Simon of Don't Ask Software and Jim Nitchals compare notes; "The Unknown Programmer," aka Don Fudge, attracted attention to camera-shy Mary Carol Smith by reposing on her lap at the Avant-Garde booth.

Fifth row: Educational software folks were in evidence in large numbers: Chuck Petts III, no relation to Apple III, showed the Software Productions offering, Micro Mother Goose; Sarah Olson of Salinas got to play *Bop-A-Bet* under the tutelage of Margaret Lowe of Sunnyside Soft; Curt Steele of Control Data showed the much respected PLATO system; Larry Bergsma and Jim Thompson of Cardinal Software introed the MicroMentor learning system; George Smith tested the CyberLogo program from Cyberthonics International; Jim Pious repped the high-quality line from SRA; Mike Harstein of Software Connections had a slightly different offering—software for local area networking in the classroom environment.



First row: Dan Dresselhaus of Dresselhaus Computer Products showed folks how to get more out of an Epson printer; C. Scott Bradley of USI showed an orange phosphor monitor; Barbara Orput of Sun-Flex introduced a new light pen; Dick Ribas, Vicki Oliver, and Kit Rosenbloom of Advanced Logic Systems showed their integrated line of products; Abe Oswai and Ralph Sweetzer of Micromax touted a new Z-80 card and an eighty-column card; Richard Slade and Jim Hinds of Stellation Two had *Wizardry* running at double time with their Pascal speed-up kit; Dr. Stan Klein and Bob Arendt of Orange County Technology displayed a disk lock device; Pete Smith of Micro Control Systems wowed attendees with a 3-D graphics system; Richard Todd Miller and Clint O'Conner of Analytical Engines were prepared to supercharge Apples with 68000 coprocessors; Bill Adler and Andy Claire showed the Echo speech synthesis program from Street Electronics.

Second row: Shirley Reinhardt of Highlands Computers stands by with helpful hints as youngsters try to tackle *Creature Venture*; other youngsters gravitated toward Bill Budge's newest, *Pinball Construction Set*; the youth movement in programming was exemplified by the executive triad of Gil Beyda, 19, David Wilkin, 20, and Greg Segal, 19—Mind Games is their company and *Desecration* is their first product; Island Graphics teased showgoers with an Atari version of a soon-to-be-released Apple product; Ron Erikson and Sue Tolley emphasized fun at the Piccadilly booth; Chad and Bob Ruble of Software by H showed a new learning system; Mike and Ken Brown of Insoft wore several hats while repping the company's diverse line; Frank F. Frazier of Frobco tantalized the hackers with a VCS program development system; E. J. and F. Lee McFadden of FlipTrack boosted their training tapes; Bernie Roach and Sharon Falkowski touted Sentinel's line of magnetic media.

Third row: Accent Software's Herve VanCleff demonstrated the *Graphic Solution*; Ed Rosenzweig of Pterodactyl had a dandy debugger at a dandier price; Randy Hyde, author of *Lisa 2.5*, showed new products at his Ark Computing booth; Joe Eliason and Diana Smith of Gourmet Software catered to the recipe-hungry; Bill Wolff, Mark O'Deen, Robert Knoll, and Ron Metaker of Omega Microware numbered *Watson* and *The Inspector* among their wonders; Val Golding, editor of *Call—A.P.P.L.E.*, helped tend the A.P.P.L.E. booth along with club prexy Dick Hubert and Kevin Donohoe; Cindy Clark, Lynn Wood, and Sue Rowland repped *Softside*; Hartley G. Lesser and Hope Currier pumped up interest in *inCider*; Earth Carol of Oakland got hands-on time under the supervision of Carol Stone in the Follett Library Book Company booth; Susan Johnson, Scott Evans, Barry Evans, and Brad Evans manned the Snave Systems booth.

Fourth row: Preston Nall of Ashton-Tate promoted the wonders of *dBase II*; Harold Gottlieb and Warren Shore of Dynatech Microsystems pushed C.O.R.P., software that writes software; real estate software from REM was the subject of discussion between customer Earl McFarland and company reps Terry Lush and William Olson; Alison and Pam Korites touted the Kern Publications line; Michael Dean, Gail Reilly, and Per Swendsen represented Infosoft; Ron West dressed the part for Professional and Medical Software; Dan Shafter, Dennis LaRue, and Thomas Simpson introduced a new statistics package from SoftCorp; Terry Bell, Jeff Coulter, Steve Cherry, and Jon Ranschoff displayed Starsoft's accounting packages; Dave Vernier repped his own software firm; Penny Globus of Computer Sight boosted a new Pascal.

Fifth row: Laird Kelly of Reactive Systems drew crowds with a real-time voter response system; Tod Chang, Susan Greiner, and John Miao of American Mitac Corporation boosted a disk drive alternative; the Renaissance booth attracted folks interested in buffered output and typewriter interfaces; Laurie Ann and Norman North of N Squared showed attendees how to analyze data ranging from stock market prices to nutrition; Ross Jimenez of Accupower exhibited a sturdier Apple power supply; Tim Chow of Micro Data Tek displayed a plethora of hardware devices; Michael Hardwick showed Eastside Software's board; Bill Halter of Lehigh Valley Computer showed a home security system based on the Apple; Rex Harper of Crabapple Systems showed the Fiddler, a device hooked to the game port that controlled a robot; Harry Sweeney of Expanding Space Software offered utility packages for the Apple III as well as the world's most complete manual on how to play *Star Trek* on the Apple II.

S·T·O·R·Y·T·A·L·K

F·I·C·T·I·O·N

THEY'RE HERE!

BY K.O. ECKLAND

It's not my desire to alarm anyone unnecessarily, but I do feel it imperative to apprise whomever I can of a situation that is becoming critical in nature. This may well be my last opportunity in communication. The realization that something was amiss came to me late one night. I had gone to bed and was lying there, idly mulling over events of the day, when I became aware of activity of some sort in the den. There was a low humming sound, punctuated by occasional mechanical clacking, which caused me to rise and put on a robe. Instinctively, I withdrew my trusty .065-caliber Lone Star water pistol from the bedside commode and stuffed it in a pocket.

From just outside the den door, the sounds were a bit louder. I grabbed the knob and threw open the door, drawing the water pistol and turning on the lights simultaneously—no easy feat for one who is used to, at most, two actions at any given moment.

"All right . . . what's going on in here?" I commanded in my best Victorian voice.

The den was quite empty. I poked randomly at the drapes, peered under the divan, even looked in a few drawers—which shows to what extremes one will go when investigating noises at night. The one thing I did notice was that I must have forgotten to turn off my Apple; a pale light glowed from below the shift key. I disregarded my oversight and snapped the switch. One more look around assured me that I'd heard some electronic switchery from the computer. Never fully understanding the inner mysteries of the thing, I allowed as how it could be fully capable of making internal noises of its own accord.

The whole incident would have been forgotten had it not been for a printed line I discovered the next morning while adjusting the Epson:

"We really must be more careful."

I didn't recall writing that the day before. And I didn't recall having to be careful of anything in particular. I dismissed the whole matter by removing the offending sheet of paper and discarding it.

Later, the same day, I was finishing up an incisive article for *PBX & Switchboard Operator's Monthly* when I became aware of a pronounced lag between my keystrokes and the printer. I pulled the sheet up to see what was happening and found there was an additional line of what appeared to be random gobbledygook, followed by an obscenity. I was lightly astonished. I didn't write that—*PBX&SOM* would never accept such language!

It crossed my mind that my Apple was—how shall we say?—going dotty on me. Perhaps Apples, like Honda Civics, had a required maintenance schedule to be adhered to, I thought. Sort of an electronic lube job, so to speak. Searching through the owner's manual provided nothing to suggest a 1,500-mile or 1,950,000-byte maintenance—whichever came first. However, on the assumption that I might have overlooked an instruction, I took it to my local Apple dealer and explained the symptoms. He accepted this with callous indifference (Apple dealers, I under-

stand, are required to complete a course in Callous Indifference), took it into the back room, returned several minutes later, and pronounced it well, considering the amount of pipe droppings under the keyboard.

But it happened again that afternoon. Another line of what, for want of better description, might be described as hash magically appeared printed midway in a letter to a friend. Since my friend was not a computer type, she would be hard pressed to understand what RR8M+sdd3P/bat5 represented in an otherwise cogent letter. I couldn't erase the hash, since it didn't appear on the screen, and a second printing of the letter showed that it was still there, so I ended up striking it out with a felt pen; messy looking at best. I spent that evening rereading the entire manual, looking for some instructions on hash removal to no avail.

Once more, a short article on the *Romance of Wallpaper* provided a framework for two more lines of hash, one on each page, both identical in content: DDI5$k/bat< 2.2dG. Retrieving a copy of the letter in subject, I compared the two hashes, finding them entirely dissimilar save for the word *bat* and the fact that both began with two capital letters.

My incredibly analytical mind leapt into action and in practically no time at all came up with nothing. After exhausting attempts at close logic, I tried entering the last hash in a complicated program that I devised from years of experience:

```
10  HOME
20  DDI5$k/bat < 2.2dG
30  END
```

Advanced programmers will recognize the deft hand of a master in that listing, but, for the neophytes among the readers, I must explain that I always start out with *home*, since it makes for a tidy screen. Fully prepared for my usual syntax error, I hit run. Nothing appeared on the screen, but the printer let off a short burst and the cursor winked its presence. I pulled out the paper and read:

"Not now. Later. 3DD."

I was well aware now that something peculiar was going on somewhere between the wall outlet and the head of the printer. This time, the whole works went into the dealership—Apple, printer, disk drive, even my game paddles. Two days later, accompanied by what I considered to be a disproportionate bill, the works were handed back to me with precious little ceremony and another bland diagnosis of good health—after they swept out the pipe droppings.

That same night, I discovered I had again left the Apple on and chided myself for a memory slowly turning to lime Jello. Yet, it gnawed on me that I really had turned the machine off. To mend this flaw in an otherwise perfect character, I devised a mnemonic system whereby I would place a shoe in the center of the bed, removing it only after ascertaining that all electronics were shut down for the night. An infallible plan, I complimented myself.

Then, the noise awoke me, this time in the early hours of the morning. With my Lone Star in hand, I again attacked the den, to discover nothing—except that the Apple was on!

I began dwelling on the problem to the point where my writing suffered more from that than it did from those rampant hash lines. I stopped shaving, drank more coffee than usual, quit watching the late news, and devoted more time to bursting into the den at odd moments, hoping to surprise someone—anyone. I even took to wearing my Lone Star in a concealed holster on my person at all times except, of course, during bathing, when it sat handy nearby. The hash ruined practically all attempts I made at communication with the outside world, including those stories and articles my editors were demanding. The random letters, numbers, and symbols increased in volume. The Apple shop began drawing their shades and putting up a "closed" sign when they saw me approaching. I was becoming a stranger to my friends, and even the cats took to avoiding me, except for around dinner time.

"I know there's someone in there," I spoke directly to the Apple. "Why are you doing this to me? Why don't you show yourself?" I even tried threatening it with my water pistol (knowing how fearful electronics are of raw water). Nothing produced any results. The hash kept coming. I tried different wall outlets, tried setting the disk drive on its side, even bought an expensive "glitch-kicker." Nothing.

One night, during a terrible thunderstorm, with the trees dramatically scraping the side of the house, the printer wrote on its own:

"Give up. There is no use resisting."

There was a flash of lightning and a boom of thunder. That really boggled my mind. I slept in the Honda that night.

The next day, a UPS truck pulled up and the driver traded my autograph for a small package. I opened it and there was a card clad in blister-wrap. No invoice, no shipping tag, no instructions, only a typed note:

"Install in slot 3."

I called the Apple shop, disguising my voice so they wouldn't hang up on me again, and asked if they had sent the card, but they denied any knowledge of it. They couldn't even tell what it was by my description or why it would go in slot 3. I went into the den and opened the lid of the computer and verified that slot 3 was vacant, but decided to delay installation. I glanced over at the Epson and there was a line printed:

"Install the card NOW."

"No," I said to it. "Not until I know more of what I'm doing."

I tried calling a friend to see if he might have sent it but the phone had gone dead. Apparently they were working on the lines after the recent storm. I noticed the clock had stopped as well. To verify the power outage, I flipped the light switch; it was quite dead. I don't know why, but on a wild hunch I went over and flicked the switch on the Apple. The light came on and it beeped as the DOS booted. I looked up at the clock; it was still stopped.

"All right, now cut that out!" I shouted at the thing. "I suppose you think you're being clever, whoever you are. . . ."

The printer clacked briefly:

"Imperative you install card NOW."

"You win. But I'm going to find out what's behind this if it kills me!" I didn't particularly like my choice of words. I jammed the card into the slot, not caring which way was front or back. "There! I hope it fries!"

Clack-clack-clack:

"Thank you."

The phone rang and I jumped a foot. It was the editor of *Sand & Gravel* asking where my article on aggregates was, as he was holding a spot open for it. This at least took my mind off the immediate problem, and I promised to have the story in the next day's mail.

I replaced the Apple's lid and flexed my fingers, trying hard to think of something exciting to write about aggregates. After a bit of thought I entered an opening sentence, then ran a test print out of curiosity. The hash was there and worse than ever:

"One tends to4R ove%rlook the na92tural wonders))dH:4 of aggr7?egates und#er our very ffffeet."

I walked away from the computer in absolute disgust. There was no way I could send copy like that to an important magazine! I went into the kitchen and turned on the coffeemaker, but it went "pffft!" and let out a cloud of blue smoke. At the same time, my cassette recorder began playing a tape backward. The phone rang and it was the editor of *Sand & Gravel* inquiring where my article on aggregates was.

"You just called me not fifteen minutes ago!" I shouted, and the phone went dead again. The clock was running twice as fast as usual and the horn in the Honda beeped.

Now, I'm frantically writin/g this not on the computer, but on my old Selectric that it replaced. I am hoping to warn all of you to be aware that something is going on; all is not right with the world. I only ho5pe that I can get this into the mail to some magazine before t7hing%s get any worse. I have no answers, no > solutions. Maybe so8BDmeone out there d##oes. SoRRmeb9ody might<<.9rR try calliYD:*ng the AaPpPpLIEe peo:::ple uP in C/u/p/e/r/t/i/n/o and TELLthemabout &&this. My 2ph5o;;ne doKKKKe3s not w+++++or++kanymor e : the damn c9ar won't sSstarRrt &and& DDfl5$k/bat<2:2dG CHR$(328)

Editor's note: Those readers who have sent letters to us for forwarding to Mr. Eckland should be informed that his mail keep3s being returned. Apparently he has moved and left no for4warding address6) an&d has no=t noti"fieD 0 this offFFice of a Nnew m(ail+in/g 44 ;add7re/ss wher—(0).e he 998(RR$M/bat6 >> <<< 0=b //v

■

Softalk Presents The Bestsellers

Cash registers hummed merrily in computer stores all over America in November as Apple computer owners geared up for what looked to be a giant Christmas season. In so doing, they reasserted the place of entertainment software in the Apple universe and helped *Choplifter* make history.

The emphasis on Christmas buying was indicated by the resurgence of games in the Top Thirty. Where only six arcade games made that list in October, ten earned mention in November. Chief beneficiaries of the holiday buying spirit were *Snack Attack*, which rose from twenty-ninth to ninth, and *Castle Wolfenstein*, which had dropped out of the Top Thirty but made it all the way back to eleventh.

The biggest shocker of all, however, was *Choplifter*. After several months as the number one program, *Choplifter* was deposed by all-time champ *VisiCalc* in October. Judging by historical sales patterns, that seemed to indicate the beginning of the end for Dan Gorlin's piece de resistance. No game had ever dropped out of first place and regained it.

But that's exactly what *Choplifter* did, narrowly edging out *VisiCalc* for the top spot. There was no weakness in *VisiCalc*'s sales. It was the reemphasis on entertainment that changed the market.

Frogger jumped from sixteenth to fifth. *Canyon Climber*, *Aztec*, and *The Mask of the Sun* came from nowhere to fourteenth, fifteenth, and sixteenth, respectively. *Zork III* debuted on the charts in twentieth. *Serpentine* regained a spot on the Top Thirty in twenty-first. *Tubeway* snuck into thirtieth in its first month of release.

Preliminary indications were that December would see more of the same. *Lunar Leeper* and *Repton* started out strong among the arcade freshmen and *Ultima II*, at last being sold, was white hot among fantasy games. *Transylvania* and *Sherwood Forest* were showing strength among the adventures. And previewers were raving about Bill Budge's *Pinball Construction Set* and Micro Fun's *Miner 2049er*.

The large number of titles in wide distribution seems to indicate that a good time will be had by all.

Nearly twice as much software was sold in November as in October, and not all of it was entertainment. What might best be called soft education packages are making a big showing. Soft education packages are those that sugar-coat the learning process, generally are not curriculum-based, and are dependent more on home buyers than on school adoptions for sales.

This fall has seen the Learning Company and Spinnaker make big splashes with soft education product. They're doing real well, but the talk of the town was the set of Children's Television Workshop packages distributed by Apple. CTW produces *Sesame Street* and it looks like they've found another medium through which to communicate to youngsters.

In recognition of the changing circumstances in educational software, *Softalk* has implemented a Home Education 10 category. The cate-

Home Education 10

1. **MasterType,** Bruce Zweig, Lightning Software
2. **Early Games for Young Children,** John Paulson, Early Game Company
3. **Apple Logo,** Apple Computer
4. **Typing Tutor,** Image Producers, Microsoft
5. **Snooper Troops I,** Tom Snyder, Spinnaker Software
6. **Facemaker,** DesignWare, Spinnaker Software
7. **Step By Step,** John Victor, Program Design Inc.
8. **Ernie's Quiz,** Children's Television Workshop, Apple Computer
9. **Snooper Troops II,** Tom Snyder, Spinnaker Software
10. **Mix & Match,** Children's Television Workshop, Apple Computer

Arcade 10

1. **Choplifter,** Dan Gorlin, Broderbund Software
2. **Frogger,** Olaf Lubeck, Sierra On-Line
3. **Snack Attack,** Dan Illowsky, DataMost
4. **The Arcade Machine,** Chris Jochumson and Doug Carlston, Broderbund Software
5. **Canyon Climber,** Steve Bjork, DataSoft
6. **Aztec,** Paul Stephenson, DataMost
7. **Cannonball Blitz,** Olaf Lubeck, Sierra On-Line
8. **Serpentine,** David Snider, Broderbund Software
9. **Star Blazer,** Tony Suzuki, Broderbund Software
10. **Tubeway,** David Arthur Van Brink, DataMost

gory is not all-encompassing. Curriculum-based software is customarily not sold through retail outlets, but through specialized educational distributors. These sales cannot be tracked and verified and are therefore not included. It is well within the realm of reason to believe that more of, say, *Algebra 3* from Edu-Ware is sold than some of the packages listed.

The first education packages were nearly all curriculum-based. But the lack of curriculum-based software should not reflect poorly on the early education publishers. It was their pioneering efforts that parents used as a justification to buy a computer in the first place—even if they did buy *Dogfight* instead of *Decimals* and *Space Eggs* instead of *Spatial Relationships*.

It should come as no surprise that *MasterType* leads the education category. Biggest eyebrow-raiser is *Early Games for Young Children*, which outsold Apple Logo and almost made the Top Thirty. One of the old guard publishers made the list as *Step By Step* from PDI took position.

Three of Spinnaker's offerings earned spots on the list and two CTW packages also made it. The higher price of the Learning Company packages apparently prevents them from being as widespread, but *Rocky's Boots* made a strong showing.

Also new this month is the Arcade 10. After two straight months in which less than ten arcade games made the Top Thirty, it was decided to implement an arcade list to present more depth. Naturally, ten such programs made the Top Thirty this month, so the list has all the value of a second appendix.

Apple III

This Month	Last Month	
1.	1.	**VisiCalc: Advanced Version,** Software Arts/Dan Bricklin and Robert Frankston, VisiCorp
2.	4.	**PFS: File,** John Page and D. D. Roberts, Software Publishing Corporation
3.	3.	**Apple Writer III,** Paul Lutus, Apple Computer
4.	—	**Word Juggler,** Tim Gill, Quark Engineering
5.	2.	**VisiCalc,** Software Arts/Dan Bricklin and Robert Frankston, VisiCorp
	9.	**General Ledger,** George Shackelford, State of the Art Inc.
7.	10.	**QuickFile,** Apple Computer
8.	—	**PFS: Report,** John Page, Software Publishing Corporation
9.	5.	**Access III,** Apple Computer
10.	—	**Micro Terminal III,** Microcom
	—	**Great Plains Hardisk Accounting Series,** Great Plains Software

Word Processors 10

This Month	Last Month	
1.	1.	**Screen Writer II,** David Kidwell, Sierra On-Line
2.	3.	**Word Handler,** Leonard Elekman, Silicon Valley Systems
3.	2.	**Apple Writer II,** Paul Lutus, Apple Computer
4.	4.	**WordStar,** MicroPro
5.	10.	**Magic Window II,** Bill Depew, Artsci
6.	9.	**PIE Writer,** Softwest, Hayden
	—	**Format II,** Kensington Microware
8.	7.	**Super-Text 40-56-70,** Ed Zaron, Muse
	—	**Sensible Speller,** Sensible Software
10.	5.	**Apple Writer II Pre-Boot Disk,** Kevin Armstrong and Mark Borgerson, Videx

In the Apple III list, *Word Juggler* made a leap into fourth and showed much broader strength than ever before. In addition, Quark's *Catalyst* just missed the cut. That the III is being used in big companies to talk to big computers is indicated by the increasing sales of Microcom's *Micro Terminal III*.

For the most part, Apple II business programs underwent only minor shuffling of order, although many dropped out of the Top Thirty. The interesting development was Microplan's appearance in ninth as Microsoft flexes its marketing muscle against VisiCorp for the electronic spreadsheet business. It's a long way from ninth to first, but the situation should be of continued interest.

Among word processors, *Word Handler* continued what now appears to be an inexorable drive for the preeminent position. After two months in third place, the program passed *Apple Writer II* and edged closer to *Screen Writer II*. That means that the two bestselling word processors do not require eighty-column boards. To enlarge upon that thought, however, *Apple Writer II* is continuing as a premium on some Apple system sales and there actually were twice as many *Apple Writers* as *Screen Writers* and *Word Handlers* combined put into the hands of users in November. Which, of course, explains the continuing strength of Videx's *Apple Writer II Pre-Boot* disk.

The Home 10 got a shakeup because the typing programs were removed from that list and placed in education. *Home Accountant* continues to dominate. SSM's marketing ploy of bundling a Source mem-

Strategy 5

This Month	Last Month	
1.	1.	**Castle Wolfenstein,** Silas Warner, Muse
2.	3.	**Flight Simulator,** Bruce Artwick, SubLogic
3.	—	**Space Vikings,** Mitchell Robbins, SubLogic
4.	—	**Sargon II,** Dan and Kathe Spracklen, Hayden
5.	—	**Spitfire Simulator,** Ted Kurtz, Mind Systems

Adventure 5

This Month	Last Month	
1.	—	**The Mask of the Sun,** Chris Anson, Alan Clark, Larry Franks, and Margaret Anson, Ultrasoft
2.	—	**Zork III,** Infocom
3.	1.	**Starcross,** Infocom
4.	2.	**Zork I,** Infocom
5.	3.	**Deadline,** Infocom

Fantasy 5

This Month	Last Month	
1.	1.	**Wizardry,** Andrew Greenberg and Robert Woodhead, Sir-tech
2.	2.	**Knight of Diamonds,** Andrew Greenberg and Robert Woodhead, Sir-tech
3.	—	**Apventure to Atlantis,** Bob Clardy, Synergistic Software
	—	**Ali Baba and the Forty Thieves,** Stuart Smith, Quality Software
5.	5.	**Prisoner 2,** David Mullich, Edu-Ware Services

Home 10

This Month	Last Month	
1.	1.	**Home Accountant,** Bob Schoenburg, Larry Grodin, and Steve Pollack, Continental Software
2.	4.	**Transend I,** Tim Dygert and Bob Kniskern, SSM
3.	4.	**Transend II,** Tim Dygert and Bob Kniskern, SSM
4.	10.	**Data Capture 4.0,** David Hughes and George McClelland, Southeastern Software
5.	—	**VisiTerm,** Tom Keith, VisiCorp
6.	4.	**Personal Finance Manager,** Jeffrey Gold, Apple Computer
7.	6.	**ASCII Express: The Professional,** Bill Blue and Mark Robbins, Southwestern Data Systems
8.	—	**Hayes Terminal Program,** Hayes Microcomputer Products
9.	—	**Dow Jones Market Analyzer,** B. C. Burch, RTR Software
10.	—	**Electric Duet,** Paul Lutus, Insoft

Hobby 10

This Month	Last Month	
1.	4.	**Bag of Tricks,** Don Worth and Pieter Lechner, Quality Software
2.	1.	**Graphics Magician,** Chris Jochumson, David Lubar, and Mark Pelczarski, Penguin Software
3.	9.	**DOS Boss,** Bert Kersey and Jack Cassidy, Beagle Bros
4.	7.	**Zoom Grafix,** Dav Holle, Phoenix Software
5.	—	**Utility City,** Bert Kersey, Beagle Bros
6.	—	**Locksmith 4.0,** Omega Microware
7.	2.	**The Complete Graphics System,** Mark Pelczarski, Penguin Software
	3.	**Apple Mechanic,** Bert Kersey, Beagle Bros
9.	—	**DOS Tool Kit,** Apple Computer
10.	—	**Nibbles Away II,** Micro-Ware Distributing Inc.

Business 10

This Month	Last Month	
1.	1.	**VisiCalc,** Software Arts/Dan Bricklin and Robert Frankston, VisiCorp
2.	2.	**PFS: File,** John Page and D. D. Roberts, Software Publishing Corporation
3.	3.	**DB Master,** Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
4.	10.	**General Ledger,** George Shackelford, State of the Art Inc.
5.	7.	**BPI General Ledger,** John Moss and Ken Debower, Apple Computer
6.	6.	**VisiTrend/VisiPlot,** Micro Finance Systems/Mitch Kapor, VisiCorp
7.	4.	**VisiFile,** Creative Computer Applications/Colin Jameson and Ben Herman, VisiCorp
	9.	**PFS: Graph,** Bessie Chin and Stephen Hill, Software Publishing Corporation
9.	—	**Multiplan,** Microsoft
10.	—	**VersaForm,** Joe Landau, Applied Software Technology

Softalk Presents The Bestsellers

bership with their communications software has been so successful that they got both second and third.

Bag of Tricks regained the number one spot in the Hobby 10, but mostly that list remains the private preserve of Beagle Bros and Penguin. Beagle Bros placed three programs and Penguin had two entries on the chart.

Three new programs made the Strategy 5 chart. *Space Vikings* scored third, *Sargon II* rejoined the list in fourth, and *Spitfire Simulator* nailed down fifth. *Castle Wolfenstein* put some distance between it and the challengers.

Wizardry and *Knight of Diamonds* remained atop the Fantasy 5, but *Ultima II* looks to challenge there in December. *Ali Baba and the Forty Thieves* and *Aventure to Atlantis* regained the list and tied for third.

And then there's Infocom. Last month they got five of the first six spots in the Adventure 5, losing only fifth place to *Escape from Rungistan*. This month they also took five of the first six spots, but *The Mask of the Sun* took number one, primarily because of Ultrasoft's introduction of new technology in screen swapping that has adventure fans agog.

The Zork family will get another chance in December, as entertainment buying reaches its peak. If you don't believe second chances, just check *Choplifter*. ■

Apple-franchised retail stores representing approximately 5.8 percent of all sales of Apple and Apple-related products volunteered to participate in the poll.

Respondents were contacted early in December to ascertain their sales for the month of November.

The only criterion for inclusion on the list was the number of units sold—such other criteria as quality of product, profitability to the computer store, and personal preference of the individual respondents were not considered.

Respondents in December represented every geographical area of the continental United States.

Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only for the month in which they are printed; readers cannot assume that an index rating of 50 in one month represents equivalent sales to an index number of 50 in another month.

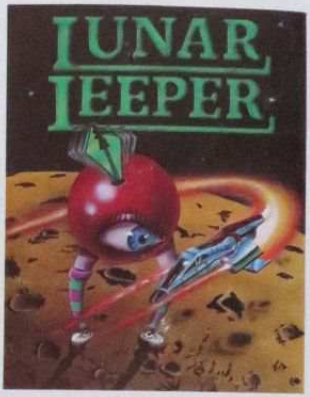
Probability of statistical error is plus or minus 4.32 percent, which translates roughly into the theoretical possibility of a change of 4.93 points, plus or minus, in any index number.

The Top Thirty

This Month	Last Month	Index	
1.	2.	123.61	**Choplifter,** Dan Gorlin, Broderbund Software
2.	1.	116.73	**VisiCalc,** Software Arts/Dan Bricklin and Robert Frankston, VisiCorp
3.	4.	99.17	**Home Accountant,** Bob Schoenburg, Larry Grodin, and Steve Pollack, Continental Software
4.	3.	95.72	**PFS: File,** John Page and D. D. Roberts. Software Publishing Corporation
5.	16.	82.29	**Frogger,** Olaf Lubeck, Sierra On-Line
6.	7.	63.35	**Wizardry,** Andrew Greenberg and Robert Woodhead, Sir-tech
7.	5.	58.54	**Screen Writer II,** David Kidwell, Sierra On-Line
8.	15.	50.27	**Word Handler,** Leonard Elekman, Silicon Valley Systems
9.	29.	35.12	**Snack Attack,** Dan Illowsky, DataMost
10.	6.	34.09	**Apple Writer II,** Paul Lutus, Apple Computer
11.	—	33.74	**Castle Wolfenstein,** Silas Warner, Muse
12.	9.	32.71	**DB Master,** Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
13.	8.	28.58	**The Arcade Machine,** Chris Jochumson and Doug Carlston, Broderbund Software
14.	—	28.24	**Canyon Climber,** Steve Bjork, DataSoft
15.	—	25.48	**Aztec,** Paul Stephenson, DataMost
16.	—	24.79	**The Mask of the Sun,** Chris Anson, Alan Clark, Larry Franks, and Margaret Anson, Ultrasoft
17.	14.	24.45	**Cannonball Blitz,** Olaf Lubeck, Sierra On-Line
18.	29.	23.76	**Transend I,** Tim Dygert and Bob Kniskern, SSM
19.	16.	21.69	**Knight of Diamonds,** Andrew Greenberg and Robert Woodhead, Sir-tech
20.	—	19.63	**Zork III,** Infocom
21.	10.	18.94	**Starcross,** Infocom
	—	18.94	**Serpentine,** David Snider, Broderbund Software
23.	20.	18.25	**Zork I,** Infocom
	—	18.25	**Bag of Tricks,** Don Worth and Pieter Lechner, Quality Software
	22.	18.25	**MasterType,** Bruce Zweig, Lightning Software
26.	—	17.56	**General Ledger,** George Shackelford, State of the Art Inc.
	18.	17.56	**Star Blazer,** Tony Suzuki, Broderbund Software
	25.	17.56	**Deadline,** Infocom
29.	23.	16.87	**BPI General Ledger,** John Moss and Ken Debower, Apple Computer
30.	—	16.53	**Tubeway,** David Arthur Van Brink, DataMost

JEEPERS CREEPERS!

DON'T KILL THEM LEEPERS!



Them leepers is cute.
Them leepers ain't kind.
Them leepers will jump up
and bite your behind.

Your job is to fly through
them leepers galore
to rescue some men
is the reason what for.

So be a good pilot
be quick and be sly
Don't kill the leepers
but shoot up the eye

Save all the men
and avoid all the leepers
fly through a cave
and shoot the lunar leepers keepers

If you want to know
what this silly poem's for
check out LUNAR LEEPERS
at your local computer store

Now on the ATARI 800
40K Disk

Lunar Leepers is a funtastic new APPLE arcade game from CHUCKLES, creator of Laff Pak. Lunar Leepers is available for $29.95 at your local computer store or directly from:

SIERRA ON-LINE, INC.
Sierra On-Line Building
Coarsegold, CA 93614
(209) 683-6858

ADD ONE DOLLAR FOR SHIPPING
VISA • MASTERCARD • CHECK • COD ACCEPTED

SIERRAVISION™

Sierra ON-LINE, INC.