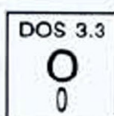


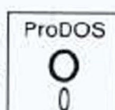


JONATHAN
GRAVES

NIBBLE MAESTRO



Turn your keyboard into an organ keyboard with this Applesoft program. Notes are drawn on a Hi-Res score as the music plays.



by David L. Smith, M.D., 1007 Gloucester Ave., Middlesboro, KY 40965

If you think an error bell represents the full extent of the Apple's audio capabilities, it's time to limber up your fingers and tune your tin ear. The error bell just scratches the surface of the Apple's musical abilities. Nibble Maestro transforms the Apple keyboard into an organ with a four-octave capacity. As you touch the keys, the corresponding notes sound and simultaneously appear in the staff on the screen.

Songs may be stored onto and replayed from disk. If you don't complete a song in one session, you can save it on disk and finish later. Or you can save a song and later edit it. Nibble Maestro lets you compose tunes using 49 notes (however, it does not play chords).

BECOME A NIBBLE MAESTRO

When you run NIBBLE.MAESTRO, a menu of nine options is presented, as shown in **Figure 1**. The first option, to listen to a computer-created tune, provides a convenient way of demonstrating the abilities of the program. When you press 1, the program randomly generates a sequence of notes and proceeds to display (**Figure 2**) and play them.

Option 2 lets you create your own tune. After selecting this option, you will be presented with a blank staff, and the program will await your input. The keyboard diagrams shown in **Figures 3 and 4** show the note which corresponds with each key for both the //e (or //c) keyboard and the II Plus keyboard. The duration of each note is controlled by pressing the digits 1-9 as shown in **Figure 5**. The zero key is used for triplets; it will shorten by a third the duration of the next three notes played.

As you press each note key, the appropriate symbol will be drawn on the staff, and the note will be played. When you reach the end of the screen, a blank staff will be displayed and you may continue your composition.

A number of keys have special meanings:

1. <ESC> stops play and displays the menu.
2. The left arrow key (used to backspace) erases the previously played note. You can continue to backspace all the way to the beginning of a screen filled with notes, but you cannot backspace to previous screens.
3. The space bar is used for rests. (It actually produces a note too high for the Apple speaker to play audibly.)
4. Any character key not included in the keyboard diagrams produces either the lowest possible note or no note at all.

The Apple tone generating routine produces high notes up to the limit of audibility, but the lowest note available is approximately B below middle C. In order to take advantage of the available four octaves of keys and notes, the notes drawn on the Hi-Res screen are transposed two octaves down from the note actually played by the Apple. For best results, tunes should be played using the lowest octaves. Thus, a tune that has a two-octave span should be played on the lower two or three octaves and will be drawn on the screen using mostly the bass clef.

Since options 1 and 2 will destroy any composition that may be in memory, both request confirmation before proceeding. Also, note that because of the time it takes to play and draw each note, notes can be lost if the keys are pressed too quickly.

DISK STORAGE

After a song is composed by the program or entered from the keyboard, it can be saved on disk with option 3. To maintain ProDOS compatibility, the program will ask you to specify a file name less than 16 characters long, which includes only letters, numbers and periods.

REPLAYING SONGS

To listen to the song in memory, choose option 4. You will first be asked if you want a printout of the song as it is played. This prompt is supplied for those who have a printer interface (such as the Grappler or Microtek Dumping) that allows direct graphics dumps of the Hi-Res screen.

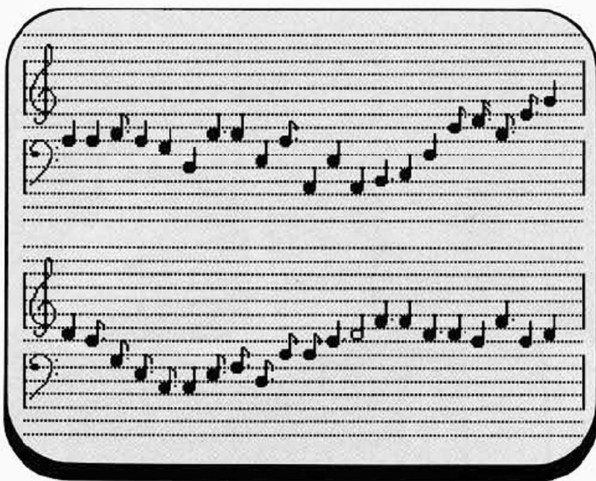
You will also be prompted to specify a delay factor. This factor determines the speed at which a song is replayed. A delay factor of zero gives maximum speed. The program will then play your song, displaying each note as it is played.

If you want to listen to a song that you previously saved on disk, use option 5 to load it into memory before using option 4. Since this selection will overwrite any song that may be in memory, confirmation will be requested before the song is loaded from disk.

EDITING SONGS

If you want to change a song, first make sure that it is in memory by either composing it or loading it from disk. When you select option 6, your song will be played until you press a key. At that point, you may re-enter the remainder of the song. If you simply want to add on to a previously created song, select option 6 and

FIGURE 2: Maestro Display



allow the song to be played through to the end. You may then enter additional notes as usual.

OTHER MENU OPTIONS

Three other options are provided to help you use NIBBLE .MAESTRO: Instructions, Catalog, and Quit. The first of these, option 7, presents several screens of information describing the operation of the program, including the keyboard layout. Option 8, the Catalog option, allows you to see a catalog of your disk. Option 9 allows you to quit, but requests confirmation before doing so.

ENTERING THE PROGRAM

To key in the Nibble Maestro, first enter the Applesoft program shown in Listing 1, and save it on disk with the command:

SAVE NIBBLE.MAESTRO

This program BLOADs two additional files when it is run: STAFF (a Hi-Res screen of a blank staff) and NOTES (the shape table for the notes). To create the first of these, enter the Applesoft program shown in Listing 2 and save it on disk with the command:

SAVE STAFF.MAKER

Run it to create the blank staff on the Hi-Res screen. After inspecting the staff for a few moments, you will be asked to confirm that the screen is correct before it will automatically be saved for you as STAFF.

Use the Monitor to enter the hex code shown in Listing 3 and save it on disk with the command:

BSAVE NOTES,A\$6000,L\$1E3

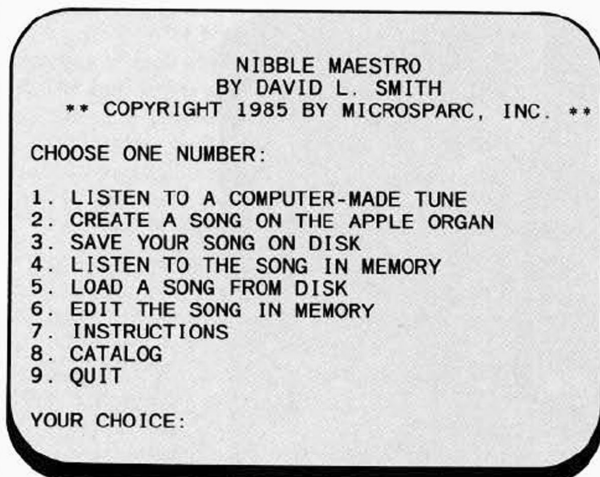
Make certain that all three files, NIBBLE.MAESTRO, STAFF, and NOTES are on the same disk when you RUN NIBBLE .MAESTRO.

The version of NIBBLE.MAESTRO shown in Listing 1 is designed for the //e or //c keyboard. To modify the program to run on the II Plus keyboard, use the lines shown in Listing 4 in place of lines 680, 690, and 810.

HOW NIBBLE MAESTRO WORKS

Nibble Maestro is a good example of the Fifth Law of Computers: Any given program will expand to fill all available memory. The core of the program is a 23-byte tone generating routine that appears

FIGURE 1: Nibble Maestro Menu



in the *Apple Red Book* (one of Apple's earliest reference manuals). The program occupies the 6K of memory below the Hi-Res screens. It gets keyboard input, draws notes on Hi-Res page one using the 483-byte shape table located just above Hi-Res page two and POKES appropriate values into the tone generating routine.

When the lower Hi-Res screen is full of notes, a memory move routine is called to replace it with the upper one, which contains a blank musical staff. The remaining 15K above the shape table holds the numerous arrays that are used as look-up tables to speed program execution, as well as the arrays that store the pitch and length of the notes. There is enough memory to hold a 1,400-note composition in RAM. Compositions can be saved to disk as standard text files.

The heart of the program is found in the menu (lines 470-530). The various subroutines are accessed through the menu options. If no variables have been initialized, the program first branches to the initialization routine in lines 750-840.

Initialization

The initialization routine first loads the staff and shape tables into memory. To speed program execution, all program constants are given variable names. Due to the slowness of Applesoft, it was necessary to initialize several arrays. They tell the program what note to play when a given key is pressed and which shape to plot at what spot on the screen. The arrays P% and L% contain the pitch and length of the notes, respectively.

The DATA statements in line 810 contain the ASCII codes for the keys that produce the 48 available notes, from lowest to highest.

Lines 820-840 contain the decimal values for the machine language routines that are POKEd into the page 3 space (starting

FIGURE 3: Keyboard Layout for //e and //c

	C	#	D	E	-	E	F	F#	G	A	-	A	B	-	B
SOPRANO:	!	@	#	\$	%	^	&	*	()	-	=			
ALTO:	Q	W	E	R	T	Y	U	I	O	P	[]			
TENOR:	A	S	D	F	G	H	J	K	L	;	:	>	RET		
BASS:	Z	X	C	V	B	N	M	,	.	>	/	?			
LOW B--:	REST: < SPACE BAR >														

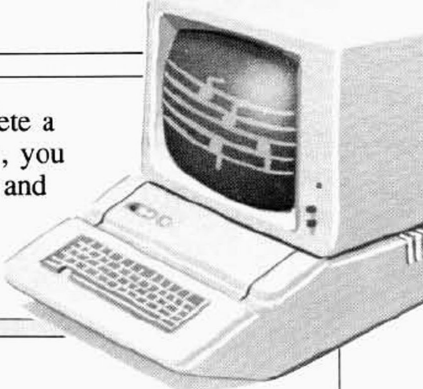
FIGURE 4: Keyboard Layout for // Plus

	C	#	D	E	-	E	F	F#	G	A	-	A	B	-	B
SOPRANO:	!	"	#	\$	%	&	'	()	:	*	-			
ALTO:	Q	W	E	R	T	Y	U	I	O	P	@	RET			
TENOR:	A	S	D	F	G	H	J	K	L	;	+	->			
BASS:	Z	X	C	V	B	N	M	,	<	.	>	/			
LOW B--:	? REST: < SPACE BAR >														

FIGURE 5: Digits and Note Duration

1-16TH	2-DOTTED 16TH
3-EIGHTH	4-DOTTED EIGHTH
5-QUARTER	6-DOTTED QUARTER
7-HALF	8-DOTTED HALF
9-WHOLE	

If you don't complete a song in one session, you can save it on disk and finish it later.



at location 770). Line 820 is the tone-generating subroutine, line 830 codes for the memory move routine from Hi-Res page 2 to page 1, and line 840 is an error handling routine from the *Applesoft II BASIC Programming Reference Manual*.

The last three lines of the DATA statements initialize the arrays that tell the program which notes to use for composing, the vertical position of notes, and whether or not to place a sharp or flat sign in front of the note.

As indicated by the ON-GOSUB statement in line 530, the major routines of the program are as follows:

Lines 710-720 create the computer generated sound.

Lines 190-270 allow you to input notes.

Lines 540-560 save the data on disk.

Lines 280-340 play back your tune and allow you to edit it if the edit flag (FL) is set.

Lines 570-610 load data from disk.

Lines 620-700 present instructions.

Line 730 displays a catalog of the disk.

MODIFICATION

If you are tempted to modify NIBBLE.MAESTRO note that, as published, it almost fills memory below Hi-Res page 1. Any modifications that you make should take this length into account.

If you do not have a printer interface that support graphics dumps, you may want to modify the prompt for obtaining a printout. One possibility would be to use the printing routine (line 150) to BSAVE a series of picture files that could be printed by a separate graphics utility.

Navigator, Nibble Maestro, and Lightning Copy are available on diskette for an introductory price of \$19.95 plus \$1.50 shipping/handling (\$2.50 outside the U.S.) from Nibble, 45 Winthrop St., Concord, MA 01742. Introductory price expires 9/31/85.

LISTING 1: NIBBLE.MAESTRO

```

10 REM *****
20 REM * NIBBLE.MAESTRO *
30 REM * BY DAVID L. SMITH *
40 REM * COPYRIGHT (C) 1985 *
50 REM * BY MICROSPARC, INC *
60 REM * CONCORD, MA. 01742 *
70 REM *****
80 LOMEM: 25088:CS = 49168:GOTO 470
90 C = C + T * (C = J AND I%(0) AND I%(0) < T
   ): IF Z + E * (C > W) > D THEN Z = S:B =
   U * (NOT B): IF NOT B THEN ON PF GOSUB
   150: CALL H:I = W
100 Y = O * G + TW + B: IF C THEN DRAW C AT
   Z,Y:Z = Z + E:I%(0) = C
110 DRAW V + J AT Z,Y:Z = Z + W2: RETURN

```

LISTING 1: NIBBLE.MAESTRO (continued)

```

120 FOR Q = TW TO A * (L%(X - R) + F * (NOT
L%(X - R))) / E: NEXT : RETURN
130 H%(I) = C + HF * (Z - T9):K%(I) = Y: RETURN

140 NN = X:X = N: POKE CS,W: RETURN
150 PRINT : PRINT D$"PR#1": PRINT CHR$(9)"
G": PRINT D$"PR#0": RETURN
160 PRINT "(Y/N)?" : POKE CS,W
170 Q = FRE (W): GET G$: PRINT G$: RETURN
180 N = NN + R: HOME : VTAB 23: POKE MS,W: PRINT
"READY TO EDIT": GOSUB 740: POKE FS,W: GOTO
200
190 HOME : VTAB 22: PRINT "PRESS <ESC> TO EN
D SONG": HGR : CALL H:N = R:A$ = "":V =
FI: GOSUB 380
200 FOR ZY = W TO R STEP W: GET G$:Q = VAL
(G$): IF Q THEN V = Q: NEXT
210 IF G$ = "0" THEN TF = G: NEXT
220 IF G$ = M$ THEN GOSUB 390: NEXT
230 IF G$ < CHR$(TT) OR G$ > CHR$(96) THEN
Q = FRE (W): NEXT
240 IF G$ = E$ OR N > 1395 THEN N = N - R:ZY
= R: NEXT : RETURN
250 L%(N) = B%(V - R * (TF > W)):TF = TF - R *
(TF > W):J = W: IF G$ = S$ THEN J = NI:P
%(N) = R:C = W
260 IF NOT J THEN X = A%(ASC (G$) - TT):P%(
N) = C%(X):O = Y%(X):C = J%(X)
270 GOSUB 90: POKE P,P%(N): POKE L,L%(N): CALL
M: GOSUB 130:N = N + R:I = I + R: NEXT
280 GOSUB 350
290 HOME : VTAB 21: PRINT A$: PRINT : PRINT
"PRESS ANY KEY TO STOP THE MUSIC": HGR :
CALL H: GOSUB 380: FOR X = R TO N:V = G
%(L%(X)):J = NI * (P%(X) = R):C = W: IF
NOT J THEN Q = F%(P%(X)):O = Y%(Q):C =
J%(Q)
300 GOSUB 90: ON A AND I GOSUB 120: POKE P,P
%(X): POKE L,L%(X): CALL M: ON FL GOSUB
130:I = I + R: IF PEEK (K) > HF THEN GOSUB
140: IF FL THEN NEXT : GOTO 180
310 NEXT :NN = N: ON FL GOTO 180: IF PF THEN
GOSUB 150: PRINT : PRINT D$"PR#1": POKE
36,40 - LEN (A$) / 2: PRINT A$: PRINT D
$"PR#0":PF = 0: RETURN
320 HOME : VTAB 21: POKE MS,W
330 PRINT : PRINT "PRESS ANY KEY TO CONTINUE
": POKE CS,W:Q = FRE (W): FOR Q = R TO
P: IF PEEK (K) > HF THEN Q = P
340 NEXT : POKE FS,W: GET G$: PRINT : HOME :
RETURN
350 PF = 0: IF NOT FL THEN PRINT "DO YOU WA
NT A PRINTOUT": GOSUB 160: IF G$ = "Y" THEN
PF = 1: GOSUB 430
360 PRINT "DELAY FACTOR = "A: POKE CS,W: INPUT
"TYPE NEW DELAY FACTOR, THEN <RETURN>," 0
R<RETURN> ALONE FOR SAME DELAY.":G$: IF
(VAL (G$) > W OR G$ = "0") THEN A = VAL
(G$)
370 RETURN
380 TF = W:Z = S:B = W:C = W: FOR X = W TO 30
:I%(X) = W: NEXT :O = 16: GOSUB 740:I =
W: POKE FS,W: RETURN
390 IF NOT I THEN PRINT B$: RETURN
400 I = I - R:N = N - R:Z = INT (H%(I) / HF)
:V = G%(L%(N)):J = NI * (P%(N) = R):C =
H%(I) - HF * Z:Y = K%(I):B = U * (Y > 92
):O = (Y - TW - B) / G:Z = Z + S: XDRAW
V + J AT Z,Y: IF C THEN Z = Z - E: XDRAW
C AT Z,Y: IF C = T THEN I%(O) = C
410 IF NOT I THEN CALL H
420 Q = FRE (W): RETURN
430 IF A$ > "" THEN PRINT "USE "A$" AS NA
ME": GOSUB 160: IF G$ = "Y" THEN RETURN
440 VTAB 18: HTAB 1: CALL - 958: PRINT "TYP
E SONG NAME (<16 LETTERS)": INPUT "":A$

```

```

450 NF = 1: IF LEN (A$) > 0 THEN FOR NC = 1
TO LEN (A$):NA = ASC (MID$(A$,NC,1)
):NF = ((NA > 64 AND NA < 91) OR (NA = 4
6) OR (NA > 47 AND NA < 58 AND NC > 1)) AND
NF = 1: NEXT : IF LEN (A$) > 15 OR NF =
0 THEN PRINT B$"INVALID NAME!": FOR NC =
1 TO 1000: NEXT : GOTO 440
460 RETURN
470 TEXT : HOME : HTAB 13: PRINT "NIBBLE MAE
STRO": HTAB 11: PRINT "BY DAVID L. SMITH
": PRINT "** COPYRIGHT 1985 BY MICROSPAR
C, INC. *": PRINT : PRINT "CHOOSE ONE
NUMBER": PRINT : PRINT "1. LISTEN TO A
COMPUTER-MADE TUNE"
480 PRINT "2. CREATE A SONG ON THE APPLE ORG
AN": PRINT "3. SAVE YOUR SONG ON DISK": PRINT
"4. LISTEN TO THE SONG IN MEMORY": PRINT
"5. LOAD A SONG FROM DISK": PRINT "6. ED
IT THE SONG IN MEMORY"
490 PRINT "7. INSTRUCTIONS": PRINT "8. CATAL
OG": PRINT "9. QUIT": POKE CS,W: PRINT :
IF NOT F THEN HTAB 5: INVERSE : PRINT
"MAKE SURE CAPS LOCK IS DOWN": NORMAL
: PRINT : HTAB 13: PRINT "PLEASE WAIT...
": GOSUB 750: VTAB PEEK (37): CALL - 8
68
500 HTAB 1: VTAB 17: CALL - 958: PRINT "YOU
R CHOICE.": GOSUB 170: IF G$ = "9" THEN
PRINT "QUIT? (Y/N)": GET G$: PRINT : IF
G$ = "Y" THEN TEXT : HOME : END
510 Q = VAL (G$): IF Q < 1 OR Q > 8 THEN 500
520 IF Q = 1 OR Q = 2 OR Q = 5 THEN VTAB 17
: CALL - 958: PRINT : PRINT "ANY SONG I
N MEMORY WILL BE LOST.": PRINT "OKAY? (Y
/N)": GET G$: PRINT : IF G$ < > "Y" THEN
500
530 FL = (Q = 6): ON Q GOSUB 710,190,540,280,
570,280,620,730: GOTO 470
540 GOSUB 430: IF A$ = "" THEN RETURN
550 ONERR GOTO 600
560 PRINT D$"OPEN"A$: PRINT D$"CLOSE": PRINT
D$"DELETE"A$: PRINT D$"OPEN"A$: PRINT D$
"WRITE"A$: PRINT N: FOR X = R TO N: PRINT
P%(X): PRINT L%(X): NEXT : PRINT D$"CLOS
E": RETURN
570 A$ = "": GOSUB 430: ONERR GOTO 600
580 IF A$ = "" THEN POKE EF,W: RETURN
590 PRINT D$"VERIFY"A$: PRINT D$"OPEN"A$: PRINT
D$"READ"A$: INPUT N: FOR X = R TO N: INPUT
P%(X): INPUT L%(X): NEXT : PRINT D$"CLOS
E": POKE EF,W: RETURN
600 CALL 818: PRINT D$"CLOSE":Q = PEEK (222
): IF Q = 6 OR Q = TT THEN PRINT B$"SOR
RY, I DON'T KNOW THAT ONE.": FOR NC = 1 TO
1000: NEXT : GOSUB 440: GOTO 580
610 PRINT B$"ERROR #"Q", LINE " PEEK (218) +
F * PEEK (219): POKE EF,W: PRINT "PRESS
ANY KEY FOR MENU": WAIT - 16384,128: POKE
- 16368,0: GOTO 470
620 TEXT : HOME : HTAB 15: PRINT "INSTRUCTIO
NS": POKE 34,2: HOME
630 PRINT "THIS PROGRAM ALLOWS YOU TO PLAY T
HE KEY-BOARD LIKE A 4 OCTAVE ORGAN (1 NO
TE AT A TIME)": PRINT : PRINT "BEFORE PL
AYING ANY NOTES, SELECT A NOTE LENGTH BY
PRESSING ANY NUMBER KEY 1-9."
640 PRINT "IF A TRIPLET IS TO BE PLAYED, ALS
O PRESS<0>. THIS SHORTENS THE LENGTH OF
THE NEXT 3 NOTES, EXCEPT FOR TRIPLETS
, A NOTE DURATION REMAINS IN EFFECT U
NTIL ANOTHER NUMBER KEY IS PRESSED."
650 PRINT : PRINT "THE DURATIONS AVAILABLE A
RE AS FOLLOWS.": PRINT " 1-16TH","2-DOTT
ED 16TH": PRINT " 3-EIGHTH","4-DOTTED EI
GHTH": PRINT " 5-QUARTER","6-DOTTED QUAR
TER": PRINT " 7-HALF","8-DOTTED HALF": PRINT
" 9-WHOLE": GOSUB 330

```

LISTING 1: NIBBLE.MAESTRO (continued)

```
660 PRINT : PRINT "IF YOU TYPE FASTER THAN T
HE ORGAN CAN PLAY, NOTES WILL BE LOST.
PLEASE WAIT FOR THE NOTE TO SOUND BEF
ORE TYPING IN ANOTHER NOTE.": PRINT
670 PRINT "IF YOU MAKE A MISTAKE, YOU CAN ER
ASE BACK TO THE TOP OF THE DISPLAYED
SCORE WITH THE (<-- ) LEFT-ARROW KEY.": GOSUB
330: PRINT "THE 4 OCTAVES ARE ARRANGED F
ROM TOP TO BOTTOM OF KEYBOARD.": PRINT
```

```
680 PRINT " C C# D E- E F# G A- A
B- B": PRINT "
-----
% ^ & * ( ) - =": PRINT " ALTO:
Q W E R T Y U I O P ": CHR$(91):"
]": REM 29 HYPHENS
690 PRINT " TENOR: A S D F G H J K L ;
' <RET>": PRINT " BASS: Z X C V B N
M . : > / ?": PRINT "LOW B- ": CHR$(
96):" REST: <SPACE BAR>": PRINT
700 PRINT "PRESSING OTHER KEYS WILL PRODUCE
LOW B FLAT OR NO NOTE AT ALL.": PRINT
: PRINT "WHEN YOUR SONG IS ENDED, PRESS
<ESC> TO RETURN TO MENU.": GOTO 330
710 X = RND ( - PEEK (78) - F * PEEK (79) )
: PRINT : FLASH : PRINT " COMPOSING ": NORMAL
: FOR N = W TO 41: Q = W:A = TW: FOR X =
W TO 6: IF N / TW ^ X = INT (N / TW ^ X
) THEN D%(X) = FI * RND (R):E%(X) = TW *
RND (R)
720 Q = Q + D%(X):A = A + E%(X): NEXT :P%(N +
R) = C%(N%(Q)):L%(N + R) = B%(A): NEXT :
A = W:A$ = "": GOTO 290
730 HOME : PRINT D$"CATALOG": GOTO 330
740 FOR Q = 0 TO 999: NEXT :Q = FRE (0): RETURN
```

```
750 D$ = CHR$(4): PRINT D$"BLOOD NOTES": POKE
232,0: POKE 233,96: PRINT D$"BLOOD STAFF
": VTAR 10: HTAR 26
760 R = 1:FJ = 4:E = 8:NI = 9:F8 = 48:F = 256
:T2 = 32:P = 768:L = 769:M = 770:K = 491
:52:HF = 127:D = 264:S = 17:T = 21:W2 = 1
:2:TW = 2:G = 3:TT = 13:U = 96:H = 791:HA
= .5:T9 = 29:SE = 7:FI = 5:FS = 49234:M
S = 49235:EF = 216
770 DIM N%(28), I%(30), C%(F8), Y%(F8), J%(F8), A
%(83), G%(192), F%(242), P%(1400), L%(1400),
H%(41), K%(41): FOR X = R TO F8:C%(X) = F
/ TW ^ (X / W2) + HA:F%(F / TW ^ (X / W
2) + HA) = X: READ Q:A%(Q - TT) = X: PRINT
": NEXT
780 G%(W) = NI:G%(W2) = R:B%(W) = W2:Q = FJ: FOR
X = R TO SE STEP TW:B%(X) = TW ^ Q:G%(TW
^ Q) = X:G%((TW ^ Q) * G / TW) = X + R:
B%(X + R) = G * B%(X) / TW:Q = Q + R: NEXT
```

```
790 B$ = CHR$(SE):E$ = CHR$(27):M$ = CHR$(
E):S$ = " ": HCOLOR= G: SCALE= R: ROT=
W: FOR X = M TO 827: READ Z: POKE X,Z: NEXT
: FOR X = R TO SE: READ N%(X): FOR N = R
TO G:N%(X + SE * N) = N%(X) + W2 * N: NEXT
: PRINT " ": NEXT
800 FOR X = W TO W2: READ Y%(X): NEXT : FOR
X = TW TO 11: READ J%(X): NEXT : FOR N =
R TO G: FOR X = R TO W2:Y%(X + W2 * N) =
Y%(X) - SE * N:J%(X + W2 * N) = J%(X): NEXT
: PRINT " ": NEXT : RETURN
```

```
810 DATA 96,90,88,67,86,66,78,77,44,46,47,6
3,65,83,68,70,71,72,74,75,76,59,39,13,81
,87,69,82,84,89,85,73,79,80,91,93,33,64,
35,36,37,94,38,42,40,41,45,61
820 DATA 173,48,192,136,208,5,206,1,3,240,9,
202,208,245,174,0,3,76,2,3,96
830 DATA 169,64,133,61,169,247,133,62,169,95
,133,63,169,32,133,67,169,0,133,60,133,6
6,160,0,76,44,254: DATA 104,168,104,166,
223,154,72,152,72,96: DATA 1,3,5,6,8,10,
12: DATA 30,29,29,28,27,26,26,25,24,2
4,23,23
840 DATA 19,0,20,0,19,0,20,0,20
```

END OF LISTING 1

KEY PERFECT 4.0		E0DA	
RUN ON		010D43	
NIBBLE.MAESTRO		0172D9	
=====	=====	=====	=====
CODE	LINE# - LINE#		
-----	-----	-----	-----
8C07	10 - 100	FA20	510 - 600
9FA6	110 - 200	02337A	610 - 700
		01DB00	710 - 800
		D8B5	810 - 840
		PROGRAM CHECK IS : 15B6	

LISTING 2: STAFF.MAKER

```
10 REM *****
20 REM * STAFF.MAKER *
30 REM * BY DAVID L. SMITH *
40 REM * COPYRIGHT (C) 1985 *
50 REM * BY MICROSPARC, INC *
60 REM * CONCORD, MA. 01742 *
70 REM *****
80 ONERR GOTO 220
90 D$ = CHR$(4): HGR2 :BO = 96: HCOLOR= 3: FOR
Y = 0 TO BO STEP BO
100 HCOLOR=1: FOR Z = Y + 18 TO Y + 42 STEP
6: HPLLOT 1,Z TO 278,Z: NEXT : FOR Z = Y +
54 TO Y + 78 STEP 6: HPLLOT 1,Z TO 278,Z:
NEXT
110 HCOLOR= 2: HPLLOT 0,Y + 6 TO 279,Y + 6: HPLLOT
0,Y + 48 TO 279,Y + 48: HPLLOT 0,Y + 84 TO
279,Y + 84: HPLLOT 0,Y + 90 TO 279,Y + 90
: HPLLOT 0,Y + 12 TO 279,Y + 12
120 HCOLOR=3: HPLLOT 0,Y + 18 TO 0,Y + 42: HPLLOT
0,Y + 54 TO 0,Y + 78: HPLLOT 279,Y + 18 TO
279,Y + 42: HPLLOT 279,Y + 54 TO 279,Y +
78
130 HPLLOT 5,Y + 44 TO 5,Y + 45: HPLLOT 6,Y +
46 TO 8,Y + 46: HPLLOT 9,Y + 45: HPLLOT 10
,Y + 44 TO 10,Y + 12: HPLLOT 11,Y + 11 TO
12,Y + 11: HPLLOT 13,Y + 12 TO 13,Y + 18:
Z = Y + 19: FOR X = 12 TO 6 STEP - 1: HPLLOT
X,Z:Z = Z + 1: NEXT
140 HPLLOT 6,Y + 26: HPLLOT 5,Y + 27 TO 5,Y +
28: HPLLOT 4,Y + 29 TO 4,Y + 30: HPLLOT 3,
Y + 31 TO 3,Y + 36: HPLLOT 4,Y + 37 TO 4,
Y + 38: HPLLOT 5,Y + 39: HPLLOT 6,Y + 40 TO
7,Y + 40: HPLLOT 8,Y + 41 TO 12,Y + 41: HPLLOT
13,Y + 40: HPLLOT 14,Y + 39
150 HPLLOT 15,Y + 38 TO 15,Y + 34: HPLLOT 14,Y
+ 33: HPLLOT 13,Y + 32: HPLLOT 12,Y + 31 TO
9,Y + 31: HPLLOT 8,Y + 32: HPLLOT 7,Y + 33
TO 7,Y + 34
160 HPLLOT 6,Y + 61 TO 4,Y + 61 TO 4,Y + 60 TO
6,Y + 60 TO 6,Y + 59 TO 4,Y + 59: HPLLOT
3,Y + 58 TO 3,Y + 57: HPLLOT 4,Y + 56: HPLLOT
5,Y + 55: HPLLOT 6,Y + 54 TO 9,Y + 54:Z =
Y + 55: FOR X = 10 TO 13: HPLLOT X,Z:Z =
Z + 1: NEXT
170 HPLLOT TO 13,Y + 62: HPLLOT 12,Y + 63 TO
12,Y + 64: HPLLOT 11,Y + 65:Z = Y + 66: FOR
X = 11 TO 3 STEP - 1: HPLLOT X,Z:Z = Z +
1: NEXT : HPLLOT 16,Y + 57: HPLLOT 16,Y +
63
180 NEXT Y
190 FOR K = 1 TO 2000: NEXT : PRINT CHR$( 7
)
200 TEXT : HOME : VTAB 12: PRINT "OKAY TO SA
VE? (Y/N)": GET G$: PRINT G$: IF G$ = "
Y" THEN PRINT D$"SAVE STAFF,A$4000,L$1
FFF"
210 TEXT : END
220 TEXT : PRINT "ERROR #" PEEK (222)" IN L
INE #" PEEK (218) + 256 * PEEK (219)
```

END OF LISTING 2

KEY PERFECT 4.0		E0DA	
RUN ON		010D43	
STAFF.MAKER		0172D9	
=====	=====	=====	=====
CODE	LINE# - LINE#		
-----	-----	-----	-----
7777	10 - 100		
01837E	110 - 200		
1416	210 - 220		
		PROGRAM CHECK IS : 046B	

LISTING 3: NOTES

```

6000- 15 00 34 00 4E 00 68 00
6008- 81 00 9A 00 B1 00 C9 00
6010- DC 00 F0 00 FC 00 11 01
6018- 27 01 3C 01 52 01 65 01
6020- 79 01 86 01 92 01 A0 01
6028- B9 01 CA 01 E4 01 00 00
6030- 00 00 00 00 09 36 0E 24
6038- 24 35 36 2E 24 24 35 36
6040- 66 24 24 24 24 24 75 0E
6048- DE 0E 0E B6 01 00 09 36
6050- 0E 24 24 35 36 2E 24 24
6058- 35 36 66 24 24 24 24 24
6060- 75 0E DE 0E 0E 96 05 00
6068- 09 36 0E 24 24 35 36 2E
6070- 24 24 35 36 66 24 24 24
6078- 24 24 75 0E 36 96 0A 00
6080- 00 09 36 0E 24 24 35 36
6088- 2E 24 24 35 36 66 24 24
6090- 24 24 24 75 0E 36 96 2A
6098- 00 00 09 36 0E 24 24 35
60A0- 36 2E 24 24 35 36 66 24
60A8- 24 24 24 24 95 92 92 00
60B0- 00 09 36 0E 24 24 35 36
60B8- 2E 24 24 35 36 66 24 24
60C0- 24 24 24 95 92 92 09 05
60C8- 00 09 4C 09 3F B7 33 0E
60D0- 2D 65 24 24 24 24 24 95
60D8- 92 92 00 00 09 4C 09 3F
60E0- B7 33 0E 2D 65 24 24 24
60E8- 24 24 4D 92 92 12 05 00
60F0- 09 4C 09 3F B7 33 0E 2D
60F8- 65 24 05 00 92 49 21 0C
6100- 64 64 64 64 BC 3F 27 37
6108- 97 3A 2E 2D 6D 89 12 00
6110- 00 92 49 21 0C 64 64 64
6118- 24 25 9F 3F 3C BE 12 37
6120- 2D 2D 4D 12 0E 00 00 92

```

```

6128- 49 21 0C 64 64 64 64 BC
6130- 17 3F 3F 3C 2C 35 4D 89
6138- 92 92 01 00 92 49 21 0C
6140- 64 64 64 24 25 9F 3E 3F
6148- 27 27 2D B6 92 52 49 29
6150- 00 00 92 49 1C 24 0C 2D
6158- 1C 1C 67 25 E5 1C 1C 4D
6160- 91 92 92 00 00 92 49 1C
6168- 24 0C 2D 1C 1C 67 25 E5
6170- 1C 1C 4D 49 92 92 72 00
6178- 00 08 08 2D 36 25 2C 36
6180- 25 2C 2D 56 00 00 41 28
6188- 35 2E 24 35 2E 24 2D B5
6190- 05 00 41 28 25 2C 36 25
6198- 2C 36 25 2C 36 B5 01 00
61A0- 92 4A 76 24 2D 3C 27 24
61A8- 27 2D 2D 3C 3F 6C 21 15
61B0- B6 12 B6 23 24 24 0D 00
61B8- 00 49 09 BE 17 17 24 24
61C0- 24 24 24 AC 92 2A 15 15
61C8- 00 00 92 4A 2E 2C 25 FF
61D0- 24 2C 2C 25 BF 23 24 4D
61D8- 89 37 35 97 32 96 21 24
61E0- 24 2C 00

```

END OF LISTING 3

KEY PERFECT 4.0 RUN ON NOTES

```

=====
CODE          ADDR# - ADDR#
-----
2643          6000 - 604F
2908          6050 - 609F
259F          60A0 - 60EF
2AA0          60F0 - 613F
2A3E          6140 - 618F
2893          6190 - 61DF
0130          61E0 - 61E2
PROGRAM CHECK IS : 01E3

```

LISTING 4: Modifications for II Plus

```

680 PRINT "          C C# D E- E F F# G A- A
B- B": PRINT "
-----": PRINT "SOPRANO: ! "; CHR$
(34); " # $ % & ' ( ) : * -": PRINT
" ALTO: Q W E R T Y U I O P @ RET
": REM 29 HYPHENS
690 PRINT " TENOR: A S D F G H J K L ;
+ -> ": PRINT " BASS: Z X C V B N
M < > /": PRINT "LOW B-: ? RES
T: <SPACE BAR>": PRINT
810 DATA 90,88,67,86,66,78,77,44,60,46,62,47
,65,83,68,70,71,72,74,75,76,59,43,21,81,
87,69,82,84,89,85,73,79,80,64,13,33,34,3
5,36,37,38,39,40,41,58,42,45

```

END OF LISTING 4