Music Maker MANUAL

subLOGIC
713 Edgebrook Drive
Champaign, IL 61820
(217) 359-8482
Telex: 206995

by Jim Baldridge

SubLOGIC / 713 Edgebrook Drive / Champaign IL 61820
October 1982

First Edition
First Printing

Introduction

MUSIC MAKER can help you make beautiful music with your Apple II! There are a lot of music programs currently available for this computer. Some of them require expensive plug-in hardware and external stereo hi-fi connections. Some allow complicated voicing of different instruments, and let you enter virtually a full-orchestra score. Others, although simpler, are intended for musical instruction instead of actually playing music. And some are just toys – fun to play with, but not very useful.

MUSIC MAKER is different. It uses sophisticated software to push your Apple to the absolute limit of what it can *already* do – without any extras. and that's quite a bit. Certainly enough for the development of elegant compositions with a minimum of musical knowledge. You can enter songs from sheet music, or create and save a library of your own music and sound effects. These songs can be played back at any time. Or, you can plug your songs into the colorful Kaleidoscopic Maestro program included with MUSIC MAKER, and watch as well as listen to your creations.

MUSIC MAKER can also be used as a sound utility in conjunction with other programs. Once you've created a song, you can take it and use it anyplace else you want. You do not need anything on the original disk to replay it, and no special programming knowledge is required to use it effectively.

Yes, MUSIC MAKER does have limitations. You won't be able to enter the entire score of Beethoven's "Moonlight Sonata" or the Beatle's greatest hits in one segment. If a song exceeds 1000 notes you must divide it and then link the parts together later. The range of fifty tones from F below Low C to F# above High C gives the notes necessary for virtually every melody ever written.

You don't have to be a musician to use this program well; a complete **Music Refresher** section is included in the manual. As an additional aid, there are several NOTES and even a few WARNINGS scattered throughout the program and documentation to keep you on the right track to successful song development.

Since many people will want to use the program quickly, the first section of instructions is simply a **Tutorial** with references to a **Technical Information** section you'll find later. A **Command Summary** card is also included so you won't have to search to find items you need quickly. You'll find some sheet music included for your practicing. Before you get there, however, you'll learn most of the commands and functions by building and playing several scales. Then, when you start on songs, you'll be able to concentrate on them rather than the fundamentals.

Tutorial

NOTE: An initialized DOS3.3 disk is required to save the data files and song modules you will be creating in this **Tutorial** section. Make sure you have a data disk before proceeding.

Song Maker/Editor

Insert MUSIC MAKER in drive #1 and load the MUSIC MAKER program. Press (ESC)ape to access the Main Menu and then select option #1 to enter the Song Maker/Editor. Any time you see the Editor prompt (—>) you can press H for HELP to view a list of the commands available. Do that now and then press any key to return to the Editor.

In order to construct a song you must INSERT each note. Press I for INSERT, and then press RETURN to begin entering notes. Your monitor will now display:

NUMBER	OCTAVE	NOTE	TYPE	MARK	TEMPO
1	3	A	T	R	160

A few words of explanation:

- NUMBER identifies each note you've entered sequentially. The program takes care of this automatically.
- Every note must be in one OCTAVE or another. If you look at the **Staff Card** included in the **Music Refresher** section you'll see that Middle C is the first note of Octave 3, and High C the start of Octave 5. Pressing any key from 1 to 5 will change the Octave accordingly.
- Every NOTE has a pitch – A, B, C, D, E, F, G – which is determined by its position on the *staff* (the horizontal lines on which music is written). Pressing any one of these keys (A – G) will change the NOTE.
- Each note has a duration; that is the number of beats or parts of a beat it receives in a *measure* (the notes or rests between two adjacent vertical bars). TYPE is used here in place of duration. Note type is changed by pressing the CTRL key while pressing the key which represents the TYPE of note you need:

CTRL W	Whole Note
CTRL H	Half Note
CTRL Q	Quarter Note
CTRL E	Eighth Note
CTRL S	Sixteenth Note
CTRL T	Thirty-second Note
CTRL G	Grace Note (64th note)
CTRL B	Half Grace Note (128th note)
CTRL A	Quarter Grace Note (256th note)

The duration of a dotted note or rest is increased by one-half its regular value. You can use the period key to dot any note type. Pressing the period key again will remove the dotted value.

- Notes will last for a relative length of time depending on the duration you assign. You can make a note last a bit longer by pressing L for Legato, or shorten it by pressing S for Staccato. R assigns a Regular length of time. These MARKs are discussed further in the **List of Commands** and **Song Development** sections of this manual.
- The MUSIC MAKER initially assigns a metronomic value of 160 beats per minute to a quarter note. TEMPO alteration is covered later in this **Tutorial** section and in the **List of Commands** immediately following.

Now you're going to build a C Major scale. Make sure that the note NUMBER is 1 and the OCTAVE is still set to 3. Press C to change the first note to a C, and then CTRL Q to make it a quarter note. Don't change anything else right now.

Press RETURN. Notice that the first note and all of its attributes are carried over to the next note to be entered. Now make the second note a D (press D) and RETURN again. Press E, RETURN, press F, RETURN, press G, RETURN, press A, RETURN, press B, RETURN. To end on C and complete the octave, change the OCTAVE to 4, press C, and RETURN. Here's how your monitor should look:

NUMBER	OCTAVE	NOTE	TYPE	MARK	TEMPO
1	3	C	Q	R	160
2	3	D	Q	R	160
3	3	E	Q	R	160
4	3	F	Q	R	160
5	3	G	Q	R	160
6	3	A	Q	R	160
7	3	B	Q	R	160
8	4	C	Q	R	160
9	4	C	Q	R	160

Want to hear what you've just entered? Press ESC to exit INSERT mode, and when the Editor prompt appears press P for PLAY. You have the option of displaying the notes as they are played (Y or N). You are then offered the option of starting at any note number; just press RETURN to accept the default value and start with note #1.

Voila... your C scale! Now press RETURN again. The notes of your scale will play back in reverse order, from end to beginning. Voila... your C scale backwards.

The left and right arrow keys can also be used to play forward or back through any sequence of notes you've entered. In addition, you can press the SPACE BAR to single-step through each note in sequence, forward or back. Any other key will return you to normal play mode. This is a great debugging tool when used in conjunction with the DISPLAY NOTES option.

Now press ESC again to exit PLAY mode. The (ESC)ape key can be used at any time, from any portion of the Song Maker/Editor, when you want to get back to the Editor prompt (—>) and enter a new command. Want to see your C scale again? Press L for LIST, and then RETURN. This function is discussed in detail later in this **Tutorial** and in the **List of Commands** section.

If your C scale looks and sounds right, it's time to save it to disk. If you've made any mistakes, you can press E for EDIT to go back and correct them. Remember to (ESC)ape back to the Editor prompt when you've finished.

Remove the MUSIC MAKER disk from drive #1 and replace it with your initialized DOS3.3 disk. Press S for SAVE and name it CSCALE. Press RETURN and then Y if correct. A ".TEXT" is appended to your title before it is saved to disk. Text files can be retrieved and loaded back into the Song Maker/Editor for further additions and alterations.

If you're certain that a song is in final form, and will not need any revisions, you can Assemble it without first saving it as a Text file. Assembled songs are an intermediate step in the creation of stand-alone song Modules. Assembled songs (or song-parts) *cannot* be loaded back into the Song Maker/Editor for revision.

Press A to ASSEMBLE your C scale. The title CSCALE will already be listed as the song name, since you've just saved it as a Text file. Forward-space the cursor over the title (or re-enter CSCALE if you prefer) and press RETURN. Make sure your initialized song disk is still in drive #1, and save this Assembled song to disk. Don't worry about using the same name as your Text file; Assembled songs do not have ".TEXT" appended onto the title. Keep in mind, however, that each Saved and/or Assembled song name must be unique.

Now you're ready to create a new scale, E Major, containing four sharps – F, G, C, and D. First press N for NEW. Yes, you *are* sure. This clears any notes you have already entered in INSERT mode and sets the note number back to 1.

NOTE: Don't forget to Save or Assemble any important song file before using this command, since NEW erases everything previously entered. MUSIC MAKER will warn you if the current Text file has not yet been Saved to disk.

Time for a new command. Press K for KEY SET and then press S for SHARPS. Now press the keys of the four sharp notes required; F, G, C, and D. Notice that the sharp sign appears below these four notes. Press RETURN when you're through.

Now press I to INSERT and follow the same procedures used for creating your C scale. Start with a quarter note E, Octave 3, and build from there. Don't forget that the C note (actually C# here) starts Octave 4. Notice that sharps are entered automatically when you press one of the four pre-set sharp notes. Sharps, flats, and naturals can also be manually entered using the #, &, and N keys respectively. Your monitor should look like this when finished:

NUMBER	OCTAVE	NOTE	TYPE	MARK	TEMPO
1	3	E	Q	R	160
2	3	F#	Q	R	160
3	3	G#	Q	R	160
4	3	A	Q	R	160
5	3	B	Q	R	160
6	4	C#	Q	R	160
7	4	D#	Q	R	160
8	4	E	Q	R	160
9	4	E	Q	R	160

PLAY the scale to see if it sounds right. This time, just ASSEMBLE and save it to your song disk under the name ESCALE. Press C for CATALOG to ensure that the disk contains the files CSCALE.TEXT, CSCALE, and ESCALE.

Re-insert MUSIC MAKER in drive #1 and CATALOG that disk. Several Text files have been included on the disk for you to play with. Press G for GET TEXT and enter the song name FLOWERS. Press RETURN and Y to load that song into the Song Maker/Editor.

Now press L to LIST the song. Only 20 notes can be displayed at one time, but you can press the period key to scroll ahead one note or the comma key to backspace one note. Use the right and left carats (SHIFT period or SHIFT comma) to scroll forward or back one full screen at a time. Notice that the TEMPO of this song is set to 200.

Press ESC and then P to PLAY the song. Now you're going to change the Tempo to see how that affects song speed. Get back to the Editor prompt and press T for TEMPO SET. This command allows you to alter the Tempo of an entire song, any part of a song, or even individual notes in the song. Press RETURN three times to modify the entire song. Want to speed it up? Press U for up. Each increment is twenty beats per minute; increment by 3 to increase the Tempo to 260. When you press RETURN the modified song will be listed automatically with the changes you've made.

PLAY this modified song to hear the difference. Then insert your song disk into drive #1 and ASSEMBLE the song as FLOWERS260.

Some SOUND EFFECTS Text files have also been included on the MUSIC MAKER disk. See if you can load SFX1.TEXT into the Song Maker/Editor, and then ASSEMBLE and save it onto your song disk as SFX1.

NOTE: Don't let all of this disk switching confuse you. Usually, the MUSIC MAKER disk is no longer required at all once the program has been loaded. You can insert a song data disk into drive #1 before entering the Song Maker/Editor, and proceed from there to the creation of stand-alone song modules. Just remember; whenever you Save or Assemble a song, your song data disk must be in drive #1.

Song-Part Linker

Now we will investigate some useful features included with MUSIC MAKER for the development of stand-alone song modules. Suppose you want to create a song longer than 1000 notes in length. Since the MUSIC MAKER can only handle 1000 notes at a time, you will have to enter and Assemble the song in several parts, and then link these parts together to form one long song.

For example, suppose you want to link CSCALE and ESCALE. Press Q for QUIT to exit the Song Maker/Editor and return to the Main Menu. Now select option #3 – LINK SONG-PARTS TOGETHER TO MAKE ONE LONG SONG. A separate list of commands govern the use of this Song-Part Linker; each must be preceded by a semicolon in order to be correctly interpreted.

Type in CSCALE, make sure your song disk is still in drive #1, and press RETURN. Then enter a ;V to ensure that CSCALE has been loaded into the Linker. Notice that up to 16 song-parts can be loaded and linked together. Use the ;H command to return to the Linker menu.

Type ESCALE over CSCALE and press RETURN to load this song as well. Now all you have tyo do is type ;S and press RETURN to link and SAVE these two scales as one longer song. Give the combined scales the name SCALES, save it to your song disk, and then type ;Q to return to the Main Menu.

Module Maker

Now you'll create a song module from the assembled songs you've saved to disk. This is the final form a song (or songs) must be in to run from Applesoft or to be incorporated into one of your own control programs. Select option #2 from the Main Menu – MAKE A MODULE FROM SONGS ON DISK. Select the Applesoft module interface to attach an Applesoft Module Header onto your assembled songs, and press Y when prompted. The Module Maker uses the same set off commands as the Song-Part Linker, and each must be preceded by a semicolon.

Type in SCALES, press RETURN, FLOWERS260, RETURN, SFX1, RETURN. (Use the ;V command if you want to check that these songs have been added to the module.) Save these three songs as one module by typing ;S. Name the module PRACTICE and press RETURN. Notice that ".MODULE" is appended to the title to indicate that this is a complete song module. Make sure your song disk is still in drive #1 before saving the module. Once it's been saved, the Start Address, End Address, and Length of your module are displayed in both decimal and hexadecimal notation. This information is useful when you want to relocate a module in memory or fit it into a larger program. Module relocation and Module Header interfaces are covered in the **Technical Information** section of this manual.

NOTE: Remember that all songs (or song-parts) must be Assembled and saved to disk before they can be incorporated into a song module.

Module Playback

To listen to your completed module, select option #6 from the Main Menu to QUIT the MUSIC MAKER, and boot the song data disk.

Now, from Applesoft, type in:

```
PRINT"(CTRL-D) BRUN PRACTICE.MODULE":S
```

You must insert a CTRL-D as the first character of the PRINT statement in order to have the Apple interpret it as a DOS command. Note that the embedded CONTROL character does not appear on the screen when this command is entered. Why the S at the end? This stands for the Speaker inside the Apple. The letter C will send the signal to the Cassette port of your Apple, where the output can be used to drive an external amplifier and speaker system. Why PRINT the statement? Because DOS doesn't like the S, and will give you a FILE NOT FOUND message if you just type it in normally.

Remember, SCALES is song #1 in this module, FLOWERS260 is song #2, and SFX1 is song #3. To play song #1, just type in:

```
X = USR(1)
```

Similarly, to play song #2 or song #3, type in:

```
X = USR(2) or
X = USR(3)
```

To play all 3 songs in sequence, you can type in:

```
X = USR(1):X = USR(2):X = USR(3)
```

Tired of listening to a song? Just press ESC. Resume by using the same statement. Changing the song number starts the new song at its beginning. Using the same number continues the song playing if it was stopped in the middle.

More information on the use of modules in your programs can be found in the **Technical Information** section of this manual.

Kaleidoscopic Maestro

Any song module you create can be run with the Kaleidoscopic Maestro program included on the MUSIC MAKER disk. (Several song modules have been included as well.) This program adds colorful graphics to your songs while they play. The Kaleidoscopic Maestro is just one example of the type of graphic program you can develop, with a little imagination, to enhance and visualize your musical creations.

Your computer must be in either Applesoft or Integer BASIC to run this program. Insert the MUSIC MAKER disk into drive #1 and simply type:

```
BRUN KALEIDOSCOPIC MAESTRO
```

remove the MUSIC MAKER disk after the program has loaded, insert your song disk, and type in PRACTICE when prompted for a module name. Notice that you are then prompted to enter the number (1-3) of the song you'd like to play. Choose #1, and press S for Speaker to hear (and watch) your SCALES play. Play songs #2 and #3 as well.

Use of the Kaleidoscopic Maestro is basically self-explanatory. Re-insert your MUSIC MAKER disk in drive #1, Catalog, and see if you can load and play the song modules included on that disk. Remember that only complete song modules can be played, not Text files or Assembled songs.

List of Commands

Most of the commands available in the Song Maker/Editor have been discussed in the **Tutorial**. A complete description of each follows, listed in alphabetical order.

Editor Mode

NOTE: Remember, use the ESC key to exit a command mode when you want to get back to the Editor prompt (—>) and enter a new command.

(A)SSEMBLE – Takes the notes you have entered and saves them off to disk as an Assembled song or song-part. You will be asked for a name to save it under. Assembled songs (or song-parts) are used in the creation of stand-alone song modules. WARNING: Assembled songs cannot be reloaded into the Song Maker/Editor for further modification.

(C)ATALOG – Self-explanatory. Use the C key to Catalog the contents of the disk in drive #1.

(D)ELETE – Allows you to delete any or all of the notes you have entered. You will be asked for a range of note numbers; if you want to delete only one, enter that number as both the beginning and the end.

(E)DIT – Make a mistake? You will be asked for a starting note number, and that note will be displayed for you to make changes. Pressing RETURN enters the note as is (or as changed) and displays the next note – or puts you back into INSERT mode if that was the last note entered. ESC ignores the note you're working on and returns you to the Editor prompt.

(G)ET TEXT – Asks you for the name of a Saved song Text file, and loads it into the Song Maker/Editor for you to work on or listen to. WARNING: This command will load the Saved song over whatever Text file (if any) you are currently working on. MUSIC MAKER will warn you if the current Text file has not yet been Saved to disk.

(H)ELP – Displays a complete list of the commands available to you in the Song Maker/Editor.

(I)NSERT – Allows you to enter notes, one by one, ranging from F below Low C to F# above High C. A short warning beep is emitted three times if you attempt to enter a note beyond this range.

The program numbers your notes automatically; if you delete or insert notes in the middle of text, it renumbers to fill the gaps. The maximum number of notes you can enter in one group is 1000. If your song is longer than that, you will have to save the song-parts and link them together later. The additional commands available in INSERT mode are described separately at the end of this section.

(K)EY SET – There's no need to keep re-entering sharps and flats. Press K, and let MUSIC MAKER assign sharps or flats to the notes you specify. They will be assigned automatically from then on in the INSERT and EDIT modes. Each time you press K, the KEY SET mode is cleared to all naturals (key of C). If no key is set the program assumes key of C.

(L)IST – Like to see your work so far? You'll be asked for a starting note number, and the notes lines will be displayed one screenful (20 notes) at a time. The comma key scrolls backward one note, the period key forward one note. The left and right carats (SHIFT comma or SHIFT period) scroll back and forward one full screen respectively.

(N)EW – Want to start a new song? This command sets the note number back to one and clears any text already entered. WARNING: Be careful not to accidently clear something you wanted to save. MUSIC MAKER will warn you if the current Text file has not yet been Saved to disk.

(P)LAY – You'll be asked what note number to start with, and whether you want a display of each note. The display slows the tempo slightly, but is very handy for debugging. Right arrow plays forward, left arrow plays backwards. The SPACE BAR plays one note at a time, forward or back (use the arrow keys). Any other key resumes real-time tempo.

(Q)UIT – Exits the Song Maker/Editor and takes you back to the Main Menu.

(R)EPEAT – More useful than it might seem at first. A lot of songs repeat measures and even long passages. You need only enter them once. You'll be asked to enter a range of note numbers, and then where to put them. The destination line number must be higher than your source range. You cannot REPEAT line numbers before or on top of themselves.

The REPEAT command is also handy for building chords from grace notes, or for producing sound effects. You can enter large numbers of repetitive notes by typing a sequence just once, doubling it with REPEAT, then doubling that, and again, etc.

(S)AVE TEXT – Saves what you've entered to disk as a binary text file. This is not the same as the ASSEMBLE command. Notes are stored in a form that you can load back into the Song Maker/Editor, and continue working on. You'll be asked to name the song, and the program will add ".TEXT" to its name on the disk Catalog.

(T)EMPO SET – MUSIC MAKER assigns a metronomic value of 160 beats per minute to a quarter note. That's a good "average" tempo that works well with most simpler music. You can choose any tempo you want, within reason, either for individual notes or the whole song, with the TEMPO SET command.

Tempo adjustments are made in increments of 20 beats per minute. TEMPO SET will ask you the note-number range, whether to increase or decrease (up or down), and by how many increments. If you go so far as to double the tempo (i.e., a setting of 320), MUSIC MAKER will compensate by keeping a setting of 160 while HALVING the value of all notes (wholes become halves, quarters become eighths, etc.).

In the same way, if you decrease tempo by half or more, MUSIC MAKER will respond by DOUBLING the value of all notes. It will keep doing this in either direction – up to the point where every note becomes a mere speaker pop, or to where it is trying to create a note longer than it can play (a Legato whole note at tempo 160). MUSIC MAKER will beep three times if you attempt to enter a note longer than this.

We recommend against drastic tempo changes before you've saved your text file to disk. If you increase tempo by 10 increments and then decrease it by 10, you will not get back the same song that you started with. TEMPO SET is relative. It will add or subtract increments of 20 beats from whatever tempo you assigned each note as you entered it. If you change your mind at the last minute, you can abort a tempo change by hitting any key while MUSIC MAKER is scrolling through the modified note listing.

Insert Mode

Now let's review the commands available to you in the INSERT and EDIT modes. When you press I for INSERT or E for EDIT, your screen displays six columns. Here are the headings and their meanings:

NUMBER	OCTAVE	NOTE	TYPE	MARK	TEMPO

NUMBER – A note number assigned by MUSIC MAKER. It's there for reference purposes only, and will change if you add or delete any notes. The maximum number of notes you can work on at one time is 1000.

OCTAVE – MUSIC MAKER offers a range of 50 notes, from F below Low C (Octave 1) to F# above High C (Octave 5). That's about the range of human voices, from a low basso-profundo to the highest soprano. Middle C is the first note in Octave 3, and the Octave changes every time you pass a C going either way. Change the Octave by pressing the appropriate number key. Use the **Staff Card**, if necessary, to determine which Octave a note is in.

NOTE – Change the note by pressing the appropriate letter key (A through G). If you've used the KEY SET command, sharps and flats will be applied automatically. To cancel one, just type an N for Natural. You can manually apply one by typing the ampersand (SHIFT 6) for a flat, or the number sign (SHIFT 3) for a sharp. Use the SPACE BAR to make a note into a rest.

TYPE – This sets the note's duration – from a whole note to a 256th note. Change it by holding down the CTRL key while pressing the corresponding letter key for the note Type desired. A grace (64th) note is about as short as the Apple can play on tune, but it's actually four times as long as the computer will handle.

So we've added two other note Types that you can use for sound effects: A and B. B is one-half the duration of a grace note, and A is one-quarter the duration. Try entering a lot of A notes running up and down the scale. Interesting game sound? Experiment with these two note Types – the possibilities are endless.

Note Type can be modified in 3 ways. Use the period key to dot any note Type; pressing the period key again will remove the dot. This dot has a relative value only, equal to half the value of the note or rest after which it is placed. In the same way, you can press the T key to make a note part of a Triplet, and "squeeze" three notes into the space of two. All three notes must be designated as parts of the Triplet. The Q key designates any note Type as part of a Quintuplet, five notes in the space of four. Toggling either of these keys again will eliminate the T or Q value.

MARK – Only three options here. R for Regular, L for Legato, and S for Staccato. These MARKs do not alter note TYPE; a quarter note is still a quarter note regardless of the MARK assigned. Rather, they determine the relative durations a note plays and rests throughout its total note length. Staccato cuts a note short, while Legato draws it out and blends it into the next note entered. R assigns a Regular note length.

TEMPO – The Tempo of an individual note can be changed as easily as an entire song. The default value is 160 beats per minute assigned to a quarter note. You can increase or decrease the tempo in subtle increments of twenty beats by pressing the right or left carats (SHIFT period and SHIFT comma) respectively.

This feature lets you vary the tempo of any part of your song for emphasis or dramatic effect. Notice, for example, that a waltz is not played with notes of exactly the same length, even though it's written that way. Rather, it "lilts" along with an emphasis on the first beat of each measure. A pianist or any solo instrumentalist adds interpretation, speeding up on runs, slowing down at ends of passages, etc. It's easy to get carried away with this, but a little variation can do wonders for otherwise mechanical computer music.

In addition to these commands just listed for **Insert Mode**, you can press P to Play each note as it is entered. Use the P key to toggle the Play option on and off.

The H key displays the same command list available to you in **Editor Mode**. Any other key returns you to INSERT mode.

RETURN enters the note you've been working on and displays an exact duplicate with the next note number. Usually, only one or two changes are necessary before you enter the next note.

ESC ignores the last note displayed and returns you to the Editor prompt, ready to enter a new command.

Song Part Linker and Module Maker

A different set of commands are available in the SONG-PART LINKER and MODULE MAKER portions of MUSIC MAKER. All are basically self-explanatory:

;C – CATALOG
;V – VIEW SONGS ENTERED SO FAR
;H – LOOK AT THIS SCREEN
;S – SAVE THE FINAL SONG TO DISK
;Q – RETURN TO THE MAIN MENU

In the Module Maker, the ;S command is modified to:

;S – SAVE MODULE TO DISK

Song Development

If you're unfamiliar or unused to reading music, we've included several sample songs along with instructions for entering them. The first thing you should do with any sheet music is number the notes. This is a great help in debugging, since you can follow along and find any mistakes easily and quickly. You will also determine if the song is longer than the 1000-note limit imposed by the Song Maker/Editor. Don't forget to number any rests you might have to enter as well. If you know nothing at all about music, you may have to refer to the **Music Refresher** section to find a note's NAME and TYPE. Use the **Staff Card** to determine which OCTAVE the note is in.

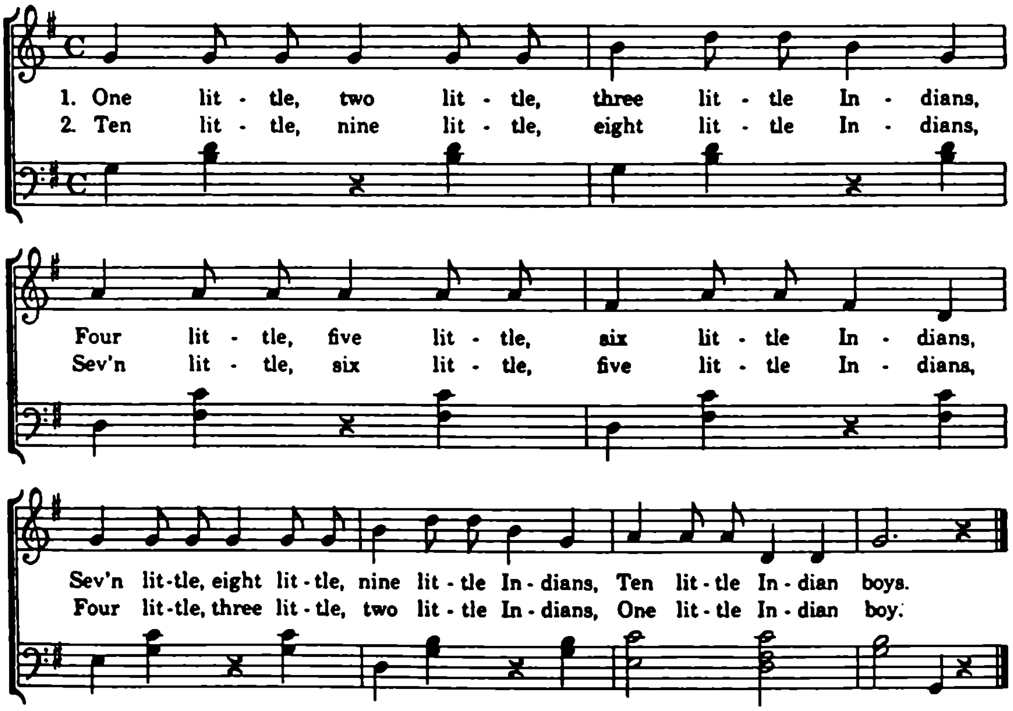
Ten Little Indians

The melody of a song is usually the top line of notes in its treble clef. That's what we're going to use here.

- Press N for NEW.
- Press K for KEY SET, and set it to the key of G Major (one F#).
- Press I for INSERT and make note #1 a G (press G). Press CTRL Q to turn it into a quarter note. RETURN.
- All you have to do with note #2 is change its TYPE to an eighth note. Press CTRL E, RETURN.
- Note #3 is the same as #2. Just press RETURN.
- Continue, remembering that you must change to OCTAVE 4 at note #8. Change back to OCTAVE 3 at note #10.
- Remember to dot the half note in the last measure.
- You should have 39 notes. You needn't enter the final rest unless you want to REPEAT the entire song for the second verse. If so, press the SPACE BAR and enter a quarter rest, giving you a total of 40 notes. RETURN. Now press ESC and and then R for REPEAT. Repeat the entire song.
- Press P for PLAY and display the notes as they're played. Does it sound all right? If not, use the SPACE BAR and the arrow keys to single-step forward or back through the song. When you've located the problem, take down the note number(s) and use the EDIT command to fix it.

Ten Little Indians

Traditional



1. One lit - tle, two lit - tle, three lit - tle In - dians,
2. Ten lit - tle, nine lit - tle, eight lit - tle In - dians,
Four lit - tle, five lit - tle, six lit - tle In - dians,
Sev'n lit - tle, six lit - tle, five lit - tle In - dians,
Sev'n lit-tle, eight lit-tle, nine lit-tle In-dians, Ten lit-tle In-dian boys.
Four lit-tle, three lit-tle, two lit-tle In-dians, One lit-tle In-dian boy.

The Miller's Flowers

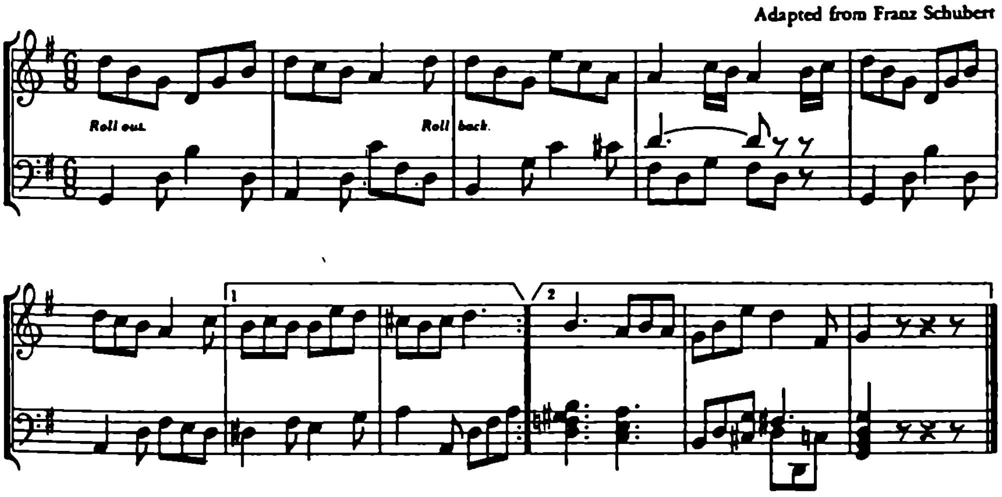
This is a nifty little song adapted from Franz Shubert that can be "jazzed up" quite nicely in a number of ways. Here's one way . . . you'll be able to think of others.

Again, we'll be using the melody line in the treble clef. This song is also conveniently included as a Text file on the MUSIC MAKER disk (FLOWERS, remember?). Try entering it as described below first, and then compare your version with the one already on disk. Notice the other changes we've made in addition to those listed here.

- Set the Key to G Major (with F sharped), and then press I to INSTERT.
- Number the first measure only. We're going to add a little "oomph" to the tune by using a grace note to emphasize the first beat of each measure. The idea is to insert a grace note exactly one octave higher than the first note in each measure.
- Note #1, then, will be a grace note (CTRL-G) in OCTAVE 5. Press D to set the note.
- Note #2 is the first note of the first measure on your sheet music, OCTAVE 4, NOTE D, TYPE E, and MARK R.
- Proceed as before. Remember to insert a grace note one octave higher immediately before the first note in each measure.
- When you've finished with the first ending, you can use the REPEAT command to save some time. Make sure you've entered the last note of the first ending, press ESC to return to the Editor prompt, and press R for REPEAT. If your numbering went the same as ours, you'll want to repeat notes 1 through 40 after note 52.
- After the repeat is done, press I to INSERT the second ending.
- Press P to PLAY and make sure all the notes are right. Use the DISPLAY NOTES and SPACE BAR single-step options to find and EDIT any mistakes.
- Press S to SAVE the Text file, if you like, onto your song disk. Give it a name other than FLOWERS to eliminate any confusion with the Text file on MUSIC MAKER.

The Miller's Flowers

Adapted from Franz Schubert



Chords and Multiple-Voice Songs

Entering chords or multiple-melody songs is very much a matter of experimentation to determine what sounds best to you. The basic procedure involves combining a series of Legato grace notes to create the chord or notes desired. This requires an understanding of how MUSIC MAKER plays the notes you've entered.

Each note is assigned a total note length according to its TYPE (whole note, half note, etc.) and the tempo you've set. The actual play length of the note varies, depending on the MARK you've assigned it (Staccato, Regular, or Legato):

Play Length + Rest Length = Total Note Length
(PL + RL) = (NL)

for a:
Staccato Note – PL = 50% of NL
Regular Note – PL = 75% of NL
Legato Note – PL = 100% of NL

In each case, RL (Rest Length) uses up the other part of NL.

Notice that a Regular note plays for 75% of its Total Note Length and rests the final quarter. This Rest Length is the pause between individual notes. Notice also that a Legato note plays for the Total Note Length. In effect, this blends it into the next note to be played. A Staccato note plays for the first half of its Total Note Length, and rests the other half.

Suppose you want to simulate a Regular quarter note with a series of grace notes. You can fit 16 grace notes into the duration of one quarter note (16/64 4). Giving these grace notes a Legato MARK will blend them together and generate a relatively steady tone. But there's more to consider. The Total Note Lengths of 16 grace notes are equal to the Total Note Length of one quarter note, but a Regular quarter note only plays for the first 75% of its Total Note Length. In order to accurately simulate a Regular quarter note, therefore, you will have to enter 12 Legato grace notes (75% of 16) followed by 4 grace rests. Verify this for yourself in the Song Maker/Editor; enter a Middle C(OCTAVE 3) quarter note with a regular MARK, followed by 12 Legato grace notes and 4 grace rests. REPEAT the entire sequence several times, then PLAY and listen to the results.

Now, suppose you want to enter a C Major chord one Regular quarter note's length in duration, containing the notes C, E, and G in OCTAVE 3. One solution is to enter this three-note series 4 times in sequence, as Legato grace notes, and follow with 4 grace rests. (These 4 grace rests can actually be replaced by one sixteenth rest to conserve space in your text file.) Try entering this chord into the Song Maker/Editor as described, PLAY it, and then try different combinations of the three notes in the series (CGE, GEC, etc.). Remember, 12 grace notes + 1 sixteenth rest = Total Note Length of a Regular quarter note.

This procedure can be applied to chords or multi-voice notes of any duration. First, determine how many grace notes and rests can fit into the Total Note Length of the note TYPE involved. Then use the MARK to calculate the ratio of grace notes to rests you'll need.

Another factor to consider is the sequence in which you enter these grace notes. Two-voice music is simply a matter of alternating between the two, but when developing 4 (or more) part harmonies we recommend against entering grace notes in straight ascending or descending order. This scaling tends to decrease the chordal effect. Again, the best advice we can offer is to experiment with different grace note combinations to determine what sounds best to you. Use the REPEAT function to build chords of the desired length, and use a single rest to fill the Rest Length portion of Total Note Length.

You might also want to try other note TYPEs during chord development. The shorter A and B notes usually play too fast to stay on pitch, but at slower tempos you may find a B-note chord preferable to one comprised of grace notes. At faster tempos a thirty-second note chord might be more appropriate.

Special Problems

Suppose you INSERT your entire song, PLAY it, and discover that you've skipped an entire measure or two. Don't panic, you won't have to do the whole thing over. The INSERT command can also be used to insert missing notes or rests into a song. Just select INSERT, enter the note number where you wish to begin inserting omitted notes, and proceed from there. These notes will be inserted into the body of your song, and all following notes will be renumbered accordingly.

If your song exceeds 1000 notes, you will have to create two or more song-parts and then combine these parts later with the Song-Part Linker portion of MUSIC MAKER. Refer to user instructions in the **Tutorial** and Linker limitations in the section on **Technical Information**.

MUSIC MAKER is protected against accidental RESET. Nevertheless, you should take extra care not to hit RESET during a disk save unless you're feeling particularly lucky. If you're running MUSIC MAKER while Integer BASIC is active, hitting RESET will put you into the Monitor. Typing 3D0G (RETURN) should get you back into the program with everything (hopefully) intact.

Music Refresher

Glossary of Music Terms

Clef – The range of pitches on a particular staff, i.e. – treble clef, bass clef.
Dot – A mark having a relative value, equal to half the value of the note or rest after which it is placed.
Double Bar – Indicates the end of a composition or an important part of it.
Duration – The length of time a note or rest is held.
First Ending – A measure or group of measures marked by ┌1‾‾‾‾╲ indicating that they should be played the first time through the piece.
Flat – Lowers a note by a half-step.
Key Signature – The sharps and flats at the start of the staff which indicate the key and the scale of the composition.
Legato – No silence between notes, indicated by a curved line (slur) extending above or below two or more notes, or a horizontal line above or below a single note.
Measure – Each group of recurring beats.
Octave – The group of notes between one note and the next note of the same letter in either direction.
Repeat – The signs ||: and :||, indicating repetition of the notes enclosed by them. If the latter is used alone, repetition starts from the beginning of the composition. The former indicates repetition from that point.
Rest – The length of silence between notes.
Scale – A succession of notes one or one-half step apart, arranged in ascending or descending order.
Second Ending – In reference to a first ending: The measure or group of measures substituted the second time through a composition, represented by the symbol ╱‾‾‾2┐.
Sharp – Raises a note by a half-step.
Staccato – Shortens the sound of a note, indicated by a dot over or under a note.
Staff – The five horizontal lines on which music is written.

Time Values

Note Values

𝅝 Whole Note

𝅗𝅥 Half Note

♩ Quarter Note

♪ Eighth Note

𝅘𝅥𝅯 Sixteenth Note

𝅘𝅥𝅰 Thirty-Second Note

𝆔 Grace Note

♩♩♩ Triplet (three notes in the space of two)

♩♩♩♩♩ Quintuplet (five notes in the space of four)

𝄻 Whole Rest

𝄼 Half Rest

𝄽 Quarter Rest

𝄾 Eighth Rest

𝄿 Sixteenth Rest

Dotted Note and Rest Values

𝅝. Whole Note and Half Dot

𝅗𝅥. Half Note and Quarter Dot

𝄼. Half Rest and Quarter Dot

𝅘𝅥. Quarter Note and Eighth Dot

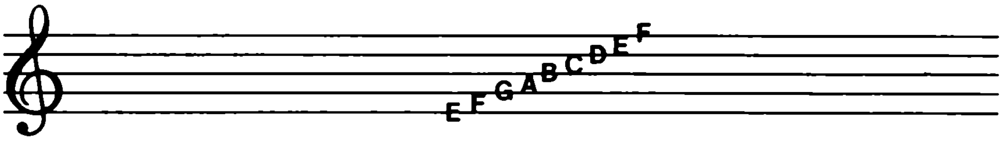
𝄽. Quarter Rest and Eighth Dot

𝅘𝅥𝅮. Eighth Note and Sixteenth Dot

Note Names and Special Markings

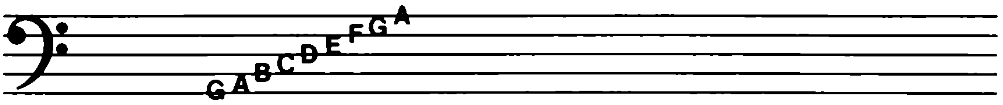
Treble Clef

(Remember: Lines can stand for "Every Good Boy Does Fine." Spaces Spell: FACE)

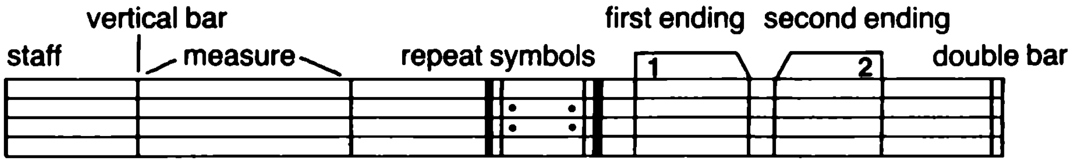


Bass Clef

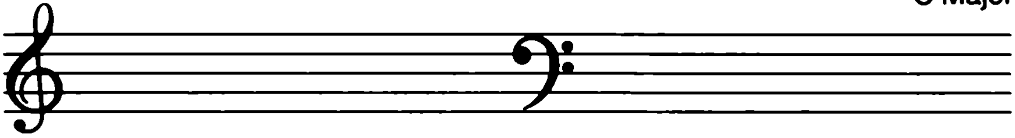
(Lines: "Good Boys Do Fine Always.")



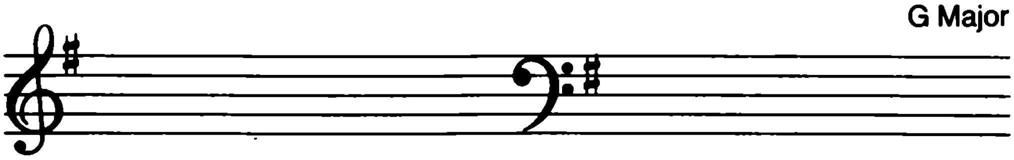
Special Markings



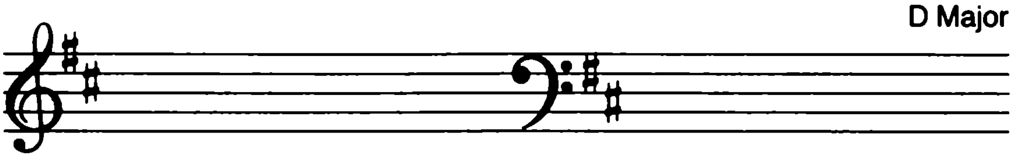
C Major



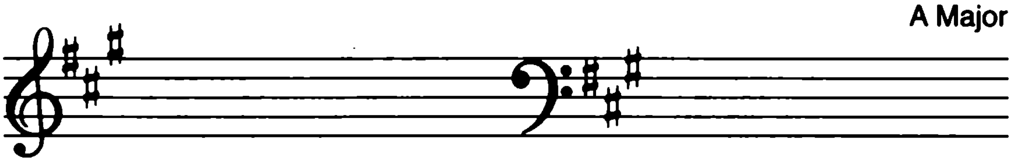
G Major



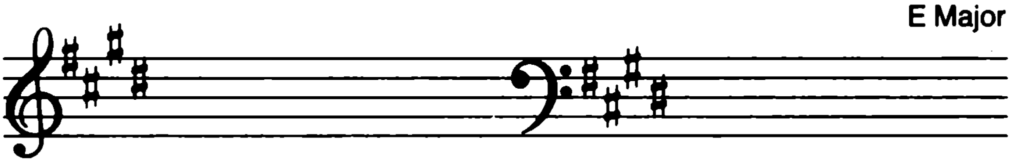
D Major



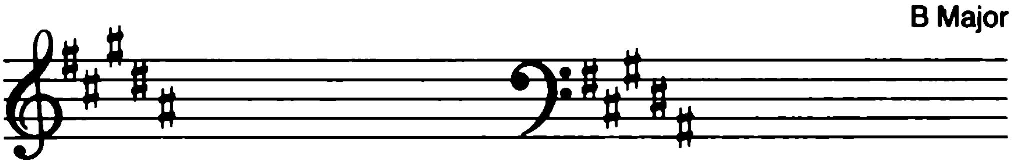
A Major



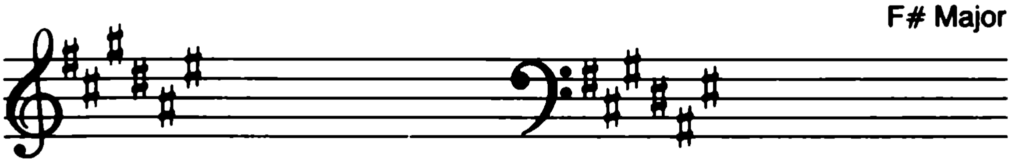
E Major



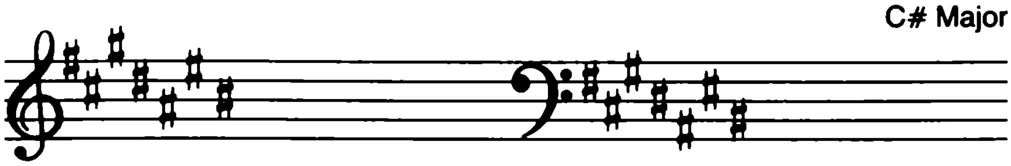
B Major



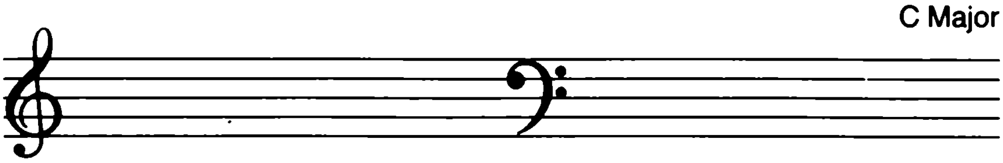
F# Major



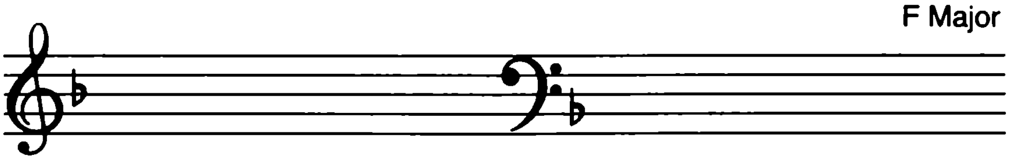
C# Major



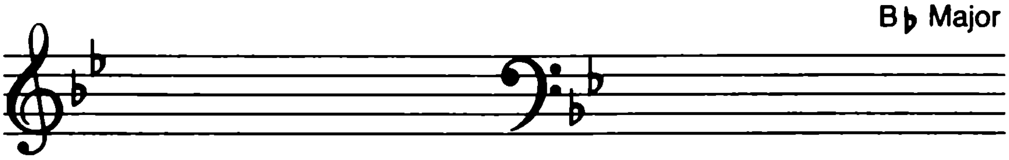
C Major



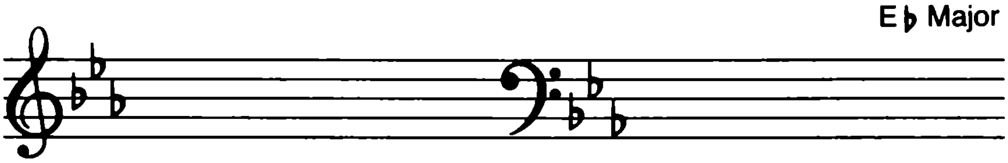
F Major



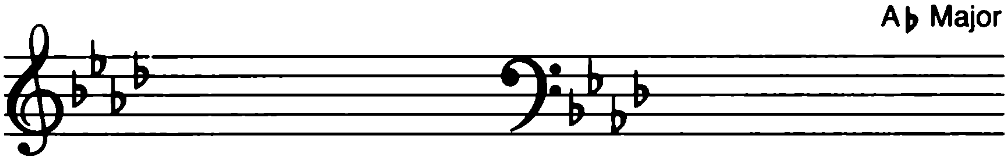
B♭ Major



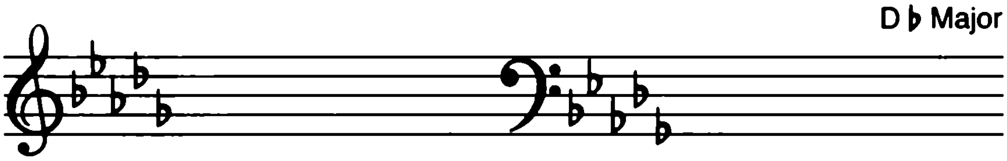
E♭ Major



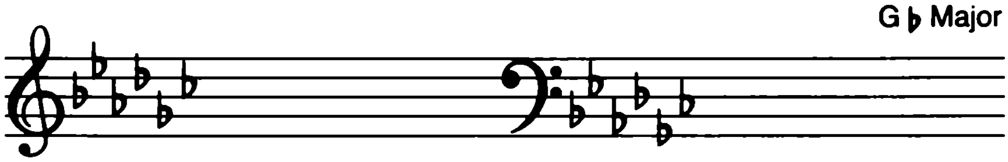
A♭ Major



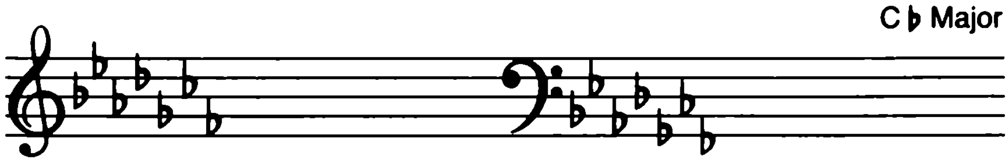
D♭ Major

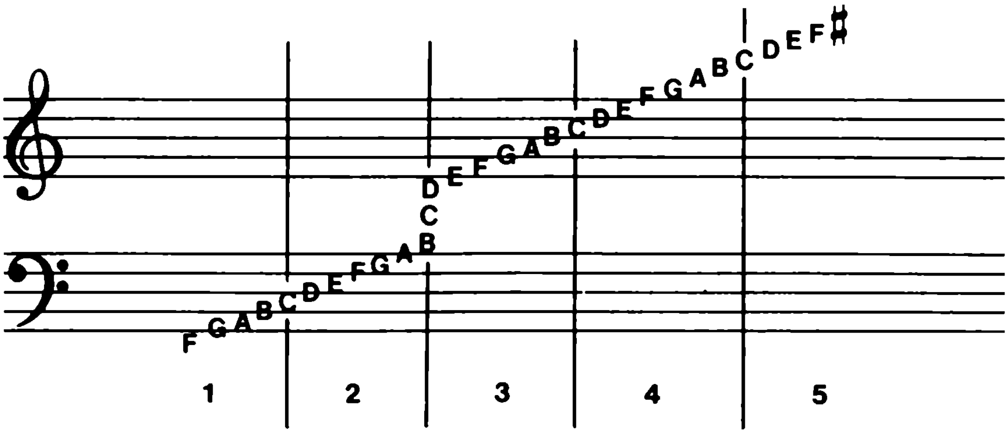


G♭ Major



C♭ Major





Octave Number

Technical Information

Song-Part Linker and Module Maker Limitations

You can load and combine song-parts in the Song-Part Linker up to a maximum length of 58 sectors on disk (14K). The Song-Part Linker will not let you load any song that would put you over this limit. Also, the maximum length of a song that you can load into the Linker is 26 sectors (6K). Rarely will this be a problem, since the maximum length that can be made with the Editor is 16 sectors (1000 notes, 3.5K). You would have to link 2 or more lengthy songs together and then try to link other songs to the end product. This length limitation can be avoided by linking the original parts together with the new songs at the same time.

The 58 sector (14K) limit also applies to the Module Maker. If the song (or songs) you want to play are too large to fit into one module, refer to the manual section on **Use and Care of Multiple Modules**.

Use of Modules in Programs

Modules were designed to be particularly painless when used from within programs. MUSIC MAKER gives you the choice of three different Module interfaces:

Applesoft
Integer BASIC
Machine Language

Applesoft

As you've probably noticed by now, the Applesoft interface uses the USR function in place of the & which is so commonly used. To activate a module from within Applesoft simply:

```
10 D$=CHR$(4):  REM CTRL-D
20 PRINT D$"BRUN modulename.MODULE":S (or C)
```

This is the same method used in the **Tutorial** to run your PRACTICE.MODULE. A string variable, D$, replaces the embedded CTRL-D character. S or C at the end of line 20 indicates you choice of either Speaker or Cassette port output.

Or, you can BLOAD a module in advance and then later CALL its start address. All non-relocated modules start at 16384. (If you need to determine a module's start address, load it into the MODULE RELOCATOR to list its current specifications.) For example:

```
20 PRINT D$"BLOAD modulename.MODULE"
30 your program here . . .
1000 CALL 16384:S (using the correct start address, and of course,
                         choosing between speaker or cassette)
1010 more of your program . . .
```

You may want to use this method instead of BRUN if:

1. You want to switch from cassette to speaker or vice-versa.
2. You make the module part of a larger utility package that can't be BRUN.
3. You're using more than one module (explained later in this section).
4. You need the USR function for something else.

5. You make the module part of your Applesoft program.
6. You saw it here and want to see if it really works.

NOTE: You cannot activate a module from within a subroutine, either by BRUNning it or CALLing. This confuses the Applesoft ROMs when the module jumps back to them. You must activate a module from the main portion of your program. You can, however, USR a module from within a subroutine.

Once a module has been activated, you can play the songs it contains in two different ways:

1. Play a whole song at once (or the rest of the song where it left off).
2. Play a song one note at a time.

To play a whole song, use the form:

```
10 X = USR (songnum)
```

Songnum can be any number from one up to the number of songs in the module, and corresponds to the number of the song in the list when you made that module. It can also be a variable. For example:

```
10 K = 3
20 X = USR(K)
```

would play song number three. You can use this USR call from within a subroutine.

When your program reaches the USR function, program excecution stops and the music plays until the song ends, or until someone hits the ESC key. This option enables you to interrupt a song in progress.

To play a song one note at a time use the form:

```
10 X = USR(-songnum)
```

Notice that that's a *negative* songnum. So an X = USR(-2) plays the first note of song number two, leaving things set up for the next note.

```
10 FOR I = 1 TO 5
20 X = USR(-2)
30 NEXT
40 X = USR(2)
```

This plays the first five notes of song number two singly, and then goes on to play the rest of the song. Something a little more useful:

```
10 GR
20 X = INT(RND)(1)*39) + 1:Y = INT(RND(1)*39) + 1:
   COLOR = INT(RND(1)*15) + 1
30 PLOT X,Y:Z = USR(-1)
40 GOTO 20
```

This routine plots random dots of random color with accompaniment. Most applications of this play-one-note-at-a-time mode will probably utilize graphics in some way. Applesoft is rather slow in execution speed, so your code should be as short and efficient as possible.

Use of the FLAG

Some of you may shy away from PEEKs and POKEs, but in your adventures with the Apple you will have to use them sooner or later. You'll find there's nothing mysterious about them after all. Memory location number 6 is a module's FLAG to signal whether it's finished playing a song or not. You can also use this memory location to force a module to restart a song, if that song was interrupted, the next time you USR it. For example:

```
120 FL = PEEK(6)
```

FL will equal 128 if the last song you were playing still has more notes left to play, and it will be 0 if the song has finished playing. Therefore:

```
100 X = USR(1):IF PEEK(6) = 128 THEN 100
```

will lock out the ESC key and force your listener to bear out the entire song. Why? When your program reaches line 100, song #1 starts to play. If somebody presses ESC, the song stops (as described earlier) and the next statement executes. Since the song wasn't finished yet, the FLAG(6) contains 128, and so we go back to line 100. The song then continues playing where it left off. When it reaches the end it stops again – only this time there's a 0 in the FLAG(6), so the next statement is false and your program continues merrily along.

Whenever you change the number of a song from one USR to another, the new song will start at its beginning. Changing the sign will not have this effect. If you want to make sure a song always begins anew when you USR it, you need to:

```
10 POKE 6,0
```

This will make the module think it's just finished a song, causing it to reset to that song's start next time around.

The USR function has the advantage of being able to return a number when it's done. What number does a module return? When you say:

```
10 X = USR(1) or
10 X = USR(-1)
```

X will equal the address of the next note to be played.

Each note is defined in three bytes:

Byte #1 – Pitch Value (0 to 50, 0 = Rest)
Byte #2 – Play Length (how long the note plays)
Byte #3 – Rest Length (how long the note rests before playing the next note)

Therefore:

```
  PEEK(X)   =  Pitch Value
PEEK(X + 1) =  Play Value
PEEK(X + 2) =  Rest Value
```

Bytes 2 and 3 added together give the total length of that note. Their relative values determine whether the note is Legato, Regular, or Staccato:

```
Play Length  +  Rest Length  =  Total Note Length
               (PL  +  RL    =  NL)

          Legato Notes: PL   =  100% of NL
         Regular Notes: PL   =   75% of NL
        Staccato Notes: PL   =   50% of NL
```

RL (Rest Length) uses up the other part of NL

You will probably never use the Play and Rest lengths. However, the Pitch value comes in handy every so often. The Kaleidoscopic Maestro, for example, uses the Pitch value to determine the location of four bars on the Lo-Res screen. You might want to use the Pitch value to add a little movement to an otherwise static scene. For example:

```
10 TEXT:HOME:VTAB 15:SG = -1:T2=1:
   K1=1:K2=39:K3=6:K4=128
20 X=USR(SG)
30 INVERSE:T1= PEEK (X) + K1:IF T1> K2 THEN T1 =K2
40 HTAB T1:PRINT" ";:NORMAL:HTAB T2:PRINT" ";:
   X=USR(SG):T2=T1:IF PEEK(K3)=K4 THEN 30
```

This program makes an inverse space jump about according to what note is being played. All constants have been replaced with variables to make the program run faster.

NOTE: X can never point to the first note in a song; when a song has finished, it points to garbage. If you're planning to use the Pitch value in a program, you may want to make the first note a short rest in the songs you're going to play. Notice in the example above that line 20 plays the first note; all others are then played in line 40 within a loop. Notice also that care has been taken to keep the dot on the same line. All notes above 38 are displayed in the 39th character position.

Integer BASIC

If you haven't read the Applesoft section yet, we suggest you do so now. Many ideas and examples are presented there that will not be repeated.

The Integer BASIC module was designed almost exclusively for program implementation. You would probably use the Integer module only when writing programs, because it's much faster than Applesoft.

To use an Integer module you need to assign the following variables as shown. They must be the FIRST variables assigned in the program:

```
10 SONGNUM=0:OUTPUT$="S" (or C):
   MODULE=0:MUSPTR=0
```

Actually, you can change the names as long as the lengths remain the same (SONGNUM, for instance, could be CHICAGO, or any other seven-letter word). Be sure to set LOMEM before assigning any of these variables, should you desire to do so.

You must assign a value for OUTPUT$ before you activate a module either by BRUNning or CALLing it, as described in the Applesoft section. Leave out the :S or :C after the statement, since OUTPUT$ handles that now.

After you've activated an Integer module, the value of MODULE equals the entry address for the module. All you have to know, however, is how to use it. Before you can play a song, you must assign:

```
SONGNUM = (songnum) or
SONGNUM = (-songnum)
```

Use of SONGNUM is the same as described in the Applesoft section.
Then to play the song:

```
20  CALL MODULE
```

These two statements have the same effect as the X = USR (songnum) function in Applesoft. The Integer CALL does not return a value as such. Instead, it sets MUSPTR equal to the address of the next note to be played, as X would be in the Applesoft line. Remember, to switch modules (explanation next) or otherwise activate a module with a CALL 16384 (or the correct start address) you must set OUTPUT$ to either "S" or "C" according to where you want the signal sent.

Use and Care of Multiple Modules

Before we explain the Machine Language module, let's cover this important point. Occasionally a song becomes too long to fit in one module (Tocatta and Fugue in D Minor, for example, included on the MUSIC MAKER disk). More often, you overload your module with too many songs. In either case, assuming that you want all your music in memory at one time, you'll need multiple modules. Usually you'll want to put one module right after another.

When you save your first module, write down the DECIMAL START = address. This is the address you will CALL to activate the module. Also write down the HEX END = address (the number with the $ in front of it). Hexadecimal is simply a method of counting in base 16:

0 1 2 3 4 5 6 7 8 9 A B C D E F —> 0

If the last two digits of the HEX address are 00, then this is your lucky day. Otherwise, add ONE to the left two digits, make the last two 00, and write this number down as well. For example:

$47B8 becomes $4800
$4C27 becomes $4D00
$5F39 becomes $6000
$69E2 becomes $6A00

Remember, 9 becomes A with no carry-over, and F becomes 0 with a carry (as in example three, above). The resulting number is known as a page boundary – that is, a multiple of 256 ($100 HEX). Modules must reside on page boundaries.

After you've rounded up to the next page boundary, go ahead and make the other module(s) you want to use. Don't worry about address numbers this time.

Now, load the second module you'll be using into the Module Relocator portion of MUSIC MAKER (option #4 on the Main Menu) and type Y to RELOCATE. What address are you going to relocate it to? You guessed it; the nearest non-overlapping page boundary following the first module, the number you went to all that trouble to get. Remember to precede any HEX number with a $.

Before you save the second module, its new address parameters will be listed. Write down the DECIMAL START= address. This is the number you'll CALL to activate the second module. If you want to add still another module, take down the new HEX END= address as well and repeat the process.

To use both modules, you must first BLOAD both of them (or BRUN the one you want to activate first, and then BLOAD the other). Use the CALL start address to activate them as desired.

To summarize:

1. Make the first module. Write down the DECIMAL START = number and the HEX END = number.
2. Move the HEX END = number up to the next page boundary (do not round it down!).
3. Relocate your second module to the page boundary address determined in step #2. Write down its new DECIMAL START= number.
4. If you've got more modules (and any memory left), write down the HEX END= number of the module you've just relocated and go back to step #2.

Machine Language

Machine Language modules can come in handy in several ways. First, they're the easiest to use from within a Machine Language program. Also, they're not particular about what kind of BASIC they're running in. You can make transportable modules if you like. Of course, there would be quite a few POKEs, PEEKs, and CALLs involved. You can't BRUN a Machine Language module because it wouldn't know where to JMP to after initialization. You must BLOAD it first. For example:

```
MOUTPUT    EQU  $2FC
CROUT      EQU  $FD8E
COUT       EQU  $FDED
           JSR  CROUT        ;MAKE SURE ON BEGINNING OF LINE
           LDY  #0           ;INIT DATA POINTER
LOOP       LDA  BLOAD,Y      ;GET CHARACTER TO "PRINT"
           BEQ  DONE         ;"0" IS THE END-OF-DATA MARKER
           JSR  COUT         ;PRINT CHARACTER
           INY               ;POINT TO NEXT
           BNE  LOOP         ;ALWAYS BRANCH
BLOAD      DFB  $84          ;CTRL-D
           ASC "BLOAD modname.MODULE"
           DFB  0            ;END OF DATA
DONE       JSR  CROUT        ;THIS DOES THE BLOAD
           LDA  $AA72        ;GET LOW BYTE START
           STA  MODJMP+1     ;ADDRESS (ASSUMING 48K DOS)
           LDA  $AA73        ;GET HIGH BYTE
           STA  MODJMP+2     ;STASH  IT
           LDA  #'S'         ;OR 'C'
           STA  MOUTPUT      ;CHOOSE SPEAKER
MODJMP     JSR  $0000        ;THIS NUMBER CHANGED BY THE
           etc.               FOUR LINES AFTER "JSR CROUT"
```

Of course, if you know in advance where the module will reside, leave out the four lines after the JSR CROUT and use JSR 'start address' for the last line. This routine BLOADs a module, sets speaker output (storing an ASCII S, high bit set, in $2FC), and activates the module. Remember to set $2FC to 'S' or 'C' before activating. Anything other than C becomes an S. After you've activated a module the following comes into effect:

Location $FD =	JUMP to module
Location $2FB =	SONGNUM
Location $6 =	FLAG (same as before)
Location $7 and $8 =	MUSPTR (in Low/High format)

Use of a Machine Language module is much like an Integer module. SONGNUM is different, however. When the high bit of SONGNUM ($2FB) is off, the whole song is played. When on, only one note is played. Using real negative (two's complement) numbers would obviously present problems. After initialization, play a song by doing a JSR $FD. The $FD location contains another JUMP to the module itself.

The Flag (see Applesoft section) is exactly the same, location number 6. To repeat, the high bit in $6 will be SET if there are still more notes to play, and RESET if the song is finished. Storing a 0 in $6 will force a song to start from the beginning.

MUSPTR, the number in X in the Applesoft X = USR(songnum), or the number in the variable MUSPTR in Integer, resides in locations $7 and $8 (Low/High) in the Machine Language module. So:

```
LDY  #0
LDA  ($7),Y = Pitch (0-50; 0 = Rest)
LDY  #1
LDA  ($7),Y = Play Length (see Applesoft)
LDY  #2
LDA  ($7),Y = Rest Length (see Applesoft)
```

Zero Page and Other Memory Usage

Key:

Temp =	You can use it, but it will be changed upon return from a module.
Permanent =	You shouldn't change it except for a good reason.
User =	Parameter passing.

LOCATION	STATUS	REMARK
$6	permanent	FLAG
$7	permanent	MUSPTR Low
$8	permanent	MUSPTR High
$19	temp	Dummy
$1A	temp	Dummy
$1E	permanent	Previous Songnum
$2C	temp	Used by tone generator
$2D	temp	Same as $2C

other locations

$2FB	user	SONGNUM
$2FC	user	MOUTPUT (Used only when module is activated)
$2FD	temp	These last two are used by each song to pass
$2FE	temp	parameters to the tone generator.

NOTE: All processor registers (A, X, Y, P, and S) remain unchanged upon return from a Machine Language module (except when activating).