

0800 2 DCM "WRITE RWTS. LIST"
0800 3 ;

LISA 2.5

DOS 3.3 RWTS

0800	4	TTL "DOS 3.3 RWTS"
B800	5	ORG \$B800
B800	6	OBJ \$800
B800	7	:
0027	8	SLOTZ EPZ \$27
0027	9	CSUM EPZ \$27
0027	10	PRI OR EPZ \$27
0026	11	WTEMP EPZ \$26
0026	12	TO EPZ \$26
0026	13	IDX EPZ \$26
0026	14	COUNT EPZ \$26
0026	15	LAST EPZ \$26
0026	16	TRKCNT EPZ \$26
002F	17	VOLFND EPZ \$2F
002E	18	TRKFND EPZ \$2E
002D	19	SECFND EPZ \$2D
002B	20	SLOTEMP EPZ \$2B
002A	21	TRKN EPZ \$2A
0036	22	PTRSDEST EPZ \$36
0035	23	DRI VNO EPZ \$35
003E	24	BUFADR EPZ \$3E
003C	25	DEVCTBL EPZ \$3C
0047	26	MONTI MEH EPZ \$47
0046	27	MONTI MEL EPZ \$46
0045	28	SYNCNT EPZ \$45
0044	29	TRACK EPZ \$44
0041	30	VOLUME EPZ \$41
0049	31	I OBPH EPZ \$49
0048	32	I OBPL EPZ \$48
0478	33	DRVOTRK EQU \$478
0478	34	CURTRK EQU \$478
04F8	35	SEEKCNT EQU \$4F8
04F8	36	DRV1TRK EQU \$4F8
0578	37	RETRYCNT EQU \$578
05F8	38	SLOT EQU \$5F8
0678	39	SLOTABS EQU \$678
06F8	40	RECALCNT EQU \$6F8
A316	41	CLOSEALL EQU \$A316
A75B	42	RSETO EQU \$A75B
AA63	43	TEMP1 EQU \$AA63
AA70	44	BYTVAL EQU \$AA70
AAB7	45	RUNINTRC EQU \$AAB7
AE7E	46	SAVFMW EQU \$AE7E
B385	47	SETERROR EQU \$B385
B39B	48	STKSAVE EQU \$B39B
B744	49	RESTART EQU \$B744
C081	50	ROM EQU \$C081
C080	51	PHASEOFF EQU \$C080
C08F	52	LC08F EQU \$C08F
C08E	53	LC08E EQU \$C08E
C08D	54	LC08D EQU \$C08D
C08C	55	LC08C EQU \$C08C
C08B	56	LC08B EQU \$C08B
C08A	57	LC08A EQU \$C08A
C089	58	LC089 EQU \$C089
C088	59	LC088 EQU \$C088
E000	60	BASIC EQU \$E000
FE93	61	SETVID EQU \$FE93
B800	62	:
B800	63	:
B800	64	:

LISA 2.5

DOS 3.3 RWTS

```

B800      65 ; PRENI BBLI ZE ROUTINE
B800      66 ;
B800      67 ; CONVERTS 256 BYTES POINTED AT BY
B800      68 ; BUFADR TO 342 6-BIT NIBBLES
B800      69 ; OF THE FORM OXXXXXXX
B800      70 ;
B800 A2 00 71 PRENI B16 LDX #0
B802 A0 02 72           LDY #2
B804      73 ;
B804      74 ; Get next user byte
B804      75 ;
B804 88   76 PRENI B1 DEY
B805 B1 3E 77           LDA (BUFADR), Y
B807      78 ;
B807      79 ; Shift H.O. two bytes into
B807      80 ; NBUF2.
B807      81 ;
B807 4A    82           LSR
B808 3E 00 BC 83           ROL NBUF2, X
B80B 4A    84           LSR
B80C 3E 00 BC 85           ROL NBUF2, X
B80F      86 ;
B80F      87 ; Put L.O. six bits (shifted left)
B80F      88 ; into NBUF1.
B80F      89 ;
B80F 99 00 BB 90           STA NBUF1, Y
B812 E8    91           INX
B813 E0 56   92           CPX #$56
B815 90 ED   93           BCC PRENI B1
B817      94 ;
B817      95 ;
B817      96 ; Done yet?
B817      97 ;
B817 A2 00   98           LDX #0
B819 98    99           TYA
B81A D0 E8   100          BNE PRENI B1
B81C      101 ;
B81C      102 ;
B81C      103 ; strip H.O. two bits of NBUF2.
B81C      104 ;
B81C A2 55   105          LDX #$55
B81E BD 00 BC 106          PRENI B2 LDA NBUF2, X
B821 29 3F   107          AND #$3F
B823 9D 00 BC 108          STA NBUF2, X
B826 CA    109          DEX
B827 10 F5   110          BPL PRENI B2
B829 60    111          RTS
B82A      112 ;
B82A      113 ;
B82A      114 ; Write subroutine
B82A      115 ;
B82A      116 ; Writes prenibblized data in
B82A      117 ; NBUF1 and NBUF2 to disk.
B82A      118 ;
B82A      119 ; note: this stuff is all time
B82A      120 ; critical .
B82A      121 ;
B82A      122 ; watch page boundries, don't

```

LISA 2.5

DOS 3.3 RWTS

B82A	123	; remove NOP's, etc.	
B82A	124	;	
B82A	125	;	
B82A	126	;	
B82A 38	127	WRITE16 SEC	; anticipate write protect
B82B 86 27	128	STX SLOTZ	
B82D 8E 78 06	129	STX SLOTABS	
B830 BD 8D CO	130	LDA LC08D, X	
B833 BD 8E CO	131	LDA LC08E, X	; sense write protect
B836 30 7C	132	BMI WEXIT	
B838 AD 00 BC	133	LDA NBUF2	
B83B 85 26	134	STA WTEMP	
B83D A9 FF	135	LDA #SFF	; sync byte
B83F 9D 8F CO	136	STA LC08F, X	; write 1st nibble
B842 1D 8C CO	137	ORA LC08C, X	
B845 48	138	PHA	
B846 68	139	PLA	
B847 EA	140	NOP	
B848 A0 04	141	LDY #4	
B84A	142	:	
B84A	143	:	
B84A	144	; Write the sync byte	
B84A	145	:	
B84A 48	146	WSYNC PHA	
B84B 68	147	PLA	
B84C 20 B9 B8	148	JSR WNLBL7	
B84F 88	149	DEY	
B850 D0 F8	150	BNE WSYNC	
B852 A9 D5	151	LDA #SD5	; first data mark
B854 20 B8 B8	152	JSR WNLBL9	
B857 A9 AA	153	LDA #SAA	; 2nd data mark
B859 20 B8 B8	154	JSR WNLBL9	
B85C A9 AD	155	LDA #SAD	; 3rd data mark
B85E 20 B8 B8	156	JSR WNLBL9	
B861 98	157	TYA	; clear checksum
B862 A0 56	158	LDY #\$56	; NBUF2 index
B864 D0 03	159	BNE >1	; always taken
B866	160	:	
B866	161	:	
B866	162	:	
B866 B9 00 BC	163	WDATAO LDA NBUF2, Y	; get prior 6-bit nibble
B869 59 FF BB	164	^1 EOR NBUF2-1, Y	; and XOR with current nibble
B86C AA	165	TAX	
B86D BD 29 BA	166	LDA NIBL, X	
B870 A6 27	167	LDX SLOTZ	
B872 9D 8D CO	168	STA LC08D, X	; write nibble
B875 BD 8C CO	169	LDA LC08C, X	
B878 88	170	DEY	; next nibble
B879 D0 EB	171	BNE WDATAO	
B87B	172	:	
B87B	173	:	
B87B	174	:	
B87B	175	; now handle NBUF1.	
B87B	176	:	
B87B	177	:	
B87B	178	; get prior nibble	
B87B	179	:	
B87B A5 26	180	LDA WTEMP	

LISA 2.5

DOS 3.3 RWTS

```

B87D EA      181      NOP
B87E          182      ;
B87E          183      ; loop to write out data in NBUF1
B87E          184      ;
B87E 59 00 BB 185      WDATA2    EOR NBUF1, Y
B881 AA      186      TAX
B882 BD 29 BA 187      LDA NI BL, X
B885 AE 78 06 188      LDX SLOTABS
B888 9D 8D CO 189      STA LC08D, X      ; write nibble
B88B BD 8C CO 190      LDA LC08C, X
B88E B9 00 BB 191      LDA NBUF1, Y
B891 C8      192      INY
B892 D0 EA      193      BNE WDATA2
B894 AA      194      TAX
B895 BD 29 BA 195      LDA NI BL, X
B898          196      ;
B898          197      ; write checksum
B898          198      ;
B898 A6 27      199      LDX SLOTZ
B89A 20 BB B8 200      JSR WNI BL
B89D          201      ;
B89D          202      ; write epilog to data
B89D          203      ;
B89D A9 DE      204      LDA #$DE      ; DM4, bit slip mark
B89F 20 B8 B8 205      JSR WNI BL9
B8A2 A9 AA      206      LDA #$SAA      ; DM5, bsm
B8A4 20 B8 B8 207      JSR WNI BL9
B8A7 A9 EB      208      LDA #$SEB      ; DM6, bsm
B8A9 20 B8 B8 209      JSR WNI BL9
B8AC          210      ;
B8AC          211      ; All done, close up the shop!
B8AC          212      ;
B8AC A9 FF      213      LDA #$FF
B8AE 20 B8 B8 214      JSR WNI BL9
B8B1          215      ;
B8B1          216      ; turn off the write mode
B8B1          217      ;
B8B1 BD 8E CO 218      LDA LC08E, X
B8B4          219      ;
B8B4          220      ; and back to the read mode
B8B4          221      ;
B8B4 BD 8C CO 222      WEXIT    LDA LC08C, X
B8B7 60          223      RTS
B8B8          224      ;
B8B8          225      ;
B8B8          226      ;
B8B8          227      ; WNI BL9 9 cycles, then write
B8B8          228      ;
B8B8 18          229      WNI BL9    CLC
B8B9          230      ;
B8B9          231      ; WNI BL7 7 cycles, then write
B8B9          232      ;
B8B9 48          233      WNI BL7    PHA
B8BA 68          234      PLA
B8BB          235      ;
B8BB          236      ;
B8BB          237      ; write nibble to disk
B8BB          238      ;

```

LISA 2.5

DOS 3.3 RWTS

```

B8BB 9D 8D C0 239 WNI BL STA LC08D, X
B8BE 1D 8C C0 240 ORA LC08C, X
B8C1 60 241 RTS
B8C2 242 ;
B8C2 243 ;
B8C2 244 ; Post nibblize routine.
B8C2 245 ;
B8C2 246 ; converts 342 nibbles of the form
B8C2 247 ;
B8C2 248 ;      00xxxxxx
B8C2 249 ;
B8C2 250 ; to eight bit bytes.
B8C2 251 ;
B8C2 252 ; The nibbles are stored in NBUF1
B8C2 253 ; and NBUF2, the 8-bit bytes will
B8C2 254 ; be stored at (BUFADR).
B8C2 255 ;
B8C2 256 ;
B8C2 A0 00 257 POSTNB16 LDY #0
B8C4 A2 56 258 POST1 LDX #$56
B8C6 CA 259 POST2 DEX
B8C7 30 FB 260 BMI POST1
B8C9 261 ;
B8C9 262 ; get byte and shift in L.O. two
B8C9 263 ; bits from NBUF2
B8C9 264 ;
B8C9 B9 00 BB 265 LDA NBUF1, Y
B8CC 5E 00 BC 266 LSR NBUF2, X
B8CF 2A 267 ROL
B8D0 5E 00 BC 268 LSR NBUF2, X
B8D3 2A 269 ROL
B8D4 270 ;
B8D4 271 ; store in user buffer
B8D4 272 ;
B8D4 91 3E 273 STA (BUFADR), Y
B8D6 C8 274 INY
B8D7 C4 26 275 CPY TO
B8D9 D0 EB 276 BNE POST2
B8DB 60 277 RTS
B8DC 278 ;
B8DC 279 ;
B8DC 280 ; READ routine, reads a sector
B8DC 281 ; from the disk and stores the
B8DC 282 ; data in NBUF1 and NBUF2.
B8DC 283 ;
B8DC A0 20 284 READ16 LDY #$20
B8DE 285 ;
B8DE 286 ; get sync bytes
B8DE 287 ;
B8DE 88 288 RSYNC DEY
B8DF F0 61 289 BEQ RDERR
B8E1 290 ;
B8E1 291 ; wait until a byte is received
B8E1 292 ; from the disk drive.
B8E1 293 ;
B8E1 BD 8C C0 294 ^1 LDA LC08C, X
B8E4 10 FB 295 BPL <1
B8E6 296 ;

```

LISA 2.5

DOS 3.3 RWTS

```

B8E6          297 ; byte received, check for DM1
B8E6          298 ;
B8E6 49 D5   299 RSYNC1 EOR #$D5
B8E8 D0 F4   300     BNE RSYNC
B8EA EA     301     NOP
B8EB          302 ;
B8EB          303 ; get next byte and check for
B8EB          304 ; DM2
B8EB          305 ;
B8EB BD 8C C0 306 ^2     LDA LC08C, X
B8EE 10 FB   307     BPL <2
B8F0 C9 AA   308     CMP #SAA
B8F2 D0 F2   309     BNE RSYNC1
B8F4 A0 56   310     LDY #$S6
B8F6          311 ;
B8F6          312 ; Check for DM3
B8F6          313 ;
B8F6 BD 8C C0 314 ^3     LDA LC08C, X
B8F9 10 FB   315     BPL <3
B8FB C9 AD   316     CMP #SAD
B8FD D0 E7   317     BNE RSYNC1
B8FF          318 ;
B8FF          319 ; Read the data from the sector
B8FF          320 ;
B8FF A9 00   321     LDA #0           ; init checksum
B901          322 ;
B901          323 ; Read stuff into NBUF2
B901          324 ;
B901 88      325 RDATA1 DEY
B902 84 26   326     STY IDX
B904 BC 8C C0 327 ^4     LDY LC08C, X
B907 10 FB   328     BPL <4
B909 59 00 BA 329     EOR MSWAIT, Y
B90C A4 26   330     LDY IDX
B90E 99 00 BC 331     STA NBUF2, Y
B911 D0 EE   332     BNE RDATA1
B913          333 ;
B913          334 ; Read stuff into NBUF1
B913          335 ;
B913 84 26   336 RDATA2 STY IDX
B915 BC 8C C0 337 ^5     LDY LC08C, X
B918 10 FB   338     BPL <5
B91A 59 00 BA 339     EOR MSWAIT, Y
B91D A4 26   340     LDY IDX
B91F 99 00 BB 341     STA NBUF1, Y
B922 C8      342    INY
B923 D0 EE   343     BNE RDATA2
B925          344 ;
B925          345 ; get and check the checksum byte
B925          346 ;
B925 BC 8C C0 347 ^6     LDY LC08C, X
B928 10 FB   348     BPL <6
B92A D9 00 BA 349     CMP MSWAIT, Y
B92D D0 13   350     BNE RDERR
B92F          351 ;
B92F          352 ; check for DM4
B92F          353 ;
B92F BD 8C C0 354 ^7     LDA LC08C, X

```

LISA 2.5

DOS 3.3 RWTS

```

B932 10 FB      355      BPL <7
B934 C9 DE      356      CMP #$DDE
B936 D0 OA      357      BNE RDERR
B938 EA         358      NOP
B939             359      ;
B939             360      ; check for DM5
B939             361      ;
B939 BD 8C CO   362 ^8     LDA LC08C, X
B93C 10 FB      363      BPL <8
B93E C9 AA      364      CMP #$SAA
B940 F0 5C      365      BEQ RDADRX
B942 38         366      RDERR SEC
B943 60         367      RTS
B944             368      ;
B944             369      ;
B944             370      ; Read address field.
B944             371      ;
B944             372      ; Reads starting address marks
B944             373      ; ($D5, $AA, $96), address info
B944             374      ; (volume/track/sector/checksum),
B944             375      ; and closing address marks
B944             376      ; ($DDE, $SAA)
B944             377      ;
B944 A0 FC      378      RDADR16 LDY #$FC
B946 84 26      379      STY COUNT
B948 C8         380      RDASYN INY
B949 D0 04      381      BNE >0
B94B E6 26      382      INC COUNT
B94D F0 F3      383      BEQ RDERR
B94F             384      ;
B94F             385      ; Read first address mark ($D5)
B94F             386      ;
B94F BD 8C CO   387 ^0     LDA LC08C, X
B952 10 FB      388      BPL <0
B954 C9 D5      389      RDASYN1 CMP #$D5
B956 D0 F0      390      BNE RDASYN
B958 EA         391      NOP
B959             392      ;
B959             393      ; Read next address mark
B959             394      ;
B959 BD 8C CO   395 ^1     LDA LC08C, X
B95C 10 FB      396      BPL <1
B95E C9 AA      397      CMP #$SAA
B960 D0 F2      398      BNE RDASYN1
B962 A0 03      399      LDY #3
B964             400      ;
B964             401      ; Read last address mark
B964             402      ;
B964 BD 8C CO   403 ^2     LDA LC08C, X
B967 10 FB      404      BPL <2
B969 C9 96      405      CMP #$96
B96B D0 E7      406      BNE RDASYN1
B96D             407      ;
B96D             408      ; init checksum and read the
B96D             409      ; address data field (four bytes)
B96D             410      ;
B96D A9 00      411      LDA #0
B96F 85 27      412      RDAFLD STA CSUM

```

LISA 2.5

DOS 3.3 RWTS

```

B971      413   ;
B971      414   ; Read 'odd' bit nibble
B971      415   ;
B971 BD 8C CO 416 ^3     LDA LC08C, X
B974 10 FB  417     BPL <3
B976 2A    418     ROL
B977 85 26  419     STA LAST
B979      420   ;
B979      421   ; Read even bit nibble.
B979      422   ;
B979 BD 8C CO 423 ^4     LDA LC08C, X
B97C 10 FB  424     BPL <4
B97E      425   ;
B97E      426   ; Merge the two.
B97E      427   ;
B97E 25 26  428     AND LAST
B980      429   ;
B980      430   ; store the data byte, update
B980      431   ; checksum, and repeat until
B980      432   ; entire address field is read.
B980      433   ;
B980 99 2C 00 434     STA SLOTTEMP+1, Y
B983 45 27   435     EOR CSUM
B985 88    436     DEY
B986 10 E7   437     BPL RDALFD
B988      438   ;
B988      439   ; Checksum (in acc) must be 0.
B988      440   ;
B988 A8    441     TAY
B989 D0 B7   442     BNE RDERR
B98B      443   ;
B98B      444   ; Read first epilogue byte ($DE)
B98B      445   ;
B98B BD 8C CO 446 ^5     LDA LC08C, X
B98E 10 FB   447     BPL <5
B990 C9 DE   448     CMP #SDE
B992 D0 AE   449     BNE RDERR
B994 EA    450     NOP
B995      451   ;
B995      452   ; Read second epilogue byte ($AA)
B995      453   ;
B995 BD 8C CO 454 ^6     LDA LC08C, X
B998 10 FB   455     BPL <6
B99A C9 AA   456     CMP #SAA
B99C D0 A4   457     BNE RDERR
B99E 18    458     RDADRX CLC
B99F 60    459     RTS
B9A0      460   ;
B9A0      461   ;
B9A0      462   ; SEEKABS routine, moves the
B9A0      463   ; disk head over the desired
B9A0      464   ; track.
B9A0      465   ;
B9A0 86 2B   466     SEEKABS STX SLOTTEMP
B9A2 85 2A   467     STA TRKN
B9A4 CD 78 04 468     CMP DRVOTRK
B9A7 F0 53   469     BEQ RTS0
B9A9 A9 00   470     LDA #0

```

LISA 2.5

DOS 3.3 RWTS

B9AB 85 26	471	STA TRKCNT
B9AD AD 78 04	472	MOVEHEAD LDA DRVOTRK
B9B0 85 27	473	STA PRI OR
B9B2 38	474	SEC
B9B3 E5 2A	475	SBC TRKN
B9B5 F0 33	476	BEQ ISTHERE
B9B7 B0 07	477	BCS >0
B9B9 49 FF	478	EOR #\$FF
B9BB EE 78 04	479	INC DRVOTRK
B9BE 90 05	480	BCC >1
B9C0 69 FE	481 ^0	ADC #\$FE
B9C2 CE 78 04	482	DEC DRVOTRK
B9C5 C5 26	483 ^1	CMP TRKCNT
B9C7 90 02	484	BCC >2
B9C9 A5 26	485	LDA TRKCNT
B9CB C9 0C	486 ^2	CMP #\$C
B9CD B0 01	487	BCS >3
B9CF A8	488	TAY
B9D0 38	489 ^3	SEC
B9D1 20 EE B9	490	JSR CHKPOS
B9D4 B9 11 BA	491	LDA ONTBL, Y
B9D7 20 00 BA	492	JSR MSWAIT
B9DA A5 27	493	LDA PRI OR
B9DC 18	494	CLC
B9DD 20 F1 B9	495	JSR CHKPOS2
B9E0 B9 1D BA	496	LDA OFFTBL, Y
B9E3 20 00 BA	497	JSR MSWAIT
B9E6 E6 26	498	INC TRKCNT
B9E8 D0 C3	499	BNE MOVEHEAD
B9EA 20 00 BA	500	ISTHERE JSR MSWAIT
B9ED 18	501	CLC
B9EE AD 78 04	502	CHKPOS LDA DRVOTRK
B9F1 29 03	503	CHKPOS2 AND #3
B9F3 2A	504	ROL
B9F4 05 2B	505	ORA SLOTTEMP
B9F6 AA	506	TAX
B9F7 BD 80 CO	507	LDA PHASEOFF, X
B9FA A6 2B	508	LDX SLOTTEMP
B9FC 60	509 RTSO	RTS
B9FD AA	510	TAX
B9FE A0 A0	511	LDY #\$AO
BA00	512 ;	
BA00	513 ;	
BA00	514 ; Head move delay subroutine	
BA00	515 ; delays ACC*100 usec	
BA00	516 ;	
BA00	517 ;	
BA00 A2 11	518 MSWAIT LDX #\$11	
BA02 CA	519 ^2 DEX ; delay 86 usec	
BA03 D0 FD	520 BNE <2	
BA05 E6 46	521 INC MONTI MEL	
BA07 D0 02	522 BNE >3	
BA09 E6 47	523 INC MONTI MEH	
BA0B 38	524 ^3 SEC	
BA0C E9 01	525 SBC #1	
BA0E D0 F0	526 BNE MSWAIT	
BA10 60	527 RTS	
BA11	528 ;	

LISA 2.5

DOS 3.3 RWTS

BA11 529 ;
BA11 530 ; PHASEON/PHASEOFF tables
BA11 531 ;
BA11 01 30 532 ONTBL HEX 0130
BA13 28 24 20 533 HEX 2824201E1D1C1C1C1C1C
BA16 1E 1D 1C
BA19 1C 1C 1C
BA1C 1C
BA1D 70 2C 26 534 OFFTBL HEX 702C26221F1E1D1C1C1C1C1C
BA20 22 1F 1E
BA23 1D 1C 1C
BA26 1C 1C 1C
BA29 535 ;
BA29 536 ;
BA29 537 ; Write translate tables
BA29 538 ;
BA29 96 97 9A 539 NI BL HEX 96979A9B9D9E9FA6A7ABACADAEAFB2B3
BA2C 9B 9D 9E
BA2F 9F A6 A7
BA32 AB AC AD
BA35 AE AF B2
BA38 B3
BA39 B4 B5 B6 540 HEX B4B5B6B7B9BABBCBDBEBFCBCDCECFD3
BA3C B7 B9 BA
BA3F BB BC BD
BA42 BE BF CB
BA45 CD CE CF
BA48 D3
BA49 D6 D7 D9 541 HEX D6D7D9DADBCDDDEDFE5E6E7E9EAEBEC
BA4C DA DB DC
BA4F DD DE DF
BA52 E5 E6 E7
BA55 E9 EA EB
BA58 EC
BA59 ED EE EF 542 HEX EDEEEFF2F3F4F5F6F7F9FAFBFCFDFFEFF
BA5C F2 F3 F4
BA5F F5 F6 F7
BA62 F9 FA FB
BA65 FC FD FE
BA68 FF
BA69 543 ;
BA69 544 ; ALIGN REMAINDER OF TABLE
BA69 545 ;
BA69 546 DFS 45
BA96 547 ;
BA96 548 ;
BA96 00 01 98 549 HEX 0001989902039C040506A0A1A2A3A4A5
BA99 99 02 03
BA9C 9C 04 05
BA9F 06 A0 A1
BAA2 A2 A3 A4
BAA5 A5
BAA6 07 08 A8 550 HEX 0708A8A9AA090AOBOCODBOB10EOF1011
BAA9 A9 AA 09
BAAC OA OB OC
BAAF OD BO B1
BAB2 OE OF 10
BAB5 11

LISA 2.5

DOS 3.3 RWTS

BAB6 12 13 B8	551	HEX 1213B81415161718191AC0C1C2C3C4C5
BAB9 14 15 16		
BABC 17 18 19		
BABF 1A C0 C1		
BAC2 C2 C3 C4		
BAC5 C5		
BAC6 C6 C7 C8	552	HEX C6C7C8C9CA1BCC1C1D1ED0D1D21FD4D5
BAC9 C9 CA 1B		
BACC CC 1C 1D		
BACF 1E D0 D1		
BAD2 D2 1F D4		
BAD5 D5		
BAD6 20 21 D8	553	HEX 2021D822232425262728E0E1E2E3E429
BAD9 22 23 24		
BADC 25 26 27		
BADF 28 E0 E1		
BAE2 E2 E3 E4		
BAE5 29		
BAE6 2A 2B E8	554	HEX 2A2BE82C2D2E2F303132F0F133343536
BAE9 2C 2D 2E		
BAEC 2F 30 31		
BAEF 32 F0 F1		
BAF2 33 34 35		
BAF5 36		
BAF6 37 38 F8	555	HEX 3738F8393A3B3C3D3E3F
BAF9 39 3A 3B		
BAFC 3C 3D 3E		
BAFF 3F		
BB00	556	:
BB00	557	:
BB00	558	:
BB00	559	; nibble buffers, must be in this
BB00	560	; order!
BB00	561	:
BB00	562	NBUF1 DFS \$100
BC00	563	NBUF2 DFS \$56
BC56	564	:
BC56	565	:
BC56	566	:
BC56	567	; Write an address field routine
BC56	568	:
BC56	569	:
BC56 38	570	WRADR16 SEC ; assume W/P error
BC57 BD 8D C0	571	LDA LC08D, X
BC5A BD 8E C0	572	LDA LC08E, X
BC5D 30 5E	573	BMI WPERROR
BC5F	574	:
BC5F	575	; output sync byte
BC5F	576	:
BC5F A9 FF	577	LDA #\$FF
BC61 9D 8F C0	578	STA LC08F, X
BC64 DD 8C C0	579	CMP LC08C, X
BC67 48	580	PHA
BC68 68	581	PLA
BC69 20 C3 BC	582	FRMSYNC JSR WAIT12
BC6C 20 C3 BC	583	JSR WAIT12
BC6F 9D 8D C0	584	STA LC08D, X
BC72 DD 8C C0	585	CMP LC08C, X

LISA 2.5

DOS 3.3 RWTS

BC75 EA	586	NOP
BC76 88	587	DEY
BC77 D0 F0	588	BNE FRMSYNC
BC79	589	:
BC79	590	; output data marks
BC79	591	:
BC79 A9 D5	592	LDA #SD5
BC7B 20 D5 BC	593	JSR WBYTE9
BC7E A9 AA	594	LDA #\$A9
BC80 20 D5 BC	595	JSR WBYTE9
BC83 A9 96	596	LDA #\$96
BC85 20 D5 BC	597	JSR WBYTE9
BC88	598	:
BC88	599	; output volume track sector
BC88	600	; and checksum
BC88	601	:
BC88 A5 41	602	LDA VOLUME
BC8A 20 C4 BC	603	JSR WBYTE
BC8D A5 44	604	LDA TRACK
BC8F 20 C4 BC	605	JSR WBYTE
BC92 A5 3F	606	LDA BUFADR+1
BC94 20 C4 BC	607	JSR WBYTE
BC97 A5 41	608	LDA VOLUME
BC99 45 44	609	EOR TRACK
BC9B 45 3F	610	EOR BUFADR+1
BC9D 48	611	PHA
BC9E 4A	612	LSR
BC9F 05 3E	613	ORA BUFADR
BCA1 9D 8D CO	614	STA LC08D, X
BCA4 BD 8C CO	615	LDA LC08C, X
BCA7 68	616	PLA
BCA8 09 AA	617	ORA #\$AA
BCAA 20 D4 BC	618	JSR WBYTE11
BCAD	619	:
BCAD	620	; output data marks 4, 5, and 6
BCAD	621	:
BCAD A9 DE	622	LDA #SDE
BCAF 20 D5 BC	623	JSR WBYTE9
BCB2 A9 AA	624	LDA #\$AA
BCB4 20 D5 BC	625	JSR WBYTE9
BCB7 A9 EB	626	LDA #\$EB
BCB9 20 D5 BC	627	JSR WBYTE9
BCBC 18	628	CLC
BCBD BD 8E CO	629	WPERROR LDA LC08E, X
BCC0 BD 8C CO	630	LDA LC08C, X
BCC3 60	631	WAIT12 RTS
BCC4	632	:
BCC4	633	:
BCC4	634	; Write a byte as two four bit
BCC4	635	; nibbles to the disk.
BCC4	636	:
BCC4 48	637	WBYTE PHA
BCC5 4A	638	LSR
BCC6 05 3E	639	ORA BUFADR
BCC8	640	:
BCC8	641	; write odd bits
BCC8	642	:
BCC8 9D 8D CO	643	STA LC08D, X

LISA 2.5

DOS 3.3 RWTS

```

BCCB DD 8C CO 644      CMP LC08C, X
BCCE 68 645      PLA
BCCF EA 646      NOP
BCDO EA 647      NOP
BCD1 EA 648      NOP
BCD2 09 AA 649      ORA #SAA
BCD4 EA 650      WBYTE11 NOP
BCD5 EA 651      WBYTE9  NOP
BCD6 48 652      PHA
BCD7 68 653      PLA
BCD8 654      ;
BCD8 655      ; write even bits
BCD8 656      ;
BCD8 9D 8D CO 657      STA LC08D, X
BCDB DD 8C CO 658      CMP LC08C, X
BCDE 60 659      RTS
BCDF 660      ;
BCDF 661      ;
BCDF 662      ; FORCE RWTS TO PAGE BOUNDARY
BCDF 663      ;
BCDF 664      DFS 33
BD00 665      ;
BD00 666      ;
BD00 667      ;
BD00 668      ; RWTS entry point
BD00 669      ;
BD00 670      ;
BD00 671      RWTS:
BD00 672      ;
BD00 673      ; upon entry, A & Y point at IOB
BD00 674      ;
BD00 84 48 675      STY IOBPL
BD02 85 49 676      STA IOBPH
BD04 677      ;
BD04 678      ; set up for one recal and 4 seeks
BD04 679      ;
BD04 A0 02 680      LDY #2
BD06 8C F8 06 681      STY RECALCNT
BD09 A0 04 682      LDY #4
BD0B 8C F8 04 683      STY SEEKCNT
BDOE 684      ;
BDOE 685      ; get slot #
BDOE 686      ;
BDOE A0 01 687      LDY #1
BD10 B1 48 688      LDA (IOBPL), Y
BD12 AA 689      TAX
BD13 690      ;
BD13 691      ;
BD13 692      ; see if slot # was changed
BD13 693      ;
BD13 A0 0F 694      LDY #$F
BD15 D1 48 695      CMP (IOBPL), Y
BD17 F0 1B 696      BEQ SAMESLOT
BD19 697      ;
BD19 698      ; if so, turn off old drive
BD19 699      ;
BD19 8A 700      TXA
BD1A 48 701      PHA

```

LISA 2.5

DOS 3.3 RWTS

BD1B B1 48	702	LDA (IOBPL), Y
BD1D AA	703	TAX
BD1E 68	704	PLA
BD1F 48	705	PHA
BD20 91 48	706	STA (IOBPL), Y
BD22 BD 8E CO	707	LDA LC08E, X
BD25	708	:
BD25	709	; delay until data is constant
BD25	710	:
BD25 A0 08	711	STI LLON LDY #8
BD27 BD 8C CO	712	LDA LC08C, X
BD2A DD 8C CO	713	NOTSURE CMP LC08C, X
BD2D DO F6	714	BNE STI LLON
BD2F 88	715	DEY
BD30 DO F8	716	BNE NOTSURE
BD32 68	717	PLA
BD33 AA	718	TAX
BD34	719	:
BD34	720	:
BD34	721	; start the motor up
BD34	722	:
BD34 BD 8E CO	723	SAMESLOT LDA LC08E, X
BD37 BD 8C CO	724	LDA LC08C, X
BD3A A0 08	725	LDY #8
BD3C BD 8C CO	726	NOTSURE2 LDA LC08C, X
BD3F 48	727	PHA
BD40 68	728	PLA
BD41 48	729	PHA
BD42 68	730	PLA
BD43 8E F8 05	731	STX SLOT
BD46 DD 8C CO	732	CMP LC08C, X
BD49 DO 03	733	BNE >3
BD4B 88	734	DEY
BD4C DO EE	735	BNE NOTSURE2
BD4E 08	736	PHP
BD4F BD 89 CO	737	LDA LC089, X
BD52	738	:
BD52	739	; move necessary pointers to
BD52	740	; page zero
BD52	741	:
BD52 A0 06	742	LDY #6
BD54 B1 48	743	PTRMOVE LDA (IOBPL), Y
BD56 99 36 00	744	STA PTRSDEST, Y
BD59 C8	745	INY
BD5A CO 0A	746	CPY #SA
BD5C DO F6	747	BNE PTRMOVE
BD5E	748	:
BD5E	749	; get other parameters
BD5E	750	:
BD5E A0 03	751	LDY #3
BD60 B1 3C	752	LDA (DEVCTBL), Y
BD62 85 47	753	STA MONTIMEH
BD64 A0 02	754	LDY #2
BD66 B1 48	755	LDA (IOBPL), Y
BD68 A0 10	756	LDY #\$10
BD6A D1 48	757	CMP (IOBPL), Y
BD6C F0 06	758	BEQ >4
BD6E 91 48	759	STA (IOBPL), Y

LISA 2.5

DOS 3.3 RWTS

BD70 28	760	PLP
BD71 A0 00	761	LDY #0
BD73 08	762	PHP
BD74 6A	763	^{^4} ROR
BD75 90 05	764	BCC >5
BD77 BD 8A C0	765	LDA LC08A, X
BD7A B0 03	766	BCS DRVSEL
BD7C BD 8B C0	767	^{^5} LDA LC08B, X
BD7F	768	:
BD7F	769	; save which drive is being used
BD7F	770	:
BD7F 66 35	771	DRVSEL ROR DRI VNO
BD81 28	772	PLP
BD82 08	773	PHP
BD83 D0 0B	774	BNE NOWAIT
BD85 A0 07	775	LDY #7
BD87	776	:
BD87	777	; wait 100 usec for old drive's
BD87	778	; timing capacitor to discharge
BD87	779	:
BD87 20 00 BA	780	STEPWAIT JSR MSWAIT
BD8A 88	781	DEY
BD8B D0 FA	782	BNE STEPWAIT
BD8D AE F8 05	783	LDX SLOT
BD90 A0 04	784	NOWAIT LDY #4
BD92 B1 48	785	LDA (IOBPL), Y
BD94 20 5A BE	786	JSR MYSEEK
BD97 28	787	PLP
BD98 D0 11	788	BNE TRYTRK
BD9A A4 47	789	LDY MONTIMEH
BD9C 10 0D	790	BPL TRYTRK
BD9E	791	:
BD9E	792	; wait for motor to come up to
BD9E	793	; speed.
BD9E	794	:
BD9E A0 12	795	MOTOROFF LDY #\$12
BDAO 88	796	^{^6} DEY
BDA1 D0 FD	797	BNE <6
BDA3 E6 46	798	INC MONTIMEL
BDA5 D0 F7	799	BNE MOTOROFF
BDA7 E6 47	800	INC MONTIMEH
BDA9 D0 F3	801	BNE MOTOROFF
BDAB	802	:
BDAB	803	; disk is now up to speed.
BDAB	804	; if not format operation,
BDAB	805	; position the head over the
BDAB	806	; proper track
BDAB	807	:
BDAB A0 OC	808	TRYTRK LDY #SC
BDAD B1 48	809	LDA (IOBPL), Y
BDAF F0 5A	810	BEQ GALLDONE
BDB1 C9 04	811	CMP #4
BDB3 F0 58	812	BEQ FORMDSK
BDB5 6A	813	ROR
BDB6 08	814	PHP
BDB7 B0 03	815	BCS TRYTRK2
BDB9	816	:
BDB9	817	; if a read operation, must

LISA 2.5

DOS 3.3 RWTS

BDB9 818 ; prenibblize first.
BDB9 819 ;
BDB9 20 00 B8 820 JSR PRENIB16
BDBC 821 ;
BDBC 822 ;
BDBC 823 ; set up for a maximum of 48
BDBC 824 ; retries.
BDBC 825 ;
BDBC A0 30 826 TRYTRK2 LDY #\$30
BDBE 8C 78 05 827 STY RETRYCNT
BDC1 AE F8 05 828 TRYADR LDX SLOT
BDC4 20 44 B9 829 JSR RDADR16
BDC7 90 24 830 BCC RDRI GHT
BDC9 CE 78 05 831 TRYADR2 DEC RETRYCNT
BDCC 10 F3 832 BPL TRYADR
BDCE 833 ;
BDCE 834 ; recalibrate disk head
BDCE 835 ;
BDCE AD 78 04 836 RECAL LDA CURTRK
BDD1 48 837 PHA
BDD2 A9 60 838 LDA #\$60
BDD4 20 95 BE 839 JSR SETTRK
BDD7 CE F8 06 840 DEC RECALCNT
BDDA F0 28 841 BEQ DRVERROR
BDDC A9 04 842 LDA #4
BDDE 8D F8 04 843 STA SEEKCNT
BDE1 A9 00 844 LDA #0
BDE3 20 5A BE 845 JSR MYSEEK
BDE6 68 846 PLA
BDE7 20 5A BE 847 RESEEK JSR MYSEEK
BDEA 4C BC BD 848 JMP TRYTRK2
BDED 849 ;
BDED 850 ;
BDED 851 ; We have just read an address
BDED 852 ; field, now check for desired
BDED 853 ; track, sector, and volume
BDED 854 ;
BDED A4 2E 855 RDRI GHT LDY TRKFND
BDEF CC 78 04 856 CPY CURTRK
BDF2 F0 1C 857 BEQ RTTRK
BDF4 858 ;
BDF4 859 ; Save dest track value
BDF4 860 ;
BDF4 AD 78 04 861 LDA CURTRK
BDF7 48 862 PHA
BDF8 98 863 TYA
BDF9 20 95 BE 864 JSR SETTRK
BDFC 68 865 PLA
BDFD CE F8 04 866 DEC SEEKCNT
BE00 D0 E5 867 BNE RESEEK
BE02 F0 CA 868 BEQ RECAL
BE04 869 ;
BE04 870 ; Bad drive error.
BE04 871 ;
BE04 68 872 DRVERROR PLA
BE05 A9 40 873 LDA #\$40
BE07 874 ;
BE07 875 ;

LISA 2.5

DOS 3.3 RWTS

```

BE07 28      876  JMPT01   PLP
BE08 4C 48 BE  877  JMP HNDLERR
BE0B F0 39      878  GALLDONE BEQ ALLDONE
BE0D 4C AF BE  879  FORMDSK  JMP DSKFORM
BE10          880  ;
BE10          881  ;
BE10          882  ; Drive is on the right track,
BE10          883  ; now check for a vol mismatch
BE10          884  ;
BE10 A0 03    885  RTTRK     LDY #3
BE12 B1 48    886  LDA (IOBPL), Y      ; get desired volume
BE14 48       887  PHA
BE15          888  ;
BE15          889  ; Save volume actually found in
BE15          890  ; RWTS IOB.
BE15          891  ;
BE15 A5 2F    892  LDA VOLFND
BE17 A0 0E    893  LDY #SE
BE19 91 48    894  STA (IOBPL), Y
BE1B          895  ;
BE1B          896  ; If volume specified was zero,
BE1B          897  ; no error.
BE1B          898  ;
BE1B 68       899  PLA
BE1C F0 08    900  BEQ CRCTVOL
BE1E          901  ;
BE1E          902  ; Otherwise, check for a volume
BE1E          903  ; mismatch error.
BE1E          904  ;
BE1E C5 2F    905  CMP VOLFND
BE20 F0 04    906  BEQ CRCTVOL
BE22 A9 20    907  LDA #$20
BE24 D0 E1    908  BNE JMPT01
BE26          909  ;
BE26          910  ; now check for the correct sector
BE26          911  ;
BE26 A0 05    912  CRCTVOL LDY #5
BE28 B1 48    913  LDA (IOBPL), Y      ; get the sector #
BE2A          914  ;
BE2A          915  ; Convert to a "soft" sector
BE2A          916  ; number by applying the software
BE2A          917  ; interleave.
BE2A          918  ;
BE2A A8       919  TAY
BE2B B9 B8 BF 920  LDA INTRLEAV, Y
BE2E          921  ;
BE2E          922  ; Are we at that sector yet?
BE2E          923  ;
BE2E C5 2D    924  CMP SECFND
BE30 D0 97    925  BNE TRYADR2
BE32          926  ;
BE32          927  ; If so, see if we are doing a
BE32          928  ; read or a write.
BE32          929  ;
BE32 28       930  PLP
BE33 90 1C    931  BCC WRI T
BE35          932  ;
BE35          933  ; Reading, so read in the 256

```

LISA 2.5

DOS 3.3 RWTS

```

BE35      934 ; bytes of data that follow.
BE35      935 ;
BE35 20 DC B8 936      JSR READ16
BE38 08   937      PHP
BE39 B0 8E  938      BCS TRYADR2
BE3B 28   939      PLP
BE3C A2 00  940      LDX #0
BE3E 86 26  941      STX TO
BE40      942 ;
BE40      943 ; Convert the nibbles to bytes.
BE40      944 ;
BE40 20 C2 B8 945      JSR POSTNB16
BE43 AE F8 05 946      LDX SLOT
BE46 18   947 ALLDONE CLC
BE47 24   948      BYT $24      ; BIT L38
BE48 38   949 HNDLERR SEC      ; OPCODE SKIPPED BY BIT L38
BE49 A0 OD   950      LDY #$D
BE4B 91 48   951      STA (IOBPL), Y
BE4D BD 88 C0  952      LDA LC088, X
BE50 60   953      RTS
BE51      954 ;
BE51      955 ; Performing a write, write the
BE51      956 ; data (already nibblized) to
BE51      957 ; the following data sector.
BE51      958 ;
BE51 20 2A B8 959 WRT      JSR WRITE16
BE54 90 F0   960      BCC ALLDONE
BE56 A9 10   961      LDA #$10
BE58 BO EE   962      BCS HNDLERR
BE5A      963 ;
BE5A      964 ;
BE5A      965 ; MYSEEK is the seek routine,
BE5A      966 ; it seeks track 'N' in slot
BE5A      967 ; X/16
BE5A      968 ;
BE5A      969 ; If DRVNO is negative - drive 0
BE5A      970 ; If DRVNO is positive - drive 1
BE5A      971 ;
BE5A 48    972 MYSEEK PHA
BE5B A0 01  973      LDY #1
BE5D B1 3C  974      LDA (DEVCTBL), Y
BE5F 6A   975      ROR
BE60 68   976      PLA
BE61 90 08  977      BCC SEEK1
BE63 0A   978      ASL
BE64 20 6B BE 979      JSR SEEK1
BE67 4E 78 04 980      LSR CURTRK
BE6A 60   981      RTS
BE6B 85 2A   982 SEEK1 STA TRKN
BE6D 20 8E BE 983      JSR XTOY
BE70 B9 78 04 984      LDA DRVOTRK, Y
BE73 24 35   985      BIT DRVNO
BE75 30 03   986      BMI WASDO
BE77 B9 F8 04 987      LDA DRV1TRK, Y
BE7A 8D 78 04 988 WASDO STA DRVOTRK
BE7D A5 2A   989      LDA TRKN
BE7F 24 35   990      BIT DRVNO
BE81 30 05   991      BMI ISDRV0

```

LISA 2.5

DOS 3.3 RWTS

```

BE83 99 F8 04    992      STA DRV1TRK, Y
BE86 10 03        993      BPL GOSEEK
BE88 99 78 04    994      ISDRV0 STA DRVOTRK, Y
BE8B 4C A0 B9    995      GOSEEK JMP SEEKABS
BE8E 8A          996      XTOY TXA
BE8F 4A          997      LSR
BE90 4A          998      LSR
BE91 4A          999      LSR
BE92 4A          1000     LSR
BE93 A8          1001     TAY
BE94 60          1002     RTS
BE95            1003     :
BE95            1004     :
BE95            1005     ; This routine sets the slot
BE95            1006     ; dependant track location
BE95            1007     :
BE95 48          1008     SETTRK PHA
BE96 A0 02        1009     LDY #2
BE98 B1 48        1010     LDA (IOBPL), Y
BE9A 6A          1011     ROR
BE9B 66 35        1012     ROR DRI VNO
BE9D 20 8E BE    1013     JSR XTOY
BEAO 68          1014     PLA
BEA1 0A          1015     ASL
BEA2 24 35        1016     BIT DRI VNO
BEA4 30 05        1017     BMI ONDRV0
BEA6 99 F8 04    1018     STA DRV1TRK, Y
BEA9 10 03        1019     BPL RTS3
BEAB 99 78 04    1020     ONDRV0 STA DRVOTRK, Y
BEAE 60          1021     RTS3 RTS
BEAF            1022     :
BEAF            1023     :
BEAF            1024     ; This is the disk formatter
BEAF            1025     ; routine.
BEAF            1026     :
BEAF A0 03        1027     DSKFORM LDY #3
BEB1 B1 48        1028     LDA (IOBPL), Y
BEB3 85 41        1029     STA VOLUME
BEB5            1030     :
BEB5            1031     ; Save timing constant in zero
BEB5            1032     ; page (for time critical section)
BEB5            1033     :
BEB5 A9 AA        1034     LDA #SAA
BEB7 85 3E        1035     STA BUFADR
BEB9 A0 56        1036     LDY #S56
BEBB            1037     :
BEBB            1038     ; Set up to start a track zero.
BEBB            1039     :
BEBB A9 00        1040     LDA #0
BEBD 85 44        1041     STA TRACK
BEBF            1042     :
BEBF            1043     :
BEBF            1044     ; zero secondary buffer.
BEBF            1045     :
BEBF 99 FF BB    1046     ^9      STA NBUF2-1, Y
BEC2 88          1047     DEY
BEC3 D0 FA        1048     BNE <9
BEC5            1049     :

```

LISA 2.5

DOS 3.3 RWTS

BEC5 1050 ; zero primary buffer.
BEC5 1051 ;
BEC5 99 00 BB 1052 ^0 STA NBUF1, Y
BEC8 88 1053 DEY
BEC9 D0 FA 1054 BNE <0
BECB 1055 ;
BECB 1056 ; Pretend we are at track 40
BECB 1057 ; (acc=2*maxtracks)
BECB 1058 ;
BECB A9 50 1059 LDA #\$50
BECB 20 95 BE 1060 JSR SETTRK
BEDO 1061 ;
BEDO 1062 ; Start with 40 bytes of self
BEDO 1063 ; sync bytes.
BEDO 1064 ;
BEDO A9 28 1065 LDA #\$28
BED2 85 45 1066 STA SYNCNT
BED4 1067 ;
BED4 1068 ; Got to the track and format it.
BED4 1069 ;
BED4 A5 44 1070 NXTTRK LDA TRACK
BED6 20 5A BE 1071 JSR MYSEEK
BED9 20 OD BF 1072 JSR DISKF2
BEDC 1073 ;
BEDC 1074 ; Init in case of DISK I/O error.
BEDC 1075 ;
BEDC A9 08 1076 LDA #8
BEDE B0 24 1077 BCS HNDERR
BEE0 A9 30 1078 LDA #S30
BEE2 8D 78 05 1079 STA RETRYCNT
BEE5 1080 ;
BEE5 1081 ; Verify the track just formatted.
BEE5 1082 ;
BEE5 38 1083 NOGOOD SEC
BEE6 CE 78 05 1084 DEC RETRYCNT
BEE9 F0 19 1085 BEQ HNDERR
BEEB 1086 ;
BEEB 1087 ; Read the address field.
BEEB 1088 ;
BEEB 20 44 B9 1089 JSR RDADR16
BEEE B0 F5 1090 BCS NOGOOD ; something wrong?
BEFO 1091 ;
BEFO A5 2D 1092 LDA SECFND
BEF2 D0 F1 1093 BNE NOGOOD
BEF4 1094 ;
BEF4 1095 ; Read the data and see if it's
BEF4 1096 ; ok.
BEF4 1097 ;
BEF4 20 DC B8 1098 JSR READ16
BEF7 B0 EC 1099 BCS NOGOOD
BEF9 1100 ;
BEF9 1101 ; A-OK, move on to the next track.
BEF9 1102 ;
BEF9 E6 44 1103 INC TRACK
BEFB 1104 ;
BEFB 1105 ; At the last track yet?
BEFB 1106 ;
BEFB A5 44 1107 LDA TRACK

LISA 2.5

DOS 3.3 RWTS

```

BEFD C9 23    1108      CMP #$23
BEFF 90 D3    1109      BCC NXTRK
BF01          1110      ;
BF01          1111      ; Force error flag (carry) off.
BF01          1112      ;
BF01 18       1113      CLC
BF02 90 05    1114      BCC DONEDSK
BF04          1115      ;
BF04          1116      ;
BF04          1117      ; Store error code in I0B.
BF04          1118      ;
BF04 A0 OD    1119      HNDERR LDY #$D
BF06 91 48    1120      STA (I0BPL), Y
BF08 38       1121      SEC
BF09          1122      ;
BF09          1123      ; Turn the disk off.
BF09          1124      ;
BF09 BD 88 CO 1125      DONEDSK LDA LC088, X
BF0C 60       1126      RTS
BF0D          1127      ;
BF0D          1128      ; Format the current track.
BF0D          1129      ;
BF0D A9 00    1130      DISKF2 LDA #0           ; init sector #
BF0F 85 3F    1131      STA BUFADR+1
BF11 A0 80    1132      LDY #$S80        ; begin track with 128 sync bytes
BF13          1133      ;
BF13 D0 02    1134      BNE >3
BF15          1135      ;
BF15          1136      ;
BF15 A4 45    1137      TRKFRM LDY SYNCNT
BF17 20 56 BC 1138      ^3      JSR WRADR16      ; write an address field
BF1A B0 6B    1139      BCS DELAY12
BF1C          1140      ;
BF1C          1141      ; Write a data field.
BF1C          1142      ;
BF1C 20 2A B8 1143      JSR WRITE16
BF1F B0 66    1144      BCS DELAY12
BF21          1145      ;
BF21          1146      ; Increment the sector number
BF21          1147      ; and see if it is 16 yet.
BF21          1148      ;
BF21 E6 3F    1149      INC BUFADR+1
BF23 A5 3F    1150      LDA BUFADR+1
BF25 C9 10    1151      CMP #$S10
BF27 90 EC    1152      BCC TRKFRM
BF29          1153      ;
BF29          1154      ; Reset sector # to 15
BF29          1155      ;
BF29 A0 0F    1156      LDY #$SF
BF2B 84 3F    1157      STY BUFADR+1
BF2D          1158      ;
BF2D          1159      ; mark the current track as
BF2D          1160      ; formatted.
BF2D          1161      ;
BF2D A9 30    1162      LDA #$S30
BF2F 8D 78 05 1163      STA RETRYCNT
BF32 99 A8 BF 1164      INITMAP STA SECMAP, Y
BF35 88       1165      DEY

```

LISA 2.5

DOS 3.3 RWTS

BF36 10 FA	1166	BPL I NI TMAP
BF38 A4 45	1167	LDY SYNCNT
BF3A 20 87 BF	1168	DELAY JSR DELAY12
BF3D 20 87 BF	1169	JSR DELAY12
BF40 20 87 BF	1170	JSR DELAY12
BF43 48	1171	PHA
BF44 68	1172	PLA
BF45 EA	1173	NOP
BF46 88	1174	DEY
BF47 D0 F1	1175	BNE DELAY
BF49 20 44 B9	1176	JSR RDADR16
BF4C B0 23	1177	BCS DOAGAIN
BF4E A5 2D	1178	LDA SECFND
BF50 F0 15	1179	BEQ ITSGOOD
BF52 A9 10	1180	LDA #\$10
BF54 C5 45	1181	CMP SYNCNT
BF56 A5 45	1182	LDA SYNCNT
BF58 E9 01	1183	SBC #1
BF5A 85 45	1184	STA SYNCNT
BF5C C9 05	1185	CMP #5
BF5E B0 11	1186	BCS DOAGAIN
BF60 38	1187	SEC
BF61 60	1188	RTS
BF62 20 44 B9	1189	NXTSEC JSR RDADR16
BF65 B0 05	1190	BCS NXTTRY
BF67 20 DC B8	1191	ITSGOOD JSR READ16
BF6A 90 1C	1192	BCC MARKMAP
BF6C CE 78 05	1193	NXTTRY DEC RETRYCNT
BF6F D0 F1	1194	BNE NXTSEC
BF71 20 44 B9	1195	DOAGAIN JSR RDADR16
BF74 B0 OB	1196	BCS ERRCNT
BF76 A5 2D	1197	LDA SECFND
BF78 C9 OF	1198	CMP #\$F
BF7A D0 05	1199	BNE ERRCNT
BF7C 20 DC B8	1200	JSR READ16
BF7F 90 8C	1201	BCC DISKF2
BF81 CE 78 05	1202	ERRCNT DEC RETRYCNT
BF84 D0 EB	1203	BNE DOAGAIN
BF86 38	1204	SEC
BF87 60	1205	DELAY12 RTS
BF88 A4 2D	1206	MARKMAP LDY SECFND
BF8A B9 A8 BF	1207	LDA SECMAP, Y
BF8D 30 DD	1208	BMI NXTTRY
BF8F A9 FF	1209	LDA #\$FF
BF91 99 A8 BF	1210	STA SECMAP, Y
BF94 C6 3F	1211	DEC BUFADR+1
BF96 10 CA	1212	BPL NXTSEC
BF98 A5 44	1213	LDA TRACK
BF9A D0 OA	1214	BNE TRKDONE
BF9C A5 45	1215	LDA SYNCNT
BF9E C9 10	1216	CMP #\$10
BFA0 90 E5	1217	BCC DELAY12
BFA2 C6 45	1218	DEC SYNCNT
BFA4 C6 45	1219	DEC SYNCNT
BFA6 18	1220	TRKDONE CLC
BFA7 60	1221	RTS
BFA8	1222	;
BFA8	1223	;

LISA 2.5

DOS 3.3 RWTS

BFA8 1224 ; SECMAP- used to mark initialized
BFA8 1225 ; sectors.
BFA8 1226 ;
BFA8 FF FF FF 1227 SECMAP HEX FFFFFFFFFFFFFFFFFFFFFFFFF
BFAB FF FF FF
BFAE FF FF FF
BFB1 FF FF FF
BFB4 FF FF FF
BFB7 FF
BFB8 1228 ;
BFB8 1229 ;
BFB8 1230 ; Interleave remapping table
BFB8 1231 ;
BFB8 00 0D 0B 1232 INTRLEAV HEX 000DOB09070503010E0COA080604020F
BFBB 09 07 05
BFBE 03 01 0E
BFC1 0C 0A 08
BFC4 06 04 02
BFC7 0F
BFC8 1233 ;
BFC8 1234 ;
BFC8 1235 ;
BFC8 1236 ; DOS 3.3 patches. WARNING! These
BFC8 1237 ; are addressed by object code
BFC8 1238 ; and should not be moved without
BFC8 1239 ; careful thought.
BFC8 1240 ;
BFC8 20 93 FE 1241 JSR SETVID
BFCB AD 81 CO 1242 LDA ROM
BFCE AD 81 CO 1243 LDA ROM
BFD1 A9 00 1244 LDA #0
BFD3 8D 00 E0 1245 STA BASIC
BFD6 4C 44 B7 1246 JMP RESTART
BFD9 1247 ;
BFD9 1248 ; UNUSED
BFD9 1249 ;
BFD9 1250 DFS 3
BFDC 1251 ;
BFDC 1252 ;
BFDC 8D 63 AA 1253 STA TEMP1
BFDF 8D 70 AA 1254 STA BYTVAL
BFE2 8D 71 AA 1255 STA BYTVAL+1
BFE5 60 1256 RTS
BFE6 20 5B A7 1257 JSR RSETO
BFE9 8C B7 AA 1258 STY RUNINTRC
BFEC 60 1259 RTS
BFED 20 7E AE 1260 JSR SAVFMW
BFF0 AE 9B B3 1261 LDX STKSAVE
BFF3 9A 1262 TXS
BFF4 20 16 A3 1263 JSR CLOSEALL
BFF7 BA 1264 TSX
BFF8 8E 9B B3 1265 STX STKSAVE
BFFB A9 09 1266 LDA #9
BFFD 4C 85 B3 1267 JMP SETERROR
C000 1268 END

***** END OF ASSEMBLY

!