THE GUILFORD 99'ER NEWSLETTER

VOL.4 NO.5 MAY 1987

Joseph Martin, President
Mike Garrett, Secretary/Treasurer
BBS: (919)274-5760(FIDO)

Mack Jones, Vice President
Robert Dobo, Program Library
(919)621-2623(ROS)

The Guilford 99'er Users' Group Newsletter is free to dues paying members (One copy per family, please). Dues are $12.00 per family, per year. Send check to P.O. Box 21691, Greensboro, NC 27420. The Software Library is for dues paying members only. (Herman Geschwind, Editor)

OUR NEXT MEETING

DATE: May 5, 1987. TIME: 7:30 PM PLACE: Glenwood Recreation Center 2010 S. Chapman Street.

Our May program will be Carl Foster's presentation on Music and the 99/4A which was originally scheduled for March. At that time much to our regret we did have some technical difficulties so that the program had to be rescheduled for a later time. In years past Carl had shown us samples of his craft for our Christmas meetings so we know that Carl will have some interesting things to demonstrate this time.

V-P PEEKS

Well, it sure looks like Spring is finally here. I am afraid that this pretty weather is going to cut into my hacking quite a bit! It's hard to sit inside while the sunshine is so inviting. I guess I will just have to burn the midnight oil with the computer, and use the sunshine in the daylight. Sure is nice to be able to do both now.

I hate that some of you missed such a good meeting. Dan gave us a real good talk and anyone that made the meeting and don't know how to access and use the Groundstar ROS Board now must have been sleeping during the meeting! Along with the speaker, Herman showed slides of an actual session on the board step by step. Everything was covered that could possibly be of use to the user of the BB. I wish I could cover it for those not able to make the meeting, but there was just too much to cover here.

The door prize disk was won by George VonSeth. I can't figure out how, since his wife drew the winning ticket!!! Just kidding Ginny! Bob Dobo says we will have a disk of the month for sale at the May meeting. It will sell for a buck, with a little going into the much demanding treasury. We saved the printing fee for last month's printing due to the kindness of Carl, who printed them for us. Since I have had no volunteers for this month's printing, I guess we will have to pay to get it done. We can't expect Carl to do it each month.

The note on the door Tuesday night read..New hours 7:00 PM, so for those of you who couldn't make the meeting, we voted on a new time for the meeting. Starting with the May meeting, the new time will be 7:30 PM. This will also help some members who have to travel quite a few more miles than some of us do. Since the daylight saving time is now in effect, it is still

daytime at 7:30 and that means the kids are playing baseball so parking is a little more hectic than before.

Well, as one or two of you know, my console gave up the ghost last month and I had it all boxed up to send back to TI. A friend I was talking to on the phone said that a kid on his block wanted to sell his TI and wanted to know if I was interested. I told him I might be according to price. I got it and an extended basic along with three other cartridges for less than TI would swap with me. So needless to say, I jumped on it. I tell you this for one reason, and that is, you had better be on the look out for any spare consoles you can pick up cheap for parts later on. Some of you may not be aware of the fact that TI is only obligated to "watch out" for us users until 1988, and that will be here before you know it. They will still swap you a factory re-conditioned console if you send them yours and $36.50. That's a $30.00 charge for the console and $6.50 postage and handling charge. Of course you must pay to send yours in so you can add another five or six dollars to that.

Before I would go IBM or a clone, however, I think I would take a long look at the new GENEVE by MYARC. From what I have read about it, it seems like the next step for 99ers when they want a better upgrade. You set your console/keyboard aside along with the snake that hooks it to your PEB. Also, your expanded memory is no longer needed, as the Card/computer has 512k of CPU Ram. You are given an IBM type keyboard with a telephone type coiled cord that connects to your PEB, which is the only part of your TI that is used. You can use 512 colors on the screen at a single time which means you can have a different shade for each pixel on the screen. You may also switch from a 40 to 80 columns which will be a boon to TI WRITER/MULTIPLAN freaks. Another thing I like about it is the Bit-Mapped-Graphics Mode. It will also handle a mouse as well as the good ol' Joystick. Also built in is a Time Clock Chip that will let you know the time and date at a touch. The computer is 3-4 times faster than the 4/A and all this plugs into a slot in your TI PE Box!! Don't get me wrong, I am not trying to sell everyone on a GENEVE, but with no support from TI, just how long will those chips we have last? I now have 5 consoles, 3 good ones and 2 to strip for any parts I need, but it would be my luck for the same part to be bad on both of them!!

I have had some correspondence with a real nice guy from Knoxville , Tennessee. His name is Mike Dodd and he is the Newsletter Editor of the K-Town 99ers. He sent some programs he has written, such as EDITRUNNER, XB version of MAX/RLE, Pack/Unpack, and a resident Disk Manager he wrote called DM99 V2.1. If we don't have them in the Library, I will see that they get in.

Herman has really been socking it to the bulletin boards and if you are not using your modems, you should. We have just about every kind of program on the boards that you could want from music to games and adventures to business programs as well as tax programs for those of you with more moxie than I have if you fill out your own tax forms! Seriously, get to using those modems and let's let the SySops know we appreciate the TI space they have given us by leaving messages and talking to one another, ok?

If you have never took part in putting a Newsletter together, you can't appreciate the time and trouble that guys like Herman go through. I don't know how many of you ever take the time to tell him, but you should personally tell Herman how much you appreciate his time and effort and also tell him just what you would like to see in print. I hope you will take advantage of the offer that he has made regarding the re- programming of the Basic game. That should be a real challenge to some of our members, and the rewards are astounding. He has no way of knowing if no one ever tells him. Better still, submit us an article to put in that you would like others to know about. On this note, I would like to thank Larry for his continuing article on LOGO. I think that this will be very interesting to us all. I tried it a few years ago and was astonished at the ease that a program could be executed. Good work Larry.

Well, I am just about out of memory, so I will try to call the Editor and see if he can download this to his Newsletter, so until I see you at the meeting, take care. (Submitted by "Mac" Jones)

TI SHOPPER

Some interesting developments have occurred this month that warrant mentioning. First of all, I had to replace my 32K (a system failure took it and a console out of service). With the literature on hand from THE CAPTAIN'S WHEEL, I went ahead and ordered their basic 32K standalone with the option bank at >6000. The service was excellent! It only took a week from the time that I mailed in my order to when the package arrived in the mail. The unit itself is about half the size of the speech synthesizer and has a "load enable" switch on the top of the unit that controls the >6000 bank. Everything works fine!

I finally heard from TACHYON SYSTEMS, 5125 S. Westwind Way, Salt Lake City, UT 84118. They are still in the 32K business and offered to fix my old unit for $5.00 (if it was the interface chip) to $15.00 (for the set of memory chips). It

sounded like a good deal so I have mailed the old unit in. One interesting item that was mentioned in the correspondence was that they still have some of their original 32K units left. These are the large ones that come with a power supply and and "on-off" switch on the back. The interesting thing about them is that they can act as a 32K RAMDISK --- as long as the power to the 32K is on, your program stays in memory! At any rate, TACHYON is trying to reduce their inventory and they are selling the 32K's (old model) for $30.00 postpaid. A real deal if you haven't expanded!

On the "fairware" market, KWIKFONT has made a rather auspicious appearance! The offering is in the vein of M6 NIGHT MISSION. The program itself is a very good character definition program but that is only a minor part of the package! It comes with a series of text files that completely unravel the source code segment by segment (ie. like NIGHT MISSION did for the XB program). Not knowing any better, George Von Seth and I decided to run the docs through the FUNNELWEB FORMATTER and print everything out. Well, when we were done we had a 78-page document! The text is excellent and is one of the better A/L tutorials that I have seen. For $6, it is one of the best "fairware bargains that I have seen around.

I just got a flyer from ASGARD SOFTWARE the other day advertising GRAPHX COMPANION IV. It is a collection of more character fonts, pictures, and clipart on a "floppy". It is now available (check past newsletter issues for the address) for $6.75.

For those of you with some "extra" money to spend, the Myarc "computer on a card", the GENEVE 9640 is now being shipped. You get a PC-type keyboard that replaces the TI keyboard. The PEB card (the computer), and 8 pieces of software. The software includes a cartridge saver, Version 4.21 of PASCAL, "Advanced BASIC", the Myarc DOS, and 80-column versions of TI-Writer and Multiplan. It is advertised to operate "more than 95%" of the existing TI programs. The price tag is right around $500.00 depending on what options you choose.

Well, that about does it from the TI marketplace for this month. It looks like there are new (and better) programs for the TI coming out all the time --despite the fact that it is an "orphan". 'Til next month. (Contributed by Bob Carmany).

LITTLE BIG LOGO

LOGO's greatest asset is the procedure. It is the workhorse of this simple, yet powerful, language which allows individuals to program individually.

In fact, when thinking in LOGO think procedure. It is central to everything that is done in LOGO programming. It's true that certain commands are used to make your individual procedures run; and these are known as the primitives of LOGO. Think of primitives not as the structure of LOGO, but as the basic vocabulary, those basic "built-in" words reserved for specific purposes. They are not the brick but are the mortar of your program. Another analogy would be that primitives are that necessary, but nondescript subsurface foundation upon which procedures are constructed to give your program structure, function and maybe even beauty. Examples of primitives are "Print," which of course directs LOGO to print something on the screen; and "Repeat" which directs LOGO to repeat some action.

Think of procedures as your new words or commands in LOGO's language. Each time you create one, you broaden the LOGO dictionary. Procedures use the primitives and other previously defined procedures to do what you want the program to do. In LOGO, you get to uniquely define the procedures in remarkably flexible ways without a lot of rules, signposts or restrictions. This freedom is nice, but the trade-off is that it's a lot easier to get lost in LOGO perhaps than it is in more structured and restrictive computer languages. Gratefully, LOGO's error messages are generally very helpful, downright friendly in some cases.

In summary, the primitives (the built-in commands) are used to construct a procedure, which can reference or call other procedures or be called by them to perform your specific tasks. Indeed, procedures can even call themselves. They can take inputs and produce outputs (in non-computerese that means "communicate"); and, amazing little-big LOGO doesn't even require you to preassign, alot or define memory locations or space. Storage space is automatically set aside when you define the procedure.

So what does a procedure look like. Well, a simple one would be:

```
TO ADD :X :Y
PRINT X + Y
```

```
END
```

This procedure takes two inputs, X and Y, when you call the procedure "ADD." It adds them, of course, and then outputs the answer to the screen. To run the procedure ADD, you would simply type ADD, a space, the two input numbers to be summed and press the enter key.

Suppose you wanted the average of X and Y. You could perform that function very simply by expanding the procedure "ADD," but for illustrative purposes consider doing it by calling another another procedure:

```
TO AVERAGE :X
PRINT (:X / 2)
END
```

You would need to rewrite "ADD" to "OUTPUT" the sum, instead of "PRINT" it. Additionally, when running the procedure "AVERAGE", you would have to give as the value of ":X", the procedure "ADD" and its two inputs. This is done parenthetically: (ADD 3 3), for example. Now, suppose you wanted the average of two sums. Try:

```
TO AVERAGETWO :X :Y
PRINT (:X + :Y) / 2
END
```

Of course, when executing "AVERAGETWO" you would have to give the two inputs for "ADD", for example: AVERAGETWO (ADD 3 3) (ADD 4 4).

This is a tedious example, unworthy of LOGO; however, it illustrates at least a couple of things. These are individual functional procedures which communicate with each other once you've provided the inputs. Other procedures could call each of these to perform yet other function, say for example, a test to see if an answer is greater or less than a certain number. The variations are endless, with the communication channels running in both directions between procedures. Notice also that LOGO isn't confused by the repeated use of the X and Y variables in the various procedures. This is LOGO's ability to keep each procedure's words or variables as unique to it in so-called "private" libraries.

So how do you go about creating a procedure. In LOGO, there are several modes, one of which is the editor. It is used to create procedures and, you guessed it, to automatically store them for call from another procedure or to call other procedures. You begin writing a procedure simply by typing "TO" and your procedure name, for example, "TO ADD." LOGO shifts to the editor (green screen) and you can now write the procedure, line by line, ending it with the primitive word "END." End separates one procedure from another, allowing LOGO to keep them and their "libraries" distinct one from another. To get out of the editor, press Function 9 (BACK). To edit an existing procedure type EDIT and the name of the procedure.

Until next time, think procedure. (Contributed by Larry Spohn)

THE FORTH FORUM

This month, we are going to do some basic work in TI-Forth. As you probably know, there are two ways to save screens to disk when you use TI-Forth. First, you can simply save them to an initialized disk with no other preparation. The result is an unrecognizable chunk of data that you can only retrieve with a track copier or TI-Forth. The screens are unreadable by conventional means (ie. DM 1000 et. al.). Here is the alternative method:

Load -SYNONYMS -EDITOR (if not already loaded)

Type in- 0 FORMAT-DISK (this initializes a disk in drive #1)

Type in- DISK-HEAD (this creates a conventional disk file header named 'SCREENS')

Type in- 1 CLEAR (this will CLEAR Screen #1 and prepare it for EDITing with the TI Forth editor)

Type in- 1 EDIT (you are ready to enter your screen data with the TI-Forth editor)

When you are done, you will have a disk with the filename 'SCREENS' on it that can be processed by any of the conventional disk utility programs. This is just one of the more poorly documented items in the TI-Forth manual.

It is interesting to note that Wycove Forth automatically produces the header for you whenever you save a screen to an initialized disk --a major convenience!

TI-Forth also has a utility word SCOPY on screen #39 that will allow you to copy screens from one location to another. If you are using a 2-drive system, you should realize that screens 0-89 are on drive 1 and 90-179 are on drive 2. You cannot access any screen number above 89 unless you have told the system that you are using two drives. To do this, type in :

180 SCRN_HI !

Or, you can squeeze this on one of the blank lines on screen #3 and make it a permanent part of your system.
To copy screen #6 to screen #110 (screen #20 on drive 2) simply enter the following code after loading screen #39.

DECIMAL 6 110 SCOPY

Here is an explanation of the code: DECIMAL tells the system that your number base is decimal (ie. base 10). The number 6 is the screen number of the screen that we wish to copy and 110 (ie. the 20th screen on drive 2) is where we want the copy of screen #6 to reside. SCOPY does the work of actually copying the screen. Just remember that if there was anything already on your destination screen, it will be over-written.

That just about does it for this month, I had planned to include a couple of Wycove Forth utility screens but I haven't gotten all of the "bugs" out of them. My system "crash" put me several weeks behind! Next month, we will have some TI-Forth and Wycove Forth screens for you. (Contributed by Bob Carmany)

MODEM TALK

ROS (919) 621-2613

Here is a quick rundown of what is new on the ROS board:

MUSARC.TI Two music programs which came to us from Australia. One is an Assembly version of Bach's Toccata. The other is a real cute program: Dueling Bytes which in turn is an adaptation of a program called Duelling Banjos. Not only is the musical treatment very nice but the author managed to come up with some interesting graphics.

EA1ARC.TI, EA2ARC.TI, EA3ARCTI. This is another program collection which came to us from Australia. The idea is to provide a collection of commonly used routines to help the Assembly programmer. The bulk of programming in any language is to write to the screen, accept input, etc. While assembly might be fast in execution, something as simple as clearing the screen might call for dozens of lines of E/A code. The idea is to establish a collection of these common routines which could be copied into or linked by your own E/A programs and save you the tedium of having to re-invent the wheel. The files include extensive documentation and sample files, both source and object code.

STARARC.TI. A collection of assembly routines that can be accessed by the XB programmers to speed up execution of certain tasks. This is a fairware program which does just about the same job of a similar program called SUPER EXTENDED BASIC which was selling for close to $100 once upon a time.

CDOCSARC.TI, CPT1ARC.TI, CPT2ARC.TI, CPART3ARC.TI. This is a complete collection of the "c" language for the TI. This not only includes the latest release of Clint Pulley's "c" compiler but also the most up-to-date versions of support routines created by other programmers. We have checked all the support libraries and extensions for the most up-to-date and current release and have consolidated everything in four disks. Includes documentation, tutorials and sample files.

ARC2ARC.TI Since almost all of the files have been packed, an ARCHIVER program is required to unpack files before running. There have ben several versions, Barry Traver's original XB which is nice but slow, Mike Dodd's which is fast (E/A) and now a version by Barry Boone, ARCHIVER II. This is an assembly language program and hence about as fast as any, it is also a very compact program (only 19 sectors). The upload includes an XB loader. My recommendation: Trash all previous versions and go with Archiver II.

SCRABARC.TI If you like to play Scrabble, this is for you. It is an assembly language program with XB loader. Among the

play options available is one to play against the computer. Documentation is included in the file. Note that we converted this program to memory image with an XB loader so that the load process is not only simpler but also faster. Disregard the load instructions in the docs.

TRACKARC.TI Another Barry Boone fairware offering: Trackmaster I. One of the fastest and best track copiers around. I have yet to find a program of which Trackmaster I will not make a legitimate back up copy. Use with discretion and don't violate the law.

CRYPTARC.TI A letter substitution game. File includes over 300 proverbs and sayings which will be flashed up at random. The challenge is to decrypt the message. A very nicely done XB program. Fairware, docs included.

FIDO (919)-274-5760

During the next several weeks there will be a significant change at the FIDO board and one which we hope will be an improvement. To manage the TI section, Ben has granted us Sysop privileges and we will have full access to all the sysop features to keep the board current and attractive.

Once this gets rolling we will maintain two types of files, the ones that we consider permanent such as Funnelweb or languages and key utilities and a second selection where we will feature various games, music programs, etc. for a limited time, in other words, these programs will be available for a period of three to four weeks and then replaced by other selections. We believe that such an approach will be attractive to TI users and also fair to other users by not taking up an excessive amount of disk space.

In the future to access the TI section, a special access code will be required. Drop me a message if you need an access code.

For both boards we urge you to consider that uploading can be fun to. Check your disk library and if there is a program which you think that others might enjoy, UPLOAD. A tip for uploading: Use ARCHIVER 2 to pack and FastTerm to upload. MassTransfer still has problems with some of the Xmodem varieties that are being used by various boards. If you are not to comfortable about uploading, how about bringing the disk to a meeting and I will do the uploading in your name. (Contributed by Herman Geschwind)

BASIC CORNER

Our feature this month is what we consider to be a very good example of educational software for youngsters of pre-school or kindergarden age.

When TI was still in the business for the last several months before "Black Friday", Bill Cosby was pitching the 99/4A as the educational computer for family use, and I believe rightly so. To attract and hold the attention of smaller children the elements that are required are music (sound), color, graphics and speech and all of these the 99/4A could provide at reasonable cost.

It is interesting to note how emphasis in computing has shifted in the last three years, now it is three dimensional spreadsheets, sophisticated relational databases and "desk top publishing" but nary a word about family and educational computing. No wonder, the "Mac" does not offer color and speech for the PC or a clone can set you back by almost two grand. Then also, these systems are much more expensive, who in his right mind would risk bubble gum in the keyboard or drive opening of his $4000 "AT"?. No wonder that kiddie computing is no longer featured - what a shame.

ADDSUM has all the features that it takes: speech, music and graphics to keep a youngster entertained. If you only have a cassette-based system (or if that is all that you prefer for the kiddies to use), modify the pre-scan in lines 1 and 2 and the program will run in Console Basic.

My apologies to the original author. I downloaded this program a good while back from some board and whoever uploaded it deleted the author attribution. I think it is a well done program and my thanks to the author, whoever he may be... . (Contributed by Herman Geschwind)

```
1 @=0 :: [=1 :: ]=2 :: _=3 : : \=4 :: GOTO 100 :: G$ :: ANSW,B,COMB,CORR,DELAY,EC,FL,JWD,K,L,LEFT,MODE,PIC,QWL,RIGHT,S,STATUS,T,U,V,W,WRONG,Z
2 CALL CHAR :: CALL CLEAR :: CALL COLOR :: CALL HCHAR :: CALL KEY :: CALL SAY :: CALL SCREEN :: CALL SOUND :: CALL VCHAR :: !@P-
100 CALL CLEAR
110 CALL SCREEN(1)
120 PRINT TAB(10);"B A S I C "
130 PRINT
140 PRINT TAB(11);"ADDITION"
150 PRINT
160 PRINT TAB(12);"A N D"
170 PRINT
180 PRINT TAB(9);"SUBTRACTION": : : : : :
190 PRINT "PRESS THE [S] KEY IF YOU AREUSING EXTENDED BASIC AND A"
200 PRINT "SPEECH SYNTHESIZER."
210 PRINT :"OTHERWISE --          PRESS ANY OTHER KEY TO BEGIN"
220 PRINT : :
230 CALL CHAR(128,"FFFFFFFFFFFFFFFF")
240 CALL HCHAR(I,[,128,32)
250 CALL HCHAR(],[,128,32)
260 CALL HCHAR(23,[,128,32)
270 CALL HCHAR(24,[,128,32)
280 CALL VCHAR(_,[,128,20)
290 CALL VCHAR(_,],128,20)
300 CALL VCHAR(_,31,128,20)
310 CALL VCHAR(_,32,128,20)
320 CALL COLOR(13,6,6)
330 CALL SCREEN(10)
340 CALL SOUND(300,131,\,1047,\)
350 CALL SOUND(300,147,\,988,\)
360 CALL SOUND(300,165,\,880,\)
370 CALL SOUND(300,175,\,784,\)
380 CALL SOUND(300,196,\,698,\)
390 CALL SOUND(300,220,\,659,\)
400 CALL SOUND(300,247,\,587,\)
410 CALL SOUND(600,262,\,523,\)
420 CALL KEY(@,JWD,STATUS)
430 IF STATUS=@ THEN 420
440 IF JWD<>83 THEN 460
450 FL=[
460 CALL CLEAR
470 QWL=@
480 PRINT "PRESS"
490 PRINT
500 PRINT "1 FOR ADDITION"
510 PRINT
520 PRINT "2 FOR SUBTRACTION"
530 FOR K=[ TO 10
540 PRINT
550 NEXT K
560 CALL KEY(@,MODE,STATUS)
570 IF STATUS=@ THEN 560
580 IF (MODE<49)+(MODE>50)THEN 560
590 V=@
600 U=@
610 PRINT "PRESS"
620 PRINT
630 PRINT "1 FOR GRAPHICS AND NUMBERS"
640 PRINT
650 PRINT " WITH GRAPHIC ANSWER GIVEN"
660 PRINT
670 PRINT "2 FOR GRAPHICS AND NUMBERS"
680 PRINT
690 PRINT " WITHOUT GRAPHIC ANSWER"
700 PRINT
710 PRINT "3 FOR NUMBERS ONLY": : : : : : :
720 CALL KEY(@,COMB,STATUS)
730 IF STATUS=@ THEN 720
740 IF (COMB<49)+(COMB>51)THEN 720
750 CALL SCREEN(16)
760 CALL CLEAR
770 G$="FFFFFFFFFFFFFFFF"
780 CALL CHAR(130,G$)
790 CALL CHAR(60,G$)
800 CALL CHAR(34,G$)
810 CALL CHAR(123,G$)
820 CALL CHAR(41,G$)
830 PRINT "PRESS [M] TO RETURN TO MENU"
840 RANDOMIZE
850 PIC=INT(5*RND)+[
860 IF COMB=51 THEN 1070
870 ON PIC GOTO 890,930,970,1010,1050
880 !
890 CALL CHAR(136,"42E7FFFFF7E3C18")
900 CALL COLOR(14,9,16)
910 GOTO 1070
920 !
930 CALL CHAR(137,"FFFFFFFFFFFFFFF")
940 CALL COLOR(14,\,\)
950 GOTO 1070
960 !
970 CALL CHAR(138,"183C3C3C3C3C7EFF")
980 CALL COLOR(14,10,16)
990 GOTO 1070
1000 !
1010 CALL CHAR(139,"3C7EFFFFFFFF7E3C")
1020 CALL COLOR(14,12,16)
1030 GOTO 1070
1040 !
1050 CALL CHAR(140,"183C7EFFFF7E3C18")
1060 CALL COLOR(14,6,16)
1070 RANDOMIZE
1080 LEFT=INT(3*RND)+[
1090 IF COMB=51 THEN 1180
1100 B=PIC+135
1110 ON LEFT GOTO 1160,1150,1140,1130,1120
1120 CALL VCHAR(10,5,B)
1130 CALL VCHAR(6,5,B)
1140 CALL VCHAR(6,7,B)
1150 CALL VCHAR(8,7,B)
1160 CALL VCHAR(8,5,B)
1170 !
1180 ON LEFT GOTO 1200,1240,1330,1400,1460
1190 !
1200 CALL COLOR(13,6,6)
1210 CALL VCHAR(15,6,130,7)
1220 GOTO 1530
1230 !
1240 CALL COLOR(13,10,10)
1250 CALL VCHAR(16,\,130,[)
1260 CALL HCHAR(15,\,130,\)
1270 CALL VCHAR(16,7,130,_)
1280 CALL HCHAR(18,\,130,_)
1290 CALL VCHAR(19,\,130,_)
1300 CALL HCHAR(21,5,130,_)
1310 GOTO 1530
1320 !
1330 CALL COLOR(13,\,\)
1340 CALL HCHAR(15,\,130,\)
1350 CALL VCHAR(16,7,130,6)
1360 CALL HCHAR(18,5,130,3)
1370 CALL HCHAR(21,\,130,_)
1380 GOTO 1530
1390 !
1400 CALL COLOR(13,12,12)
1410 CALL VCHAR(15,\,130,\)
1420 CALL HCHAR(18,5,130,3)
1430 CALL VCHAR(15,7,130,7)
1440 GOTO 1530
1450 !
1460 CALL COLOR(13,8,8)
1470 CALL HCHAR(15,\,130,\)
1480 CALL VCHAR(16,\,130,[)
1490 CALL HCHAR(17,\,130,\)
1500 CALL VCHAR(18,7,130,\)
1510 CALL VCHAR(20,\,130,[)
1520 CALL HCHAR(21,\,130,_)
1530 IF MODE=50 THEN 1630
1540 !
1550 CALL COLOR(\,],16)
1560 CALL VCHAR(17,11,60,_)
1570 CALL HCHAR(18,10,60,_)
1580 IF COMB=51 THEN 1670
1590 CALL VCHAR(6,11,60,_)
1600 CALL HCHAR(7,10,60,_)
1610 GOTO 1670
1620 !
1630 CALL COLOR(\,],16)
1640 CALL HCHAR(18,10,60,_)
1650 IF COMB=51 THEN 1670
1660 CALL HCHAR(7,10,60,_)
1670 RANDOMIZE
1680 IF MODE=49 THEN 1710
```

```
1690 RIGHT=INT(LEFT*RND)+1
1700 GOTO 1740
1710 RIGHT=INT(5*RND)+1
1720 IF (RIGHT=5)*(LEFT=5)TH
EN 1730 ELSE 1740
1730 RIGHT=1
1740 IF COMB=51 THEN 1820
1750 ON RIGHT GOTO 1800,1790
,1700,1770,1760
1760 CALL VCHAR(10,15,8)
1770 CALL VCHAR(6,15,8)
1780 CALL VCHAR(6,17,8)
1790 CALL VCHAR(8,17,8)
1800 CALL VCHAR(8,15,8)

1810 !
1820 ON RIGHT GOTO 1840,1880
,1970,2040,2100

1830 !
1840 CALL COLOR(1,6,16)
1850 CALL VCHAR(15,16,34,7)
1860 GOTO 2180

1870 !
1880 CALL COLOR(1,10,16)
1890 CALL VCHAR(16,15,34,1)
1900 CALL HCHAR(15,15,34,1)
1910 CALL VCHAR(16,18,34,3)
1920 CALL HCHAR(18,15,34,1)
1930 CALL VCHAR(19,15,34,3)
1940 CALL HCHAR(21,15,34,1)
1950 GOTO 2180

1960 !
1970 CALL COLOR(1,1,16)
1980 CALL HCHAR(15,15,34,1)
1990 CALL VCHAR(16,18,34,6)
2000 CALL HCHAR(18,16,34,3)
2010 CALL HCHAR(21,15,34,_)
2020 GOTO 2180

2030 !
2040 CALL COLOR(1,12,16)
2050 CALL VCHAR(15,15,34,1)
2060 CALL HCHAR(18,16,34,3)
2070 CALL VCHAR(15,18,34,7)
2080 GOTO 2180

2090 !
2100 CALL COLOR(1,8,16)
2110 CALL HCHAR(15,15,34,1)
2120 CALL VCHAR(16,15,34,3)
2130 CALL HCHAR(17,15,34,1)
2140 CALL VCHAR(18,18,34,1)
2150 CALL VCHAR(20,15,34,3)
2160 CALL HCHAR(21,15,34,1)

2170 !
2180 CALL COLOR(12,3,16)
2190 CALL HCHAR(17,21,123,_)
2200 CALL HCHAR(19,21,123,_)
2210 IF COMB=51 THEN 2240
2220 CALL HCHAR(6,21,123,_)
2230 CALL HCHAR(8,21,123,_)
2240 IF MODE=49 THEN 2280
2250 ANSW=LEFT-RIGHT
2260 IF ANSW=0 THEN 2460
2270 GOTO 2300
2280 ANSW=LEFT+RIGHT
2290 QWL=0
2300 IF (COMB=50)+(COMB=51)T
HEN 2460
2310 IF QWL<>0 THEN 2340
2320 QWL=ANSW
2330 IF QWL=0 THEN 2920
2340 ON QWL GOTO 2440,2430,2
420,2410,2400,2390,2380,2370
,2360,2350
2350 CALL VCHAR(11,26,8)
2360 CALL VCHAR(9,30,8)
2370 CALL VCHAR(7,30,8)
2380 CALL VCHAR(5,30,8)
2390 CALL VCHAR(9,28,8)
2400 CALL VCHAR(9,26,8)
2410 CALL VCHAR(5,26,8)
2420 CALL VCHAR(5,28,8)
2430 CALL VCHAR(7,28,8)
2440 CALL VCHAR(7,26,8)
2450 IF COMB=50 THEN 2920
2460 CALL SOUND(300,1200,1)
2470 IF FL=1 THEN 3630

2480 !
2490 CALL KEY(0,QWL,STATUS)
2500 IF STATUS=0 THEN 2490
2510 IF STATUS=-1 THEN 2490
2520 IF QWL=77 THEN 460
2530 IF (QWL<48)+(QWL>57)THE
N 2490
2540 QWL=QWL-48
2550 RANDOMIZE
2560 FOR S=1 TO 10
2570 T=INT(3000*RND)+110
2580 CALL SOUND(5,T,1)
2590 NEXT S
2600 IF QWL=ANSW THEN 2860
2610 CALL SOUND(600,-1,_)
2620 GOTO 2900
2630 IF FL=1 THEN 4180
2640 FOR EC=26 TO 30
2650 IF (COMB=49)+(COMB=51)T
HEN 2670
2660 CALL VCHAR(1,EC,32,8)
2670 CALL VCHAR(15,EC,32,7)
2680 NEXT EC
2690 GOTO 2480
2700 U=U+1
2710 GOTO 4060

2720 V=V+1
2730 IF V=10 THEN 2780
2740 IF FL=1 THEN 4170
2750 FOR DELAY=1 TO 1000
2760 NEXT DELAY
2770 GOTO 750
2780 W=V-U

2790 CALL CLEAR
2800 PRINT "YOUR SCORE IS";W
2810 PRINT
2820 PRINT "OUT OF 10"
2830 FOR L=1 TO 1000
2840 NEXT L
2850 GOTO 460
2860 FOR Z=1 TO 6
2870 CALL SOUND(20,2000,_)
2880 CALL SOUND(20,1700,_)
2890 NEXT Z

2900 !
2910 IF COMB=50 THEN 2330
2920 ON QWL+1 GOTO 2930,3000
,3040,3130,3200,3260,3350,34
40,3500,3580
2930 CALL COLOR(1,15,16)
2940 CALL HCHAR(15,26,41,1)
2950 CALL VCHAR(16,26,41,6)
2960 CALL HCHAR(21,27,41,_)
2970 CALL VCHAR(16,29,41,5)
2980 GOTO 3630

2990 !
3000 CALL COLOR(1,6,16)
3010 CALL VCHAR(15,27,41,7)
3020 GOTO 3630

3030 !
3040 CALL COLOR(1,10,16)
3050 CALL VCHAR(16,26,41,1)
3060 CALL HCHAR(15,26,41,1)
3070 CALL VCHAR(16,29,41,_)
3080 CALL HCHAR(18,26,41,_)
3090 CALL VCHAR(19,26,41,_)
3100 CALL HCHAR(21,27,41,_)
3110 GOTO 3630

3120 !
3130 CALL COLOR(1,1,16)
3140 CALL HCHAR(15,26,41,1)
3150 CALL VCHAR(16,29,41,6)
3160 CALL HCHAR(18,27,41,1)
3170 CALL HCHAR(21,26,41,_)
3180 GOTO 3630

3190 !
3200 CALL COLOR(1,12,16)
3210 CALL VCHAR(15,26,41,1)
3220 CALL HCHAR(18,26,41,_)
3230 CALL VCHAR(15,29,41,7)
3240 GOTO 3630

3250 !
3260 CALL COLOR(1,8,16)
3270 CALL HCHAR(15,26,41,1)
3280 CALL VCHAR(16,26,41,1)
3290 CALL HCHAR(17,26,41,1)
3300 CALL VCHAR(18,29,41,1)
3310 CALL VCHAR(20,26,41,1)
3320 CALL HCHAR(21,26,41,_)
3330 GOTO 3630

3340 !
3350 CALL COLOR(1,10,16)
3360 CALL HCHAR(15,26,41,1)
3370 CALL VCHAR(16,29,41,1)
3380 CALL VCHAR(16,26,41,6)
3390 CALL HCHAR(18,27,41,_)
3400 CALL VCHAR(19,29,41,_)
3410 CALL HCHAR(21,27,41,1)
3420 GOTO 3630

3430 !
3440 CALL COLOR(1,9,16)
3450 CALL VCHAR(16,26,41,1)
3460 CALL HCHAR(15,26,41,1)
3470 CALL VCHAR(16,29,41,6)
3480 GOTO 3630

3490 !
3500 CALL COLOR(1,12,16)
3510 CALL HCHAR(15,26,41,1)
3520 CALL HCHAR(18,27,41,1)
3530 CALL HCHAR(21,26,41,1)
3540 CALL VCHAR(16,26,41,5)
3550 CALL VCHAR(16,29,41,5)
3560 GOTO 3630
```

```
3570 !
3580 CALL COLOR(1,5,16)
3590 CALL HCHAR(15,26,41,\)
3600 CALL VCHAR(16,26,41,1)
3610 CALL HCHAR(18,26,41,_)
3620 CALL VCHAR(16,29,41,6)
3630 IF ANSW=ONE THEN 3940 E
LSE 3840

3640 !
3650 CALL SAY("WHAT+IS")
3660 IF MODE=50 THEN 3740
3670 ON LEFT GOTO 3680,3690,
3700,3710,3720
3680 CALL SAY("ONE"):: GOTO
3730
3690 CALL SAY("TWO"):: GOTO
3730
3700 CALL SAY("THREE"):: GOT
O 3730
3710 CALL SAY("FOUR"):: GOTO
 3730
3720 CALL SAY("FIVE")
3730 IF MODE=49 THEN 3810 EL
SE 3830
3740 ON RIGHT GOTO 3750,3760
,3770,3780,3790
3750 CALL SAY("ONE"):: GOTO
3800
3760 CALL SAY("TWO"):: GOTO
3800
3770 CALL SAY("THREE"):: GOT
O 3800
3780 CALL SAY("FOUR"):: GOTO
 3800
3790 CALL SAY("FIVE")
3800 IF MODE=50 THEN 3820 EL
SE 3830
3810 CALL SAY("AND"):: GOTO
3740
3820 CALL SAY("FROM"):: GOTO
 3670
3830 GOTO 2480
3840 IF FL=1 THEN 3860
3850 GOTO 2630
3860 WRONG=INT(5*RND)+1
3870 ON WRONG GOTO 3880,3890
,3900,3910,3920
3880 CALL SAY("I+AM+SORRY"):
: GOTO 3930
3890 CALL SAY("UHOH, THAT+IS
+NOT+THE1+RIGHT ANSWER"):: G
OTO 3930
3900 CALL SAY("THAT+IS NOT R
IGHT"):: GOTO 3930
3910 CALL SAY("THAT IS IN+CO
RRECT"):: GOTO 3930
3920 CALL SAY("THAT IS+NOT+T
HE ANSWER")
3930 GOTO 2630
3940 IF FL<>1 THEN 4030
3950 CORR=INT(5*RND)+1
3960 ON CORR GOTO 3970,3980,
3990,4000,4010
3970 CALL SAY("THAT+IS+RIGHT
 YOU+HAVE+THE1+CORRECT ANSWE
R"):: GOTO 4020
3980 CALL SAY("GOOD WORK"):
 GOTO 4020
3990 CALL SAY("THAT IS CORRE
CT"):: GOTO 4020
4000 CALL SAY("VERY GOOD"):
 GOTO 4020
4010 CALL SAY("YOU+ARE EXACT
LY RIGHT")
4020 CALL SAY("THE ANSWER IS
 ")

4030 !
4040 IF FL<>1 THEN 2720
4050 ON ANSW+1 GOTO 4060,407
0,4080,4090,4100,4110,4120,4
130,4140,4150
4060 CALL SAY("ZERO"):: GOTO
 2720
4070 CALL SAY("ONE"):: GOTO
2720
4080 CALL SAY("TWO"):: GOTO
2720
4090 CALL SAY("THREE"):: GOT
O 2720
4100 CALL SAY("FOUR"):: GOTO
 2720
4110 CALL SAY("FIVE"):: GOTO
 2720
4120 CALL SAY("SIX"):: GOTO
2720
4130 CALL SAY("SEVEN"):: GOT
O 2720
4140 CALL SAY("EIGHT"):: GOT
O 2720
4150 CALL SAY("NINE"):: GOTO
 2720
4160 CALL SAY("TEN"):: GOTO
2720
4170 CALL SAY("NOW+TRY+THIS"
):: GOTO 750
4179 !@P+
4180 CALL SAY("TRY AGAIN"):
 GOTO 2640
```