JERRY WHITE'S

MUSIC LESSONS

A Comprehensive SOUND EFFECTS and MUSIC Tutorial

SWIFTWARE®

LIMITED WARRANTY ON MEDIA

SWIFTWARE computer programs are recorded on quality media to demanding quality control standards. The media only (cassette or diskette) is warranted to the original retail purchaser to be free from defects in materials and workmanship for a period of ninety (90) days from the date of retail sale.

All warranties of merchantability or fitness for intended use, either express or implied, and any liability for incidental or consequential damages resulting from the use of any SWIFTWARE media is disclaimed and excluded from the retail sales transaction.

SWIFTWARE media which proves defective in materials and workmanship during the warranty period will be replaced, free of charge, if the purchaser returns the defective media, postpaid, along with proof of original retail purchase to: SWIFTY SOFTWARE, Inc., 64 Broadhollow Road, Melville, N. Y. 11747.

After the warranty period, damaged or defective cassettes or diskettes will be replaced for $5.00 each, plus $1.00 shipping and handling, if returned, postpaid, to SWIFTY SOFTWARE.

NOTICE

SWIFTWARE computer programs, documentation and instructions (the "software") are sold "AS IS" and "WITH ALL FAULTS."

All warranties of merchantability or fitness for intended use, either express or implied, and any liability for incidental or consequential damages resulting from the use of SWIFTWARE computer programs, documentation or instructions is disclaimed and excluded from the retail sales transaction. At the time of the original retail sale and delivery the purchaser, not the manufacturer, distributor or retailer, assumes the entire risk and responsibility for determining the suitability and fitness of the computer programs, documentation or instructions for the particular use intended.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitations and exclusions may not apply to you.

SWIFTWARE™

PRESENTS:

Jerry White's

&

PLAYER PIANO

A SOUND AND MUSIC TUTORIAL
FOR THE ATARI 400 & 800 HOME COMPUTER SYSTEMS

PUBLISHED BY:

SWIFTY SOFTWARE, INC.
64 BROADHOLLOW ROAD
MELVILLE, NEW YORK 11746

— TABLE OF CONTENTS —

PROGRAM LISTINGS

APPENDIX I

Jerry White's

&

PLAYER PIANO

INTRODUCTION

The SINGALONG ATARI BASIC SOUND & MUSIC TUTORIAL, provided on side one of the Master Diskette or on sides one, two and three of the cassette version, is designed to provide demonstrations, in ATARI BASIC, of the ATARI HOME COMPUTER system's SOUND capabilities. This extensive documentation provides a complete tutorial on the use of ATARI BASIC to produce a wide variety of sounds. The SINGALONG song programs play songs in up to four part harmony while displaying lyrics on the screen. The fully REMed listings of the programs which make up this package and appear at the end of this manual should prove an invaluable aid and resource to the BASIC programmer.

Side two of the Master Diskette and side four of the cassette version contains PLAYER PIANO, a program which turns the ATARI HOME COMPUTER into a twenty note mini-piano including a full, graphically accurate, screen display of the piano keyboard. Detailed instructions and documentation for PLAYER PIANO appear beginning at page 36 of this manual.

It is assumed that the user will have some understanding of ATARI BASIC commands and of music, generally. These programs are designed to aid the person who desires to learn to create music and various sound effects using ATARI BASIC.

SYSTEM REQUIREMENTS

Minimum RAM: 16K Cassette 24K Disk
ATARI BASIC Language Cartridge
ATARI 810 Disk Drive
or
ATARI 410 Program Recorder
One ATARI Joystick
Any ATARI or equivalent printer: optional

PLAYER PIANO

Minimum RAM: 24K Cassette 32K Disk
ATARI BASIC Language Cartridge
ATARI 810 Disk Drive
or
ATARI 410 Program Recorder
Any ATARI or equivalent printer: optional

PART I:

CHAPTER 1: STARTING UP

STARTING UP - DISK VERSION

Turn on your monitor or TV and BOOT (cold start) your computer using side one of the Master Diskette and the MENU program will run automatically. MENU displays the names of the programs available, assigning a number to each. You may then select the desired program by typing that program's number and pressing the <RETURN> key. The selected program will then be RUN. Each program contains a return to MENU option eliminating the need to enter RUN commands.

STARTING UP - CASSETTE VERSION

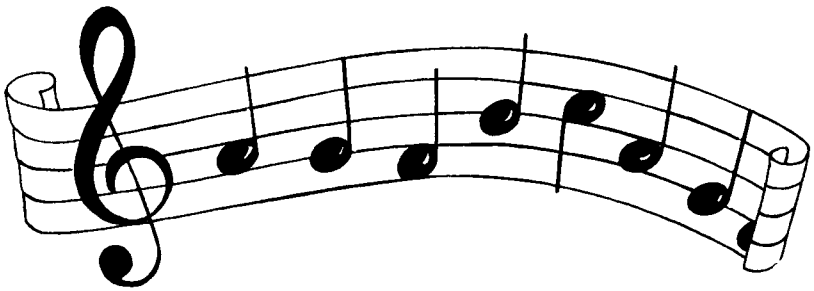
The MUSIC LESSONS programs are organized on the cassettes as follows:

Side 1 - SOUNDEMO, EFFECTS, FORNEXT & POKSOUND
Side 2 - KEYOFC, KEYATOC, KEYDTOG, CHORDS12 & SPEAKER
Side 3 - BIRTHDAY, JINGLE, SILENT & SONGRITE
Side 4 - PLAYER PIANO (SEE PAGE 36)

Turn on your computer and monitor or TV and insert the

selected cassette into your program recorder. Make sure the cassette is fully rewound and the program counter is zeroed. Fully depress the PLAY key on program recorder, then type CLOAD on the computer and press the <RETURN> key twice. The first program on the cassette will now load. When the READY prompt is again displayed on your screen, type RUN and press <RETURN> to run the program you just loaded. To run the next program on the tape, press <SYSTEM RESET> and again type CLOAD and press <RETURN> twice and proceed as above.

The cassette programs are located sequentially on the tape. In order to avoid the necessity of loading all the programs which appear before the desired program you should carefully note and record the index number on the program recorder at the beginning of each program (which is the reason you zeroed your counter). After loading all the programs on each tape and recording the index numbers at the beginning of each, you will be able to randomly access any program on the tapes by fast forwarding to the appropriate index number, and following the above loading instructions.



SOUNDEMO permits experimentation with the wide range of sounds available using the BASIC SOUND command.

EFFECTS is a self-running demonstration of a wide range of sound effects and simple BASIC animation.

FORNEXT permits experimentation with the BASIC FOR/NEXT loop to create sound effects. The resulting BASIC instructions are displayed and may optionally be printed or saved on diskette.

POKSOUND permits experimentation with the extended range of sounds made possible by using the POKE command.

KEYOFC demonstrates the C major scale and chord using colorful Graphics Mode 7 while generating the sound of each note, and displaying the appropriate SOUND commands.

KEYATOC and KEYDTOG use high resolution Graphics Mode 8 to display all the major chords and related information.

CHORDS12 demonstrates the SOUND commands required to generate deep bassnote chords.

SPEAKER demonstrates the use of the console speaker. The user may enter parameters that will alter a machine language subroutine capable of generating a wide range of sounds.

BIRTHDAY, JINGLE, and SILENT play the well known songs "Happy Birthday to You", "Jingle Bells" and "Silent Night" in up to four part harmony as the lyrics are

displayed on the screen.

SONGRITE is a skeleton singalong program with no song data. A user may insert notes and lyrics to create additional SINGALONG songs.

The cassette version of these programs ends with a BASIC END command. The disk versions have an additional routine that provides three options. By using the OPTION or SELECT button to highlight the desired option, you may then press the START key to rerun the current program, return to BASIC as in the cassette versions, or RUN the disk MENU program.

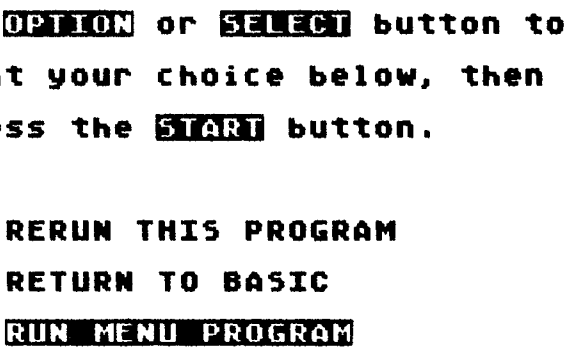


Figure 1 - End Screen (Disk Version)

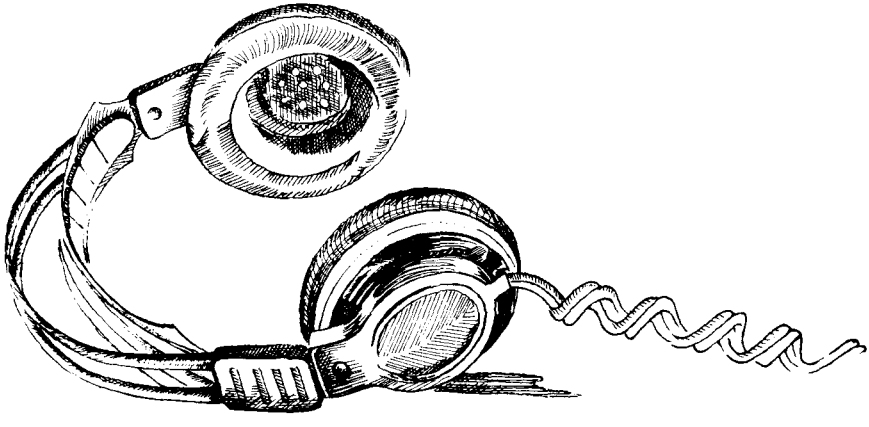
The printed program listing reflects the disk version of each program. Cassette users should ignore program lines numbered above 30000.

CHAPTER 3: THE SOUNDEMO PROGRAM

This program provides an easy way to experiment with the ATARI BASIC SOUND command. The SOUND command must be followed by four parameters. These parameters may be numeric constants or numeric variables. The following is a sample SOUND command:

SOUND VOICE,PITCH,DISTORTION,VOLUME

The first parameter is called the VOICE. Your ATARI computer has four sound channels or voices numbered zero through three. The second parameter is called the PITCH. The pitch can be any integer value from 0 through 255. When the clear sound of distortion level 10 is used, increasing the pitch value makes the sound deeper or lower, and decreasing the pitch value makes the sound higher. The third parameter is called DISTORTION. This value can be any even number from 0 through 14. A clear or undistorted sound is obtained by using a value of 10 or 14. The fourth parameter is called VOLUME. This value may be any integer from 0 through 15. A value of zero shuts off a sound. As the value increases, the sound becomes louder.



As the program runs, it provides it's own instructions on the screen. Just make sure you are not using lower case or inverse video.

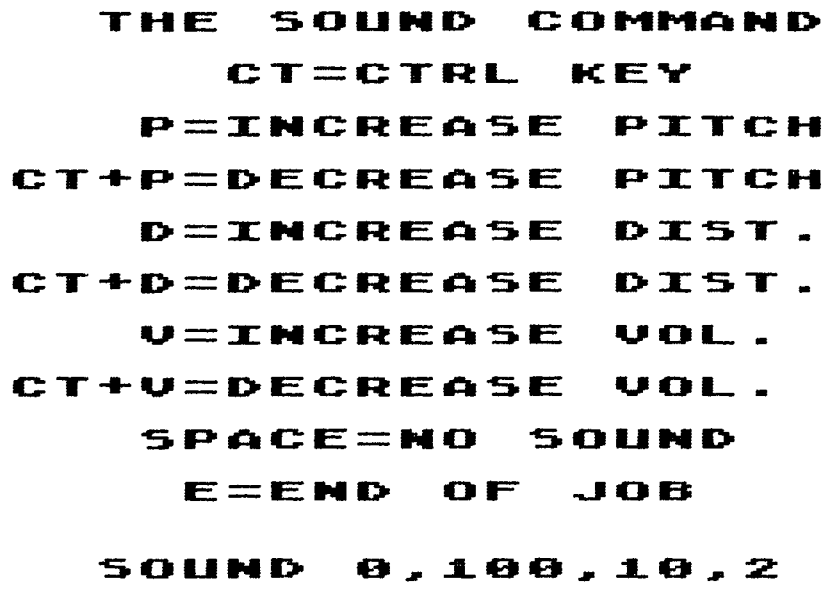


Figure 2 - SOUNDEMO Screen

If you look at the SOUNDEMO program listing, you will find the program to be small and simple. The variables P=Pitch, D=Distortion, and V=Volume.

Note that the program looks at location 764 to find the internal value of the last key pressed, and reacts accordingly.

CHAPTER 4: THE EFFECTS PROGRAM

The EFFECTS program demonstrates a wide range of sound effects and simple graphics and is the largest program in this package. It requires no operating instructions; just turn up the volume on your TV or monitor, watch, and listen.

There is a great deal that can be learned by studying the EFFECTS program listings. Each section of the program is labeled using inverse video REM statements. To explain each instruction would require a prohibitive amount of text, but to supply no explanation at all would likewise be unacceptable. The following documentation therefore, is a compromise. We assume you understand the function of each BASIC command so that we can briefly describe the routines rather than the individual instructions.

Before we begin, there are a few methods of debugging I'd like to explain that might prove helpful in case of confusion. The one I find most effective is to act as if you are the computer. Some guy named BASIC is your boss and he is giving you detailed instructions. You take each instruction, one at a time, and do exactly as you are told. The computer probably has a better memory than you have, so keep a pencil and paper handy. The instructions that you, (the computer), execute, will change things. Write these things down and make changes as needed.

For example, if the numeric variable X were equal to 5, you should have "X=5" written down. Then if BASIC told you that X=X+1, draw a line through "X=5" and write "X=6".

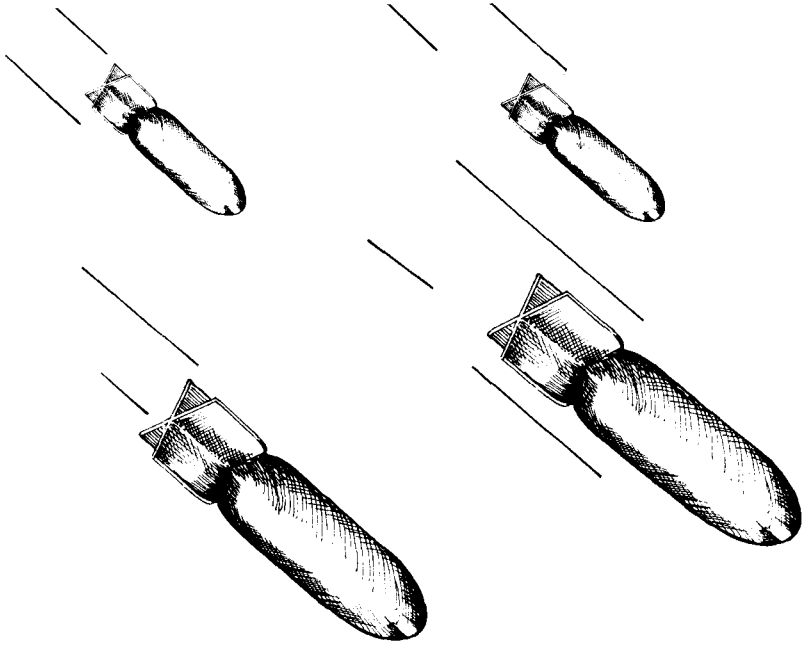
You might also BREAK into the program as it is running, make changes, and then rerun the program or routine, to

see how your changes affected the program. If you BREAK into a program while sounds are on, you may wish to shut them off. There are a number of ways to do this. The three easiest ways are to enter an "END" command and press <RETURN>, press the <SYSTEM RESET> key, or turn off the volume on your TV or monitor. The latter will not effect the SOUND channels in the computer because the computer won't care, even if you should turn off the monitor completely.

O.K., let's take a look at the EFFECTS program listing. I will skip the AUTHOR INTRODUCTION section because it uses routines which will be discussed in the documentation of the FORNEXT and POKSOUND programs. Notice that line 320 says GOTO 350. This was done to skip over two subroutines. At line 350 we begin by using the subroutine at line 340. This simply sets up good old standard text mode, sets the colors to black and white, and turns off the cursor. RETURNing to line 350, we display the word "TARGET" near the bottom of the screen.

Lines 360 thru 380 contain a FOR/NEXT loop that will change the variable PITCH from 20 to 200, by increments of two. As the loop is executed, the word "BOMB" is displayed on the screen along with B$(1,4). B$ contains 30 blanks, so B$(1,4) is equal to four blanks. This is used to erase the word "BOMB" without erasing the entire screen. As the BOMB drops, the change in the value of the variable PITCH will lower the frequency in the SOUND command in line 380.

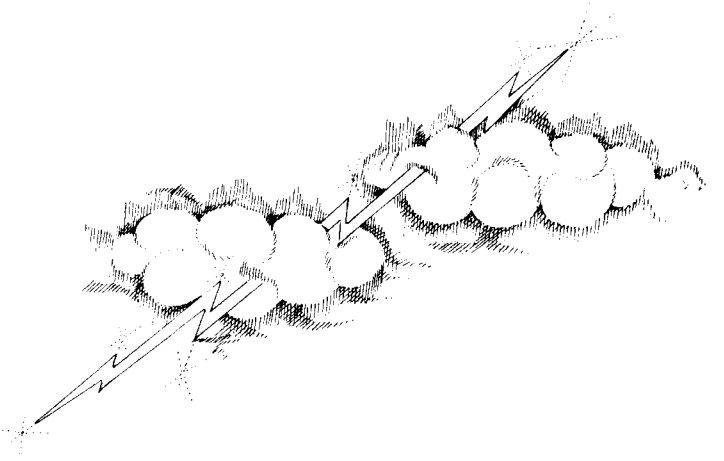
At line 390, we erase the screen and begin repositioning the word "TARGET." Note the POKE 755,4 at the end of line 390. This creates what is called vertical reflect or mirror image. The word TARGET is spread over three lines of the screen, crudely simulating a hole left by the bomb.



The FOR/NEXT loop in lines 410 and 420 alters the PITCH and VOLUME of a SOUND command, and the luminance of a SETCOLOR command, creating an explosive sound and red flash on the screen. As the smoke clears, you will notice the word TARGET displayed in vertical reflect video. It will stay that way until the POKE 755,2 sets things back to normal in line 430. By the way, some interesting effects are demonstrated when a 1 or a 3 is poked into location 755 while inverse video characters are displayed on the screen. Notice, also, the WAIT loop in that same line. All it does is kill time and is NOT the best way to accomplish a delay. It is, however, effective in this case and easily adjusted. The problem with FOR/NEXT loops is that they execute faster if they are near the beginning of a program. A more exact method of timing will be explained later on when we get to the "THAT'S ALL FOLKS" routine.

If the FOR/NEXT loop has you a bit confused, don't let it get you down. There is a program dedicated to the FOR/NEXT loop, called FORNEXT of all things, which you will get to shortly. After some experimentation using that program and possibly a bit of trial and error testing on your own, it will all make sense.

The LIGHTNING routine uses GRAPHICS MODES 21 and 23 which are the same as modes 5 and 7, except that they do not have a text window. Lines are displayed on the screen to crudely simulate lightning. After each line is drawn, the subroutine at line 330 is executed.



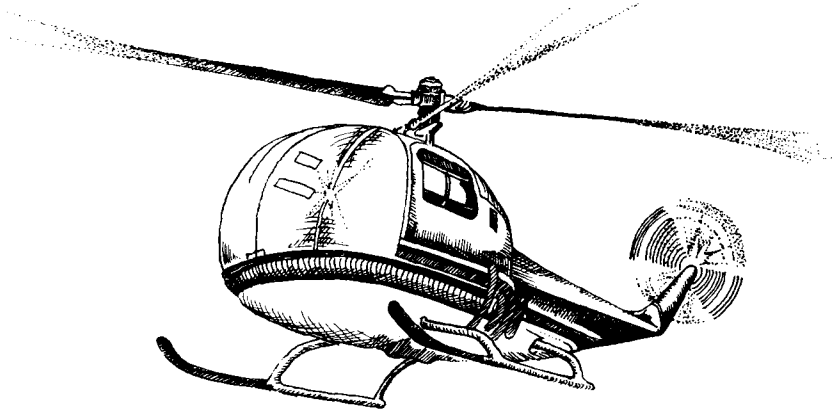
Why didn't we just put this routine in the next sequential line you may ask? Subroutines are often used to save memory and typing. This subroutine is used four times and by using four GOSUB commands, we only need to enter the instructions found in line 330 once.

Next we have our sputtering helicopter courtesy of Dave Culbertson and C. E. Software. In order to make the graphics easy to understand, we have defined seven strings. Again we use FOR/NEXT loops to alter the sound and the helicopter's appearance and position on the screen.

There are other ways to position character graphics on the screen. Since this is a tutorial on sound, I won't dwell on this or on the Player/Missile Graphics routine. I would just like to point out one alternative method. The special function keys such as the arrows can be used within a string. They must be preceded by pressing the

<ESC> key when they are entered into a program. Once they are part of a string, they will be executed as if someone were actually holding the <CTRL> key while using the arrows to position the cursor.

The preceeding paragraph was an example of a programmer going off into a subroutine. I'll try not to let that happen too often. Now, where were we?



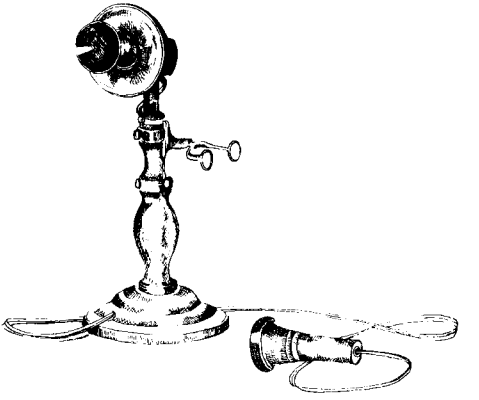
By the sound of that helicopter, it probably crashed by now so let's go on to the ALERT SIREN routine. There's no animation here, the nested FOR/NEXT loops change only the sound and color of the word ALERT. NESTED??? Nested refers to one or more loops within another loop. Again, we will discuss this in depth when we look into the FORNEXT program.

The BOUNCING BALL uses the ATARI ball character (CTRL T) to demonstrate simple animation. As the ball rises, the sound increases in pitch and decreases in volume. This combination provides what I call a "BOING" effect.

This brings us to a very complex subject called Player/Missile (P/M) graphics. Again, I will refrain from expounding on this subject since it requires a

separate tutorial of it's own.

The TELEPHONE RINGING routine demonstrates that sound effect, plus the sound of a teletype machine. These results are achieved using what I call short sounds. Identical SOUND commands can be made to sound different when the duration of the sound is limited. By now you may be wondering how to select the pitch, distortion, volume, and duration, to create other sound effects. I know of only one way: you must experiment and use the old trial and error method.



To provide more food for thought, we now come to the section labeled "DISPLAY FOR NEXT LOOPS." This is a series of nested FOR/NEXT loops that will be seen as BASIC instructions on the screen as these instructions are executed.

Finally we arrive at the "THAT'S ALL FOLKS" and "END OF JOB" routines. The first is a single voice tune you may have heard before. The pitch and duration is determined by reading the DATA found in lines 2710, 2720, and 2730. Playing music often requires a more exact method of timing than the FOR/NEXT loop can provide. There are several clocks or timers built into your computer. This program uses what is called a

countdown timer at RAM location 540. This timer counts backwards to zero at the rate of 60 per second. One sixtieth of a second is called a "jiffy."

At line 2750, a TRAP is set to line 2760. Since this is a subroutine and line 2760 contains a RETURN command, the program will RETURN from this subroutine when an error occurs. Following the TRAP command is the instruction READ N,T. Look at the DATA in line 2710. When the READ instruction is executed, N will be equal to 40 and T will be equal to 19. The next instruction on line 2770 is GOSUB 2770 followed by GOTO 2750. In other words, when we RETURN from the subroutine beginning at line 2770, the GOTO 2750 command will create a loop back to the beginning of the current line number. In other words, we will be reading DATA, going to a subroutine, then repeating this procedure until an error occurs. The error will occur when there is no more DATA to read.

The subroutine beginning at line 2770 pokes the value of T into location 540. In this case, T=19. The countdown timer will immediately begin decrementing the value in location 540 at the rate of 60 per second. Also on line 2770 are is a SETCOLOR and a SOUND command. The value of N will be used to change the background color and provide the Note or Pitch for the SOUND command.

In line 2780, we check to see if our countdown timer has reached zero. If not, we just loop there until it does. In this case, it effectively causes the sound and background color to remain the same for 19/60ths or about 1/3 of a second. I used 19 instead of 20 to allow an extra jiffy for turning sounds on and off, etc.

Once we PEEK into location 540 and find a zero, we fall through to line 2790 which shuts off the SOUND and provides the RETURN command we need to get back to line 2750.

The final END OF JOB ROUTINE uses the teletype effect

and lets you know that this demonstration program is over.

CHAPTER 5: THE FORNEXT PROGRAM

This program is designed to allow the user to experiment with nested FOR/NEXT loops for generating sound effects. The user supplies parameters for an inner loop that controls pitch, and an outer loop that controls volume. The BASIC instructions that create each sound subroutine are displayed on the screen. These instructions may then be printed or saved on diskette for later use.

When you run the FORNEXT program, instructions will be displayed on the screen. Don't be concerned if they appear to be confusing. This program will let you learn by trial and error. After reading the introductory screen, press any key to continue.

```
FORNEXT PROGRAM VARIABLES:

DIST=DISTORTION variable (0 thru 14)
even numbers only.

PITCH=any number from 0 thru 255.
We will change the pitch in loop L2.

L1=Outer loop 1 VOLUME.
Type any integer from 0 thru 15
At prompts FROM, TO, and STEP.

L2=Inner loop 2 PITCH.
Type any number from 0 thru 255
At prompts FROM, TO, and STEP.


PRESS ANY KEY TO BEGIN
```

Figure 3 - FORNEXT Screen 1

You will now be asked to enter a series of parameters. A

parameter is simply a value that you supply to a program or instruction. You can experiment by entering various parameters later on. For now, make the entries that appear below by typing the number found within the quotation marks, and pressing the <RETURN> key after each entry.

```
SOUND TEST

TYPE DIST     ?10
TYPE L1 FROM  ?15
TYPE L1 TO    ?0
TYPE L1 STEP  ?-0.5
TYPE L2 FROM  ?100
TYPE L2 TO    ?10
TYPE L2 STEP  ?-10
```

Figure 4 - FORNEXT Screen 2

The first prompt asks you to "TYPE DIST". DIST is the distortion level to be used. Enter "10".

The second prompt asks you to "TYPE L1 FROM". L1 is the outer loop and will control volume. FROM indicates the starting volume. Enter "15".

The third prompt asks you to "TYPE L1 TO". This is the ending volume. In this case we are trying for an effect that begins relatively loud and decays until there is no sound at all. Enter "0".

The fourth prompt asks you to "TYPE L1 STEP". The larger the step, the faster we will get from the beginning of this routine to the end. Let's make this a very slow rate by entering "-0.5".

The fifth prompt asks you to "TYPE L2 FROM". This is

the starting value of the inner loop which controls pitch. Enter "100".

The sixth prompt asks you to "TYPE L2 TO". This is the ending value of our pitch. Enter "10".

The seventh and final prompt asks you to "TYPE L2 STEP". We want the pitch to go from 100 to 10 rather quickly. Enter "-10".

You should now hear the sound effect generated using the parameters you have just entered. The BASIC instructions that were used will be displayed near the top of the screen. Further down you will see that you now have five choices.

```
YOUR SOUND SUBROUTINE:

101 FOR L1=15 TO 0 STEP -0.5
102 FOR L2=100 TO 10 STEP -10
103 NEXT L2
104 NEXT L1
105 RETURN


TYPE 1 TO SAVE SUBROUTINE
TYPE 2 TO PRINT SUBROUTINE
TYPE 3 TO REPEAT SUBROUTINE
TYPE 4 TO ENTER PARAMETERS
TYPE 5 TO RUN MENU
```

Figure 5 - FORNEXT Screen 3

Number 3 will allow you to hear that sound subroutine again, so type the number 3. There is no need to press the <RETURN> key since this routine just checks the last

key pressed.

You will see three new options on the screen. To repeat the sound subroutine exactly the way you just heard it, type 1. Option 2 will disable screen interrupts before repeating the sound subroutine. Since the computer will not have to bother updating your screen sixty times per second, execution will be much faster. By choosing option number 3, the SOUND command will be displayed on the screen so you will be able to see the variables change as the subroutine is executed. This will slow down the execution and will also be a great help in understanding exactly what is happening, and how nested FOR/NEXT loops work.

If you want to save a subroutine so that you may use it in one of your own programs at a later date, use option number 1. This option will LIST your sound routine to disk so that it may be ENTERed into another program. DO NOT write on your master diskette. It has been write protected for your own safety. Make sure a DOS 2.0S formatted diskette is ready in Disk Drive 1 before continuing, if you decide to use this option.

We must have some name for this disk file. Try to think of a word of up to eight letters that describes the sound to you. I suggest you follow this filename with the extention .LST to indicate that it is a LISTed or untokenized file. If you can't think of a filename, enter TEST1.LST then press <RETURN>. The subroutine will be LISTed onto the diskette in Disk Drive 1.

Option number 2 will allow you to print the sound subroutine. Any 40 or 80 column printer will do.

Now, if you'd like to experiment using other parameters, use option number 4. When you're ready exit this program, use option number 5. Remember, if you have removed your master program diskette, you must put it back in Drive 1 before typing the number 5.

CHAPTER 6: THE CHORDS12 PROGRAM

Up until now, all of our music has been played using SOUND commands with a Distortion variable of 10. You may have noticed that there were no deep bass sounds. Using distortion 10, the lowest note possible isn't very low.

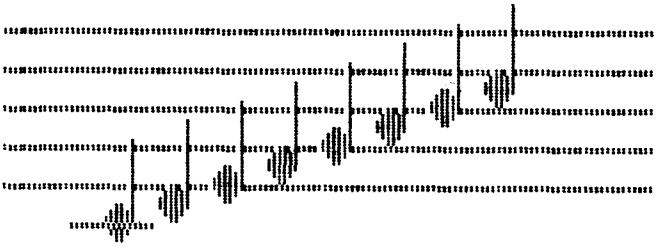
So how do you get bass notes? I'm glad you asked. I just happen to have included a program to demonstrate bass chords called CHORDS12. The key to deep bass sounds is that number is 12 used in the distortion variable.

The program begins by playing a very low A major chord. Each time you press the <START> key, the next highest major chord will be played and the SOUND commands used to create each note will be displayed. Don't bother trying to memorize them, a separate chart of chords is provided in the appendix section of this manual.

When we reach the A major chord again, we will be one octave higher than when we started. When you press <START> this time, every pitch from 255 to 0 in distortion level 10 will be displayed on the screen, one at a time. At the conclusion of this FOR/NEXT loop you will be returned to the MENU program.

CHAPTER 7: THE KEYOFC, KEYATOC & KEYDTOG PROGRAMS

There are three programs that demonstrate chords by drawing notes on the screen, displaying the SOUND commands required to create each note and executing these SOUND commands.



THE C SCALE

C D E F G A B C
PRESS ANY KEY TO CONTINUE.

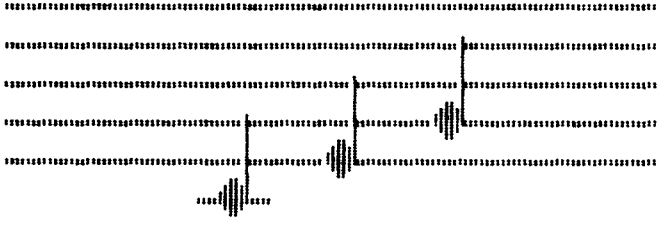
Figure 6 - The C Major Scale

The KEYOFC program demonstrates the key of "C" only and will not only show you a C major chord, but will also show the C major scale. These programs do not teach music but merely use the treble clef to show what many notes look like in sheet music form, while standard major chords are played.

You should be able to understand how the KEYOFC program works by reading the program listing. The variable X is used to store the horizontal screen

coordinate while Y is used for vertical positioning. Notes are drawn by specifying the X and Y coordinates, and then using the subroutine beginning at line 160 to draw the note on the screen. Although seven PLOT and seven DRAWTO commands were used along with addition and subtraction, the notes are drawn rather quickly. Who says ATARI BASIC is slow?

The KEYATOC (Keys of A, A#, B, C, and C#), and KEYDTOG (Keys of D, D#, E, F, F#, G, and G#), programs, use Graphics Mode 8. The BASIC code is rather difficult to read because so many variables are used in order to save memory. They are easy to understand when you run them, so further documentation will not be required.



THE C MAJOR CHORD
C + E + G
PRESS ANY KEY TO CONTINUE.

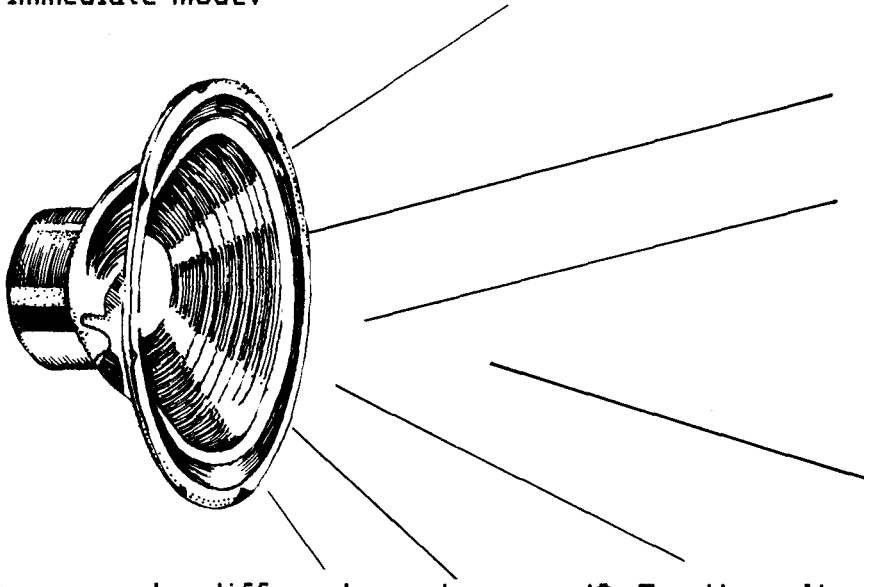
Figure 7 - The C# Major Chord

I do not deny that languages like FORTH and ASSEMBLER are much faster than BASIC, and provide the keys to doors that BASIC cannot enter. This tutorial package,

however, does prove that BASIC is fast enough to play music and create a wide range of SOUND effects.

CHAPTER 8: THE SPEAKER PROGRAM

As if having four sound channels weren't enough, there is one other way your ATARI computer can generate sound. As you may have noticed there's a tiny speaker built into the computer's console. When you enter a CLOAD or CSAVE command, the buzzer sound you hear comes from this speaker. You can use this speaker yourself be executing the BASIC command PRINT CHR$(253) or by pressing the CTRL and the number 2 key in immediate mode.



Do you want a different speaker sound? Try these lines of BASIC in immediate mode, one at a time:

```
POKE 66,1:PRINT CHR$(253):POKE 66,0

FOR TIME=0 TO 9:POKE 53279,0:NEXT TIME

FOR NUMBER=0 TO 255:POKE 53279,NUMBER:NEXT
NUMBER
```

As far as I know, these are all the sounds BASIC can get

out of that speaker. But remember that I said Assembler can open doors not available to BASIC. This program presents you with one of those keys in the form of an Assembler subroutine.

Forgive me, but I cannot teach Assembler, or explain this routine in great detail. It's not that I don't want to; to tell you the truth, I'm not exactly sure myself. I have a confession to make. I stole this routine from the ATARI Operating System, altered it somewhat, put it into a Basic String and then figured out which bytes alter the resulting sound coming from the speaker. Correct me if I'm wrong, but it appears that the tenth byte of JW$ affects SPEED or duration, the twentieth byte seems to control the loop of speaker on and byte 28 the loop of speaker off.

So what does all this mean to the BASIC programmer? It means that if you take lines 10 and 40 out of this program and put them into your own, you can generate various sounds with the speaker by changing these three bytes as demonstrated.

RUN the SPEAKER program now. Press the <START> key to hear a demonstration of a chord played at various speeds. At this point you can press the <OPTION> key to return to the MENU program, or press the <START> key to edit or experiment. Be brave, press <START>.

You may now enter various values that will alter those three bytes we spoke of earlier, and execute the assembler subroutine at line 10. You may enter any integer from 0 thru 255. For example, try a DURATION of 20, SPEAKER ON=80 and SPEAKER OFF=50. After the speaker squawks, you may press the <OPTION> key to repeat that squawk, or the <START> key to continue entering experimental data.

```
ENTER NUMBERS FROM 0 TO 255 RETURN
DURATION?20
SPEAKER ON?80
SPEAKER OFF?50
```

```
Press OPTION to repeat
Press START to continue
```

CHAPTER 9: THE POKSOUND PROGRAM

The POKSOUND program allows you to experiment with sound by using a joystick to POKE values directly into the SOUND registers. The 'greater than' symbol ">" points to the currently selected location. Selection is made by pushing the joystick forward to move up on the screen or pulling back on the joystick to move down. When the ">" is positioned next to the desired location, press the red trigger button. Now, push forward on the joystick to increase the value being POKEd into the selected location, or pull back to decrease the value. When you have reached the desired value, press the trigger to select a new POKE location.

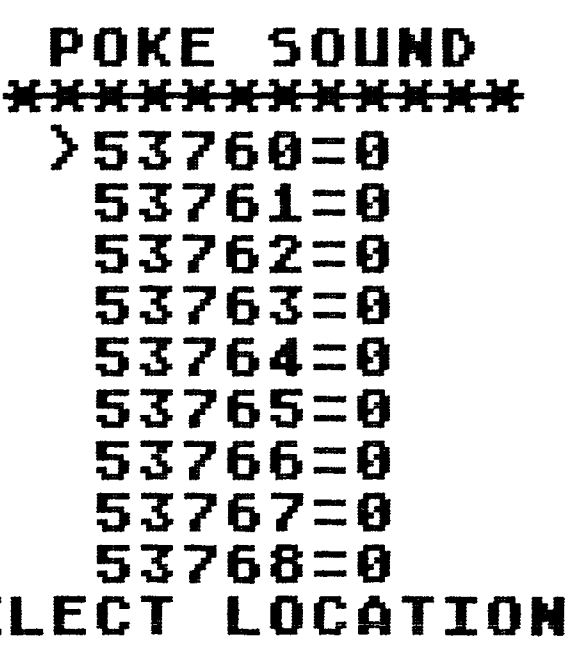


Figure 9 - The POKSOUND Screen

Pressing the space bar will instantly shut off all sounds by poking a zero into locations 53760 thru 53768. Press the ESC key to exit this program.

Detailed information on the use of these sound registers may be found in the SOUND Chapter of De Re ATARI, published by the ATARI PROGRAM EXCHANGE (APEX).

Locations 53760, 53762, 53764 and 53766 are audio frequency registers. Locations 53761, 53763, 53765 and 53767 are audio control registers. Location 53768 is an audio control register that effects all four sound channels. It's effect depends on the individual bits which are set by the value being POKEd into it. This is the reason why binary values are being displayed along with decimal values. The use of this register requires extensive background knowledge. Again, I'd recommend De Re ATARI to those who wish to obtain additional information.

These registers are called "Write Only" because they are updated constantly by the Operating System. In other words, if you POKE a number into one of these locations and then PEEK at that same location, the odds are 256 to 1 against finding the value you have just POKEd in.

For purposes of experimentation, you may use this program to get a feel for the various sounds made possible using POKE commands.

CHAPTER 10

THE SINGALONG PROGRAMS

This package includes three singalong programs called BIRTHDAY, JINGLE and SILENT. Each plays a song while the words to the song are displayed on the screen. BIRTHDAY plays the Happy Birthday Song, JINGLE plays Jingle Bells and SILENT plays Silent Night. There is also a program named SONGRITE which is similar to the other singalong programs except that the DATA required to play music and display lyrics has deliberately been left out. By carefully studying the singalong programs, you can learn to enter data into SONGRITE and create your own singalong song programs.

Some knowledge of music is required. The following documentation will explain how the BIRTHDAY program works. This will provide the BASIC knowledge you will need to create other singalong songs.

BIRTHDAY uses four string variables. NAME$ holds the name of the person to whom our song will be dedicated. WORD$ holds the words of the song as they are read from DATA statements. DISP$ will hold the words as they will be displayed on the screen. BLANK$ holds a string of twenty blanks and is used to center the words on the screen. Each string is dimensioned at twenty characters which is the maximum number we can display on the screen at any given time. The words are displayed in GRAPHICS MODE 18 which is MODE 2 without a text window.

As the program begins, we are in GRAPHICS MODE 0 and must enter the name of our birthday person. Our string dimensions are made in lines 20 and 40, then we skip over some subroutines and begin reading data at line 300. For each change of sound and display, we will read a pitch for each of the four voices and store this data in the variables V0, V1, V2 and V3. We will also read data to be stored in the variables HOLD, SWITCH and WORD$. HOLD will tell the program how long to hold the current sounds and display in 60ths of a second. SWITCH will be used to note special functions such as when to display the birthday persons name instead of the words found in our DATA. WORD$ will store the words as they are read from the DATA. It is most important to note that each line of data MUST contain seven items, separated by commas.

After each set of seven items of data are read, we check the SWITCH variable to see if we should do something special. In this program, I check for 1, 3 or 9. If SWITCH=3 then I want to display the person's name and not the word, "NAME". If SWITCH=1 then I want to clear the screen, and go onto line 360. If SWITCH=9 then the song is over and I want to GOTO line 9000.

The routine in lines 360 and 370 does some string manipulation and eventually winds up with the words as they will be displayed in the string DISP$. This string contains the leading and trailing blanks necessary to display our words neatly centered on the screen. At line 380, we put the words on the screen, GOSUB 100, then back to line 300 to get more DATA and do it all over again.

The subroutine at line 100 POKEs the value stored in the variable HOLD, into the countdown timer at location 540. Then we turn on all four voices using the pitch values we have just read. Notice that I have voice zero (SOUND 0), twice as loud as any other. This is my melody voice which is louder so that it stands out. You may change volumes

as you see fit. Remember, however, that you should not exceed a total volume of 32 or distortion and decrease in volume will occur.

We loop at line 101 until our countdown timer has gone back to zero. Then we turn off the four voices at line 102 and return from this subroutine.

The changes in the color of the words on the screen is coordinated with the current syllable being sung. This is done by keeping the syllables yet to be sung in inverse video. A quick look at the words to our BIRTHDAY song will show you how this works.

The hardest part of entering the DATA is determining the pitches for the four voices. You will either have to use the old trial and error method, or read sheet music and enter the data accordingly. There is however no need to use four part harmony. To use only one voice, remove the SOUND commands for voices 1, 2 and 3 in lines 100 and 102 and the reading of V1, V2, and V3 from line 300.

You may notice that, in most cases, I am only using two different pitches in the birthday song. Only the final two notes actually use four part harmony. Throughout the rest of the song, I'm putting the same melody note into V0 and V1, and the note one octave lower, into V2 and V3.

When we need a REST (a period with no sound at all) we can specify pitch zero. This is almost cheating but it works because that pitch is beyond the normal range of human hearing. If you have a dog, Rover may begin to sing during rests.

About that HOLD variable, again, trial and error will be the best teacher. There is a method to my madness. The reason I used numbers line 19 and 39 instead of 20 and 40, was to allow that extra jiffy to shut off sounds and get more data, etc. As it turns out, I probably should have allowed a bit more time since I'm also checking

switches and centering text on the screen. The method I used does seem to work, although I'm sure many of you will eventually find better ways. If I were a better musician, I probably wouldn't be writing this documentation right now.

The JINGLE program is somewhat different. Notice that location 20 is being used as a timer instead of location 540. This timer counts up instead of down, and is somewhat more reliable than the countdown timers. Graphics Mode 7 is used to display the Christmas tree, and P/M graphics are used to create the lights on the tree. Since the computer has more display work to do in mode seven, and we are adding instructions to change the colors of the players, I had to sacrafice text centering. The syllable color changes have been replaced by simple phrase displays. Of course we could have the best of both worlds by adding some machine language, but I wanted this program to be 100% BASIC. It was intended to illustrate how much can be done without resorting to machine language. Notice that the SOUND commands have been replaced by POKEs. This was done for the sake of speed, and to demonstrate an alternative method of turning sounds on and off.

I can give you a few tips from my experience in entering the data to the song programs in this package. Don't enter more than one or two lines of DATA without testing, and test by running the program and listening carefully. Use the BIRTHDAY song as a guide because it is short and slow. The whole notes last for 80 jiffies or

one and one-third seconds. To get the values of halfnotes and quarternotes, simple division does the trick. Half of 80 is 40, a quarter of 80 is 20. Subtracting 1 provides the durations 79, 39 and 19. If you come out to a noninteger, make like ATARI BASIC and round off to the lower number.

PART II:

PLAYER PIANO

CHAPTER 11: OVERVIEW

Here's a program that turns your ATARI Home Computer into a twenty-note minipiano. Everyone can enjoy this versatile program, regardless of musical ability or training, PLAYER PIANO can also introduce young children to computer applications beyond game playing.

This automatically loaded program displays the twenty black and white piano keys, and a musical note symbol jumps across the keys as you play your tune. The computer keyboard keys corresponding to the piano keys display in the lower part of the screen. With PLAYER PIANO, you create tunes by pressing a key, causing the note to play until you press either another key or the space bar to rest the note. A series of menu options let you save your tune as you create it, modify it as desired (now or later), play all or part of it back at any time, and store it on cassette or diskette for recall at a later time. You can build tunes having as many as 400 notes, composed of whole, half, quarter, and eighth notes, and having a variety of tempos.

An auxiliary program lets you display on your TV screen or print a data listing of your longer, more complex songs for analysis and modification.

THE SPECIAL FUNCTION KEYS

The only function keys that PIANO PLAYER uses are <ESC>, <DELETE/BACK S>, and <RETURN>. The program has disabled all other function keys (except <SYSTEM RESET>) to guard against user errors.

THE <ESC> KEY

Use the <ESC> key to leave your current option or menu selection. As a rule of thumb, if you don't see what you want, press the <ESC> key.

THE <DELETE/BACK S> KEY

Use the <DELETE/BACK S> key to backspace and erase the previous key displayed in the text window at the bottom of the PIANO screen display. Note that you don't need to use the <CTRL> key to use the <DELETE/BACK S> key.

INVERSE VIDEO AND LOWER CASE

Enter all your information in uppercase and in normal video. If you accidentally press the ATARI inverse video key or the <CAPS/LOWER> key, the program automatically resets your input to uppercase, normal video mode when you press another valid key.

THE <RETURN> KEY

Throughout PLAYER PIANO, you don't need to press the <RETURN> key when you enter a single-character response anticipated by the program. (However, no harm is done if you do press the <RETURN> key in these instances.) Since your response can vary in the number of keystrokes, then you must press the <RETURN> key after typing in your response.

CHAPTER 12: LOADING PLAYER PIANO

Be sure your ATARI BASIC Language Cartridge is installed in your computer. If you plan to use your printer, turn it and your interface module (if Applicable) ON now. Make sure your printer is in ONLINE mode.

STARTING UP - DISK VERSION

Be sure your computer is turned OFF. Turn your disk drive ON. When the BUSY light goes out, open the disk drive door and insert the PLAYER PIANO diskette (Side 2 of the package) and close the door. Now turn your computer ON. The program will load into computer memory and start automatically. If you have not already done so, turn ON your TV also.

STARTING UP - CASSETTE VERSION

Turn on your TV set. Now turn ON your computer and connect your program recorder to the computer and to a wall outlet. Slide the PLAYER PIANO cassette (Side 4) into the program recorder's cassette holder and press REWIND on the recorder until the tape rewinds completely. Reset the program counter to zero then press PLAY. Now type CLOAD on your computer and press the <RETURN> key two times. The tape will load PLAYER PIANO into computer memory.

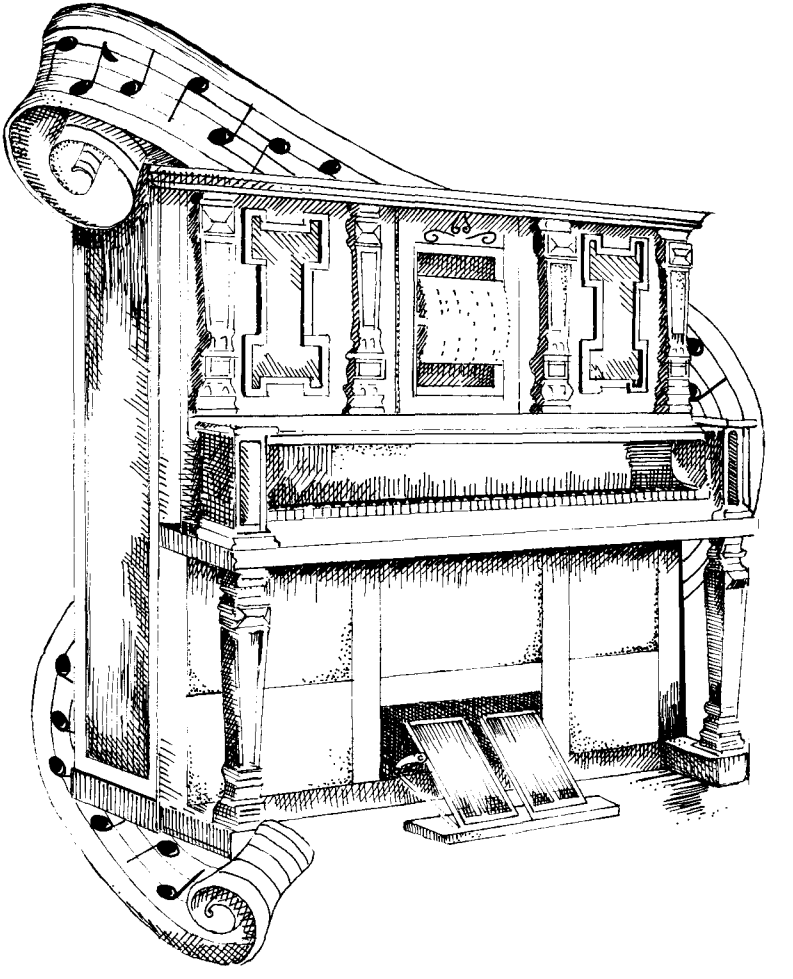
After the tape finishes loading, the word READY will display on your TV screen. Type RUN and press the <RETURN> key. The first display screen will appear on your TV screen.

IMPORTANT NOTE: The program FILEDUMP is <u>not</u> loaded with PLAYER PIANO on the initial load and must be loaded seperately. Make careful note of the number which appears on the program counter following the loading of PLAYER PIANO. FILEDUMP is located on the cassette

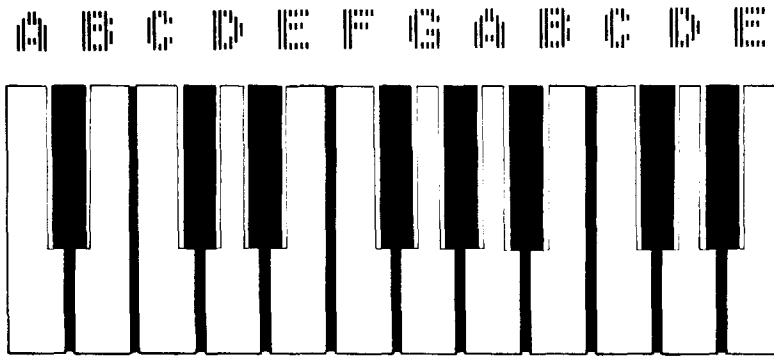
immediately following PLAYER PIANO and may be CLOADed by returning to the correct point on the cassette using the program counter. Instructions for use of FILEDUMP are contained in Chapter 14.

THE FIRST DISPLAY SCREEN

The title, author, and copyright information display briefly. Then, the program displays the menu described in the next section.



CHAPTER 13: USING PLAYER PIANO




```
  W     R  T     U  I  O     -  =
A  S  D  F  G  H  J  K  L  ;  +  *
     PRESS SPACE BAR FOR REST
     PRESS ESC FOR OPTIONS
```

PLAYER PIANO MENU

The program's menu looks as follows:

```
TYPE OPTION NUMBER

1=PLAY PIANO

2=READ DATA

3=CREATE DATA

4=END PROGRAM
```

Within each option are sub-options letting you create songs, save or load song data files using cassette or diskette, fix or change as many as 400 notes in computer memory, and play all or part of a song. The same suboptions are available in options 1, 2, or 3.

Choose your option according to the first activity you want to do. That is, use option 1 to experiment with tunes first. Use option 2 to load in a song data file from cassette or diskette first. Use option 3 to store a series of notes in computer memory immediately. Aside from the different initial activity, the discussion under option 1 below applies to these other two options as well. Option 4 returns you to the READY prompt in ATARI BASIC. Choose an option by typing its corresponding number.

OPTION 1 - PLAY PIANO

Within option 1, PLAY PIANO, you use suboptions to create and revise a song, save your song in memory while you create other notes, play back your saved song, add more notes to your saved song, save your song on cassette or diskette, read in a song data file from cassette or diskette, and erase the song saved in memory and save a new one.

PIANO DISPLAY

When you type a "1" in response to the OPTION prompt, the program draws a piano on the screen and a display in the text window of the keyboard keys corresponding to the piano notes.

The center two rows on your keyboard activate musical notes. The middle row of letter keys, from A through *, play the natural (white) notes. Some of the top row of letter keys, as indicated in the text window, play the sharps and flats (black keys). Use the SPACE BAR at the bottom of the keyboard to create a REST - to shut off a note. For example, press the letter "A" to play the lowest note on the PIANO display. You can get a very short note by typing A and then pressing the SPACE BAR immediately.

After a little musical note jumps across the keys as it plays, it's your turn. Experiment for a while to get

used to using the keyboard and the SPACE BAR.

If you just want to play tunes, you needn't use any of the suboptions for PLAY PIANO. However, if you want to do some of the activities mentioned earlier, such as retaining a tune in computer memory as you work it out, then you're ready to explore the suboptions.

ENTERING A TUNE INTO COMPUTER MEMORY

Let's enter a song into computer memory as we create it. To do so, we'll refer to the chart in the appendix for the BIRTHDAY song as our source for data. (The BASIC utility program, FILEDUMP, described in a later section, created this chart.)

To enter a song into computer memory, we need to go to the first of two suboption menus. which display in the text window when we press the ESC key. The first suboption menu looks like this:

```
TYPE 1 TO ADD DATA TO MEMORY
TYPE 2 TO SAVE MEMORY
TYPE 3 TO PLAY MEMORY DATA
TYPE 4 TO FIX MEMORY DATA
```

Pressing the <ESC> key again displays the second suboption menu:

```
TYPE 1 TO PLAY KEYBOARD PIANO
TYPE 2 TO READ A DATA FILE
TYPE 3 TO CREATE A NEW DATA FILE
TYPE 4 TO END THIS PROGRAM
```

We can use either of two suboptions, ADD DATA TO MEMORY (in the Suboption 1 display) or CREATE A NEW DATA FILE (in the Suboption 2 display), to enter a song into computer memory. The two differ in that we would use the ADD DATA TO MEMORY suboption if we

wanted to append notes to song data currently stored in computer memory, whereas we would use the CREATE A NEW DATA FILE to erase any song data currently stored in computer memory and start over again from NOTE #1. Assuming you haven't yet entered anything into computer memory, we can in this case use either option. We'll type a 3 to CREATE A NEW DATA FILE.

ENTERING OUR SONG DATA

The text window asks you to:

TYPE DURATION (1-120) [RETURN]

Look at NOTE #1 on the BIRTHDAY chart in the appendix. You'll see four columns, the third labeled "DURATION". The value for NOTE #1 is 30, representing 30/60 (or 1/2) second. Duration always represents sixtieths of a second. (See the section on "Advanced Information" for a more detailed discussion about selecting duration.) Thus, by typing 30 in response to the prompt, we're telling the program to hold the first note for a half second. Notice you need to press the <RETURN> key after entering your data, because the program can't tell whether you'll be typing one or two digits.

Next, the prompt for entering a note displays in the text window, along with the keyboard display:

NOTE #1?

Type the note indicated under the column labeled KEY on the chart -D- in the same way as in PLAY PIANO mode. If all went well, that's one note down and 27 to go. The prompt for the duration of the next key then displays and we enter duration and key for each note.

FIXING MISTAKES

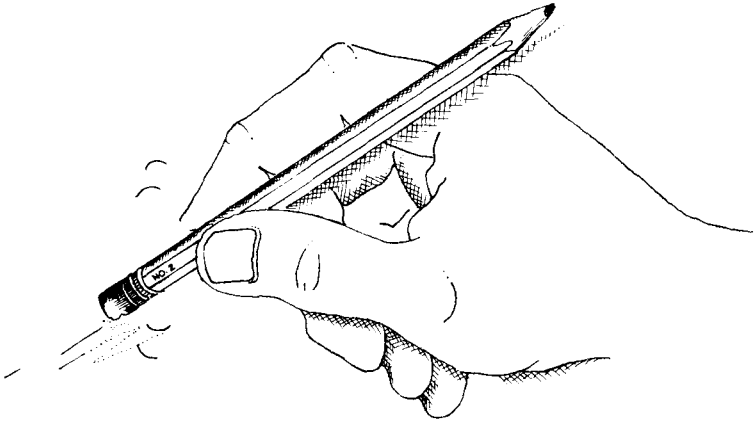
If you make a mistake, don't worry, because you can use the FIX MEMORY DATA suboption to repair the

damage. Use the <ESC> key to re-display the Suboption 1 display and type a 4 to select the FIX MEMORY DATA suboption. The prompt ("#" represents a number):

THERE ARE # NOTES IN MEMORY

TYPE NOTE NUMBER TO BE FIXED

asks you which note number you want to fix. Type the appropriate number and press the <RETURN> key.



A SUGGESTION

Before you enter too many notes, it's helpful to replay the notes often as you add data. If you do make a mistake, finding and correcting the data is much easier while it's still fresh in your memory. Referring to the BIRTHDAY chart, enter the first seven notes and then press the <ESC> key to return to the Suboption 1 display. (Note: On the BIRTHDAY chart, NOTE #7 is SP, meaning SPACE BAR. Between the sixth and eighth notes is a full second rest. So, type the duration of 60 and press the SPACE BAR for NOTE #7.)

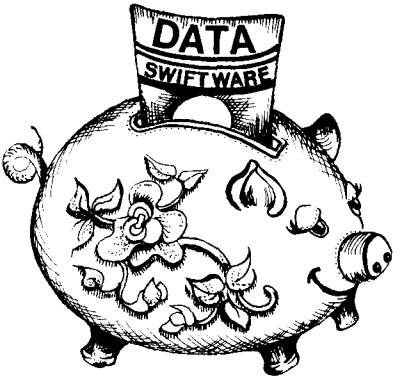
PLAYING BACK NOTES STORED IN COMPUTER MEMORY

Let's play what we have so far. Type 3 on the Suboption 1 display for PLAY MEMORY DATA. The

display tells us that we have 7 notes in memory and it asks us which ones we'd like to start and end with. We'll replay all seven notes by typing the starting number of 1 and the ending number of 7. Again, we need to press the <RETURN> key after typing each number, because these numbers can be between one and three digits (up to note #400, the limit for notes saved in computer memory). The program now plays the seven notes. If they sound like "Happy Birthday" to you, continue adding data using the ADD DATA TO MEMORY suboption (press the ESC key to obtain the suboption displays.) If some notes sound strange, locate the wrong one(s) and use the FIX MEMORY DATA suboption to correct them. (You may abort a song in progress by pressing any key.)

SAVING DATA ON CASSETTE OR DISKETTE

Use the SAVE MEMORY DATA suboption to store your data currently in computer memory out on cassette or diskette. After selecting this suboption, type a "C" for "cassette" or a "D" for "diskette" (or press the <ESC> key if you change your mind.)



If you're using a cassette, use a blank tape and note the position number on the program recorder. Then press the PLAY and RECORD buttons on the recorder

before you press the <RETURN> key on the keyboard to begin saving your data.

If you're using a diskette, you need to specify the file name in response to the prompt and then press the <RETURN> key. Be sure the notch is uncovered on the diskette you're using. Also make sure the diskette has enough free sectors to hold your data file. Disk drive 1 is assumed, so you need specify only the file name and perhaps an extension such as .DAT for Data File or .PPM for Player Piano Music file. Thus, you could save the BIRTHDAY song file on diskette by typing D in response to the prompt for cassette or diskette storage and then typing the file name BIRTHDAY and pressing <RETURN> in response to the file name prompt. You may use any valid file name, but remember that you'll erase a file already on the diskette if you specify the same file name for the file you want to save. Important: DO NOT try to save your files on the original PLAYER PIANO diskette, which is write-protected for safety.

READING SONG FILES INTO COMPUTER MEMORY

Use the READ A DATA FILE suboption to read a song file into computer memory from cassette or diskette. Loading a file into memory erases any data currently in memory. If you want to keep such a file, save it on cassette or diskette before loading in another file.

CHAPTER 14: ADVANCED INFORMATION

SELECTING DURATION

Look again in the appendix at the BIRTHDAY chart produced by the FILEDUMP program. Notice the duration of each note is either 30 or 60. Notes with a duration of 60 are held for a full second; those with a duration of 30 are held for half a second. A note held for more than a full second is rare. Most songs you create will probably use a faster tempo.

Tempo is based on what is called a "whole" note. In the BIRTHDAY song, the whole notes are those with a duration of 60. Those with a duration of 30 are "half" notes. Most songs use whole notes, half notes, quarter notes, and eighth notes. If the BIRTHDAY song required quarter notes, the duration would be 15 (1/4 of 60). Now, the problem is to divide 15 in half for the eighth notes. To do so, we drop any fractions. Thus, to produce an eighth note, we enter a duration of 7.

You'll find that this program can replay notes as short as a duration of around 6, but not less than that. The program takes about a tenth of a second to interpret the data into a sound, position the little note on the proper key, and display the note number in the text window. If you specify a duration of 6 for one note, then specify a duration of 1 for the following note--you'll see and hear the same duration.

Before entering a song of your own, think about its tempo. If it's slow, like the BIRTHDAY song, use 60 as your whole note and calculate the shorter notes based on fractions of the whole. If the tempo is faster, try 40 as the whole note duration, 20 for the half note, and so on.

FILE LAYOUT AND THE FILEDUMP PROGRAM

The FILEDUMP program comes in handy when you write longer, more complex songs. After making sure no program is already in computer memory (verify by typing the direct mode BASIC command NEW), Load and RUN the FILEDUMP program. Owners of the PLAYER PIANO Disk version should insert their Master Diskette and type: RUN "D:FILEDUMP" and press <RETURN>. Owners of the Cassette version should cue their program recorder to the beginning of the FILEDUMP program, press the PLAY key, type CLOAD and press the <RETURN> key twice. FILEDUMP will now load. Don't forget to type RUN and press <RETURN> after the program loads.

You don't need a printer. FILEDUMP lets you either print or display files on your TV screen. The program also accepts song files from both cassette and diskette.

PLAYER PIANO song files are data files of numbers. Each note in the file has two numbers--KEYCODE and DURATION. Note that the first two numbers in the file aren't for NOTE #1. The first number specifies the total number of notes in the file, and the second number is a dummy zero. After reading this first set of numbers, PLAYER PIANO and FILEDUMP know how many more sets of numbers to read.

With one exception, the keycode is the ATASCII value of the key pressed, minus 42. Don't confuse this value with the computer's internal keycodes. The keycodes used by PLAYER PIANO and FILEDUMP are generated specifically for these programs. The exception to this rule is 2 for the SPACE BAR keycode (which saves a little memory in the PLAYER PIANO program by keeping smaller the array of keycodes and their corresponding sounds and screen positions.)

If you do some programming, you may wish to write a

small program to increase or decrease the duration of each note by a given percentage to permit the changing of tempo in a song. Use the FILEDUMP program as a model.

PLAYER PIANO has twenty notes, plus the SPACE BAR (keycode of 2). The keycodes for the notes, from lowest (far left A) to highest (far right E) are as follows:

A	A♯	B	C	C♯	D♯	D	D♯	E	F
23	45	41	26	40	28	42	29	30	43

G	G♯	A	A♯	B	C	C♯	D	D♯	E
32	31	33	37	34	17	3	1	19	0

```
0 REM SOUNDEMO (c) 1982 by Jerry White[6/3/82]
100 P=100:D=10:V=2:GRAPHICS 17
120 ? #6:? #6;"   THE sound COMMAND"
130 ? #6:? #6;"      CT=CTRL KEY"
140 ? #6:? #6;"    p=increase pitch"
160 ? #6:? #6;"CT+p=decrease pitch"
180 ? #6:? #6;"    d=increase dist."
200 ? #6:? #6;"CT+d=decrease dist."
220 ? #6:? #6;"    v=increase vol."
240 ? #6:? #6;"CT+v=decrease vol."
260 ? #6:? #6;"    space=no sound"
280 ? #6:? #6;"      e=end of job"
320 POSITION 2,22:? #6;"sound 0,";P;",";D;",";V;"    ":SOUND
0,P,D,V:GOTO 400
340 POSITION 2,22:? #6;"sound 0,0,0,0       ":SOUND 0,0,0,0
400 POKE 764,255
420 IF PEEK(764)=10 AND P<255 THEN P=P+1:GOTO 320
440 IF PEEK(764)=138 AND P>0 THEN P=P-1:GOTO 320
460 IF PEEK(764)=16 AND V<15 THEN V=V+1:GOTO 320
480 IF PEEK(764)=144 AND V>0 THEN V=V-1:GOTO 320
500 IF PEEK(764)=58 AND D<14 THEN D=D+2:GOTO 320
520 IF PEEK(764)=186 AND D>0 THEN D=D-2:GOTO 320
540 IF PEEK(764)=33 THEN 340
560 IF PEEK(764)=42 OR PEEK(764)=28 THEN 30000
580 GOTO 420
30000 SOUND 0,0,0,0:GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 8
2,2:POKE 201,9
30005 ? "%%%  Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "  press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020

30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "%":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM EFFECTS (c) 1982 by Jerry White [6/3/82]
100 GRAPHICS 18:POSITION 7,4:? #6;"SOUND":POSITION 6,6:? #6;
"EFFECTS":GOTO 120
110 POKE 53768,1:POKE 53769,1:RETURN
120 FOR X=10 TO 100 STEP 2:SOUND 0,X,10,X/10:GOSUB 110
130 SOUND 1,X+2,10,X/10:GOSUB 110
140 SOUND 2,X+4,10,X/10:GOSUB 110
150 SOUND 3,X+6,10,X/10:GOSUB 110
160 SETCOLOR 0,X,10:SETCOLOR 4,X,2:SETCOLOR 2,X+4,10:NEXT X
170 FOR X=100 TO 0 STEP -2
180 SOUND 0,X,10,X/10:GOSUB 110
190 SOUND 1,X+2,10,X/10:GOSUB 110
200 SOUND 2,X+4,10,X/10:GOSUB 110
210 SOUND 3,X+6,10,X/10:GOSUB 110
220 SETCOLOR 0,X,10:SETCOLOR 4,X,2:SETCOLOR 2,X+4,10:NEXT X:
GOTO 260
230 FOR V=10 TO 0 STEP -5:SOUND 0,4,13,V:SOUND 1,2,8,V:NEXT
V:RETURN
240 FOR V=14 TO 0 STEP -0.5:SOUND 0,13,4,V:NEXT V:RETURN
250 FOR OFF=0 TO 3:SOUND OFF,0,0,0:NEXT OFF:RETURN
254 REM
255 REM *** AUTHOR INTRODUCTION ***
256 REM
260 GRAPHICS 18:SETCOLOR 1,4,8:POSITION 8,3:? #6;"b ";:GOSUB
 240:? #6;"y":GOSUB 240:SETCOLOR 2,7,8
270 POSITION 7,5:? #6;"J";:GOSUB 240:? #6;"E";:GOSUB 240:? #
6;"R";:GOSUB 240:? #6;"R";:GOSUB 240:? #6;"Y":GOSUB 240
280 POSITION 7,7:? #6;"W";:GOSUB 240:? #6;"H";:GOSUB 240:? #
6;"I";:GOSUB 240:? #6;"T";:GOSUB 240:? #6;"E":GOSUB 240
285 DIM L$(40)
290 FOR ME=15 TO 0 STEP -0.5:FOR JW=6 TO 0 STEP -1:SOUND 0,J
W,2,ME:NEXT JW:SETCOLOR 4,4,ME:NEXT ME
300 FOR ME=1 TO 10:FOR V=14 TO 0 STEP -2:SOUND 0,2,2,V:SOUND
 1,0,4,V:NEXT V:NEXT ME:? CHR$(253)
310 DIM B$(30):B$="                              "
320 GOTO 350
324 REM
325 REM *** B O M B ***
326 REM
330 FOR PITCH=0 TO 240 STEP 5:SOUND 0,PITCH,0,14-PITCH/20:SO
UND 1,PITCH,8,14-PITCH/20:NEXT PITCH:RETURN
340 GRAPHICS 0:SETCOLOR 1,0,8:SETCOLOR 2,0,0:SETCOLOR 4,0,0:
POKE 752,1:RETURN
350 GOSUB 340:POSITION 15,21:? "TARGET"
360 FOR PITCH=20 TO 200 STEP 2
370 POSITION 16,PITCH/10:? B$(1,4):POSITION 16,PITCH/10-1:?
B$(1,4):POSITION 16,PITCH/10:? "BOMB"
380 SOUND 0,PITCH,10,2:NEXT PITCH
390 GRAPHICS 0:POKE 752,1:SETCOLOR 1,8,8:POSITION 15,21:? "T
    T":POKE 755,4
400 POSITION 16,22:? "A  E":POSITION 17,23:? "RG";
410 FOR PITCH=20 TO 200 STEP 2:SOUND 0,PITCH,0,10-(PITCH/20)


420 SETCOLOR 2,4,10-(PITCH/20):NEXT PITCH
430 POKE 755,2:FOR WAIT=1 TO 200:NEXT WAIT
434 REM
435 REM *** LIGHTNING ***
436 REM
440 GRAPHICS 21:SETCOLOR 0,1,14:COLOR 1:PLOT 5,5:DRAWTO 75,3
0:GOSUB 330
450 GRAPHICS 21:SETCOLOR 0,2,14:COLOR 1:PLOT 75,5:DRAWTO 5,3
0:GOSUB 330
460 GRAPHICS 23:SETCOLOR 0,3,14:COLOR 1:PLOT 5,5:DRAWTO 150,
60:GOSUB 330
```

```
470 GRAPHICS 23:SETCOLOR 0,4,14:COLOR 1:PLOT 150,5:DRAWTO 5,
60:GOSUB 330
480 GRAPHICS 0:SETCOLOR 2,6,0:POKE 752,1
484 REM
485 REM *** HELICOPTER ***
486 REM
490 DIM H1$(12),H2$(10),H3$(10),H4$(10),H5$(10),H6$(10),H7$(
10)
500 H1$="     - ┬- -"
510 H2$=" ♣    ┌┴┐  "
520 H3$="  ╲__╱▀╲ "
530 H4$="   ▬▬▬┬╱ "
540 H5$="      ●  "
550 H6$=" x    ┌┴┐ "
560 H7$="    - -+ - "
570 SOUND 2,240,12,4
580 X=2:FOR Y=10 TO 0 STEP -1:POSITION X,10+Y:? "           "
:GOSUB 230
590 POSITION X,5+Y:? H1$:POSITION X,6+Y:? H2$:POSITION X,7+Y
:? H3$:POSITION X,8+Y:? H4$:GOSUB 230
600 POSITION X,9+Y:? H5$:GOSUB 230:POSITION X,5+Y:? H7$:POSI
TION X,6+Y:? H6$:GOSUB 230
610 POSITION X,5+Y:? H1$:POSITION X,6+Y:? H2$:GOSUB 230
620 NEXT Y
630 FOR X=2 TO 28:GOSUB 230
640 POSITION X,5:? H1$:POSITION X,6:? H2$:POSITION X,7:? H3$
:POSITION X,8:? H4$:GOSUB 230
650 POSITION X,9:? H5$:GOSUB 230:POSITION X,5:? H7$:POSITION
 X,6:? H6$:GOSUB 230
660 POSITION X,5:? H1$:POSITION X,6:? H2$:GOSUB 230
670 NEXT X
889 REM
890 REM *** ALERT SIREN ***
891 REM
900 FOR ME=1 TO 6:FOR J=1 TO 20:SOUND 0,80+J,12,8:NEXT J
905 GRAPHICS 18:SETCOLOR 0,ME,8
906 POSITION 7,4:? #6;"ALERT"
910 SOUND 0,80,10,12:SOUND 1,100,10,12:SOUND 2,13,4,12
920 FOR W=1 TO 100:NEXT W:GOSUB 250:NEXT ME
940 FOR V=12 TO 0 STEP -0.1:SOUND 0,(20-V)*10,10,V:SOUND 1,(
20-V)*10+20,10,V:SOUND 2,13,4,V:NEXT V
1000 REM
1001 REM *** BOUNCING BALL ***
1002 REM
1005 ? "↰":SETCOLOR 2,4,2:POKE 752,1
1006 FOR ME=1 TO 3
1010 FOR Y=22 TO 10 STEP -0.5
1020 POSITION 18,Y+1:? " ":POSITION 18,Y:? "●"
1030 SOUND 0,Y*10,14,Y/2-5:NEXT Y
1035 FOR Y=1 TO 9:NEXT Y
1040 FOR Y=10 TO 22 STEP 0.5:POSITION 18,Y-1:? " ":POSITION
18,Y:? "●":NEXT Y:NEXT ME
1049 REM
1050 REM *** P/M FLYING SAUCERS ***
1051 REM
1100 GRAPHICS 0:POKE 752,1:? " "
1110 SETCOLOR 2,0,0:X=20:Y=80
1120 POSITION 5,3:? ".":POSITION 27,6:? "."
1130 FOR ME=1 TO 10:RX=RND(0)*37+1:RY=RND(0)*21+1:POSITION R
X,RY:? ".":NEXT ME
1140 A=PEEK(106)-8:POKE 54279,A:PMBASE=256*A
1150 POKE 559,46:POKE 53277,3:POKE 53248,X:POKE 53256,3
1170 FOR I=PMBASE+512 TO PMBASE+640:POKE I,0:NEXT I:POKE 704
,100
1180 FOR I=PMBASE+512+Y TO PMBASE+516+Y:READ A:POKE I,A:NEXT
 I
```

```
1190 DATA 60,126,256,126,60
1200 SOUND 0,13,4,2
1210 FOR X=20 TO 80 STEP 2:POKE 53248,X
1220 FOR I=0 TO 6:POKE PMBASE+511+Y+I,PEEK(PMBASE+512+Y+I):N
EXT I:Y=Y-1:NEXT X
1230 SOUND 0,13,4,10:FOR X=82 TO 220:POKE 53248,X:NEXT X:POK
E 704,20
1240 FOR V=10 TO 0 STEP -0.5:SOUND 0,13,4,V:NEXT V
1250 FOR ME=0 TO 10:SOUND 1,0,2,ME+2:FOR I=0 TO 6:POKE PMBAS
E+511+Y+I,PEEK(PMBASE+512+Y+I):NEXT I
1260 SOUND 1,0,0,0:Y=Y-1:NEXT ME:POKE 53248,220
1270 SOUND 0,25,4,2:POKE 53256,1:FOR X=220 TO 0 STEP -1:POKE
 53248,X:NEXT X:SOUND 0,0,0,0:POKE 53248,120
1280 FOR ME=15 TO 0 STEP -1:POKE 53277,3:POKE 53256,3:FOR P=
0 TO 10:SOUND 0,P,2,ME:SOUND 1,P,2,ME
1290 NEXT P:POKE 704,ME*16+6:POKE 53277,1:RX=RND(0)*37+1:POS
ITION RX,22:? ".":? :NEXT ME
1700 GOSUB 250:? "K":POKE 752,1:?
1749 REM
1750 REM *** TELEPHONE RINGING ***
1751 REM
1800 FOR R=1 TO 2:FOR ME=1 TO 30:SOUND 0,25,10,1:SOUND 1,0,2
,1:FOR J=1 TO 2:SOUND 0,25,10,8:SOUND 1,0,2,8
1805 NEXT J:SOUND 0,0,0,0:SOUND 1,0,0,0:SETCOLOR 2,ME,4:NEXT
 ME
1810 FOR V=7 TO 0 STEP -0.1:SOUND 0,25,10,V:SOUND 1,0,2,V:NE
XT V:GOSUB 8200:NEXT R
1820 L$="PARDON ME WHILE I ANSWER THE PHONE":? :? :FOR J=1 T
O LEN(L$):? L$(J,J);:GOSUB 230:NEXT J
1830 FOR ME=1 TO 10:SOUND 0,25,10,2:SOUND 1,0,2,1:FOR J=1 TO
 2:SOUND 0,25,10,8:SOUND 1,0,2,8
1840 NEXT J:NEXT ME:GOSUB 250:GOSUB 8200
1842 L$="HELLO":POSITION 15,8:FOR J=1 TO LEN(L$):? L$(J,J);:
GOSUB 230:NEXT J
1850 GOSUB 8000:POSITION 12,12:L$="IT'S FOR YOU.":FOR J=1 TO
 LEN(L$):? L$(J,J);:GOSUB 230:NEXT J:GOSUB 8200
2100 FOR X=1 TO 24:? :SOUND 0,X,2,2:SOUND 1,X,4,2:SOUND 2,X,
6,2:SOUND 3,X,12,6
2110 FOR Y=1 TO 5:NEXT Y:NEXT X:GOSUB 250
2200 FOR X=14 TO 0 STEP -0.5:SOUND 0,91,12,X:NEXT X
2300 REM
2301 REM *** DISPLAY FOR NEXT LOOPS ***
2302 REM
2400 GRAPHICS 0:POKE 752,1:POKE 82,2:POKE 83,39:SETCOLOR 2,0
,0
2410 ? :? :? "            ┌────────────────────┐":? "          │ A
TARI BASIC SOUNDS │":? "            └────────────────────┘"
2420 ? CHR$(253):GOSUB 2480:LIST 2490,2500:GOSUB 2490
2430 GOSUB 2480:LIST 2530,2540:GOSUB 2530:GOSUB 2480:LIST 25
50,2560:GOSUB 2550:GOSUB 2480:LIST 2570,2580
2440 GOSUB 2570:GOSUB 2480:LIST 2610,2620:GOSUB 2610:GOSUB 2
480
2450 LIST 2630,2640:GOSUB 2630:GOSUB 2480:LIST 2650,2670:GOS
UB 2650:GOSUB 2480:LIST 2680:GOSUB 2680:GOSUB 2480
2460 LIST 2690:GOSUB 2690:GOSUB 2480:LIST 2700:GOSUB 2700:GO
SUB 2710:GOTO 6000
2480 FOR ME=0 TO 3:SOUND ME,0,0,0:NEXT ME:POSITION 2,6:? "↑↑
↑↑↑↑↑↑":RETURN
2490 FOR J1=0 TO 15:FOR J2=200 TO 206 STEP 2:SOUND 0,J2,10,J
1:NEXT J2
2500 FOR J2=208 TO 214 STEP 2:SOUND 1,J2,10,J1:NEXT J2
2510 FOR J2=216 TO 222 STEP 2:SOUND 2,J2,10,J1:NEXT J2
2520 FOR J2=224 TO 234 STEP 2:SOUND 3,J2,10,J1:NEXT J2:NEXT
J1:RETURN
2530 FOR J2=0 TO 4:FOR J1=14 TO 0 STEP -7:SOUND 0,13,4,J1:SO
UND 1,13,4,J1:SOUND 2,13,4,J1
```

```
2540 SOUND 0,100,0,J1:SOUND 1,100,0,J1:SOUND 2,100,0,J2:NEXT
 J1:NEXT J2:RETURN
2550 FOR J1=15 TO 0 STEP -1:FOR J2=10 TO 20:SOUND 0,J2,10,J1
:NEXT J2
2560 FOR J2=30 TO 20 STEP -1:SOUND 1,J2,10,J1:NEXT J2:NEXT J
1:RETURN
2570 FOR J1=14 TO 0 STEP -1:FOR J2=10 TO 20:SOUND 0,J2,12,J1
:SOUND 1,J2*2,12,J1:NEXT J2
2580 FOR J2=0 TO 5:SOUND 2,J2,2,J1:SOUND 3,J2+2,2,J1:NEXT J2
:NEXT J1:RETURN
2610 FOR J1=15 TO 0 STEP -0.5:FOR J2=3 TO 10:SOUND 0,J2,10,J
1:NEXT J2
2620 FOR J2=13 TO 25 STEP 12:SOUND 1,J2,10,J1:NEXT J2:NEXT J
1:RETURN
2630 FOR J1=10 TO 0 STEP -1:FOR J2=20 TO 0 STEP -2:SOUND 0,J
2,10,J1:NEXT J2
2640 FOR J2=40 TO 10 STEP -2:SOUND 1,J2,10,J1:NEXT J2:NEXT J
1:RETURN
2650 FOR J1=14 TO 0 STEP -1:FOR J2=10 TO 60 STEP 5:SOUND 0,J
2,10,J1:NEXT J2
2660 FOR J2=70 TO 140 STEP 10:SOUND 1,J2,10,J1:NEXT J2
2670 FOR J2=210 TO 10 STEP -50:SOUND 2,J2,10,J1:NEXT J2:NEXT
 J1:RETURN
2680 FOR J1=15 TO 0 STEP -0.5:FOR J2=0 TO 9:SOUND 0,J2,2,J1:
SOUND 1,J2*5,10,J1:NEXT J2:NEXT J1:RETURN
2690 FOR J1=0 TO 250 STEP 2:SOUND 0,J1,10,15:SOUND 1,255-J1,
12,15:NEXT J1:RETURN
2700 FOR J1=15 TO 0 STEP -0.05:SOUND 0,51,12,J1:SOUND 1,102,
12,J1:SOUND 2,51,12,J1:SOUND 3,102,12,J1:NEXT J1:RETURN
2705 REM
2706 REM *** THAT'S ALL FOLKS ***
2707 REM
2710 DATA 40,19,29,9,31,9,35,9,40,9,47,19,0,9,40,9,29,3,29,3
,31,9,35,9,33,9,31,19,0,9
2720 DATA 31,9,31,9,40,9,35,9,40,3,40,3,31,3,31,3,40,9,35,9
2730 DATA 40,3,37,3,35,3,31,3,29,3,23,3,26,3,29,3,35,3,40,3,
29,3,26,9,23,9
2740 GRAPHICS 18:? #6:? #6:? #6:? #6;"        that's":? #6:?
#6;"          ALL":? #6:? #6;"        FOLKS"
2750 TRAP 2760:READ N,T:GOSUB 2770:GOTO 2750
2760 RETURN
2770 POKE 540,T:SETCOLOR 4,N,0:SOUND 0,N,10,14
2780 IF PEEK(540)<>0 THEN 2780
2790 SOUND 0,0,0,0:RETURN
5899 REM
5900 REM *** END OF JOB ROUTINE ***
5901 REM
6000 ? "5":POKE 752,1:POSITION 4,4:L$="This concludes our SO
UND EFFECTS"
6100 FOR J=1 TO LEN(L$):? L$(J,J);:GOSUB 230:NEXT J
6200 POSITION 12,6:L$="demonstration.":FOR J=1 TO LEN(L$):?
L$(J,J);:GOSUB 230:NEXT J
6300 FOR ME=1 TO 10:POKE 755,3:GOSUB 230:POKE 755,2:SOUND 0,
ME,2,8:NEXT ME:SOUND 0,0,0,0
6600 SOUND 1,0,0,0:FOR J=200 TO 0 STEP -5:SOUND 0,J,10,8:SET
COLOR 2,J,4:SOUND 1,J,12,8:NEXT J
6700 GOSUB 250:GOTO 30000
8000 FOR WAIT=1 TO 300:NEXT WAIT:RETURN
8100 FOR WAIT=1 TO 500:NEXT WAIT:RETURN
8200 FOR WAIT=1 TO 100:NEXT WAIT:RETURN
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9
```

```
30005 ? "↖↓↓↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▸ press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020

30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "↖":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM FORNEXT (c) 1982 by Jerry White  [6/3/82]
210 GRAPHICS 0:POKE 559,0:DIM NAME$(12),FN$(14),X$(1),BL$(20
):BL$="                    ":SETCOLOR 2,0,0
212 DIM L1$(50),L2$(50),L3$(11),L4$(11),L5$(10):L3$="103 NEX
T L2":L4$="104 NEXT L1":L5$="105 RETURN"
220 ? :? "    ■■■ FORNEXT PROGRAM VARIABLES: ■■■":? :? "DIST=DISTO
RTION variable (0 thru 14)":? "even numbers only."
230 ? :? "PITCH=any number from 0 thru 255.":? "We will chan
ge the pitch in loop L2."
240 ? :? "L1=Outer loop 1 VOLUME.":? "Type any integer from
0 thru 15":? "At prompts FROM, TO, and STEP."
250 ? :? "L2=Inner loop 2 PITCH.":? "Type any number from 0
thru 255":? "At prompts FROM, TO, and STEP."
260 ? "↓↓↓↓↓ ■■■ PRESS ANY KEY TO BEGIN ■■■";:POKE 53279,0:POKE 559
,34:POKE 764,255:POKE 53279,8
262 IF PEEK(764)=255 AND PEEK(53279)=7 THEN 262
270 GRAPHICS 0:POKE 559,0:? :? ,"■■■ SOUND TEST ■■■"
280 POKE 752,1:C=RND(0)*16:REPEAT=0:POKE 764,255
290 SETCOLOR 1,C,0:SETCOLOR 2,C,8:SETCOLOR 4,C,0:POKE 752,0:
POKE 559,34
300 POSITION 9,3:? "TYPE DIST    ";:TRAP 320:INPUT D:TRAP 40
000
310 IF D=0 OR D=2 OR D=4 OR D=6 OR D=8 OR D=10 OR D=12 OR D=
14 THEN 360
320 POSITION 9,3:? BL$:GOTO 300
360 POSITION 9,5:? "TYPE L1 FROM ";:TRAP 380:INPUT F1:TRAP 4
0000
370 IF F1<16 AND F1>-0.01 THEN 390
380 GOSUB 1400:POSITION 9,5:? BL$:GOTO 360
390 POSITION 9,7:? "TYPE L1 TO   ";:TRAP 410:INPUT T1:TRAP 4
0000
400 IF T1<16 AND T1>-0.01 THEN 420
410 GOSUB 1400:POSITION 9,7:? BL$:GOTO 390
420 POSITION 9,9:? "TYPE L1 STEP ";:TRAP 440:INPUT S1:TRAP 4
0000
430 IF S1<16 THEN 450
440 GOSUB 1400:POSITION 9,9:? BL$:GOTO 420
450 POSITION 9,11:? "TYPE L2 FROM ";:TRAP 470:INPUT F2:TRAP
40000
460 F2=INT(F2):IF F2<256 AND F2>-1 THEN 480
470 GOSUB 1400:POSITION 9,11:? BL$:GOTO 450
480 POSITION 9,13:? "TYPE L2 TO   ";:TRAP 500:INPUT T2:TRAP
40000
490 IF T2<256 AND T2>-1 THEN 510
500 GOSUB 1400:POSITION 9,13:? BL$:GOTO 480
510 POSITION 9,15:? "TYPE L2 STEP ";:TRAP 530:INPUT S2:TRAP
40000
520 IF S2<256 THEN 540
530 GOSUB 1400:POSITION 9,15:? BL$:GOTO 510
540 GOSUB 720
550 SOUND 0,0,0,0:POKE 82,5:? CHR$(125):POKE 559,0:POKE 752,
1
560 ? :? "  ■■■ YOUR SOUND SUBROUTINE: ■■■":?
570 L1$="101 FOR L1=":L1$(LEN(L1$)+1)=STR$(F1):L1$(LEN(L1$)+
1)=" TO "
580 L1$(LEN(L1$)+1)=STR$(T1):L1$(LEN(L1$)+1)=" STEP ":L1$(LE
N(L1$)+1)=STR$(S1)
590 L2$="102 FOR L2=":L2$(LEN(L2$)+1)=STR$(F2):L2$(LEN(L2$)+
1)=" TO "
600 L2$(LEN(L2$)+1)=STR$(T2):L2$(LEN(L2$)+1)=" STEP ":L2$(LE
N(L2$)+1)=STR$(S2)
610 ? L1$:? L2$:? L3$:? L4$:? L5$:POKE 82,5:? :POKE 559,34
620 FOR TIME=1 TO 3:POSITION 5,18:? "■■■ TYPE 1 TO SAVE SUBROU
TINE ■■■":? "■■■ TYPE 2 TO PRINT SUBROUTINE ■■■"
```

```
630 ? "   TYPE 3 TO REPEAT SUBROUTINE ":? "   TYPE 4 TO ENTER
PARAMETERS ":? "   TYPE 5 TO EXIT                    ";
640 POKE 755,3:FOR DELAY=1 TO 10:NEXT DELAY
650 POKE 755,2:FOR DELAY=1 TO 10:NEXT DELAY:NEXT TIME:SOUND
0,0,0,0:POKE 764,255
660 POKE 82,2:POKE 559,34:CHOICE=PEEK(764):IF CHOICE=255 THE
N 660
665 IF CHOICE=26 THEN REPEAT=REPEAT+1:GOTO 1200
670 IF CHOICE=31 THEN 900
680 IF CHOICE=29 THEN 30000
690 IF CHOICE=30 THEN 800
700 IF CHOICE=24 THEN 270
710 FOR TIME=1 TO 10:POKE 53279,0:NEXT TIME:GOTO 640
720 FOR L1=F1 TO T1 STEP S1
730 FOR L2=F2 TO T2 STEP S2
740 TRAP 760:SOUND 0,L2,D,L1:TRAP 40000
750 NEXT L2:NEXT L1:REPEAT=0:POKE 764,255:RETURN
760 ? :? "      INVALID SOUND, TRY AGAIN.      ":SOUND 0,0,0,0
770 FOR DELAY=1 TO 250:NEXT DELAY:RUN
800 TRAP 1000:LPRINT "     SOUND SUBROUTINE":LPRINT :LPRINT L
1$:LPRINT L2$:LPRINT L3$:LPRINT L4$:LPRINT L5$
810 LPRINT :TRAP 40000:GOTO 550
900 ? "■":? :? "  YOUR SUBROUTINE WILL BE LISTED TO":? :? "D
ISK DRIVE 1.  INSERT DATA DISKETTE."
910 ? :? "TYPE FILE NAME THEN PRESS RETURN":? :POKE 764,255:
INPUT NAME$
920 LN=LEN(NAME$):IF LN=0 THEN 910
930 FN$="D:":FN$(LEN(FN$)+1)=NAME$:TRAP 1100:CLOSE #1:OPEN #
1,8,0,FN$
940 ? :? L1$:? #1;L1$:? L2$:? #1;L2$:? L3$:? #1;L3$:? L4$:?
#1;L4$:? L5$:? #1;L5$:CLOSE #1:GOTO 550
1000 ? "■■":? :? "YOUR PRINTER ISN'T READY":? :? "TYPE 1 TO
PRINT OR 2 FOR OPTIONS":POKE 764,255
1010 IF PEEK(764)=255 THEN 1010
1020 IF PEEK(764)=31 THEN 800
1030 IF PEEK(764)=30 THEN 550
1040 ? :? "■PLEASE TYPE 1 OR 2":GOTO 1010
1100 ? "■":? :? "I WAS UNABLE TO LIST YOUR SUBROUTINE":? :?
"ONTO DISK DRIVE 1."
1110 FOR DELAY=1 TO 250:NEXT DELAY:CLOSE #1:GOTO 550
1200 ? "■↓↓↓↓↓":? "TYPE 1 TO REPEAT ORIGINAL SOUND":? :? "TY
PE 2 TO DISABLE SCREEN INTERRUPTS"
1210 ? :? "TYPE 3 TO REPEAT SOUND WHILE THE":? "CURRENT SOUN
D COMMAND IS DISPLAYED":POKE 764,255
1215 IF PEEK(764)=255 THEN 1215
1220 IF PEEK(764)=31 THEN REPEAT=REPEAT+1:GOTO 540
1230 IF PEEK(764)=30 THEN REPEAT=REPEAT+1:POKE 559,0:GOTO 54
0
1240 IF PEEK(764)<>26 THEN POKE 764,255:GOSUB 1400:GOTO 1215


1250 TRAP 1400:GRAPHICS 18:FOR L1=F1 TO T1 STEP S1:FOR L2=F2
 TO T2 STEP S2:SOUND 0,L2,D,L1
1260 POSITION 2,4:? #6;"SOUND 0,";INT(L2);",";INT(D);",";INT
(L1);"    ":NEXT L2:NEXT L1
1270 SOUND 0,0,0,0:GRAPHICS 0:SETCOLOR 2,0,0:POKE 752,1:POKE
 764,255:? :GOTO 550
1400 FOR X=1 TO 12:POKE 53279,0:NEXT X:RETURN
1500 ? :? "■  I COULDN'T FIND THE MENU PROGRAM.":? "▸ INSERT
 PROGRAM DISKETTE":? "▸ PRESS START WHEN READY";
1510 IF PEEK(53279)=6 THEN 680
1520 GOTO 1510
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9
```

```
30005 ? "↖↓↓↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▸ press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020

30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
 PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "↖":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM POKSOUND (c) 1982 by Jerry White [7/12/82]
100 FOR ME=1536 TO 1574:READ IT:POKE ME,IT:NEXT ME:GOTO 360
120 POSITION 0,2:? #6;">";Y=2
130 FOR ME=0 TO 8:POSITION 1,ME+2:? #6;53760+ME;"=";N(ME);"
  00000000":NEXT ME:GOTO 140
135 FOR ME=0 TO 8:POSITION 1,ME+2:? #6;53760+ME;:NEXT ME
140 POSITION 0,11:? #6;"  select location  "
150 S=STICK(0):IF S=15 AND STRIG(0) AND PEEK(764)=255 THEN 1
50
155 IF S=14 AND Y>2 THEN POSITION 0,Y:? #6;" ":Y=Y-1:POSITIO
N 0,Y:? #6;CHR$(30):GOSUB 340
160 IF PEEK(764)=28 THEN 390
170 IF PEEK(764)=33 THEN RUN
180 IF S=13 AND Y<10 THEN POSITION 0,Y:? #6;" ":Y=Y+1:POSITI
ON 0,Y:? #6;CHR$(30):GOSUB 340
190 IF STRIG(0) THEN 150
200 GC=Y-2:GOSUB 350
210 IF  NOT STRIG(0) THEN 210
220 P=53760+GC:L$=STR$(P):FOR ME=1 TO 5:IT=ASC(L$(ME,ME)):IT
=IT+128:L$(ME,ME)=CHR$(IT):NEXT ME
230 POSITION 1,Y:? #6;L$
240 POSITION 0,11:? #6;"    change poke  "
250 S=STICK(0):IF S=15 AND STRIG(0) AND PEEK(764)=255 THEN 2
50
255 IF S=14 AND N(GC)<255 THEN N(GC)=N(GC)+1
260 IF PEEK(764)=28 THEN 390
270 IF PEEK(764)=33 THEN RUN
280 IF S=13 AND N(GC)>0 THEN N(GC)=N(GC)-1
290 POKE P,N(GC):POSITION 7,Y:? #6;"   ":POSITION 7,Y:? #6;N
(GC)
295 BIN=USR(1536,N(GC),ADR(BIN$)):POSITION 11,Y:? #6;BIN$;
300 IF STRIG(0) THEN 250
310 GOSUB 350
320 IF  NOT STRIG(0) THEN 320
330 POKE 77,0:GOTO 135
340 FOR ME=1 TO 10:POKE 53279,0:POKE 53279,8:NEXT ME:RETURN


350 POSITION 0,11:? #6;"  release trigger  ":RETURN
360 GRAPHICS 18:DIM N(8),L$(5),BIN$(8):POKE 712,144:POKE 710
,156:POKE 709,14:POKE 708,28:BIN$="        "
370 POKE 711,190:? #6;"    POKE  SOUND ":? #6;" decimal   5
digit":POKE 764,255:SOUND 0,0,0,0
380 FOR ME=0 TO 8:N(ME)=0:NEXT ME:CLOSE #1:OPEN #1,4,0,"K:":
GOTO 120
390 FOR ME=53760 TO 53768:POKE ME,0:NEXT ME
400 DATA 104,104,104,133,203,104,133,205,104,133,204,160,0,1
62,8,165,203,10,133,203,32,27,6
410 DATA 202,208,245,96,176,6,169,48,145,204,200,96,169,49,2
08,248
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9


30005 ? "K↑↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▸ press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020
```

```
30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
 PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "K";? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM KEYOFC (c) 1982 by Jerry White  [6/3/82]
120 GOTO 480
140 FOR WAIT=1 TO 200:NEXT WAIT:RETURN
160 COLOR 1:PLOT X+1,Y:DRAWTO X+2,Y:PLOT X,Y+1:DRAWTO X+3,Y+
1
180 PLOT X-1,Y+2:DRAWTO X+4,Y+2:PLOT X-1,Y+3:DRAWTO X+4,Y+3
200 PLOT X,Y+4:DRAWTO X+3,Y+4:PLOT X+1,Y+5:DRAWTO X+2,Y+5
220 PLOT X+4,Y-10:DRAWTO X+4,Y+2:RETURN
240 FOR HOLD=1 TO 500:NEXT HOLD:FOR OFF=0 TO 3:SOUND OFF,0,0
,0:NEXT OFF:RETURN
260 ? "      PRESS ANY KEY TO CONTINUE.";:POKE 764,255
280 IF PEEK(764)=255 AND PEEK(53279)=7 THEN 280
300 GOSUB 380:RETURN
320 ? "      PRESS ANY KEY TO CONTINUE.";:POKE 764,255
340 IF PEEK(764)=255 AND PEEK(53279)=7 THEN 340
360 RETURN
380 GRAPHICS 7:SETCOLOR 0,7,8:SETCOLOR 1,12,8:SETCOLOR 2,2,1
2:SETCOLOR 3,4,6:SETCOLOR 4,2,12
400 COLOR 2:PLOT 20,20:DRAWTO 140,20:PLOT 20,26:DRAWTO 140,2
6:PLOT 20,32:DRAWTO 140,32
420 POKE 752,1:PLOT 20,38:DRAWTO 140,38:PLOT 20,44:DRAWTO 14
0,44:RETURN
440 COLOR 1:PLOT X+3,Y+6:DRAWTO X+5,Y-1:PLOT X+7,Y+6:DRAWTO
X+9,Y-1
460 PLOT X+2,Y+1:DRAWTO X+10,Y+1:PLOT X+2,Y+4:DRAWTO X+10,Y+
4:RETURN
480 GOSUB 380:? ,"  THE C SCALE":? :? ,"C D E F G A B C"
500 X=40:Y=47:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,12
1,10,VOL:NEXT VOL
520 COLOR 2:PLOT 33,50:DRAWTO 48,50:GOSUB 140:COLOR 1
540 X=50:Y=44:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,10
8,10,VOL:NEXT VOL:GOSUB 140
560 X=60:Y=41:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,96
,10,VOL:NEXT VOL:GOSUB 140
580 X=70:Y=38:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,91
,10,VOL:NEXT VOL:GOSUB 140
600 X=80:Y=35:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,81
,10,VOL:NEXT VOL:GOSUB 140
620 X=90:Y=32:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,72
,10,VOL:NEXT VOL:GOSUB 140
640 X=100:Y=29:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,6
4,10,VOL:NEXT VOL:GOSUB 140
660 X=110:Y=26:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,6
0,10,VOL:NEXT VOL:GOSUB 140
680 GOSUB 260
700 ? ,"THE C MAJOR CHORD":? ,"    C + E + G"
720 X=60:Y=47:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,12
1,10,VOL:NEXT VOL
740 PLOT 55,50:DRAWTO 68,50
760 X=80:Y=41:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,96
,10,VOL:NEXT VOL
780 X=100:Y=35:GOSUB 160:FOR VOL=10 TO 0 STEP -0.2:SOUND 0,8
1,10,VOL:NEXT VOL:GOSUB 320
800 ? :? :? ,"SOUND 0,121,10,8":? ,"SOUND 1,96,10,8":? ,"SOU
ND 2,81,10,8"
820 SOUND 0,121,10,8:SOUND 1,96,10,8:SOUND 2,81,10,8:GOSUB 2
40
840 GOSUB 320
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9
```

```
30005 ? "↖↓↓↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▸ press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020

30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
 PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "↖":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM ATOC (c) 1982 by Jerry White [6/3/82]
3 Q0=0:Q1=1:Q2=2:Q3=3:Q4=4:Q5=5:Q6=0.2:Q8=8:Q10=10:Q12=12:Q1
3=13:Q16=16:Q17=17:Q19=19:Q20=20:Q22=22:Q23=23:Q26=26:Q29=29
:Q32=32
4 Q35=35:Q38=38:Q40=40:Q41=41:Q42=42:Q44=44:Q45=45:Q47=47:Q5
0=50:Q53=53:Q55=55:Q57=57:Q60=60:Q64=64:Q68=68:Q70=70:Q72=72
:Q75=75
5 Q76=76:Q80=80:Q81=81:Q85=85:Q91=91:Q95=95:Q96=96:Q102=102:
Q108=108:Q114=114:Q121=121:Q122=122:Q130=130:Q138=138:Q154=1
54
6 Q178=178:Q255=255:Q256=256:Q300=300:Q764=764:Q100=100:Q7=6


20 DIM ZA$(Q100):GOTO 150
40 ZL=PEEK(560)+PEEK(561)*Q256
42 ZM=PEEK(ZL+Q4)+PEEK(ZL+Q5)*Q256:FOR ZW=Q1 TO LEN(ZA$):ZT=
57344+((ASC(ZA$(ZW,ZW))-Q32)*Q8):ZC=ZM+ZY*Q40+ZX+ZW-Q1
44 FOR ZR=Q0 TO 7:POKE ZC+ZR*Q40,PEEK(ZT+ZR):NEXT ZR:ZY=ZY+Z
Z:NEXT ZW:RETURN
50 COLOR Q1:PLOT X+Q1,Y:DRAWTO X+Q2,Y:PLOT X,Y+Q1:DRAWTO X+Q
3,Y+Q1
52 PLOT X-Q1,Y+Q2:DRAWTO X+Q4,Y+Q2:PLOT X-Q1,Y+Q3:DRAWTO X+Q
4,Y+Q3
53 PLOT X,Y+Q4:DRAWTO X+Q3,Y+Q4:PLOT X+Q1,Y+Q5:DRAWTO X+Q2,Y
+Q5
55 PLOT X+Q4,Y-Q10:DRAWTO X+Q4,Y+Q2:RETURN
60 FOR HOLD=Q1 TO 500:NEXT HOLD:FOR OFF=Q0 TO Q3:SOUND OFF,Q
0,Q0,Q0:NEXT OFF:RETURN
70 ? "      PRESS ANY KEY TO CONTINUE.";:POKE Q764,Q255
71 IF PEEK(Q764)=Q255 AND PEEK(53279)=7 THEN 71
72 RETURN
80 GRAPHICS Q8:SETCOLOR Q0,Q2,Q0:SETCOLOR Q1,Q2,Q0:SETCOLOR
Q2,Q2,Q8:SETCOLOR Q4,Q2,Q8:POKE 752,Q1:COLOR Q1
81 PLOT Q20,Q20:DRAWTO Q300,Q20:PLOT Q20,Q26:DRAWTO Q300,Q26
:PLOT Q20,Q32:DRAWTO Q300,Q32:PLOT Q20,Q38:DRAWTO Q300,Q38
85 PLOT Q20,Q44:DRAWTO Q300,Q44:ZX=Q13:ZY=125:ZA$="BASIC CHO
RDS":GOSUB Q40:RETURN
100 FOR VOL=Q10 TO Q0 STEP -Q6:SOUND Q0,P,Q10,VOL:NEXT VOL:R
ETURN
150 GRAPHICS Q17:SETCOLOR Q2,Q8,Q10:? #Q7:? #Q7:? #Q7;"     5
ELECT CHORD":? #Q7;"      BY NUMBER":? #Q7
160 ? #Q7:? #Q7;"      1= C MAJOR":? #Q7:? #Q7;"      2= DB M
AJOR"
170 ? #Q7:? #Q7;"      3= D MAJOR":? #Q7:? #Q7;"      4= E M
AJOR"
180 ? #Q7:? #Q7;"      5= EB MAJOR"
200 ? #Q7:? #Q7;"      6= Q U I T":P=Q10:GOSUB Q100
210 POKE Q764,Q255:CLOSE #Q1:OPEN #Q1,Q4,Q0,"K:":GET #Q1,GC:
CLOSE #Q1:GC=GC-48
215 IF GC=Q7 THEN 30000
220 IF GC<Q1 OR GC>Q5 THEN 210
230 GOSUB Q80:ON GC GOSUB 3100,3400,3700,300,700
240 GOTO 150
300 ZA$="THE C MAJOR CHORD":ZX=Q12:ZY=Q75:GOSUB Q40:? ,"500
SOUND 0,122,10,2"
440 ZA$="C":ZX=Q16:ZY=Q55:GOSUB Q40:P=Q121:GOSUB Q100:X=Q130
:Y=Q47:GOSUB Q50
460 PLOT Q122,Q50:DRAWTO Q138,Q50
500 ? ,"510 SOUND 1,96,10,2":ZA$="E":ZX=Q19:ZY=Q55:GOSUB Q40
:P=Q96:GOSUB Q100
530 X=Q154:Y=Q41:GOSUB Q50:? ,"520 SOUND 2,81,10,2":ZA$="G":
ZX=Q22:ZY=Q55:GOSUB Q40
620 P=Q81:GOSUB Q100:X=Q178:Y=Q35:GOSUB Q50
650 ZA$="C+E+G=C MAJOR":ZX=Q13:ZY=Q95:GOSUB Q40
```

```
660 SOUND 0,0121,010,02:SOUND 01,096,010,02:SOUND 02,081,01
0,02:GOSUB 060:GOSUB 070:RETURN
700 ZA$="THE C# MAJOR CHORD":ZX=012:ZY=075:GOSUB 040:? ,"500
 SOUND 0,114,10,2":ZA$="C#":ZX=016:ZY=055:GOSUB 040
742 P=0114:GOSUB 100:X=0130:Y=047:GOSUB 050:PLOT 0122,050:DR
AWTO 0138,050
760 ZA$="#":ZX=017:ZY=047:GOSUB 040:? ,"510 SOUND 1,91,10,2"
:ZA$="F":ZX=019:ZY=055:GOSUB 040
820 P=091:GOSUB 0100:X=0154:Y=038:GOSUB 050:? ,"520 SOUND 2,
76,10,2"
910 ZA$="G#":ZX=022:ZY=055:GOSUB 040:P=076:GOSUB 0100:X=0178
:Y=035:GOSUB 050
940 ZA$="#":ZX=023:ZY=035:GOSUB 040:ZA$="C#+F+G#=C# MAJOR":Z
X=013:ZY=095:GOSUB 040
960 SOUND 0,0114,010,02:SOUND 01,091,010,02:SOUND 02,076,01
0,02:GOSUB 060:GOSUB 070:RETURN
3100 ZA$="THE A MAJOR CHORD":ZX=012:ZY=075:GOSUB 040:? ,"500
 SOUND 0,72,10,2":ZA$="A":ZX=016:ZY=055:GOSUB 040
3142 FOR VOL=010 TO 00 STEP -06:SOUND 00,072,010,VOL:NEXT VO
L:X=0130:Y=032:GOSUB 050:? ,"510 SOUND 1,57,10,2"
3210 ZA$="C#":ZX=019:ZY=055:GOSUB 040:FOR VOL=010 TO 00 STEP
 -06:SOUND 00,057,010,VOL:NEXT VOL:X=0154:Y=026:GOSUB 050
3240 ZA$="#":ZX=020:ZY=026:GOSUB 040:? ,"520 SOUND 2,47,10,2
":ZA$="E":ZX=022:ZY=055:GOSUB 040
3320 FOR VOL=010 TO 00 STEP -06:SOUND 00,047,010,VOL:NEXT VO
L:X=0178:Y=020:GOSUB 050
3350 ZA$="A+C#+E=A MAJOR":ZX=013:ZY=095:GOSUB 040
3360 SOUND 00,072,010,02:SOUND 01,057,010,02:SOUND 02,047,01
0,02:GOSUB 060:GOSUB 070:RETURN
3400 ZA$="THE A# MAJOR CHORD":ZX=012:ZY=075:GOSUB 040:? ,"50
0 SOUND 0,68,10,2":ZA$="A#":ZX=016:ZY=055:GOSUB 040
3442 FOR VOL=010 TO 00 STEP -06:SOUND 00,068,010,VOL:NEXT VO
L:X=0130:Y=032:GOSUB 050:ZA$="#":ZX=017:ZY=032:GOSUB 040
3500 ? ,"510 SOUND 1,53,10,2":ZA$="D":ZX=019:ZY=055:GOSUB 04
0:FOR VOL=010 TO 00 STEP -06:SOUND 00,053,010,VOL:NEXT VOL
3530 X=0154:Y=023:GOSUB 050:? ,"520 SOUND 2,45,10,2":ZA$="F"
:ZX=022:ZY=055:GOSUB 040
3620 FOR VOL=010 TO 00 STEP -06:SOUND 00,045,010,VOL:NEXT VO
L:X=0178:Y=017:GOSUB 050
3650 ZA$="A#+D+F=A# MAJOR":ZX=013:ZY=095:GOSUB 040
3660 SOUND 00,068,010,02:SOUND 01,053,010,02:SOUND 02,045,01
0,02:GOSUB 060:GOSUB 070:RETURN
3700 ZA$="THE B MAJOR CHORD":ZX=012:ZY=075:GOSUB 040:? ,"500
 SOUND 0,64,10,2":ZA$="B":ZX=016:ZY=055:GOSUB 040
3742 FOR VOL=010 TO 00 STEP -06:SOUND 00,064,010,VOL:NEXT VO
L:X=0130:Y=029:GOSUB 050:? ,"510 SOUND 1,50,10,2"
3810 ZA$="D#":ZX=019:ZY=055:GOSUB 040:FOR VOL=010 TO 00 STEP
 -06:SOUND 00,050,010,VOL:NEXT VOL:X=0154:Y=023:GOSUB 050
3840 ZA$="#":ZX=020:ZY=023:GOSUB 040:? ,"520 SOUND 2,42,10,2
":ZA$="F#":ZX=022:ZY=055:GOSUB 040
3920 FOR VOL=010 TO 00 STEP -06:SOUND 00,042,010,VOL:NEXT VO
L:X=0178:Y=017:GOSUB 050:ZA$="#":ZX=023:ZY=017:GOSUB 040
3950 ZA$="B+D#+F#=B MAJOR":ZX=013:ZY=095:GOSUB 040
3960 SOUND 00,064,010,02:SOUND 01,050,010,02:SOUND 02,042,01
0,02:GOSUB 060:GOSUB 070:RETURN
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9
```

```
30005 ? "K↓↓↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▶ press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020

30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
 PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "K":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM DTOG (c) 1982 by Jerry White [6/3/82]
3 Q0=0:Q1=1:Q2=2:Q3=3:Q4=4:Q5=5:Q6=0.2:Q8=8:Q10=10:Q12=12:Q1
3=13:Q16=16:Q17=17:Q19=19:Q20=20:Q22=22:Q23=23:Q26=26:Q29=29
:Q32=32
4 Q35=35:Q38=38:Q40=40:Q41=41:Q42=42:Q44=44:Q45=45:Q47=47:Q5
0=50:Q53=53:Q55=55:Q57=57:Q60=60:Q64=64:Q68=68:Q70=70:Q72=72
:Q75=75
5 Q76=76:Q80=80:Q81=81:Q85=85:Q91=91:Q95=95:Q96=96:Q102=102:
Q108=108:Q114=114:Q121=121:Q122=122:Q130=130:Q138=138:Q154=1
54
6 Q178=178:Q255=255:Q256=256:Q300=300:Q764=764:Q100=100:Q7=6


20 DIM ZA$(Q100):GOTO 150
40 ZL=PEEK(560)+PEEK(561)*Q256
42 ZM=PEEK(ZL+Q4)+PEEK(ZL+Q5)*Q256:FOR ZW=Q1 TO LEN(ZA$):ZT=
57344+((ASC(ZA$(ZW,ZW))-Q32)*Q8):ZC=ZM+ZY*Q40+ZX+ZW-Q1
44 FOR ZR=Q0 TO 7:POKE ZC+ZR*Q40,PEEK(ZT+ZR):NEXT ZR:ZY=ZY+Z
Z:NEXT ZW:RETURN
50 COLOR Q1:PLOT X+Q1,Y:DRAWTO X+Q2,Y:PLOT X,Y+Q1:DRAWTO X+Q
3,Y+Q1
52 PLOT X-Q1,Y+Q2:DRAWTO X+Q4,Y+Q2:PLOT X-Q1,Y+Q3:DRAWTO X+Q
4,Y+Q3
53 PLOT X,Y+Q4:DRAWTO X+Q3,Y+Q4:PLOT X+Q1,Y+Q5:DRAWTO X+Q2,Y
+Q5
55 PLOT X+Q4,Y-Q10:DRAWTO X+Q4,Y+Q2:RETURN
60 FOR HOLD=Q1 TO 500:NEXT HOLD:FOR OFF=Q0 TO Q3:SOUND OFF,Q
0,Q0,Q0:NEXT OFF:RETURN
70 ? "      PRESS ANY KEY TO CONTINUE.";:POKE Q764,Q255
71 IF PEEK(Q764)=Q255 AND PEEK(53279)=7 THEN 71
72 RETURN
80 GRAPHICS Q8:SETCOLOR Q0,Q2,Q0:SETCOLOR Q1,Q2,Q0:SETCOLOR
Q2,Q2,Q8:SETCOLOR Q4,Q2,Q8:POKE 752,Q1:COLOR Q1
81 PLOT Q20,Q20:DRAWTO Q300,Q20:PLOT Q20,Q26:DRAWTO Q300,Q26
:PLOT Q20,Q32:DRAWTO Q300,Q32:PLOT Q20,Q38:DRAWTO Q300,Q38
85 PLOT Q20,Q44:DRAWTO Q300,Q44:ZX=Q13:ZY=125:ZA$="BASIC CHO
RDS":GOSUB Q40:RETURN
100 FOR VOL=Q10 TO Q0 STEP -Q6:SOUND Q0,P,Q10,VOL:NEXT VOL:R
ETURN
150 GRAPHICS Q17:SETCOLOR Q2,Q8,Q12:? #Q7:? #Q7:? #Q7;"     S
ELECT CHORD":? #Q7;"      BY NUMBER":? #Q7
160 ? #Q7:? #Q7;"      1= D  MAJOR":? #Q7:? #Q7;"      2= DB  M
AJOR"
170 ? #Q7:? #Q7;"      3= E  MAJOR":? #Q7:? #Q7;"      4= F  M
AJOR"
180 ? #Q7:? #Q7;"      5= FB MAJOR":? #Q7:? #Q7;"      6= G  M
AJOR":? #Q7:? #Q7;"      7= GB MAJOR"
200 ? #Q7:? #Q7;"      8= Q U I T":P=Q10:GOSUB Q100
210 POKE Q764,Q255:CLOSE #Q1:OPEN #Q1,Q4,Q0,"K:":GET #Q1,GC:
CLOSE #Q1:GC=GC-48
215 IF GC=Q8 THEN 30000
220 IF GC<Q1 OR GC>7 THEN 210
230 GOSUB Q80:ON GC GOSUB 1000,1300,1600,1900,2200,2500,2800


240 GOTO 150
1000 ZA$="THE D MAJOR CHORD":ZX=Q12:ZY=Q75:GOSUB Q40:? ,"500
 SOUND 1,108,10,2":ZA$="D":ZX=Q16:ZY=Q55
1042 GOSUB Q40:P=Q108:GOSUB Q100:X=Q130:Y=Q44:GOSUB Q50:? ,"
510 SOUND 1,85,10,2"
1110 ZA$="F#":ZX=Q19:ZY=Q55:GOSUB Q40:P=Q85:GOSUB Q100:X=Q15
4:Y=Q38:GOSUB Q50
1140 ZA$="A":ZX=Q20:ZY=Q38:GOSUB Q40:? ,"520 SOUND 2,72,10,2
":ZA$="A":ZX=Q22:ZY=Q55:GOSUB Q40
1220 P=Q72:GOSUB Q100:X=Q178:Y=Q32:GOSUB Q50
```

```
1250 ZA$="D+F#+A=D MAJOR":ZX=Q13:ZY=Q95:GOSUB Q40
1260 SOUND Q0,Q108,Q10,Q2:SOUND Q1,Q85,Q10,Q2:SOUND Q2,Q72,Q
10,Q2:GOSUB Q60:GOSUB Q70:RETURN
1342 P=Q102:GOSUB Q100:X=Q130:Y=Q44:GOSUB Q50:ZA$="#":ZX=Q17
:ZY=Q44:GOSUB Q40
1400 ? ,"510 SOUND 1,81,10,2":ZA$="G":ZX=Q19:ZY=Q55:GOSUB Q4
0:P=Q81:GOSUB Q100
1430 X=Q154:Y=Q35:GOSUB Q50:? ,"520 SOUND 2,68,10,2":ZA$="A#
":ZX=Q22:ZY=Q55:GOSUB Q40
1520 P=Q68:GOSUB Q100:X=Q178:Y=Q32:GOSUB Q50:ZA$="#":ZX=Q23:
ZY=Q32:GOSUB Q40
1550 ZA$="D#+G+A#=D# MAJOR":ZX=Q13:ZY=Q95:GOSUB Q40
1560 SOUND Q0,Q102,Q10,Q2:SOUND Q1,Q81,Q10,Q2:SOUND Q2,Q68,Q
10,Q2:GOSUB Q60:GOSUB Q70:RETURN
1600 ZA$="THE E MAJOR CHORD":ZX=Q12:ZY=Q75:GOSUB Q40:? ,"500
 SOUND 0,96,10,2":ZA$="E":ZX=Q16:ZY=Q55:GOSUB Q40
1642 P=Q96:GOSUB Q100:X=Q130:Y=Q41:GOSUB Q50:? ,"510 SOUND 1
,76,10,2"
1710 ZA$="G#":ZX=Q19:ZY=Q55:GOSUB Q40:P=Q76:GOSUB Q100:X=Q15
4:Y=Q35:GOSUB Q50
1740 ZA$="#":ZX=Q20:ZY=Q35:GOSUB Q40:? ,"520 SOUND 2,64,10,2
":ZA$="B":ZX=Q22:ZY=Q55:GOSUB Q40
1820 P=Q64:GOSUB Q100:X=Q178:Y=Q29:GOSUB Q50
1850 ZA$="E+G#+B=E MAJOR":ZX=Q13:ZY=Q95:GOSUB Q40
1860 SOUND Q0,Q96,Q10,Q2:SOUND Q1,Q76,Q10,Q2:SOUND Q2,Q64,Q1
0,Q2:GOSUB Q60:GOSUB Q70:RETURN
1900 ZA$="THE F MAJOR CHORD":ZX=Q12:ZY=Q75:GOSUB Q40:? ,"500
 SOUND 0,91,10,2":ZA$="F":ZX=Q16:ZY=Q55:GOSUB Q40
1942 P=Q91:GOSUB Q100:X=Q130:Y=Q38:GOSUB Q50:? ,"510 SOUND 1
,72,10,2"
2010 ZA$="A":ZX=Q19:ZY=Q55:GOSUB Q40:P=Q72:GOSUB Q100:X=Q154
:Y=Q32:GOSUB Q50
2100 ? ,"520 SOUND 2,60,10,2":ZA$="C":ZX=Q22:ZY=Q55:GOSUB Q4
0:P=Q60:GOSUB Q100
2130 X=Q178:Y=Q26:GOSUB Q50:ZA$="F+A+C=F MAJOR":ZX=Q13:ZY=Q9
5:GOSUB Q40
2160 SOUND Q0,Q91,Q10,Q2:SOUND Q1,Q72,Q10,Q2:SOUND Q2,Q60,Q1
0,Q2:GOSUB Q60:GOSUB Q70:RETURN
2200 ZA$="THE F# MAJOR CHORD":ZX=Q12:ZY=Q75:GOSUB Q40:? ,"50
0 SOUND 0,85,10,2":ZA$="F#":ZX=Q16:ZY=Q55:GOSUB Q40
2242 P=Q85:GOSUB Q100:X=Q130:Y=Q38:GOSUB Q50:ZA$="#":ZX=Q17:
ZY=Q38:GOSUB Q40
2300 ? ,"510 SOUND 1,68,10,2":ZA$="A#":ZX=Q19:ZY=Q55:GOSUB Q
40:P=Q68:GOSUB Q100
2330 X=Q154:Y=Q32:GOSUB Q50:ZA$="#":ZX=Q20:ZY=Q32:GOSUB Q40:
? ,"520 SOUND 2,57,10,2":ZA$="C#":ZX=Q22:ZY=Q55:GOSUB Q40
2420 P=Q57:GOSUB Q100:X=Q178:Y=Q26:GOSUB Q50:ZA$="#":ZX=Q23:
ZY=Q26:GOSUB Q40
2450 ZA$="F#+A#+C#=F# MAJOR":ZX=Q13:ZY=Q95:GOSUB Q40
2460 SOUND Q0,Q85,Q10,Q2:SOUND Q1,Q68,Q10,Q2:SOUND Q2,Q57,Q1
0,Q2:GOSUB Q60:GOSUB Q70:RETURN
2500 ZA$="THE G MAJOR CHORD":ZX=Q12:ZY=Q75:GOSUB Q40:? ,"500
 SOUND 0,81,10,2":ZA$="G":ZX=Q16:ZY=Q55:GOSUB Q40
2542 P=Q81:GOSUB Q100:X=Q130:Y=Q35:GOSUB Q50:? ,"510 SOUND 1
,64,10,2"
2610 ZA$="B":ZX=Q19:ZY=Q55:GOSUB Q40:P=Q64:GOSUB Q100:X=Q154
:Y=Q29:GOSUB Q50
2700 ? ,"520 SOUND 2,53,10,2":ZA$="D":ZX=Q22:ZY=Q55:GOSUB Q4
0:P=Q53:GOSUB Q100
2730 X=Q178:Y=Q23:GOSUB Q50:ZA$="G+B+D=G MAJOR":ZX=Q13:ZY=Q9
5:GOSUB Q40
2760 SOUND Q0,Q81,Q10,Q2:SOUND Q1,Q64,Q10,Q2:SOUND Q2,Q53,Q1
0,Q2:GOSUB Q60:GOSUB Q70:RETURN
```

```
2800 ZA$="THE G# MAJOR CHORD":ZX=Q12:ZY=Q75:GOSUB Q40:? ,"50
0 SOUND 0,76,10,2":ZA$="G#":ZX=Q16:ZY=Q55:GOSUB Q40
2842 P=Q76:GOSUB Q100:X=Q130:Y=Q35:GOSUB Q50:ZA$="#":ZX=Q17:
ZY=Q35:GOSUB Q40
2900 ? ,"510 SOUND 1,60,10,2":ZA$="C":ZX=Q19:ZY=Q55:GOSUB Q4
0:P=Q60:GOSUB Q100
2930 X=Q154:Y=Q26:GOSUB Q50:? ,"520 SOUND 2,50,10,2":ZA$="D#
":ZX=Q22:ZY=Q55:GOSUB Q40
3020 P=Q50:GOSUB Q100:X=Q178:Y=Q23:GOSUB Q50:ZA$="#":ZX=Q23:
ZY=Q23:GOSUB Q40
3050 ZA$="G#+C+D#=G# MAJOR":ZX=Q13:ZY=Q95:GOSUB Q40
3060 SOUND Q0,Q76,Q10,Q2:SOUND Q1,Q60,Q10,Q2:SOUND Q2,Q50,Q1
0,Q2:GOSUB Q60:RETURN
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9


30005 ? "K↓↓↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▸ press the START button.";FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020


30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
 PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "K":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM CHORDS12 (c) 1982 by Jerry White [6/3/82]
3 REM MAJOR CHORDS IN DISTORTION LEVEL 12 BASS RANGE
4 REM ALL PITCHES IN DISTORTION LEVEL 10 CLEAR SOUND
5 REM FOUR TEXT COLORS USING GRAPHICS MODE 18
20 DIM A$(8),A5$(8),B$(8),C$(8),C5$(8),D$(8),D5$(8),E$(8),F$
(8),F5$(8),G$(8),G5$(8)
22 DIM P1(13),P2(13),P3(13),MC$(11):MC$="MAJOR CHORD"
30 A$="A    C# E ":A5$="A# D   F ":B$="B   D#  F#":C$="C   E  G
";C5$="C# F   G#"
32 D$="D   F# A ":D5$="D# G   A#":E$="E   G# B ":F$="F   A   C ":
F5$="F# A# C#"
34 G$="G   B   D ":G5$="G# C   D#":GOTO 200
35 ? #6;"     A ";MC$:RETURN
40 ? #6;"    A# ";MC$:RETURN
45 ? #6;"     B ";MC$:RETURN
50 ? #6;"     C ";MC$:RETURN
55 ? #6;"    C# ";MC$:RETURN
60 ? #6;"     D ";MC$:RETURN
65 ? #6;"    D# ";MC$:RETURN
70 ? #6;"     E ";MC$:RETURN
75 ? #6;"     F ";MC$:RETURN
80 ? #6;"    F# ";MC$:RETURN
85 ? #6;"     G ";MC$:RETURN
90 ? #6;"    G# ";MC$:RETURN
95 ? #6;"     A ";MC$:RETURN
99 FOR WAIT=1 TO 200:NEXT WAIT:RETURN
100 DATA 75,60,51,72,57,48,67,55,45,63,51,42,60,48,40,57,45,
37,55,42,36
102 DATA 51,40,33,48,37,31,45,36,30,42,33,28,40,31,27,37,30,
25
200 FOR CHORD=1 TO 13:READ P1,P2,P3:P1(CHORD)=P1:P2(CHORD)=P
2:P3(CHORD)=P3:NEXT CHORD
1000 FOR CHORD=1 TO 13:GRAPHICS 18:POKE 559,0:? #6
1020 ON CHORD GOSUB 35,40,45,50,55,60,65,70,75,80,85,90,95
1040 POKE 559,34:? #6:? #6;"    sound 1,";P1(CHORD);",12,4"
1050 SOUND 1,P1(CHORD),12,4:GOSUB 99
1060 ? #6:? #6;"    sound 2,";P2(CHORD);",12,2"
1070 SOUND 2,P2(CHORD),12,2:GOSUB 99
1080 ? #6:? #6;"    sound 3,";P3(CHORD);",12,2"
1090 SOUND 3,P3(CHORD),12,2:GOSUB 99
1400 FOR HOLD=1 TO 200:NEXT HOLD
1500 SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0
1600 ? #6:? #6;"     Press Start"
1700 IF PEEK(53279)<>6 THEN 1700
1800 NEXT CHORD
2000 GRAPHICS 18:? #6:? #6;"    THE CLEAR sound"
2100 FOR PITCH=255 TO 0 STEP -1:POSITION 3,4
2120 SETCOLOR 0,PITCH,6:SETCOLOR 3,PITCH+8,6:SETCOLOR 1,PITC
H+4,6
2200 ? #6;"SOUND 0,";PITCH;",10,V  "
2220 FOR VOLUME=14 TO 0 STEP -2:SOUND 0,PITCH,10,VOLUME:NEXT
 VOLUME:NEXT PITCH
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9


30005 ? "↖↓↓↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▸ press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020
```

```
30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
 PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "K":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM SPEAKER (c) 1982 Jerry White [6/3/82]
5 GOSUB 20:FOR SPEED=21 TO 1 STEP -10:? :? ,"   SPEED=";SPEE
D:JW$(10,10)=CHR$(SPEED):GOSUB 11
6 FOR WAIT=1 TO 300:NEXT WAIT:NEXT SPEED:GOTO 70
10 JW=USR(ADR(JW$)):RETURN
11 JW$(28,28)=CHR$(77):JW$(20,20)=CHR$(77):GOSUB 10
12 JW$(28,28)=CHR$(61):JW$(20,20)=CHR$(61):GOSUB 10
13 JW$(28,28)=CHR$(51):JW$(20,20)=CHR$(51):GOSUB 10
14 JW$(28,28)=CHR$(38):JW$(20,20)=CHR$(38):GOSUB 10:RETURN
20 GRAPHICS 0:POKE 82,2:POKE 83,39:POKE 752,1:POKE 710,144:?
 "↓↓          CONSOLE SPEAKER DEMO"
25 POKE 66,1:? CHR$(253):POKE 66,0:? "↓↓ This routine will p
lay a four note"
30 ? :? "  chord using the console speaker.":? :? "      Pres
s any key to begin..."
40 DIM JW$(39):JW$="h[illegible]"
:POKE 764,255
50 IF PEEK(764)=255 AND PEEK(53279)=7 THEN 50
60 RETURN
70 FOR QUICK=1 TO 4:GOSUB 11:NEXT QUICK
80 GOSUB 800:POSITION 9,20:? "Press OPTION to exit"
90 POSITION 9,21:? "Press START to edit":POKE 710,0
100 IF PEEK(53279)=6 THEN 200
110 IF PEEK(53279)=3 THEN 30000
120 POKE 755,1:POKE 755,2:GOTO 100
200 GOSUB 800:? "K":POKE 710,16:? :? "ENTER NUMBERS FROM 0 T
O 255 RETURN":POKE 764,255:POKE 752,0
210 NUM=-1:LINE=210:TRAP 900:? :? "DURATION";:INPUT NUM:GOTO
 900
220 JW$(10,10)=CHR$(NUM)
230 NUM=-1:LINE=230:TRAP 900:? :? "SPEAKER ON";:INPUT NUM:GO
TO 900
240 JW$(20,20)=CHR$(NUM)
250 NUM=-1:LINE=250:TRAP 900:? :? "SPEAKER OFF";:INPUT NUM:G
OTO 900
260 JW$(28,28)=CHR$(NUM):POKE 752,1
270 GOSUB 10:POSITION 9,20:? "Press OPTION to repeat":POSITI
ON 9,21:? "Press START to continue":POKE 710,0
280 IF PEEK(53279)=3 THEN 270
290 IF PEEK(53279)=6 THEN ? "K":GOTO 80
300 POKE 755,1:POKE 755,2:GOTO 280
800 IF PEEK(53279)<>7 THEN 800
810 RETURN
900 NUM=INT(NUM):IF NUM<0 OR NUM>255 THEN ? "[illegible]":? "PLEASE EN
TER A NUMBER FROM 0 TO 255.":GOTO LINE
910 GOTO LINE+10
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9


30005 ? "K↓↓↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▶ press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020
```

```
30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
 PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM";? :? ,"RETURN TO
 BASIC";? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "K";? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC";? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM HAPPY BIRTHDAY (c) 1982 BY JERRY WHITE [6/3/82]
20 GRAPHICS 0:DIM NAME$(20):? :? "BIRTHDAY SONG PROGRAM:":?
:? "ENTER NAME:";:INPUT NAME$
40 DIM WORD$(20),DISP$(20),BLANK$(20):BLANK$="
     ":GOTO 300
100 POKE 540,HOLD:SOUND 0,V0,10,8:SOUND 1,V1,10,4:SOUND 2,V2
,10,4:SOUND 3,V3,10,4
101 IF PEEK(540)<>0 THEN 101
102 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0:
RETURN
300 READ V0,V1,V2,V3,HOLD,SWITCH,WORD$
310 IF SWITCH=3 THEN WORD$=NAME$:GOTO 360
330 IF SWITCH=1 THEN GRAPHICS 18:GOTO 360
340 IF SWITCH=9 THEN 9000
360 LW=LEN(WORD$):DISP$=BLANK$:IF LW>18 THEN DISP$=WORD$:GOT
O 380
370 FP=11-INT(LW/2):DISP$(FP,FP+LW-1)=WORD$
380 POSITION 0,4:? #6;DISP$:GOSUB 100:GOTO 300
500 DATA 121,121,243,243,19,1,HAPPY BIRTHDAY
510 DATA 121,121,243,243,19,0,HAPPY BIRTHDAY
520 DATA 108,108,217,217,39,0,HAPPY BIRTHDAY
530 DATA 121,121,243,243,39,0,HAPPY BIRTHDAY
540 DATA 91,91,182,182,39,1,TO YOU
550 DATA 96,96,193,193,79,0,TO YOU
600 DATA 121,121,243,243,19,1,HAPPY BIRTHDAY
610 DATA 121,121,243,243,19,0,HAPPY BIRTHDAY
620 DATA 108,108,217,217,39,0,HAPPY BIRTHDAY
630 DATA 121,121,243,243,39,0,HAPPY BIRTHDAY
640 DATA 81,81,162,162,39,1,TO YOU
650 DATA 91,91,182,182,79,0,TO YOU
700 DATA 121,121,243,243,19,1,HAPPY BIRTHDAY
710 DATA 121,121,243,243,19,0,HAPPY BIRTHDAY
720 DATA 60,60,121,121,39,0,HAPPY BIRTHDAY
730 DATA 72,72,144,144,39,0,HAPPY BIRTHDAY
740 DATA 91,91,182,182,39,1,DEAR
750 DATA 96,96,193,193,39,3,NAME
760 DATA 108,108,217,217,79,3,NAME
770 DATA 0,0,0,0,19,3,NAME
800 DATA 68,68,136,136,19,1,HAPPY BIRTHDAY
810 DATA 68,68,136,136,19,0,HAPPY BIRTHDAY
820 DATA 72,72,144,144,39,0,HAPPY BIRTHDAY
830 DATA 91,91,182,182,39,0,HAPPY BIRTHDAY
840 DATA 81,96,121,162,39,1,TO YOU
850 DATA 91,121,144,182,79,0,TO YOU
8000 DATA 0,0,0,0,0,9,END
9000 GRAPHICS 18:SETCOLOR 2,4,8:SETCOLOR 0,12,6:SETCOLOR 4,4
,0
9100 POSITION 3,4:? #6;"HAPPY BIRTHDAY"
9200 FOR HOLD=1 TO 500:NEXT HOLD:GOTO 30000
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9


30005 ? "K↓↓↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▶ press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020
```

```
30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "K":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM JINGLE (c) 1982 BY JERRY WHITE [8/12/82]
20 GRAPHICS 7:DIM PM$(2048),PLL$(5),PLR$(5),WORD$(40):GOSUB
2000:GOTO 300
100 POKE 20,0:POKE 53760,V0:POKE 53762,V1:POKE 53764,V2:POKE
 53766,V3
101 IF V0=0 THEN 110
102 POKE 704,V0:POKE 705,V1:POKE 706,V2:POKE 707,V3
110 IF PEEK(20)<HOLD THEN 110
120 POKE 53760,0:POKE 53762,0:POKE 53764,0:POKE 53766,0
130 POKE 704,14:POKE 705,14:POKE 706,14:POKE 707,14:RETURN
300 READ V0,V1,V2,V3,HOLD,SWITCH,WORD$
340 IF SWITCH=9 THEN 9500
350 TRAP 360:IF TIME$=1 AND WORD$="DASHING THRU" THEN 9000
360 IF WORD$="*" OR SWITCH=2 THEN 400
380 ? CHR$(125),WORD$:GOSUB 100:GOTO 300
400 GOSUB 100:GOTO 300
600 DATA 47,60,81,96,19,1,JINGLE BELLS
610 DATA 47,60,81,96,19,0,*
620 DATA 47,60,81,96,29,0,*
625 DATA 0,0,0,0,9,2,REST
630 DATA 47,60,81,96,19,1,JINGLE BELLS
640 DATA 47,60,81,96,19,0,*
650 DATA 47,60,81,96,29,0,*
655 DATA 0,0,0,0,9,2,REST
660 DATA 47,60,81,96,19,1,JINGLE
670 DATA 40,47,60,81,19,0,*
680 DATA 60,81,96,121,29,1,ALL THE WAY
690 DATA 53,53,108,108,9,0,*
700 DATA 47,60,81,96,79,0,*
710 DATA 0,0,0,0,9,2,REST
720 DATA 45,60,72,91,19,1,OH WHAT FUN
730 DATA 45,60,72,91,19,0,*
740 DATA 45,60,72,91,29,0,*
750 DATA 45,60,72,91,9,1,IT IS TO RIDE
760 DATA 45,60,72,91,19,0,*
770 DATA 47,60,81,96,19,0,*
780 DATA 47,60,81,96,19,0,*
790 DATA 47,60,81,96,9,0,IN A ONE HORSE
800 DATA 47,60,81,96,9,0,*
810 DATA 47,60,81,96,19,0,*
820 DATA 53,64,81,108,19,0,*
830 DATA 53,64,81,108,19,1,OPEN SLEIGH
840 DATA 47,60,81,96,19,0,*
850 DATA 53,64,81,108,39,0,*
855 DATA 0,0,0,0,4,2,REST
860 DATA 40,47,60,162,39,0,*
870 DATA 0,0,0,0,4,2,REST
900 DATA 47,60,81,96,19,1,JINGLE BELLS
910 DATA 47,60,81,96,19,0,*
920 DATA 47,60,81,96,29,0,*
925 DATA 0,0,0,0,9,2,REST
930 DATA 47,60,81,96,19,1,JINGLE BELLS
940 DATA 47,60,81,96,19,0,*
950 DATA 47,60,81,96,29,0,*
955 DATA 0,0,0,0,9,2,REST
960 DATA 47,60,81,96,19,1,JINGLE
970 DATA 40,47,60,81,19,0,*
980 DATA 60,81,96,121,29,1,ALL THE WAY
990 DATA 53,53,108,108,9,0,*
992 DATA 47,60,81,96,79,0,*
994 DATA 0,0,0,0,9,2,REST
1000 DATA 45,60,72,91,19,1,OH WHAT FUN
1010 DATA 45,60,72,91,19,0,*
1020 DATA 45,60,72,91,29,0,*
```

```
1030 DATA 45,60,72,91,9,1,IT IS TO RIDE
1040 DATA 45,60,72,91,19,0,*
1050 DATA 47,60,81,96,19,0,*
1060 DATA 47,60,81,96,19,0,*
1070 DATA 47,60,81,96,9,1,IN A ONE HORSE
1080 DATA 47,60,81,96,9,0,*
1090 DATA 40,53,64,81,19,0,*
1100 DATA 40,53,64,81,19,0,*
1110 DATA 45,53,64,81,19,1,OPEN SLEIGH
1120 DATA 53,64,81,108,19,0,*
1130 DATA 60,81,96,121,79,0,*
1140 DATA 0,0,0,0,9,2,REST
1200 DATA 81,96,121,162,19,1,DASHING THRU
1210 DATA 47,47,96,96,19,0,*
1220 DATA 53,53,108,108,19,0,*
1230 DATA 60,60,121,121,19,1,THE SNOW
1240 DATA 81,96,121,162,59,0,*
1245 DATA 0,0,0,0,9,2,REST
1250 DATA 81,96,121,162,9,1,IN A ONE HORSE
1260 DATA 81,96,121,162,9,0,*
1270 DATA 81,96,121,162,19,0,*
1280 DATA 47,47,96,96,19,0,*
1290 DATA 53,53,108,108,19,1,OPEN SLEIGH
1300 DATA 60,60,121,121,19,0,*
1310 DATA 72,91,121,144,79,0,*
1320 DATA 0,0,0,0,9,2,REST
1330 DATA 72,91,121,144,19,1,OVER THE FIELDS
1340 DATA 45,45,91,91,19,0,*
1350 DATA 47,47,96,96,19,0,*
1360 DATA 53,53,108,108,19,1,WE GO
1370 DATA 64,81,108,128,79,0,*
1380 DATA 0,0,0,0,9,2,REST
1390 DATA 40,47,60,81,19,1,LAUGHING
1400 DATA 40,47,60,81,19,0,*
1410 DATA 45,45,91,91,19,1,ALL THE WAY
1420 DATA 53,53,108,108,19,0,*
1430 DATA 47,60,81,96,79,0,*
1440 DATA 0,0,0,0,9,2,REST
1500 DATA 81,96,121,162,19,1,BELLS ON BOB
1510 DATA 47,47,96,96,19,0,*
1520 DATA 53,53,108,108,19,0,*
1530 DATA 60,60,121,121,19,1,TAILS RING
1540 DATA 81,96,121,162,79,0,*
1545 DATA 0,0,0,0,9,2,REST
1550 DATA 81,96,121,162,19,1,MAKING SPIRITS
1560 DATA 47,47,96,96,19,0,*
1570 DATA 53,53,108,108,19,0,*
1580 DATA 60,60,121,121,19,0,*
1590 DATA 72,91,121,144,79,1,BRIGHT
1600 DATA 0,0,0,0,9,2,REST
1610 DATA 72,91,121,144,9,1,WHAT FUN IT IS
1620 DATA 72,91,121,144,19,0,*
1630 DATA 45,45,91,91,19,0,*
1640 DATA 47,47,96,96,19,0,*
1650 DATA 53,53,108,108,19,1,TO RIDE AND SING
1660 DATA 40,53,64,81,19,0,*
1670 DATA 40,53,64,81,19,0,*
1680 DATA 40,53,64,81,29,0,*
1700 DATA 40,53,64,81,9,1,A SLEIGHING
1710 DATA 35,35,72,72,9,0,*
1720 DATA 40,53,64,81,19,0,*
1730 DATA 45,45,91,91,19,1,SONG TONITE
1740 DATA 53,53,108,108,19,0,*
1750 DATA 60,81,96,121,79,0,*
2000 POKE 752,1:POKE 559,62:I=PEEK(106)-32:POKE 54279,I:POKE
 623,33
```

```
2040 VTAB=PEEK(134)+PEEK(135)*256
2050 ATAB=PEEK(140)+PEEK(141)*256
2060 OFFS=I*256+1024-ATAB
2070 HI=INT(OFFS/256):LO=OFFS-HI*256
2080 POKE VTAB+2,LO:POKE VTAB+3,HI
2100 DIM MC$(42):MC$="hh[illegible]":CLEAR=USR(ADR(MC$),ADR(PM$),1024):GOTO 10000
8000 DATA 0,0,0,0,0,9,END
9000 POKE 201,10:? CHR$(125)," MERRY CHRISTMAS":? ,"from Jer
ry White"
9200 FOR HOLD=0 TO 231 STEP 0.5:POKE 704,HOLD:POKE 705,HOLD+
8:POKE 706,HOLD+16:POKE 707,HOLD+24:NEXT HOLD
9300 GOSUB 20000:POKE 53277,0:GOTO 30000
9500 TIME$=TIME$+1:FOR HOLD=1 TO 40:NEXT HOLD:RESTORE :GOTO
300
10000 PLL$(1,1)=CHR$(32):PLL$(2,2)=CHR$(112):PLL$(3,3)=CHR$(
248):PLL$(4,4)=PLL$(2,2):PLL$(5,5)=PLL$(1,1)
10003 PLR$(1,1)=CHR$(4):PLR$(2,2)=CHR$(14):PLR$(3,3)=CHR$(31
):PLR$(4,4)=PLR$(2,2):PLR$(5,5)=PLR$(1,1)
10004 FOR ME=70 TO 150 STEP 20:PM$(ME+256,ME+260)=PLR$:PM$(M
E+266,ME+270)=PLL$
10005 PM$(ME+512,ME+516)=PLR$:PM$(ME+522,ME+526)=PLL$:NEXT M
E
10006 PM$(114,118)=PLR$:PM$(134,138)=PLR$:PM$(154,158)=PLL$
10007 PM$(882,892)=PLL$:PM$(902,906)=PLL$:PM$(922,926)=PLR$
10008 POKE 708,196:POKE 710,65:POKE 712,14:POKE 709,72:POKE
201,15:POKE 65,0:POKE 82,2:? :GOSUB 20000
10009 SOUND 0,0,0,0:POKE 53761,168:POKE 53763,164:POKE 53765
,164:POKE 53767,164
10010 POKE 765,1:FOR X=2 TO 6:A=8*X+80:B=11*X:C=80-8*X
10020 COLOR 1:PLOT 81,9:PLOT A-16,B-13:DRAWTO A,B:PLOT C-1,B
+1
10030 POSITION C+16,B-13:XIO 18,#6,12,0,"S:"
10040 COLOR 2:PLOT A,B:DRAWTO A-1,B+1:PLOT C,B:DRAWTO C+1,B+
1:DRAWTO C-1,B+1
10042 PLOT A,B+1:DRAWTO A+1,B+1:NEXT X
10050 COLOR 0:PLOT 81,9:COLOR 3:POKE 765,3:PLOT 90,67
10060 DRAWTO 90,79:DRAWTO 70,79:POSITION 70,67:XIO 18,#6,12,
0,"S:"
10070 COLOR 2:POKE 765,2:PLOT 80,1:DRAWTO 85,10:DRAWTO 83,9
10080 POSITION 74,4:XIO 18,#6,12,0,"S:":DRAWTO 86,4:DRAWTO 7
5,10
10090 DRAWTO 80,1:DRAWTO 80,5:DRAWTO 76,9:PLOT 83,5:POKE 201
,13:POKE 53277,2
11000 POKE 53248,102:POKE 53249,116:POKE 53250,132:POKE 5325
1,146:RETURN
20000 FOR ME=53248 TO 53251:POKE ME,0:NEXT ME:RETURN
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9

30005 ? "[illegible] Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? " press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020
```

```
30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
 PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "K":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM SILENT NIGHT (C) 1982 by Jerry White [6/3/82]
20 GRAPHICS 18:SETCOLOR 2,4,8:SETCOLOR 0,12,6:SETCOLOR 4,4,0


40 DIM WORD$(20),DISP$(20),BLANK$(20):BLANK$="
     ":GOTO 300
100 POKE 540,HOLD:SOUND 0,V0,10,8:SOUND 1,V1,10,4:SOUND 2,V2
,10,4:SOUND 3,V3,10,4
101 IF PEEK(540)<>0 THEN 101
102 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0:
RETURN
300 READ V0,V1,V2,V3,HOLD,SWITCH,WORD$
330 IF SWITCH=2 THEN WORD$=DISP$
340 IF SWITCH=9 THEN 9000
360 LW=LEN(WORD$):DISP$=BLANK$:IF LW>18 THEN DISP$=WORD$:GOT
O 380
370 FP=11-INT(LW/2):DISP$(FP,FP+LW-1)=WORD$
380 POSITION 0,4:? #6;DISP$:GOSUB 100:GOTO 300
600 DATA 81,96,121,162,89,1,SILENT NIGHT
610 DATA 72,72,144,144,29,0,SILENT NIGHT
620 DATA 81,96,121,162,59,0,SILENT NIGHT
630 DATA 96,121,81,193,119,0,SILENT NIGHT
640 DATA 0,0,0,0,59,2,REST
650 DATA 81,96,121,162,89,1,HOLY NIGHT
660 DATA 72,72,144,144,29,0,HOLY NIGHT
670 DATA 81,96,121,162,59,0,HOLY NIGHT
680 DATA 96,121,81,193,119,0,HOLY NIGHT
690 DATA 0,0,0,0,59,2,REST
700 DATA 53,64,81,108,119,1,ALL IS CALM
710 DATA 53,64,81,108,59,0,ALL IS CALM
720 DATA 64,81,108,128,119,0,ALL IS CALM
730 DATA 0,0,0,0,59,2,REST
740 DATA 60,81,96,121,119,0,ALL IS BRIGHT
750 DATA 60,81,96,121,59,0,ALL IS BRIGHT
760 DATA 81,96,121,162,119,0,ALL IS BRIGHT
770 DATA 0,0,0,0,59,2,REST
800 DATA 72,91,121,144,119,1,ROUND YON
810 DATA 72,91,121,144,59,0,ROUND YON
820 DATA 60,72,91,121,89,1,VIRGIN
830 DATA 64,64,128,128,29,0,VIRGIN
840 DATA 72,71,121,144,59,0,VIRGIN
850 DATA 81,96,121,162,89,1,MOTHER AND CHILD
860 DATA 72,72,144,144,29,0,MOTHER AND CHILD
870 DATA 81,96,121,162,59,0,MOTHER AND CHILD
880 DATA 96,121,81,193,119,0,MOTHER AND CHILD
890 DATA 0,0,0,0,59,2,REST
900 DATA 72,91,121,144,119,1,HOLY INFANT SO
910 DATA 72,91,121,144,59,0,HOLY INFANT SO
920 DATA 60,72,91,121,89,0,HOLY INFANT SO
930 DATA 64,64,128,128,29,0,HOLY INFANT SO
940 DATA 72,71,121,144,59,0,HOLY INFANT SO
950 DATA 81,96,121,162,89,1,TENDER AND MILD
960 DATA 72,72,144,144,29,0,TENDER AND MILD
970 DATA 81,96,121,162,59,0,TENDER AND MILD
980 DATA 96,121,81,193,119,0,TENDER AND MILD
990 DATA 0,0,0,0,59,2,REST
1000 DATA 53,64,81,108,119,1,SLEEP IN
1010 DATA 53,64,81,108,59,0,SLEEP IN
1020 DATA 45,53,64,81,89,1,HEAVENLY PEACE
1030 DATA 53,64,81,108,29,0,HEAVENLY PEACE
1040 DATA 64,81,108,128,59,0,HEAVENLY PEACE
1050 DATA 60,81,96,121,179,0,HEAVENLY PEACE
1060 DATA 47,60,81,96,119,0,HEAVENLY PEACE
1070 DATA 0,0,0,0,59,2,REST
1100 DATA 60,81,96,121,59,1,SLEEP IN
```

```
1110 DATA 81,96,121,162,59,0,SLEEP IN
1120 DATA 96,121,162,193,59,0,SLEEP IN
1130 DATA 81,108,128,162,89,1,HEAVENLY PEACE
1140 DATA 91,91,182,182,29,0,HEAVENLY PEACE
1150 DATA 108,128,162,217,59,0,HEAVENLY PEACE
1160 DATA 121,162,193,243,179,0,HEAVENLY PEACE
8000 DATA 0,0,0,0,0,9,END
9000 GRAPHICS 18:SETCOLOR 2,4,8:SETCOLOR 0,12,6:SETCOLOR 4,4
,0
9100 POSITION 3,4:? #6;"MERRY CHRISTMAS"
9200 FOR HOLD=1 TO 500:NEXT HOLD:GOTO 30000
30000 GRAPHICS 0:POKE 752,1:POKE 710,48:POKE 82,2:POKE 201,9


30005 ? "K↓↓↓ Use the OPTION or SELECT button to":? :? " hig
hlight your choice below, then"
30010 ? :? "▸ press the START button.":FOR ME=0 TO 8:POKE 53
279,ME:NEXT ME:GOSUB 31000:SEL=11
30020 POSITION SEL,SEL:? "RERUN THIS PROGRAM"
30100 BUTTON=PEEK(53279):IF BUTTON=7 THEN 30100
30110 GOSUB 32000:IF CHOICE=6 THEN 31500
30120 SEL=SEL+2:IF SEL>15 THEN SEL=11:GOSUB 31000:GOTO 30020


30130 IF SEL=13 THEN GOSUB 31000:POSITION 11,SEL:? "RETURN T
O BASIC":GOTO 30100
30140 IF SEL=15 THEN GOSUB 31000:POSITION 11,SEL:? "RUN MENU
 PROGRAM":GOTO 30100
30150 GOTO 30100
31000 POSITION 11,11:? "RERUN THIS PROGRAM":? :? ,"RETURN TO
 BASIC":? :? ,"RUN MENU PROGRAM":RETURN
31500 TRAP 30000:POKE 201,10:IF SEL=15 THEN ? "K":? :? ," LO
ADING MENU":RUN "D:MENU":TRAP 40000
31510 IF SEL=13 THEN GRAPHICS 0:? :? "BASIC":? "IS";:POKE 75
2,0:TRAP 40000:END
31520 TRAP 40000:RUN
32000 GOSUB 32200
32010 CHOICE=BUTTON:BUTTON=PEEK(53279):IF BUTTON<>7 THEN 320
10
32100 GOSUB 32200:RETURN
32200 FOR ME=0 TO 8:POKE 53279,ME:NEXT ME:RETURN
```

```
0 REM SONG WRITER [6/3/82]
1 REM (c) 1982 by Jerry White
20 GRAPHICS 0:LIST 0,1:? :? "ENTER DATA BEGINNING AT LINE 50
0.":? :? "DELETE LINES 20 & 30 BEFORE TESTING."
30 ? :? "TO RUN THE MENU PROGRAM":? :? "TYPE RUN";CHR$(34);"
D:MENU";CHR$(34):? :? "THEN PRESS RETURN":LIST 500:END
40 DIM WORD$(20),DISP$(20),BLANK$(20):BLANK$="
      ":GOTO 300
100 POKE 540,HOLD:SOUND 0,V0,10,8:SOUND 1,V1,10,4:SOUND 2,V2
,10,4:SOUND 3,V3,10,4
101 IF PEEK(540)<>0 THEN 101
102 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0:
RETURN
300 READ V0,V1,V2,V3,HOLD,SWITCH,WORD$
360 LW=LEN(WORD$):DISP$=BLANK$:IF LW>18 THEN DISP$=WORD$:GOT
O 380
370 FP=11-INT(LW/2):DISP$(FP,FP+LW-1)=WORD$
380 POSITION 0,4:? #6;DISP$:GOSUB 100:GOTO 300
500 DATA
8000 DATA 0,0,0,0,0,0,9,END
```

```
10 RESTORE 1002:POKE 559,0:GOTO 1000
12 TRAP Q12:GET #Q1,GC:SOUND Q2,Q2,Q2,Q2:TRAP 22:GC=GC-42:IF
 GC=-Q15 THEN GOTO Q101
13 IF PX(GC)=Q2 THEN GOTO Q12
15 SOUND Q2,P(GC),Q10,Q15:Z=USR(ADR(ER$),PM):PL$(Q4,Q4)=CHR$
(PY(GC)):Z=USR(ADR(PM$),ADR(PL$),14,PM):POKE Q248,PX(GC)
16 SOUND Q2,P(GC),Q10,Q7:GOTO Q12
22 Z=USR(ADR(ER$),PM):POKE 694,Q2:POKE 702,Q64:GOTO Q12
25 TRAP Q14:Z=USR(ADR(ER$),PM):SETCOLOR Q1,Q2,Q12:SOUND Q2,Q
2,Q2,Q2:RETURN
120 GOSUB 4000
128 ? "K":FOR ME=J1 TO J2:GC=FILE(ME,Q2):POKE 540,FILE(ME,Q1
):SOUND Q2,P(GC),Q10,Q15:Z=USR(ADR(ER$),PM):POKE Q248,PX(GC)

132 PL$(Q4,Q4)=CHR$(PY(GC)):Z=USR(ADR(PM$),ADR(PL$),14,PM):?
 "↑↑NOTE #";ME:SOUND Q2,P(GC),Q10,Q7
134 IF PEEK(Q764)<>Q255 THEN GOTO Q101
136 IF PEEK(540)<>Q2 THEN 136
138 SOUND Q2,Q2,Q2,Q2:NEXT ME
140 N=FILE(Q2,Q2):GOSUB Q25:GOTO Q101
150 RESTORE 5000:FOR J=Q1 TO Q20:READ GC
151 SOUND Q2,P(GC),Q10,Q15:PL$(Q4,Q4)=CHR$(PY(GC)):GOSUB Q25
:Z=USR(ADR(PM$),ADR(PL$),14,PM):POKE Q248,PX(GC)
152 SOUND Q2,Q2,Q2,Q2:NEXT J:GOSUB Q25:RETURN
160 N$="":K=Q2:TRAP 160
161 POKE Q764,Q255:POKE Q766,Q1:K=K+Q1:GET #Q1,GC:IF GC>47 A
ND GC<58 THEN N$(K,K)=STR$(GC-48):? N$(K,K);:GOTO 161
162 TRAP 165:IF GC=155 OR GC=Q27 THEN POKE Q766,Q2:N$(K,K)="
 ":RETURN
163 IF GC=126 AND K>Q1 THEN POKE Q766,Q2:? "4";:K=K-Q1:N$(K,
K)=" "
164 N$(K,K)=" ":K=K-Q1:POKE 694,Q2:POKE 702,Q64:GOTO 161
165 TRAP Q101:RETURN
170 TRAP Q170:POKE Q764,Q255:POKE Q766,Q1:GET #Q1,GC
171 IF GC>41 AND GC<91 THEN POKE Q766,Q2:RETURN
172 IF GC=155 OR GC=Q27 OR GC=32 THEN POKE Q766,Q2:RETURN
174 POKE 694,Q2:POKE 702,Q64:GOTO Q170
190 N$="":K=Q2:TRAP 190
191 POKE Q764,Q255:POKE Q766,Q1:K=K+Q1:GET #Q1,GC:IF GC>45 A
ND GC<91 THEN N$(K,K)=CHR$(GC):? N$(K,K);:GOTO 191
192 TRAP 195:IF GC=155 THEN POKE Q766,Q2:N$(K,K)=" ":RETURN

193 IF GC=Q27 THEN POKE Q766,Q2:N$(K,K)=" ":POP :GOTO Q101
194 IF GC=126 AND K>Q1 THEN POKE Q766,Q2:? "4";:K=K-Q1:N$(K,
K)=" "
195 N$(K,K)=" ":K=K-Q1:POKE 694,Q2:POKE 702,Q64:GOTO 191
196 TRAP Q101:RETURN
200 SETCOLOR Q4,Q9,Q4:SETCOLOR Q9,Q9,Q4:POKE Q752,Q1:GOSUB Q
25
201 ? "K     TYPE C TO READ FROM CASSETTE":? "     TYPE D TO R
EAD FROM DISKETTE";:GOSUB Q170
202 IF GC=Q27 THEN GOTO Q101
210 IF GC=68 THEN 250
212 IF GC<>67 THEN ? "■";:GOTO 200
220 ? "K↓  INSERT DATA CASSETTE":? "↓  PRESS PLAY THEN":? "↓
  PRESS RETURN";
230 TRAP 8600:OPEN #Q4,Q9,Q2,"C:":TRAP 8000
231 ? "K";:POKE Q752,Q1:? "↓  READING CASSETTE FILE":GET #Q4
,N:FILE(Q2,Q2)=N:GET #Q4,X
232 IF N>Q400 OR N<Q1 OR X<>Q2 THEN 8700
240 FOR ME=Q1 TO FILE(Q2,Q2):GET #Q4,X:FILE(ME,Q2)=X:GET #Q4
,X:FILE(ME,Q1)=X:NEXT ME:GOTO Q101
250 POKE Q752,Q2:? "K     TYPE DISK FILE NAME?";:GOSUB 190:D$
="D:":D$(LEN(D$)+Q1)=N$
```

```
260 TRAP 8600:? "{CLR}";:POKE 0752,01:? "{DOWN}     READING ";N$:CLOSE
 #Q16:OPEN #Q16,Q9,Q2,D$:TRAP 8000
262 GET #Q16,N:GET #Q16,X:IF N)0400 OR N<Q1 OR X<>Q2 THEN 87
00
264 FILE(Q2,Q2)=N:FOR ME=Q1 TO FILE(Q2,Q2):GET #Q16,X:FILE(M
E,Q2)=X:GET #Q16,X:FILE(ME,Q1)=X
280 NEXT ME:CLOSE #Q16:GOTO Q101
290 ? "{CLR}{INV}      THE FILE IN MEMORY IS LOST":? "{DOWN}WHEN YOU CREAT
E NEW DATA.":POKE 0752,01
292 ? "      TYPE 1 TO CREATE NEW DATA                TYPE 2 TO
SAVE DATA FILE";:GOSUB 0170
293 IF GC=027 THEN GOTO 0101
294 GC=GC-48:IF GC=01 THEN N=02:FILE(02,02)=02:GOTO 0300
296 IF GC=04 THEN 460
298 ? "{INV}";:GOTO 290
300 SETCOLOR 04,015,04:SETCOLOR 09,015,04:GOSUB 025:IF FILE(
02,02)=0400 OR N=0400 THEN GOTO 0400
370 ? "{CLR}";:POKE 0752,02:? "    TYPE DURATION (1-120) {INV}RETURN{INV}?"
;:GOSUB 160:IF GC=027 THEN GOTO 0101
375 TRAP 370:D=VAL(N$):TRAP 014
380 D=INT(D):IF D<01 OR D>0120 THEN ? "{INV}";:GOTO 370
381 POKE 0752,01:? K$
382 N=N+01:? "{DOWN}     TYPE NOTE #";N;:GOSUB 0170
383 SOUND 02,02,02,02:TRAP 383:GC=GC-42:IF GC=-015 THEN GOTO
 0101
384 IF GC=-010 THEN GC=04:GOTO 390
385 IF PX(GC)=02 THEN N=N-01:? "{INV}":GOTO 0300
386 SOUND 02,P(GC),010,015:PL$(04,04)=CHR$(PY(GC)):Z=USR(ADR
(ER$),PM):Z=USR(ADR(PM$),ADR(PL$),14,PM)
387 POKE 0248,PX(GC):SOUND 02,P(GC),010,07:POKE 540,D
388 IF PEEK(540)<>02 THEN 388
390 SOUND 02,02,02,02:FILE(N,01)=D:FILE(02,02)=N:FILE(N,02)=
GC:GOTO 0300
398 FILE(N,01)=D:FILE(02,02)=N:FILE(N,02)=GC
400 GOSUB 025:? "{INV}{CLR}{DOWN}    MEMORY DATA FILE FULL                TYPE
 1 TO CREATE NEW MEMORY DATA "
410 ? "     TYPE 2 TO SAVE MEMORY DATA               TYPE 3 TO P
LAY MEMORY DATA";
420 POKE 0764,0255:GOSUB 0170:GC=GC-48:GOSUB 025:IF GC=-21 T
HEN GOTO 0101
422 IF GC<01 OR GC>016 THEN ? "{INV}";:GOTO 420
430 IF GC=01 THEN GOTO 290
440 IF GC=016 THEN GOTO 0120
460 FILE(02,02)=N:SETCOLOR 04,011,04:SETCOLOR 09,011,04:POKE
 0752,01:IF FILE(02,02)=02 THEN 8800
461 ? "{CLR}      TYPE C TO SAVE ON CASSETTE":? "      TYPE D TO S
AVE ON DISKETTE";:GOSUB 170
462 IF GC=027 THEN GOTO 0101
470 IF GC=68 THEN 500
472 IF GC<>67 THEN ? "{INV}";:GOTO 460
480 ? "{CLR}{DOWN}  INSERT DATA CASSETTE":? "{DOWN}    PRESS REC & PLAY":?
"{DOWN}    PRESS {INV}RETURN{INV}";
482 TRAP 014:CLOSE #04:OPEN #04,08,02,"C:":? "{CLR}";:POKE 0752,
01:? "{DOWN} SAVING CASSETTE FILE"
484 FOR ME=02 TO FILE(02,02):PUT #04,FILE(ME,02):PUT #04,FIL
E(ME,01):NEXT ME:CLOSE #04:GOTO 0101
500 POKE 0752,02:? "{CLR}      TYPE DISK FILE NAME?";:POKE 0764,0
255:GOSUB 190:D$="D:";D$(LEN(D$)+01)=N$
510 TRAP 500:? "{CLR}";:POKE 0752,01:? "{DOWN}     SAVING ";N$:CLOSE #
Q16:OPEN #Q16,08,02,D$
520 FOR ME=02 TO FILE(02,02):PUT #Q16,FILE(ME,02):PUT #Q16,F
ILE(ME,01):NEXT ME:CLOSE #Q16:GOTO 0101
805 DATA 60,130,65,144,66,110,87,136,72,130,83,128,84,130,68
,121,90,110,82,114,96,130,70,108,102,110,84,102
810 DATA 108,130,71,96,120,130,72,91,126,110,85,85.132,130,7
4,81,138,110,73,76,144,130,75,72
```

```
820 DATA 150,110,79,68,156,130,76,64,168,130,59,60,174,110,4
5,57,180,130,43,53,186,110,61,50,192,130,42,47
1000 READ Q1,Q2,Q4,Q7,Q8,Q9,Q10,Q11,Q12,Q14,Q15,Q16,Q18,Q19,
Q23,Q24,Q25,Q101,Q120,Q248,Q255,Q280,Q300,Q752,Q764,Q3
1001 READ Q20,Q27,Q64,Q170,Q766,Q400:GOSUB 12100
1002 DATA 1,0,2,6,8,4,10,11,12,11000,15,3,18,7,3000,24,25,71
00,120,53248,255,2280,300,752,764,8200
1003 DATA 20,27,64,170,766,400
1004 GOSUB Q3:DIM PL$(23),PM$(81),ER$(Q24),PX(45),PY(45),P(45),D$(14),FILE(Q400,Q1),K$(74),N$(Q20):TRAP Q14
1008 K$="↖    W      R   T      U   I   O       -   =       A   S   D   F
  G   H   J   K   L   ;   +   *"
1010 RESTORE 805:FOR ME=Q1 TO 45:PX(ME)=Q2:PY(ME)=Q2:P(ME)=Q
2:NEXT ME:FILE(Q2,Q2)=Q2
1050 POKE Q248,Q2:GOSUB 9000:POKE 82,Q4:POKE 83,39
1100 FOR ME=15 TO Q2 STEP -Q.2:SOUND Q2,150-(ME*Q10),Q8,ME:S
OUND Q1,51,Q12,ME:SOUND Q4,102,Q12,ME
1120 SOUND Q16,150-(ME*Q10),Q8,ME:SETCOLOR Q2,ME,Q10:NEXT ME
:POKE 106,P106
1200 POKE Q248,Q2:GRAPHICS Q18:GOSUB 12100:SETCOLOR Q4,Q19,Q
8:SETCOLOR Q2,Q9,Q10
1210 ? #Q7:? #Q7;" TYPE OPTION NUMBER":? #Q7:? #Q7;"    1= PL
AY PIANO":X=29:GOSUB Q23
1220 ? #Q7:? #Q7;"    2= READ DATA":X=23:GOSUB Q23:? #Q7:? #Q
7;"    3= CREATE DATA":X=19:GOSUB Q23
1222 ? #Q7:? #Q7;"    4= END PROGRAM":X=14:GOSUB Q23
1230 CLOSE #Q1:OPEN #Q1,Q9,Q2,"K:":GOSUB Q170:OP=INT(GC-48):
IF OP<Q1 OR OP>Q9 THEN ? "■":GOTO 1200
1240 TRAP Q14:IF OP=Q1 THEN POSITION Q7,Q16:? #Q7;"PLAY PIAN
O":X=29:GOSUB Q23
1250 IF OP=Q4 THEN POSITION Q7,5:? #Q7;"READ DATA":X=23:GOSU
B Q23
1260 IF OP=Q16 THEN POSITION Q7,Q19:? #Q7;"CREATE DATA":X=19
:GOSUB Q23
1270 IF OP=Q9 THEN POSITION Q7,9:? #Q7;"END PROGRAM":X=14:GO
SUB Q23
1300 FOR ME=Q1 TO Q20:READ J1,J2,J3,J4:CO=J3-42:PX(CO)=J1:PY
(CO)=J2:P(CO)=J4:NEXT ME
1310 IF OP=Q9 THEN POKE Q248,Q2:GOTO 10300
1900 GRAPHICS Q19:SETCOLOR Q4,Q19,Q4:SETCOLOR Q9,Q19,Q4:SETC
OLOR Q2,Q2,Q2:SETCOLOR Q1,Q2,Q12:P106=PEEK(106)
2000 POKE 53277,Q2:PMA=P106-32:POKE 559,62:POKE 54279,PMA:J=
Q2:POKE 704,31:PM=(PMA+Q9)*256
2100 PL$="♥d◣◤ ◣◢◢◢8HH0♥":GOSUB 12100:POKE 53277,Q16
2200 GOSUB 31000
2266 IF OP<>Q1 THEN SETCOLOR Q1,Q2,Q12:GOTO 2500
2268 ? K$:POKE Q752,Q1:GOSUB Q25
2270 ? "▸ PRESS SPACE BAR FOR REST                    PRESS ESC FO
R OPTIONS";:IF ME=-Q1 THEN RETURN
2272 GOTO 2500
2500 IF OP=Q4 THEN GOSUB 150:GOTO 200
2510 IF OP=Q16 THEN GOSUB 150:GOTO Q300
2520 GOSUB 150:GOTO Q12
3000 FOR ME=Q15 TO Q2 STEP -Q1:SOUND Q2,X,Q10,ME:NEXT ME:RET
URN
4000 SETCOLOR Q4,5,Q4:SETCOLOR Q9,5,Q4:? "↖    THERE ARE ";FI
LE(Q2,Q2);" NOTES IN MEMORY":POKE Q764,Q255
4010 IF FILE(Q2,Q2)=Q2 THEN ? "■":FOR J1=Q2 TO 100:NEXT J1:J
1=Q2:N=Q2:POKE Q764,Q255:GOTO 7000
4020 ? "    ENTER STARTING NOTE#";:GOSUB 160:IF GC=Q27 THEN G
OTO Q101
4030 TRAP 4000:J1=VAL(N$):IF J1<Q1 OR J1>FILE(Q2,Q2) THEN ?
"■";:GOTO 4000
4040 ? :? "    ENTER ENDING NOTE#";:GOSUB 160:TRAP 4000:J2=VA
L(N$)
4060 IF J2<J1 OR J2<Q1 OR J2>FILE(Q2,Q2) THEN ? "■":GOTO 400
0
```

```
4080 TRAP Q14:RETURN
5000 DATA 23,45,41,26,40,28,42,29,30,43,32,31,33,37,34,17,3,
1,19,0
7000 J2=Q2:SETCOLOR Q4,Q19,Q4:SETCOLOR Q9,Q19,Q4:POKE Q752,Q
1:GOSUB Q25:TRAP 7000
7005 ? "K     TYPE 1 TO PLAY KEYBOARD PIANO            TYPE 2 TO
 READ A DATA FILE"
7010 ? "     TYPE 3 TO CREATE A NEW DATA FILE        TYPE 4 TO
END THIS PROGRAM";
7020 GOSUB Q170:OP=INT(GC-48):IF OP=-21 THEN GOSUB Q25:GOTO
Q101
7022 IF OP<Q1 OR OP>Q9 THEN ? "■";:GOTO 7020
7050 IF OP=Q1 THEN ME=-Q1:? :? :GOSUB 2268:GOTO Q12
7060 IF OP=Q4 THEN 200
7070 IF OP=Q16 AND FILE(Q2,Q2)=Q2 THEN N=Q2:GOTO Q300
7072 IF OP=Q16 THEN 290
7080 IF OP=Q9 AND FILE(Q2,Q2)=Q2 THEN 1310
7090 ? "K";:POKE Q752,Q1:? "      TYPE 1 TO SAVE MEMORY DATA"
:? "      TYPE 2 TO END THIS PROGRAM":GOSUB 170:GC=GC-48
7091 IF GC=-21 THEN GOTO Q101
7092 IF GC=Q1 THEN 200
7094 IF GC=Q4 THEN 1310
7096 ? "■";:GOTO 7090
7100 J2=Q2:SETCOLOR Q4,Q19,Q4:SETCOLOR Q9,Q19,Q4:POKE Q752,Q
1:GOSUB Q25:TRAP Q101
7105 ? "K     TYPE 1 TO ADD DATA TO MEMORY             TYPE 2 TO
 SAVE MEMORY DATA"
7110 ? "     TYPE 3 TO PLAY MEMORY DATA               TYPE 4 TO
FIX  MEMORY DATA";
7120 GOSUB Q170:GC=GC-48:IF GC=-21 THEN GOSUB Q25:GOTO 7000
7122 IF GC<Q1 OR GC>Q9 THEN ? "■";:GOTO 7120
7130 IF GC=Q16 THEN GOTO Q120
7140 IF GC=Q4 THEN 460
7150 IF GC=Q1 THEN N=FILE(Q2,Q2):GOTO Q300
7160 SETCOLOR Q4,9,Q9:SETCOLOR Q9,9,Q9:POKE Q752,Q2:? "K   T
HERE ARE ";FILE(Q2,Q2);" NOTES IN MEMORY"
7162 IF FILE(Q2,Q2)=Q2 THEN GOSUB Q25:POKE Q752,Q1:? "■":GOS
UB 8500:GOTO Q101
7165 ? "   TYPE NOTE NUMBER TO BE FIXED";:GOSUB 160
7170 TRAP 7160:IF GC=Q27 THEN GOTO Q101
7176 TRAP Q101:FN=VAL(N$):TRAP Q14:IF FN>FILE(Q2,Q2) OR FN<Q
1 OR FN>Q400 THEN ? "■";:GOTO 7160
7200 ? "K";:POKE Q752,Q2:? "    TYPE DURATION (1-120) RETURN?
";:TRAP 7200:GOSUB 160:D=VAL(N$):TRAP Q14
7210 D=INT(D):IF D<Q1 OR D>Q120 THEN ? "■";:GOTO 7200
7230 POKE Q752,Q1:? K$:? "▶     TYPE NOTE #";FN;:GOSUB Q170:P
OKE Q752,Q2
7240 SOUND Q2,Q2,Q2,Q2:TRAP 7240:GC=GC-42:IF GC=-Q15 THEN GO
TO Q101
7245 IF GC=-Q10 THEN GC=Q4:GOTO 7290
7250 IF PX(GC)=Q2 THEN N=N-Q1:? "■":GOTO Q101
7255 SOUND Q2,P(GC),Q10,Q15:PL$(Q4,Q4)=CHR$(PY(GC)):Z=USR(AD
R(ER$),PM):Z=USR(ADR(PM$),ADR(PL$),14,PM)
7260 POKE Q248,PX(GC):SOUND Q2,P(GC),Q10,Q7:POKE 540,D
7265 IF PEEK(540)<>Q2 THEN 7265
7290 SOUND Q2,Q2,Q2,Q2:FILE(FN,Q2)=GC:FILE(FN,Q1)=D:GOTO Q10
1
8000 IF PEEK(195)=136 THEN GOTO Q101
8100 GOTO 8700
8200 FOR ME=Q12 TO Q2 STEP -Q1:SOUND Q2,Q2,Q4,ME:NEXT ME:RET
URN
8500 FOR ME=Q2 TO 200:NEXT ME:RETURN
8600 ? "K■";:? ,"FILE NOT FOUND":CLOSE #Q4:CLOSE #Q16:GOSUB 8
500:GOTO Q101
```

```
8700 ? "K]":? ,"BAD DATA FILE":CLOSE #Q16:CLOSE #Q4:GOSUB 85
00:GOTO Q101
8800 ? "K]":? ,"0 RECORDS IN MEMORY":GOSUB 8500:GOTO Q101
9000 GRAPHICS Q18:P106=PEEK(106):POKE 709,Q12:POKE 711,56:PO
KE 710,106:GOSUB 12100
9080 PMA=P106-32:POKE 559,62:POKE 54279,PMA:J=Q2:POKE 704,41
:PM=(PMA+Q9)*256
9130 ER$="hh.Mh.L H ♥) ♥.LHP♦fMJPV♦"
9140 PM$="hh.Lh.Khh.Mh.Nh ♥ ♥LK ♥/HDMPV┘=♥/eN.Ph=♥/.Oh=♥/.Qh
 ♥=♥/.OhHFQZQI♥Pg=♥/I♥P/h ♥┘ZN♦":POKE 53277,Q16
9200 Z=USR(ADR(ER$),PM):POSITION Q9,Q4:? #Q7;" L YER  I NO"
9210 PL$="♥d♧♥♥||ffff||♦♦♦♦♥♥♥":PL$(Q4,Q4)=CHR$(40):POKE Q24
8,80
9220 FOR ME=30 TO Q64 STEP Q4:PL$(Q4,Q4)=CHR$(ME):Z=USR(ADR(
PM$),ADR(PL$),21,PM):NEXT ME:POSITION Q9,Q4:? #Q7;"P"
9230 GOSUB Q3:FOR ME=Q64 TO 80 STEP Q4:PL$(Q4,Q4)=CHR$(ME):Z
=USR(ADR(PM$),ADR(PL$),21,PM):NEXT ME
9235 FOR ME=80 TO 136:POKE Q248,ME:NEXT ME:FOR ME=80 TO Q64
STEP -Q4:PL$(Q4,Q4)=CHR$(ME)
9240 Z=USR(ADR(PM$),ADR(PL$),21,PM):NEXT ME:POSITION Q11,Q4:
? #Q7;"P":GOSUB Q3:Z=USR(ADR(ER$),PM):POKE Q248,Q2
9250 PL$="♥d♧♥♥┴┴<<ffff◀◀ff♥♥♥":PL$(Q4,Q4)=CHR$(Q64):POKE Q2
48,96
9260 FOR ME=30 TO Q64 STEP Q4:PL$(Q4,Q4)=CHR$(ME):Z=USR(ADR(
PM$),ADR(PL$),Q20,PM):NEXT ME:POSITION Q7,Q4:? #Q7;"A"
9270 GOSUB Q3:FOR ME=Q64 TO 80 STEP Q4:PL$(Q4,Q4)=CHR$(ME):Z
=USR(ADR(PM$),ADR(PL$),Q20,PM):NEXT ME
9280 FOR ME=96 TO 152:POKE Q248,ME:NEXT ME:FOR ME=80 TO Q64
STEP -Q4:PL$(Q4,Q4)=CHR$(ME)
9285 Z=USR(ADR(PM$),ADR(PL$),Q20,PM):NEXT ME:POSITION 13,Q4:
? #Q7;"A":GOSUB Q3:Z=USR(ADR(ER$),PM):POKE Q248,Q2
9300 ? #Q7:? #Q7;"     (C) 1981 BY":FOR ME=250 TO Q2 STEP -Q1
0:SOUND Q2,ME,Q10,Q10:NEXT ME
9310 SOUND Q2,Q2,Q2,Q2:POSITION Q9,Q7:? #Q7;"R":GOSUB Q3:POS
ITION 5,Q7:? #Q7;"G":GOSUB Q3
9320 POSITION Q7,Q7:? #Q7;"G":GOSUB Q3:POSITION 7,Q7:? #Q7;"
G":GOSUB Q3:POSITION Q8,Q7:? #Q7;"M":GOSUB Q3
9330 POSITION Q10,Q7:? #Q7;"M":GOSUB Q3:POSITION Q11,Q7:? #Q
7;"A":GOSUB Q3:POSITION Q12,Q7:? #Q7;"I"
9340 GOSUB Q3:POSITION 13,Q7:? #Q7;"E":GOSUB Q3
9400 PL$="♥d♧♥♥◀◀♦♦||♦♦♦♦◀◀♥♥♥":POKE 704,Q12:PL$(Q4,Q4)=CHR$
(128):POKE Q248,88
9410 Z=USR(ADR(PM$),ADR(PL$),Q20,PM)
9430 FOR ME=128 TO 144 STEP Q4:PL$(Q4,Q4)=CHR$(ME):Z=USR(ADR
(PM$),ADR(PL$),Q20,PM):NEXT ME
9440 FOR ME=88 TO 160:POKE Q248,ME:NEXT ME
9450 FOR ME=144 TO 128 STEP -Q4:PL$(Q4,Q4)=CHR$(ME):Z=USR(AD
R(PM$),ADR(PL$),Q20,PM):NEXT ME
9460 POSITION Q10,Q7:? #Q7;"white":GOSUB Q3:Z=USR(ADR(ER$),P
M):POKE Q248,Q2:POKE 53277,Q2:RETURN
10000 DATA 40,20,29,10,31,10,35,10,40,10,47,20,0,10,40,10,29
,4,29,4,31,10,35,10,33,10,31,20,0,10
10100 DATA 31,10,31,10,40,10,35,10,40,4,40,4,31,4,31,4,40,10
,35,10
10200 DATA 40,4,37,4,35,4,31,4,29,4,23,4,26,4,29,4,35,4,40,4
,29,4,26,10,23,10
10300 RESTORE 10000:GRAPHICS Q18:SETCOLOR Q16,5,Q10:? #Q7:?
#Q7:? #Q7:? #Q7;"       that's":? #Q7:? #Q7;"        ALL"
10400 ? #Q7:? #Q7;"       folks":TRAP 10999
10500 READ ME,J3:POKE 540,J3:SOUND Q2,ME,Q10,Q10:SETCOLOR Q9
,ME,Q4
10600 IF PEEK(540)<>Q2 THEN 10600
10700 SOUND Q2,Q2,Q2,Q2:GOTO 10500
10999 POKE 106,P106:END
11000 ? "K]▸  ERROR ";PEEK(195);" AT LINE ";PEEK(186)+PEEK(1
87)*256:? :? "▸  PRESS START FOR OPTIONS"
```

```
11111 IF PEEK(53279)<>07 THEN 11111
12000 TRAP 014:GOTO 7000
12100 POKE 16,64:POKE 53774,112:RETURN
31000 TRAP 31300:CLOSE #01:OPEN #01,09,02,"D:PIANO.SCR"
31100 POKE 850,7:POKE 852,244:POKE 853,PEEK(89)+1:POKE 856,2
55:POKE 857,255
31200 X=USR(ADR("hhh■LV■"),16):CLOSE #01:OPEN #01,09,02,"K:"

31220 COLOR 2:PLOT 14,10:PLOT 15,10:RETURN
31300 GRAPHICS 02:? :? "UNABLE TO LOAD PIANO SCREEN FROM DIS
K.":END
32000 REM PLAYER PIANO (C) 1981/82 by Jerry White Version 2.
0 9/19/82
```

ATARI BASIC CHORDS

* C H O R D *	DIST	FIRST	THIRD	FIFTH	DIST	FIRST	THIRD	FIFTH
A Major	10=	144	114	96	12=	75	60	51
A Minor	10=	144	121	96	12=	75	63	51
A# or Bb Major	10=	136	108	91	12=	72	57	48
A# or Bb Minor	10=	136	114	91	12=	72	60	48
B Major	10=	128	102	85	12=	67	52	45
B Minor	10=	128	108	85	12=	67	57	45
C Major	10=	121	96	81	12=	63	51	42
C Minor	10=	121	102	81	12=	63	52	42
C# or Db Major	10=	114	91	76	12=	60	48	40
C# or Db Minor	10=	114	96	76	12=	60	51	40
D Major	10=	108	85	72	12=	57	45	37
D Minor	10=	108	91	72	12=	57	48	37
D# or Eb Major	10=	102	81	68	12=	52	42	36
D# or Eb Minor	10=	102	85	68	12=	52	45	36
E Major	10=	96	76	64	12=	51	40	33
E Minor	10=	96	81	64	12=	51	42	33
F Major	10=	91	72	60	12=	48	37	31

ATARI BASIC CHORDS

(Continued)

x C H O R D x	DIST	FIRST	THIRD	FIFTH	DIST	FIRST	THIRD	FIFTH
F Minor	10=	91	76	60	12=	48	40	31
F# or Gb Major	10=	85	68	57	12=	45	36	30
F# or Gb Minor	10=	85	72	57	12=	45	37	30
G Major	10=	81	64	53	12=	42	33	28
G Minor	10=	81	68	53	12=	42	36	28
G# or Ab Major	10=	76	60	50	12=	40	31	26
G# or Ab Minor	10=	76	64	50	12=	40	33	26

NOTES

NOTES

NOTES

JERRY WHITE'S MUSIC LESSONS is a comprehensive sound effects and music tutorial package that will teach you everything you ever wanted to know about ATARI sound. Includes 13 separate programs exploring the full capabilities of your computer and PLAYER PIANO that turns your keyboard into a 20 note mini-piano.

INCLUDED:

SOUND DEMO
EFFECTS
FORNEXT
POKSOUND
KEY OF C
KEY A TO C
KEY D TO G
CHORDS 12
SPEAKER
BIRTHDAY
JINGLE
SILENT
SONGRITE
FILEDUMP

and PLAYER PIANO

MUSIC LESSONS 16K CASSETTE / 24K DISK

PLAYER PIANO 24K CASSETTE / 32K DISK

SWIFTY SOFTWARE, INC.
64 BROAD HOLLOW ROAD
MELVILLE, NEW YORK 11747