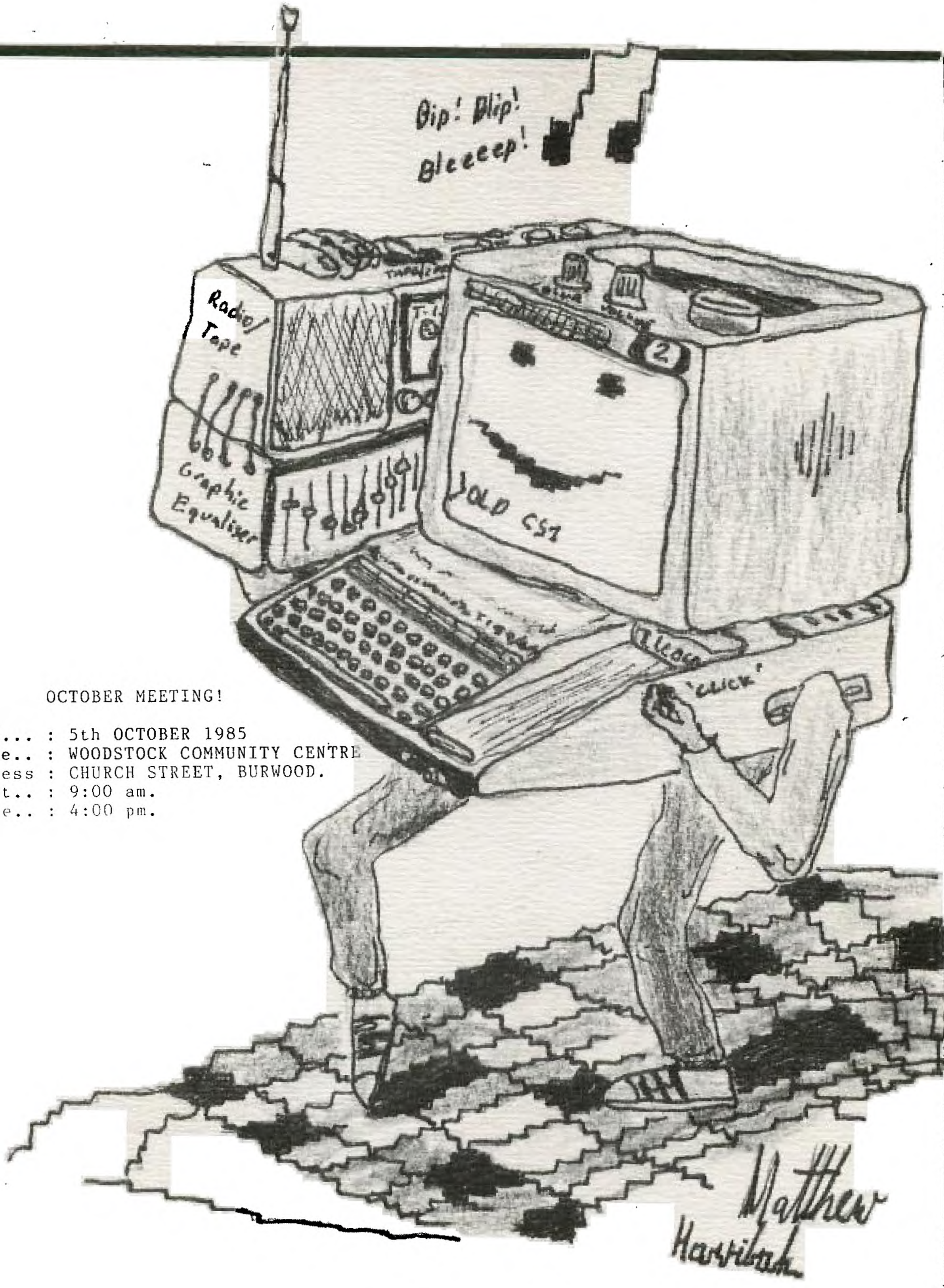
Cct TI shug

SYDNEY NEWS DIGEST



OCTOBER MEETING!

```
Date... : 5th OCTOBER 1985
Venue.. : WOODSTOCK COMMUNITY CENTRE
Address : CHURCH STREET, BURWOOD.
Start.. : 9:00 am.
Close.. : 4:00 pm.
```

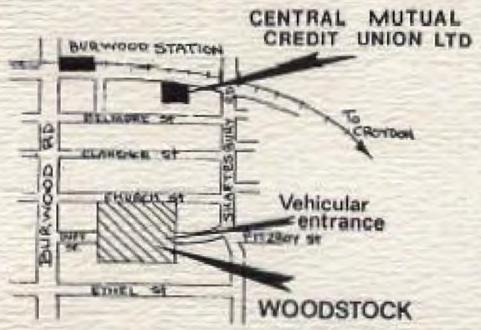
COMPUTER MUSIC !

TISHUG
PO BOX 149
PENNANT HILLS
N.S.W.
REGISTERED BY AUSTRALIA POST
PUBLICATION No NBH5933

OCT '85 $2

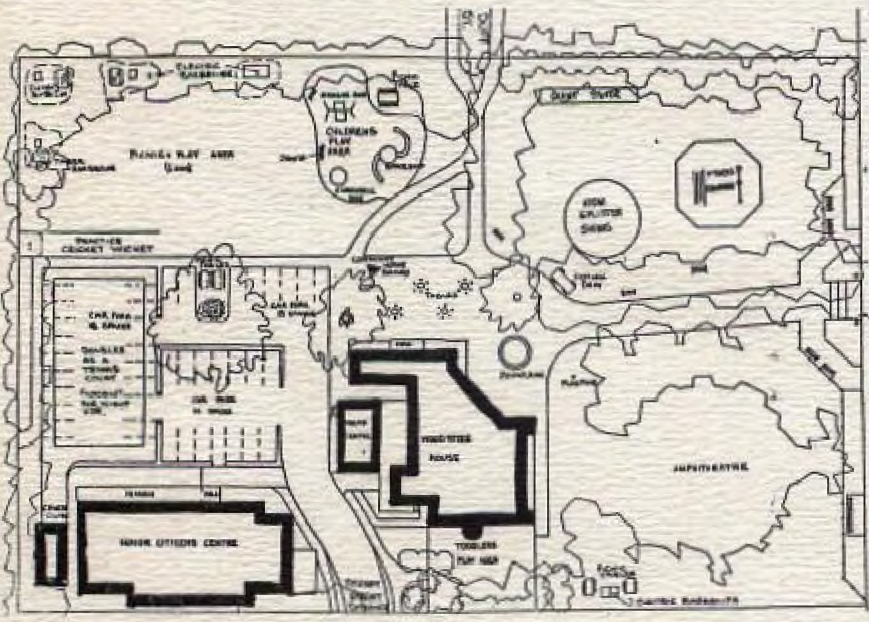
INVITATION TO A TEDDY BEARS PICNIC AND ALL DAY COMPUTER TUTORIAL-TUTORIAL-TUTORIAL

Please accept this as our personal invitation to you and your family,to visit WOODSTOCK on saturday the 5th of October 1985.Built in 1873 this property consisting of buildings and land of 2.4 HA is just a short walk from Burwood station,a real island of tranquility.



Where we will be taking over the whole ground floor area and consisting of a number of rooms which will be devided into separate tutorial classes for
BASIC
EXTENDED BASIC
EDITOR ASSEMBLES and many other topics of interest.

However for the younger children we have a wonderland of a PICNIC area including.Bar-be-cue facilities,swings,slides,spaceship and carousell.All in a totally safe clean enviroment



WOODSTOCK GROUNDS

We wil! be providing drinks and do nuts for all children bringing their teddy bears,plus special prizes for selected teddy bear catagories (ie,biggest,fattest,nicest etc) Please come and bring your picnic gear and goodies,you will be pleasantly supprised at WOODSTOCK and surroundings

SO COME ONE COME ALL

YOURS
PETER VARGA
CO-ORDINATOR

In an unprecedented move your TI.SHUG Co-Ordinating Committee resolved that this meeting should be held during the public holiday week-end so that our country members would have the opportunity, and, the time to make the journey to Sydney in order to attend.

The program for the day includes:-

1. Basic programming.
-Program planning and flowcharts.
-Graphic programming and control.
-Input/Output programming and control.
-Developing and using Subprograms
-and, Display techniques.

2. Assembly programming.
-Introduction to Assembly programming.
-Mixing X/B and Assembly.
-Advanced Assembly workshop.

3. Forth programming.
-Introduction to Forth.

4. Technical workshop.

5. Music programming.

6. Applications workshop.
- Multiplan.
- TI - Writer.
- Communications.

7. Program downloading.

8. Publications Library Use.

9. Shopping.

By spending the day with us we are sure that you will not only learn but have the chance to get to know your fellow TI Users a little better.

We will have the use of a number of computers so, if you have a problem or would just like to show us something that you believe may be of interest then please bring it along, recorded either on tape or disk.

DAY STARTS 9:00 AM SHARP WITH REGISTRATION.

THE LECTURES AND WORKSHOPS START 9:30AM - ALSO SHARP!

SOME OF THE HELPERS ARE:-
ROSS MUDIE,KEVIN GARDNER, SHANE FERRET,RUSSEL WELHAM, SHANE ANDERSEN, FRED MORRIS, TERRY JOHNSEN,TERRY PHILLIPS ROSS HARDY.

THE TI-99 MODEM/RS232 BOX

A mini review by Shane...

This is a Stand-alone expansion box made by TISHUG member Peter Schubert which has taken over a year to develop resulting in a well built and professional looking product that combines some of the features of two cards that are made for the P.E. expansion box. These are the RS232 Card and the MULTI-MODEM Card. The latter is also made by Peter.

However this is made as a stand-alone unit to eliminate the need for a P.E. box as many of us do not have this and it is no longer available.

This box has its own power lead and it fits against the side of the console. It measures 190x110mm and is 60mm high.

The RS232 socket at the back can be used as a standard RS232 port for example to connect to a printer (if it is an RS232 printer). If you wish to use a Centronics type printer a small attachment will be made that will convert the socket to a PIO port. This will be sold as an accessory.

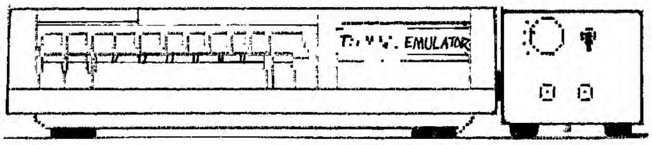
As a MODEM it has the following modes;
300 Answer
300 Originate
VIATEL (75/1200)
1200/75

The last mode means that it would be possible with suitable terminal emulator program to send at 1200 baud to another similarly equipped TI.

Two lamps show READY FOR SENDING(green) and CARRIER DETECTED(red). A standard Telecom plug is provided and Telecom Permit has been applied for. The line cord is wired for use with a Y connector so that any Phone can be used. I use a phone with 10 number memory so that I can store the most often needed numbers of BBS services which is so handy when the one you are after is constantly engaged.

For 300 baud communicating as used by our TEXPAC BBS you will need the TE2 MODULE. All of its features are available with this unit.

To use this unit to log onto a BBS is very simple. First you select OPTION 2 on the TE2 which gives you a parameter selection screen.The default options can be accepted on all but the PORT which must be changed from port 1 to PORT 2. You then hit enter till you pass the bottom of this screen and then you will get a square cursor at the top of a blank screen. Then you dial the number and when you hear the carrier tone you switch to DATA on the modem. This will cause the carrier light to come on and then you hit enter a few times until it responds and commence to log on.



For VIATEL access at this stage you will need a MINIMEM MODULE and the Viatel software available from Robert Craigo on Cassette.This can be ordered from Robert once you have received you Viatel Membership from Telecom by writing to the following address and sending your payment of $25 + $1.50 postage.

Robert Craigo
100 Victoria Road
West Pennant Hills. 2120

You must also include your Viatel Customer ID number as this is inserted into the program. Note however that this program is not in colour. The colour version which is available on disc will not load to cassette as it needs too much memory.

If you have the Corcomp Expansion with disc drive then you can of course use the colour Viatel program with this stand-alone modem. However you must ask Robert to alter the program to use Port 2.

I had no problems loading the Viatel program from cassette as it is all Assembly using option 1 on the MM. Entering the program name gives a nice VIDEOTEX title screen which tells you to dial 01955. Then when you hear the tone you switch to DATA and wait. The program will then automatically send your 10 digit ID number and you will get a Viatel title screen that asks you to enter your personal password. If you are a new member I would suggest you change this password as soon as possible for your security.

So here you have an accessory for your TI of real quality. I have logged on to various bulletin boards and always received error-free reception and if by chance a problem did occur with the unit I am assured by Peter that he will check it out and repair if neccessary free of charge so he is obviously confident of its reliability.

Peter also tells me the 32K BOX he also makes can be used in conjunction with this unit. With MM that would sure make a powerful combination. Perhaps not to be compared to a full disc system but then its not intended to compete against this but to offer a viable low-cost alternative.

The Modem/RS232 Box can be ordered now until the end of October for $280 as a special price to TI user group members. The normal price will then be $320. It can be ordered by phoneing Peter on (02)358 5602 AH or writing to:-

Peter Schubert
C/-TI.S.H.U.G.
P.O. Box 595
Marrickville. 2204

Next month I hope to have news of a powerful WORD PROCESSOR that can be built up of accessories to provide full screen editing 80 columns with 32K memory storage and printer port for less than $450 onto your basic console and cassette. Or if you added this Modem box for only an extra $120 approx plus printer.

```
100 REM ******************
110 REM ****  ROADER  ****
120 REM ****          ****
130 REM *BY  JUSTIN TWISS*
140 REM ****   TIUP   ****
150 REM ***DATE WRITTEN***
160 REM ****          ****
170 REM **** 13/9/84  ****
180 REM ******************
190 F=12
200 GOTO 290
210 FOR VOL=1 TO 30 STEP 10
220 CALL SOUND(-1000,-7,VOL)
230 CALL SCREEN(INT(VOL/2.5)
+1)
240 F=19-F
250 CALL COLOR(9,F,1)
260 NEXT VOL
270 CALL COLOR(9,7,1)
280 RETURN
290 CALL CHAR(99,"223E2A0808
2A3E2A")
300 CALL CHAR(37,"FFFFFFFFFF
FFFFFF")
310 CALL CHAR(100,"447C54100
A2E3F7F")
320 CALL CHAR(101,"00080C1E1
E3E3F7F")
330 CALL CHAR(104,"001818181
83C3C00")
340 CALL CHAR(120,"FFFFFFFFF
FFFFFFF")
350 CALL CHAR(128,"00FF00FF0
0FF00FF")
360 CALL COLOR(9,5,1)
370 CALL COLOR(10,6,1)
380 CALL COLOR(11,9,1)
390 CALL COLOR(12,2,1)
400 CALL COLOR(13,9,2)
410 CALL CLEAR
420 CALL SCREEN(15)
430 PRINT
440 PRINT TAB(8):"* ROADER *
"
450 FOR T=1 TO 6
460 PRINT
470 NEXT T
480 PRINT "STEER WITH THE <
AND > KEYS"
490 PRINT
500 PRINT "OF JOYSTICK '1'"
510 FOR T=1 TO 5
520 PRINT
530 NEXT T
540 FOR I=110 TO 120 STEP .5
550 CALL SOUND(-150,I,2)
560 NEXT I
570 FOR I=120 TO 110 STEP -.
3
580 CALL SOUND(-150,I,2)
590 NEXT I
600 FOR T=110 TO 120 STEP .8
610 CALL SOUND(-75,T,2)
620 NEXT T
630 GOSUB 210
640 CALL SCREEN(15)
650 CALL COLOR(9,5,1)
660 FOR T=1 TO 250
670 NEXT T
680 CALL CLEAR
690 CALL COLOR(9,8,2)
700 OSKILL=1
710 PRINT TAB(2):"ENTER YOUR
 SKILL LEVEL...."
720 FOR T=1 TO 3
730 PRINT
740 NEXT T
750 PRINT TAB(4);"ENTER :"
760 PRINT
770 PRINT
780 PRINT TAB(8);"1) NOVICE.
"
```

(CONTINUED NEXT PAGE)

TECHO TIME

WITH ROBERT

As promised in the AUGUST News Digest I am giving the details of the tr CENTRONICS interface. You will see that there are quite rigid (relatively speaking) timing and handshaking requirements.

Now to answer some of Barry Ridgeway's questions. To operate a modem you require some form of software. This software can be in the form of the TE II of an assembly language based programme. To communicate on 300 BAUD the TE II is suitable. For VIATEL an assembly language programme is required.
The method mentioned regarding quieting the PEB fan does so, but in so doing compromises the overall cooling efficiency to reduce noise. You could try to use a pot but it would have to be of a wire wound design causing it to be rather expensive. Another method of slowing down the fan would be to use a suitably A.C. rated non polarised capacitor (of approx. 1-2 uF) in series with the fan. In all cases cooling is affected and to do so is up to YOU. Finally commercial units are available for converting 12 V DC to 240 V AC. These units usualy very expensive and not all that efficient. One company to try would be WARBURTON FRANKI.

Into RTTY/HAM/AMATEUR interfacing for the TI ? Then contact Peter Shelling 42 Balmoral Dr. Gorakan 2203. VK2CPS.

PLEASE NOTE. If you have a second RS 232 card and try to set it up as RS 232/3 and 4 it will still respond to RS 232/1 and 2 unless the first card is fitted. So to test it out you MUST have the first card in the PEB.

Technical manuals (which include circuits of the computer and expansion system) are available for $15 at club meetings or from the club Librarian (please add $3.50 postage & handling).

If anyone has a technical problem or wishes to contribute technical articles then contact me (ROBERT) on - 602-4168 between 5.00-8.30 PM (PLEASE STICK TO THESE TIMES) or send a letter to

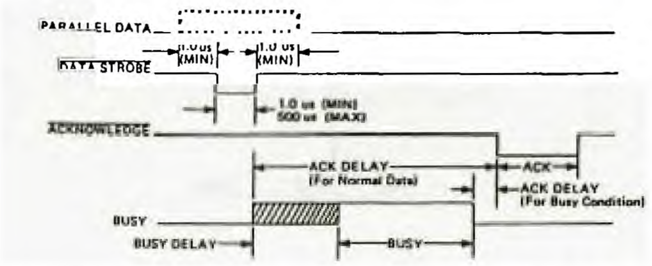
TECHO TIME
P.O. BOX 595
MARRACKVILLE 2204

ANYONE WISHING TO OBTAIN INFORMATION, PLEASE SEND A SELF-ADDRESSED ENVELOPE (WITH SUFFICIENT POSTAGE STAMPS) TO THE ABOVE ADDRESS.

INTERFACE TIMING

In general, the parallel data transfer sequence consists of the input device placing the appropriate code on the data lines to the printer and then generating a data strobe pulse. The printer, after a slight delay, responds with an acknowledge pulse, or if the received data causes a busy condition, the printer first activates the busy line for the duration of the busy condition and then responds with an acknowledge pulse.

The timing involved in normal data transfer and the transfer of data which causes a busy condition is as follows:



NORMAL DATA INPUT TIMING

ACK DELAY = 2.5-10.0 usec
ACK = 2.5-5.0 usec

BUSY CONDITION TIMING

BUSY DELAY = 0-1.5 usec
ACK DELAY = 0-10.0 usec
ACK = 2.5-5.0 usec

A busy indication (buffer full line from the printer) occurs only during the following conditions:

	Busy Duration
DELETE	100-400 usec
DESELECT	Until printer is selected.
FAULT	Until fault is cleared.
BUFFER FULL*	Until buffer reaches its lower limit.

```
790 PRINT
800 PRINT TAB(8);"2) PROFESS
IONAL."
810 PRINT
820 PRINT TAB(8);"3) EXPERT.
"
830 PRINT
840 PRINT TAB(8);"4) PERFECT
."
850 FOR D=1 TO 3
860 PRINT
870 NEXT D
880 CALL KEY(1,K,S)
890 RANDOMIZE
900 IF S=0 THEN 880
910 IF K<>10 THEN 940
920 SKILL=OSKILL
930 GOTO 960
940 SKILL=-(K=19)-(K=7)*2-(K
=8)*3-(K=9)*4
950 IF SKILL=0 THEN 880
960 OSKILL=SKILL
970 PRINT
980 PRINT TAB(8);"HERE WE GO
!!!"
990 PRINT
1000 PRINT
1010 FOR T=1 TO 400
1020 NEXT T
1030 CALL CLEAR
1040 CALL SCREEN(3)
1050 I=.1
1060 IF (SKILL<>2)*(SKILL<>4
)THEN 1080
1070 I=.2
1080 N=24
1090 J=0
1100 OLDN=24
1110 B$="%%%%%%%"
1120 FOR C=1 TO 4
1130 PRINT TAB(18);"h";B$;"h
"
1140 NEXT C
1150 FOR Q=1 TO 4
1160 FOR C=9 TO 6 STEP -1
1170 IF C<>0 THEN 1190
1180 B$="%%%%%%%"
1190 FOR A=0 TO 6.25 STEP I
1200 Y=COS(A)
1210 J=J+1
1220 PRINT TAB(8*Y+10);"h";B
$;"h"
1230 IF (RND>.5)+(SKILL=1)+(
SKILL=2)THEN 1260
1240 IF J>25 THEN 1260
1250 CALL HCHAR(23,28*RND+2,
128)
1260 CALL GCHAR(20,N,G)
1270 CALL HCHAR(19,OLDN,120)
1280 IF (G=104)+(G=128)+(G=3
2)THEN 1500
1290 CALL HCHAR(20,N,99)
1300 OLDN=N
1310 CALL KEY(0,K,S)
1320 IF S<>0 THEN 1340
1330 CALL JOYST(1,XR,YR)
1340 N=N+(K=44)-(K=46)+XR/4
1350 NEXT A
1360 B$=SEG$(B$,1,C-2)
1370 FOR D=110 TO 129-C STEP
 .5
1380 CALL SOUND(-150,D,2)
1390 NEXT D
1400 NEXT C
1410 NEXT Q
1420 CALL CLEAR
1430 CALL SCREEN(11)
1440 PRINT TAB(5);"YOU MADE
IT, MARIO !!"
1450 FOR T=1 TO 10
1460 PRINT
1470 NEXT T
```

(CONTINUED NEXT PAGE)

*Both buffer full and buffer empty limits are customer selectable.

All standard Centronics printers are supplied with an Amphenol No. 5740360, 36-pin interface connector (Centronics No. 31310019). The pin assignments, name, source and description for each interface signal are listed below. Active low signals are specified by a line over the signal name. Active high signals have no line.

Parallel Interface Connector	Signal Name	Source	Description
Pin 1,19*	$\overline{\text{DATA STROBE}}$	Input Device	A 1.0 usec pulse (min.) used to clock data from the processor to the printer logic.
2,20 3,21 4,22 5,23 6,24 7,25 8,26 9,27	DATA 1 DATA 2 DATA 3 DATA 4 DATA 5 DATA 6 DATA 7 DATA 8	Input Device Input Device Input Device Input Device Input Device Input Device Input Device Input Device	Input data levels. A high represents a binary ONE, a low represents a ZERO. All printable characters (i.e., codes having a ONE in DATA 6 or DATA 7) are stored in the printer buffer. Control characters (i.e., codes having a ZERO in both DATA 6 and DATA 7), are used to specify special control functions. These codes are not stored in the buffer except when they specify a print command and are preceded by at least one printable character in that line.
10,28	$\overline{\text{ACKNLG}}$	Printer	Acknowledge pulse indicates the input of a character into memory or the end of a functional operation.
11,29	BUSY	Printer	A level indicating that the printer cannot receive data. For conditions causing BUSY, refer to preceding page.
12	PE	Printer	A level indicating that the printer is out of paper.
13	$\overline{\text{SLCT}}$	Printer	A level indicating that the printer is selected.
14	±OV	Printer	Signal Ground
15	OSCXT	Printer	100-200 KHz signal
16	±OV		Signal ground
17	Chassis Gnd		Frame ground
18	+5V		+5 Volt power buss
31,30	$\overline{\text{INPUT PRIME}}$	Input Device	A level which causes the printer to be primed.

*Second pin number indicates twisted pair return (±OV).

Parallel Interface Connector	Signal Name	Source	Description
32	$\overline{\text{FAULT}}$	Printer	A level that indicates a paper empty, light detect, or a deselect condition.
34	Line Count Pulse		Both sides of the line count switch appear at the interface connector. This switch is opened and closed during each line feed operation. A level delivered to the switch would be pulsed off and on each time a line feed operation is performed.
35	Line Count Pulse Return		Return line.
36	Not Used		

DATA INPUT

Inputs to the printer consist of seven standard parallel data lines (DATA 1 - DATA 7) and an active low data strobe (DSTA (N)). The seven data lines represent the 7-bit USASCII code (refer to the operators manual). The data strobe is used to synchronize the input data with the printer electronics.

In response to received data, the printer generates an active low acknowledge (ACK) pulse to acknowledge reception of a character. If the received character caused the printer to perform some function such as paper

```
1480 GOTO 1590
1490 CALL HCHAR(20-1,N,101)
1500 CALL HCHAR(20,N,100)
1510 GOSUB 210
1520 CALL SCREEN(3)
1530 FOR T=1 TO 500
1540 NEXT T
1550 CALL CLEAR
1560 CALL COLOR(9,8,2)
1570 GOTO 1620
1580 FOR T=1 TO 500
1590 NEXT T
1600 CALL CLEAR
1610 CALL SCREEN(15)
1620 PRINT TAB(20);"YOUR SCO
RE IS";JESKILL
1630 FOR T=1 TO 5
1640 PRINT
1650 NEXT T
1660 PRINT TAB(9);"PLAY AGAI
N ?"
1670 PRINT
1680 PRINT
1690 PRINT TAB(8);"<FIRE BUT
TON> OR <S>"
1700 PRINT TAB(8);"- FOR SAM
E LEVEL"
1710 PRINT
1720 PRINT TAB(4);"<C> TO CH
ANGE LEVELS"
1730 PRINT
1740 PRINT TAB(4);"<E> TO EN
D PLAY"
1750 CALL KEY(1,K,S)
1760 IF S=0 THEN 1750
1770 IF (K=18)+(K=2)THEN 103
0
1780 IF (K<>5)*(K<>14)THEN 1
750
1790 IF K=14 THEN 680
1800 END
```

movement, character printing, etc., the printer responds with a BUSY signal and does not acknowledge until termination of the busy condition.

DATA INPUT TIMING

In general, the data transfer sequence consists of the input device placing the appropriate code on the data lines to the printer and then generating a data strobe pulse. The printer, responds with an acknowledge pulse, or if the received data caused a busy condition, the printer first activates the busy line for the duration of the busy condition and then responds with an acknowledge pulse.

Refer to Appendix for the timing involved in transferring data, which does not cause a busy condition, and the timing involved in receiving any character, which causes a busy condition in the printer.

Secretary's Notebook
with JR

Hi! First a summary of the discussions at the September Committee Meeting. Attending these meetings as a visitor can be a hazard... ask Keir Wells, he was appointed to the committee to replace Greg Hope and take over the responsibilities of Advertising Manager. We need a member with a TI Professional computer to join the committee to ensure that the interests of PC owners are properly looked after. If you are such a person please leave a message on the answering service 848-0956. The reprinting of the constitution has been delayed pending the completion of the Incorporation of the club, and drafting of the code of conduct for software. The monthly software entries were distributed for adjudication. Plans for the all day tutorial to be held at Woodstock, in October were formulated. We decided to hold this event during the holiday weekend to give our country members a chance to attend. This event has the potential of being our best Tutorial to date. That's the end of the summary except to say we would like to see more members at these committee meetings, which are held at Woodstock, Burwood on the third Tuesday of every month except January, commencing at 6.30 pm.

FREEWARE... does this name mean anything to you? Actually coined in the USA it is defined as a marketing technique used by a software author to distribute his/her product. It works like this: the program is produced and sent to organisations such as User Groups around the world. A notice on the title states that the program may be copied and given to anyone. Now how does the author get paid? The documentation for the software is provided in the form of text files on the disk. This information can be dumped on to your printer to obtain a hard copy. On reading the documentation you will find a notice saying if you like the program and you intend to use it then send a payment to the author. The payment expected is usually around $10 - $20. So you can see that the more it is copied the more chances the program will be used by people, who recognise that talent should be rewarded and will send their money to the author. All of us must realise that unless programmers receive some monetary incentive, they will stop producing software for our computers. So if you come across some of this software don't forget to make out that cheque or purchase a bank draft. There is no such thing as a free lunch....

Received in the mail in early September an invitation to the club to participate at the PC86- The Sixth Australian Personal Computer Show to be held at Centrepoint March 12-15 1986. Christmas can't be far away!

Running out of memory...,

Happy Computing the TI way,

John Robinson.

Hon. Secretary.

PROGRAM CHALLENGE by Ed York.

From Cin-Day User Group

I have received numerous enquiries for either a program or a method that will allow you to list a program to the printer in the maximum number of characters allowed using condensed printing. I believe that the standard range for most printers, when printing is condensed, is somewhere between 132 and 136 characters per line. I know that it is not possible to permanently change the built in default of 80 characters per line,but you can use a basic, extended basic, assembly language or forth program to achieve the desired results. I understand that there is an assembly language program available on Compuserve, that was written by Mac McCormic that will allow you to list your program to the printer, in 132 characters per line. I, therefore, issue a challenge to anyone who wishes to accept it, to write an equivalent program in extended basic. I wonder who will accept it? How about it Jim Peterson of Tigercub Software.

NEW PRODUCTS by Ed York.

I would like to announce that Tarik Isani, owner of Starsoft, now has seven fantastic assembly language utilities available for the TI 99/4A. i have had the opportunity to try each of the programs and I will state personally that the programs are excellent. I understand that you may order these products directly from Starsoft at the following address:
STARSOFT / 601 ALLEGHANY STREET /
BLACKBURG, VA / 24060 / (703) 953
1490.

I understand that Starsoft will accept either VISA or MASTERCARD. Should you order by credit card, money order or certified cheque, your order will be shipped on the next buisness day. However, should you order by personal cheque, please allow two weeks for the cheque to be cleared. I also understand that there is no shipping and handling charges on Starsoft orders. I have listed below a brief description of the programs being offered by Starsoft.

TI WRITER LOADER by Tarik Isani.

The TI WRITER LOADER is a menu driven assembly language program that allows you to load and execute the TI WRITER text editor, text formatter and utility programs, without the TI WRITER cartridge. All of the text editor and formatter commands , except the "show directory" (SD), are fully functional. The loader also features a custom Starsoft-designed true lowercase character set, to make TI WRITER text more readable. The updated public domain TI WRITER text editor (EDITA1 and EDITA2) and text formatter (FORMA1 and FORMA2) files are also included on the disk at no extra charge, making this package a complete word processing system. NOTE: There are no utility programs included. The system requirements are:- 32K memory expansion, 1 or more disk drives, and either the EXTENDED BASIC, MINI-MEMORY or EDITOR/ASSEMBLER cartridge. TI WRITER LOADER is available only on disk, and retails for $19.95

TE-PLUS by Tarik Isani.

TE-PLUS is a high speed terminal emulator written in assembly language. Full menu driven protocol setup is implemented, including port number, baud rate (110-9600), parity, stop bits, data bits and full or half duplex (local echo). Further setup is provided for screen and text colours, file system record length, and automatic linefeed. Incoming text may be saved (downloaded) in a 24k circular buffer, that can be dumped to disk, printer or any other device, with user selected record length (1-254) and optional filtering of linefeed and other control characters. TE PLUS also has the ability to upload text files from disk drive to an RS232 port, and to simultaneously send incoming text to a printer, 80 column card, or other peripheral. In addition TE PLUS has a true lower case character set built in. The system requirements are:- 32K memory expansion, 1 or more disk drives, RS232 interface with modem, and either the EXTENDED BASIC, MINI-MEMORY, or EDITOR/ASSEMBLER cartridge. TE PLUS is available only on disk and retails for $19.95

NIBBLER by Tarik Isani

NIBBLER is the fastest 1 or 2 drive sector-by-sector disk copier on the market. This assembly language program is capable of copying all sectors of a disk, or for faster copying, only the "used" sectors. When copying NIBBLER can write to different sectors than the ones being copied from the source disk. NIBBLER reads, writes and formats ranging from single sided/single density to double sided/double density, depending on the capabilities of your disk controller card. In addition to copying all the disks that the DISK MANAGER can copy, NIBBLER can create backups of protected disks. Single sided/single density backups are created in at most three passes, and double sided/double density backups in at most six passes (for full disks). This certainly makes NIBBLER ideal for double, as well as single drive backups. The system requirements are:- 32K memory expansion, 1 or 2 disk drives, and either the EXTENDED BASIC, MINI-MEMORY or EDITOR/ASSEMBLER cartridge. NIBBLER is available only on disk and retails for $19.95

STARSOFT DISASSEMBLER by Tarik Isani.

The STARSOFT DISASSEMBLER is an extremely fast assembly language program capable of disassembling 4K of memory to disk in 1 minute and 20 seconds. The program can disassemble in text, data, and mnemonic instruction format. All three formats are compatible with the TI editor/assembler cartridge. The disassembly can be output to the screen, printer, disk drive or any other peripheral. Optionally, relocatable labels may be output with the disassembly. The disassembler may be used to disassemble system ROM routines, as well as programs loaded from disk.The disassembler is a must for debugging and developing TMS 9900 assembly language programs. The system requirements are:- 32K memory expansion, 1 or 2 disk drives, and the EDITOR/ASSEMBLER cartridge. The STARSOFT DISASSEMBLER is available only on disk and retails for $19.95

DISK TOR by Tarik Isani.

The DISK EDITOR is an assembly language program that allows the user to edit a disk or individual sectors rather than by filename. Individual bytes within any sector of a disk may be displayed and edited. Editing is through a full-screen editor, that displays the contents of a sector on the screen and allows editing in both hexadecimal, as well as, ASCII character formats. The DISK EDITOR may also be used to move data from one sector to another and to attempt to fix damaged disk directories. The system requirements are:- 32K memory expansion, 1 or 2 disk drives, and either the EXTENDED BASIC, MINI-MEMORY, or EDITOR/ASSEMBLER cartridge. DISK EDITOR is available only on disk and retails for $19.95

UNPROTECTOR by Tarik Isani.

The UNPROTECTOR is an assembly language program that allows the user to "unprotect" protected extended basic programs. Programs are "protected" if an attempted list, edit or save command produces the "protection violation" error. UNPROTECTOR will "unprotect" these programs after they have been loaded into memory from tape or disk. After that, the "unprotected" program may be listed, edited, and saved to tape or disk. The UNPROTECTOR may be used to recover accidently protected programs, to backup purchased extended basic programs that are protected, and to transfer protected programs from tape to disk. The system requirements are:- 32K memory expansion, 1 disk drive, and the EXTENDED BASIC cartridge. UNPROTECTOR is available only on disk and retails for $14.95

→

TI 99/4A CONSOLE HARDWARE.

by Henry Clark. TI-TIMES Magazine No 6 1984) Being an inquisitive sort of person, I like to know how things work. So soon after acquiring a TI 99/4A I set about trying to discover what made the machine work. By a process of trial and error, I worked out the joystick and cassette port pin connections, but was rather daunted by the 44 connections to the expansion port. More in hope than in expectation I rang TI at Bedford. I was pleasantly surprised to find that there is a manual available called "TI 99/4A Console and Peripheral Expansion System, Technical Data" for UK8.97 including p&p. from TI, P.O. Box 50, Market Harborough, Leics. This manual gives a wealth of information about the internal operation of the console, the expansion system and design guidelines for using the peripheral expansion port. Having read through it a few times, I realised how little I knew about microprocessor techniques, despite having used and designed industrial electronic equipment for several years. I felt, however, that many TI*MES readers would like to know a bit about their computers, so I fool- ishly volunteered to write something about the TI 99/4A hardware. So, here goes ...... THE CONSOLE ----------- Before we look at the insides of the console, here are the pin connections for the external connections. Cassette Port ------------- Pin Function 1. CS1 Control +ve 2. CS1 Control -ve 3. Data out, screen 4. Audio gate (not used) 5. Data out 6. CS2 Control +ve 7. CS2 Control -ve 8. Data in 9. Data in, screen Joystick Port ------------- Pin Function. 1. Not connected. 2. Joyst 2 common. 3. Up. 4. Fire. 5. Left. 6. Not connected. 7. Joyst 1 common. 8. Down. 9. Right. The joystick contacts are diode isolated to prevent interaction giving odd results. Remember, the Alpha Lock button must be up for the joystick to work correctly. The fire button is scanned using a call key command where it returns a value of 18 with key units 1 and 2 for joysticks 1 and 2 respect- ively. Video Port ---------- The manual shows a TMS9918A as the video display processor, whereas the European TI 99/4A uses a TMS9929A; this is virtually the same except for the colour video section which is suitable for the PAL modulator. The connections are believed to be as follows:- Pin Function 1. 12V +ve power 2. red-yellow 3. sound 4. yellow 5. blue-yellow 6. common Power Supply Plug ----------------- Pin Function 1. 16V AC 2. Common 3. 8V AC 4. Not connected Expansion Port --------------- Pin Notaion I/O Description 1 +5V 0 50mA max for speech synthesiser 2 SBE 0 Speech Block Enable 3 *RESET 0 Master reset, low active. 4 *EXTINT I External interrupt low active 5 A5 0 Address bit 5 6 A10 0 Address bit 10 7 A4 0 Address bit 4 8 A11 0 Address bit 11 9 DBIN 0 Data Bus In, buffers & 9900 in input mode 10 A3 0 Address bit 3 11 A12 0 Address bit 12 12 READY I External memory ready 13 *LOAD I To 9900 Load pin 14 A8 0 Address bit 8 15 A13 0 Address bit 13 16 A14 0 Address bit 14 17 A7 0 Address bit 7 18 A9 0 Address bit 9 19 A15 0 Address bit 15 (LSB), CRU output

20	A2	0	Address bit 2
21	GND	0	Signal ground, OV
22	*CRUCLK	0	CRU clock-inverted
23	GND	0	Signal ground OV
24	*03	0	Clock phase 3 (inverted)
25	GND	0	Signal ground OV
26	*WE	0	Write Enable
27	GND	0	Signal ground
28	*MBE	0	Memory Block Enable
29	A6	0	Address bit 6
30	A1	0	Address bit 1
31	A0	0	Address bit 0(MSB)
32	*MEMEN	0	Memory Enable
33	CRUIN	I	CRU data in, input data line
34	D7	I/O	Data bus line 7, (LSB)
35	D4	I/O	Data bus line 4
36	D6	I/O	Data bus line 6
37	D0	I/O	Data bus line 0 (MSB)
38	D5	I/O	Data bus line 5
39	D2	I/O	Data bus line 2
40	D1	I/O	Data bus line 1
41	IAQ	0	Instruction Acquisition, CPU is acquiring an instruction.
42	D3	I/O	Data bus line 3
43	-5V	0	50mA max for speech synthesiser
44	AUDIO IN	I	Audio input from speech synthesiser

Notes
-----

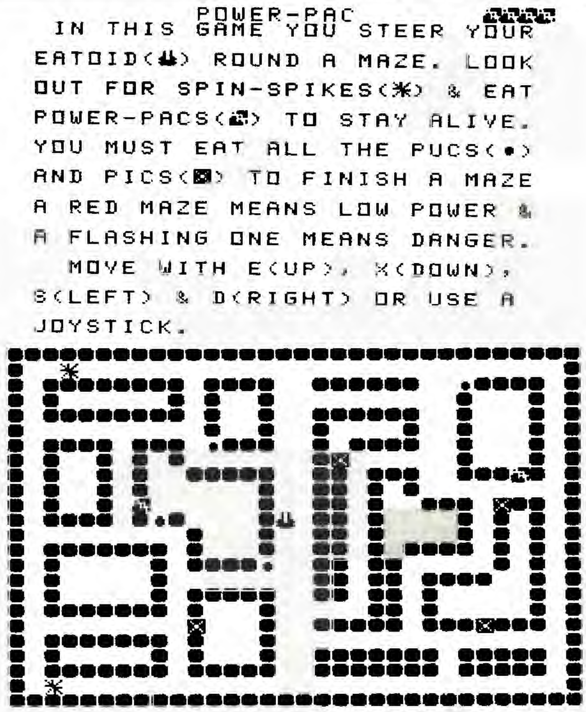
1. A function marked "*" indicates that this is active in the low state, 0. 2. I indicates an input to the console. 0 indicates an output. I/O indicates a bidirectional line. 3. Pins 1 and 43 are not intended for use by anything but the speech synthesiser. Overloading may blow up the console power supply. 4. Beware of connecting anything to the expansion port unless you are sure that it is designed to work with the TI 99/4A. Serious, ie. expensive damage could result. Peripheral Expansion Philosophy. ------------------------------- The expansion port gives access to the 16 bit address bus, the 8 bit data bus, bus control, memory control, timing and control signals, and speech module signals. Any peripheral connected to the console must have a device service routine (DSR) ROM, in order to be recognised by the TI 99/4A. On start up, either software or hardware, the TI 99/4A interrogates the I/O port for the presence of peripherals. The DSR of any peripheral will respond giving information about the nature of the device and the relevant communications protocol. This system has the advantage that types of peripherals are not restricted by console hardware limitations. The disadvantage is that it makes it difficult for amateurs like me to make peripherals, as I cannot write the DSRs or program the ROMs which would contain them, without learning 9900 assembly language and getting access to a PROM programmer. In future issues I hope to describe more about the operation of the consoles 40 odd integrated circuits, plus, if possible, information about how to build your own low cost peripherals. Meanwhile, keep on supporting your TI 99/4A, it is a good little computer which deserves a better fate than TI felt forced to give it.

→

MICROKEY by Tarik Isani.

MICROKEY is an interrupt-driven assembly langusge program that runs concurrently with basic or extended basic, allowing the user to define 10 independent function keys. Commonly used commands, up to 28 characters each, may be assigned to the keys <CTRL>-0 through <CTRL>-9. When one of the function keys is pressed, the corresponding command is displayed on the screen, followed by an automatic <ENTER>, just as if the command had been typed from the keyboard, followed by a depression of the <ENTER> key. For example, if the string "CALL CLEAR" had been assigned to function key 1, then whenever the user holds down the control key(<CTRL>), and presses "1", the assigned string is displayed on the screen and the screen clears. The definitions of the function keys may be displayed on the screen, loaded into memory and saved to disk. Any legal commands , or other strings, may be assigned to the function keys, thereby simplifying the basic programming process. The system requirements are:- 32K memory expansion, 1 or 2 disk drives, and either the EXTENDED BASIC, MINI-MEMORY, or EDITOR/ASSEMBLER cartridge. MICROKEY is available only on disk and retails for $19.95

==================



```
         POWER-PAC       ▨▨▨▨
 IN THIS GAME YOU STEER YOUR
EATOID(⯊) ROUND A MAZE. LOOK
OUT FOR SPIN-SPIKES(✱) & EAT
POWER-PACS(▨) TO STAY ALIVE.
YOU MUST EAT ALL THE PUCS(•)
AND PICS(☒) TO FINISH A MAZE
A RED MAZE MEANS LOW POWER &
A FLASHING ONE MEANS DANGER.
  MOVE WITH E(UP), X(DOWN),
S(LEFT) & D(RIGHT) OR USE A
JOYSTICK.
```

```
100 REM *********************
110 REM   P O W E R - P A C
120 REM     by Toby Grays
130 REM  TI.S.H.U.G.Under'18
140 REM  YOUNGER SET MEMBER
150 REM  (requires Ex/Basic)
160 REM       October'85
170 REM *********************
180 CALL CLEAR :: CALL SCREE
N(12):: RANDOMIZE :: DIM SX(
19),SY(19),R$(5)
190 FOR I=1 TO 5 :: READ R$(
I):: NEXT I
200 DATA "MAZE-NOVICE","MAZE
-PLAYER","MAZE-HERO","MAZE-S
TAR","MAZE-MASTER"
210 FOR I=1 TO 8 :: READ C,A
$ :: CALL CHAR(C,A$):: NEXT
I
220 DATA 138,0000183C3C3C18,
96,24242466E77EFF7E1818183C7
E7EFF7E0A1FFF0707FF1F0A020F1
FFFFF1F0F02
230 DATA 100,7EFF7EE76624242
47EFF7E7E3C18181850F8FFE0E0F
FF85040E0F8FFFFF8E04
240 DATA 136,915234F81F2C4A8
9,137,894A2C1FF8345291,130,1
F5F036BE0E9FDFC,129,FFBDDBE7
E7DBBDFF,110,7CFEFEFEFEFE7C
250 CALL COLOR(9,7,1,10,5,1,
11,14,1,12,13,1,13,16,7)
260 A$=RPT$(" ",11):: A$=A$&
"POWER-PAC"&A$ :: FOR I=2 TO
 28
270 CALL HCHAR(1,I-1,ASC(SEG
$(A$,I-1,1))):: CALL HCHAR(1
,I,130,4):: NEXT I
280 RESTORE 300 :: FOR I=2 T
O 22 STEP 2 :: READ A$
290 DISPLAY AT(I,1):A$ :: NE
XT I :: DISPLAY AT(24,3):"PR
ESS ANY KEY TO PLAY."
300 DATA " IN THIS GAME YOU
STEER YOUR",EATOID(`) ROUND
A MAZE. LOOK,OUT FOR SPIN-SP
IKES0 & EAT
310 DATA POWER-PACS() TO ST
AY ALIVE.,YOU MUST EAT ALL T
HE PUCS(
        ),AND PICS() TO FI
NISH A MAZE
```

```
320 DATA A RED MAZE MEANS LO
W POWER &,A FLASHING ONE MEA
NS DANGER.," MOVE WITH E(UP
), X(DOWN),","S(LEFT) & D(RI
GHT) OR USE A",JOYSTICK.
330 CALL KEY(0,K,S):: IF S=0
 THEN 330
340 CALL CLEAR :: RESTORE 36
0 :: FOR I=1 TO 65 :: READ W
,X,Y,Z :: IF W=Y THEN 360
350 CALL HCHAR(X,W,110,Y-W+1
):: GOTO 370
360 CALL VCHAR(X,W,110,Z-X+1
)
370 NEXT I :: GOTO 480
380 DATA 1,1,32,1,32,1,32,24
,1,24,32,24,1,1,1,24,3,3,10,
3,12,3,15,3,18,3,23,3,26,3,3
0,3
390 DATA 30,4,30,9,30,11,30,
19,30,21,30,22,26,22,29,22,1
8,22,24,22,11,22,14,22,3,22,
9,22
400 DATA 3,20,3,21,3,14,3,18
,3,7,3,12,3,4,3,5,15,3,15,7,
15,9,15,15,15,17,15,22,18,5,
18,17
410 DATA 10,4,10,5,4,5,9,5,1
2,4,12,7,13,7,15,7,4,7,6,7,6
,8,6,12,4,12,5,12
420 DATA 8,7,10,7,10,8,10,8,
11,9,14,9,11,15,14,15,11,13,
11,14,8,12,10,12,8,8,8,11
430 DATA 4,14,9,14,9,15,9,18
,4,18,8,18,4,20,9,20,9,21,9,
22,11,17,15,17,11,18,11,21
440 DATA 19,5,23,5,23,6,23,7
,20,7,22,7,26,4,26,9,27,9,30
,9,19,8,19,17
450 DATA 21,9,21,18,22,9,23,
9,23,10,23,10,24,11,26,11,26
,12,26,14,23,14,25,14
460 DATA 28,11,30,11,24,19,2
9,19,24,16,24,18,25,16,27,16
,28,12,28,15
470 DATA 18,19,22,19,22,15,2
2,18,18,21,24,21,26,21,29,21
480 SR=0 :: LL=1
490 EN=80-LL*5 :: NS=LL*2-1
500 FOR I=0 TO NS
510 SX(I)=INT(1.5*I+3):: SY(
I)=2-21*(I/2=INT(I/2))
520 CALL HCHAR(SY(I),SX(I),1
36):: CALL SOUND(10,110,0)
```

```
530 NEXT I
540 YY=12 :: YX=16 :: YC=32
:: YP=96 :: N=(LL+7)/2 :: FO
R I=1 TO N*2
550 X=INT(30*RND)+2 :: Y=INT
(22*RND)+2 :: CALL GCHAR(Y,X
,P)
560 IF P<>110 THEN 550 :: CA
LL HCHAR(Y,X,129-9*(I>N))
570 NEXT I :: FOR I=1 TO N/2
580 X=INT(30*RND)+2 :: Y=INT
(22*RND)+2 :: CALL GCHAR(Y,X
,P):: IF P<>110 THEN 580
590 CALL HCHAR(Y,X,130):: CA
LL SOUND(100,1000,0):: NEXT
I
600 CALL HCHAR(YY,YX,YP)
610 FOR I=0 TO NS :: GOSUB 7
20
620 X=SX(I):: Y=SY(I):: P=13
7+(EN/2=INT(EN/2))
630 X1=SGN(YX-X):: Y1=SGN(YY
-Y)
640 CALL GCHAR(Y+Y1,X+X1,P1)
:: CALL GCHAR(Y+Y1,X,P2):: C
ALL GCHAR(Y,X+X1,P3)
650 IF (P1=32)+((P1>95)*(P1<
104))THEN 690
660 IF (P2=32)+((P2>95)*(P2<
104))THEN X1=0 :: GOTO 690
670 IF (P3=32)+((P3>95)*(P3<
104))THEN Y1=0 :: GOTO 690
680 CALL HCHAR(Y,X,P):: X1=0
 :: Y1=0 :: GOTO 710
690 CALL GCHAR(Y+Y1,X+X1,P1)
:: CALL HCHAR(Y,X,32):: CALL
 HCHAR(Y+Y1,X+X1,P)
700 IF P1<>32 THEN 960
710 SY(I)=Y+Y1 :: SX(I)=X+X1
 :: NEXT I :: GOTO 610
720 EN=EN-.5 :: IF EN<26 THE
N CALL COLOR(10,7,1):: IF EN
<11 THEN CALL COLOR(10,16,1)
730 CALL KEY(3,K,S):: IF (K<
>69)*(K<>68)*(K<>83)*(K<>88)
THEN 790
740 X=0 :: Y=0 :: IF K=69 TH
EN Y=-1
750 IF K=68 THEN X=1
760 IF K=83 THEN X=-1
770 IF K=88 THEN Y=1
780 GOTO 830
790 X=0 :: Y=0 :: CALL JOYST
(1,W,Z):: IF W=0 AND Z=0 THE
N CALL JOYST(2,W,Z)
800 IF W<>0 THEN X=W/4 :: GO
TO 830
810 IF Z<>0 THEN Y=-(Z/4)::
GOTO 830
820 GOTO 880
830 CALL GCHAR(YY+Y,YX+X,P):
: IF (P=110)THEN 880
840 IF X=0 THEN C=99+Y*2 ELS
E C=101+X*2
850 CALL HCHAR(YY,YX,32):: C
ALL HCHAR(YY+Y,YX+X,C+(EN=IN
T(EN)))
860 IF (P<>32)*((P<136)+(P>1
37))THEN 900
870 YY=YY+Y :: YX=YX+X :: IF
 (P>135)*(P<138)THEN 960
880 IF EN=0 THEN 950
890 RETURN
900 CALL HCHAR(YY+Y,YX+X,110
):: CALL HCHAR(YY,YX,96):: X
=0 :: Y=0
910 CALL SOUND(100,1000,10,-
1,0)
920 IF P=130 THEN EN=EN+INT(
LL+20*RND)+1 :: SR=SR+20 ::
CALL COLOR(10,5,1):: P=3
2 :: GOTO 870
930 SR=SR+(INT(LL*RND)+1)*10
 :: PE=PE+1 :: IF PE=N*2 THE
N 1030
940 P=32 :: GOTO 870
```

```
950 DISPLAY AT(12,8):"NO POW
ER!!" :: FOR D=1 TO 200 :: N
EXT D
960 FOR I=1 TO 30 :: CALL SO
UND(-100,-5,I):: NEXT I
970 CALL CLEAR :: PE=0 :: CA
LL COLOR(10,5,1)
980 DISPLAY AT(3,3):"YOU SCO
RED";SR;"POINTS.": :"   YOU
GOT UP TO LEVEL";LL
990 DISPLAY AT(8,8):"RATING:
":RPT$(" ",14-LEN(R$(LL/2))/
2);R$(LL/2)
1000 IF LL=10 THEN DISPLAY A
T(10,12):"SUPREME"
1010 FOR D=1 TO 200 :: NEXT
D :: DISPLAY AT(24,1):"PRESS
 ANY KEY TO PLAY AGAIN."
1020 GOTO 330
1030 FOR I=2 TO 16 :: CALL S
OUND(-200,I*60,0):: CALL COL
OR(10,I,1,9,I-1,1):: NEXT I
1040 SR=SR+INT(EN)*10 :: IF
LL=10 THEN LL=9
1050 CALL COLOR(10,5,1,9,7,1
):: LL=LL+1 :: PE=0 :: CALL
HCHAR(YY,YX,32):: FOR I=0 TO
 NS
1060 CALL HCHAR(SY(I),SX(I),
32):: SX(I)=0 :: SY(I)=0 ::
SC(I)=0 :: NEXT I :: GOTO 49
0
```

```
50 REM ********************
51 REM *     B U L L        *
52 REM *                    *
53 REM * In Extended Basic*
54 REM *                    *
55 REM ********************
56 CALL CLEAR
100 CALL SCREEN(7)
110 DISPLAY AT(12,9):"BULL"
120 FOR I=1 TO 14 :: CALL CO
LOR(I,2,7):: NEXT I
130 CALL CLEAR
140 FOR C=33 TO 140
150 READ C$
160 CALL CHAR(C,C$)
170 NEXT C
180 DATA FFFFFFFFFFFFFFFF,,0
001070F1F3F7F7F,40C080000000
808,00000000003C4582,0000030
4081020E,7FC
190 DATA C0303F080402,000087
68101008,0000806C12473804,00
04060703030307,0000000080C0E
0F,E0FFFFFFFFFFFFFF
200 DATA 0102FFFEFFFFFAFC,05
489020C08,0000030301110E,008
080000CF3,070F3F2F271D0602,F
0FCFFFFFFFF1F0D
210 DATA 0000FFFFFFFFFFFF,0F
1FFFFFFFFFFFFF,FCFCFCFCFCFCF
CFC,7F7F7F3F1F1F2F2,FFFFFFFF
FCF0C,FCF9FA0D
220 DATA 7080384488102021,03
0001020404040 2,438C304040818
282,0E166EBF7E,FFFFFFFF0F030
1,F8F0F0E0C08
230 DATA 00000000106040E0F,20
204183071F7FFF,008000808CFFF
FFF,0000806A7FFFFFFF,2224455
EFFFEFFFE
240 DATA 01FD03798503010D,84
B42424241C0101,0C08304080902
0C,0080707C3E3E1F1F,0000101C
3E3FFFFF
250 DATA 38300E81406,0000008
0C020100C,1F1F3F3F7F7F797,FF
FFFFFFFFFCFAFD,FEFFFCFCF8588
10B,749C2008A8F8FCFC
260 DATA 8040404020204010008,
1F0F0F0F070707E7,070301,FFFF
FF7F,F4E9CB830F070707,17FFFF
F9FDFCFEFE
270 DATA 0F000818FCFCFCFC,F8
C8070060906,38D890187C94E407
,FFFFFFFEF090909,FFFFFF7F3F1
F272,FFFFFFFFFFFFCF8
280 DATA 0303030301010101,7F
7F7DF8E0FFFFFF,0707030101010
303,FFFFFFCFCFCEFCF1,9090A0A
06040C09,202020202020202
290 DATA FEFCF8F0E0C0C081,10
1020204043941 8,204040808,7F7
F3F3F3F1F1F0F,FCF8F0E0E0E6FF
FF,00001F205F84C7E
300 DATA 0404848480C0F3FF,00
0000000C020101,1202020204040 8
08,0F0F0707070737C7,FFFFFFB3
8080F0FF,E0E0C08000003FFF
310 DATA 7F7F7F3E1C0080F,808
000181C1E3F7E,101020204080 80
38,FFFF3F3F3F3F1F1F,FEFEFEFC
F0F2F1F,0F0F,FF7F,FFF8
320 DATA F0E,078040201805038
1,00E040808,4222120A06020101
,86463A01,80000000003050505,0
00003FC,408
330 DATA 080808101010202,080
804040404040 4,00000080402010
1,000007080A0A04,03334280081
0204,0E708001020C106
340 DATA 8000402010101 02,800
10200808080C,A0100F,00010638
C,8080407807,4040808080808 08
,808080808E513E2
350 DATA 000007182020404,00C
0201000000001
360 PRINT TAB(6);"#$ %&'()*
 +,"
370 PRINT TAB(6);"!-./  0123
456"
380 PRINT TAB(6);"789:  ;< =
>!?"
390 PRINT TAB(5);"@ABCDE  FG
HIJK"
400 PRINT TAB(5);"L!!MNOP
Q!!5,"
410 PRINT TAB(6);"RSTU!VWX#Y
Z[S"
420 PRINT TAB(8);"\!]6 ^_`a"
430 PRINT TAB(9);"!!6 \bcd"
440 PRINT TAB(9);"e!fghij"
450 PRINT TAB(9);"k!lmnop"
460 PRINT TAB(10);"q!!!r"
470 PRINT TAB(10);"s tuv"
480 FOR I=1 TO 5 :: PRINT ::
 NEXT I
490 CALL SOUND(-2500,-7,21)
500 FOR I=1 TO 25
510 READ X,Y,C
520 CALL HCHAR(X,Y,C)
530 NEXT I
540 DATA 18,17,119,18,18,120
,19,17,121,20,18,122,19,18,1
23,20,19,124,20,20,125,19,20
,126
550 DATA 18,20,127,17,20,128
,17,19,129,18,11,130,18,10,1
31,19,11,132,20,11,125,20,10
,134
560 DATA 19,10,133,20,9,135,
20,8,136,19,8,137,18,8,138,1
7,8,139,17,9,39,17,10,140,1,
1,32
570 RESTORE 540
580 FOR I=1 TO 25
590 READ X,Y,C
600 CALL HCHAR(X,Y,32)
610 NEXT I
620 RESTORE 540
630 GOTO 490
```

```
100 REM ********************
110 REM *                  *
120 REM * T R I F E C T A  *
130 REM *                  *
140 REM * TE#2 & Speech Syn*
150 REM *        required. *
160 REM *   -TI.S.H.U.G-   *
170 REM *    October'85    *
180 REM ********************
190 CALL CLEAR
200 OPEN #1:"SPEECH",OUTPUT
210 PRINT #1:"_HELLO, PUNTER
 MAY I ASSIST YOU IN EVALUAT
ING A ^TRY FECTA. "
220 OPTION BASE 1
230 DIM M$(10,5),M(10,18),V(
18),RD(18)
240 INPUT "RACE MEETING & DA
TE?":RCN$
250 PRINT
260 INPUT "RACE NO.":NO
270 R=R+1
280 RD(R)=NO
290 PRINT
300 INPUT "NO. OF ENTRIES IN
 FIELD?":NE
310 FOR ENT=1 TO NE
320 COUNT=COUNT+1
330 IF COUNT>4 THEN 370
340 PRINT "NAME OF NO.";COUN
T;"FAV-"
350 INPUT NN$
360 M$(NO,ENT)=NN$
370 PRINT "ODDS ON ENTRY";EN
T;"-"
380 INPUT "V1,V2-":V1,V2
390 ODDS=V1/V2
400 M(NO,ENT)=1/ODDS
410 V(NO)=V(NO)+M(NO,ENT)
420 NEXT ENT
430 COUNT=0
440 PRINT "NEW RACE Y OR N"
450 CALL KEY(0,K,S)
460 IF S=0 THEN 450
470 IF K=89 THEN 1110
480 IF K<>78 THEN 450
490 PRINT "DO YOU WISH TO ED
IT Y OR N"
500 CALL KEY(0,K,S)
510 IF S=0 THEN 500
520 IF K=89 THEN 1160
530 IF K<>78 THEN 500
540 CALL CLEAR
550 PRINT RCN$
560 PRINT
570 PRINT #1:"EVALUATING RES
ULTS FOR";RCN$
580 FOR EVAL=1 TO R
590 NO=RD(EVAL)
600 TU1=M(NO,1)*1/V(NO)
610 TU2=M(NO,2)*1/(V(NO)-M(N
O,1))
620 TU3=M(NO,3)*1/(V(NO)-M(N
O,1)-M(NO,2))
630 TU4=M(NO,4)*1/(V(NO)-M(N
O,1)-M(NO,2)-M(NO,3))
640 PRINT "-----------------
-----------"
650 PRINT "RACE NUMBER";NO
660 PRINT #1:"RACE NUMBER";N
O
670 IF (V(NO)>1.3)+(V(NO)<.8
0)THEN 790
680 IF (TU1<.25)+(TU2<.25)+(
TU3<.25)THEN 840
690 REM SBR COMPARES ODDS
700 IF (M(NO,3)-M(NO,4))/M(N
O,4)<.2 THEN 1070
710 IF ((M(NO,1)-M(NO,2))/M(
NO,2)>.5)*((M(NO,2)-M(NO,3))
/M(NO,3)>.5)THEN 910
720 IF (M(NO,3)-M(NO,4))/M(N
O,4)<.50 THEN 950 ELSE 1000
730 GOTO 1040
740 NEXT EVAL
```

```
750 INPUT "ENTER 1 TO REPEAT
,2 TO ADD,    OR 3 TO EDIT!":
ADR
760 IF (ADR<>1)*(ADR<>2)*(AD
R<>3)THEN 750
770 ON ADR GOTO 540,1280,116
0
780 STOP
790 PRINT ".8<TOT ODDS>1.3 ".
800 PRINT "TOT ODDS=";V(NO)
810 PRINT #1:"WARNING TOTAL
ODDS _OUT OF ^RANGE. RACE MA
Y BE _INVALID. EVALUATING US
ING _TRUE ODDS"
820 PRINT
830 GOTO 680
840 PRINT "TRUE ODDS>.25"
850 PRINT "TO1=";TU1
860 PRINT "TO2=";TU2
870 PRINT "TO3=";TU3
880 PRINT #1:"NO VALUE IN TH
IS RACE. TRUE ODDS ON FAVORI
TES _TO ^SMALL"
890 PRINT
900 GOTO 740
910 PRINT "FAVOURITES UNBOXE
D"
920 PRINT
930 PRINT #1:M$(NO,1),: : :M
$(NO,2),: : :M$(NO,3),: :"UN
BOCKST. THIS IS A GOOD VALUE
 RACE"
940 GOTO 740
950 PRINT "FAVOURITES BOXED
4TH FAV IS     THE DANGER"
960 PRINT
970 PRINT #1:M$(NO,1),: : :M
$(NO,2),: : :M$(NO,3): : :,"
BOCKST. ",: :M$(NO,4),"IS TH
E DANGER"
980 PRINT #1:"CHECK ODDS ^BE
FORE PLACING BET"
990 GOTO 740
1000 PRINT "FAVOURITES BOXED
"
1010 PRINT
1020 PRINT #1:M$(NO,1),: : :
M$(NO,2),: : :M$(NO,3): : :,
"BOCKST. "
1030 GOTO 740
1040 PRINT "NO VALUE AT THIS
 TIME"
1050 PRINT #1:"NO _VALUE AT
_THIS ^TIME"
1060 GOTO 740
1070 PRINT "4TH FAV TOO CLOS
E TO 3RD"
1080 PRINT #1:"NO VALUE";: :
M$(NO,4);"TO CLOSE _TOO";: :
M$(NO,3)
1090 PRINT
1100 GOTO 740
1110 REM SBR FOR NEW RACE
1120 CALL CLEAR
1130 PRINT RCN$
1140 GOTO 260
1150 STOP
1160 REM EDIT SBR
1170 CALL CLEAR
1180 PRINT RCN$
1190 PRINT
1200 INPUT "RACE NO. TO EDIT
.":EDT
1210 V(EDT)=0
1220 CALL CLEAR
1230 PRINT "RACE NO.";EDT
1240 NO=EDT
1250 GOTO 300
1260 STOP
1270 REM SBR FOR ADDING A RA
CE
1280 FOR TEST=1 TO 20
1290 IF RD(TEST)=0 THEN 1310
1300 NEXT TEST
1310 LET R=TEST-1
1320 CALL CLEAR
1330 PRINT RCN$
1340 GOTO 250
1350 STOP
```

```
100 CALL CLEAR :: CALL SCREE
N(4):: CALL CHAR(128,"FF8181
81818181FF",129,RPT$("F",16)
):: CALL COLOR(13,2,16)
110 FOR R=9 TO 15 :: CALL HC
HAR(R,11,128,9):: NEXT R
120 X=1 :: FOR R=9 TO 15 ::
DISPLAY AT(R,7)SIZE(2):STR$(
X):: X=X*2 :: NEXT R :: FOR
C=9 TO 17 :: DISPLAY AT(8,C)
SIZE(1):STR$(C-8):: NEXT C
130 DISPLAY AT(2,9):"TIGERCU
B'S" :: DISPLAY AT(4,1):"GEM
INI CHARACTER DOWNLOADER" !p
rogrammed by Jim Peterson fo
r the Public Domain
140 DISPLAY AT(17,1):" Move
cursor with W,E,R,S,D,":"Z,X
 and C keys. Toggle on":"and
 off with Q key. Press":"Ent
er when finished.": : :"Pres
s any key"
150 CALL KEY(0,K,ST):: IF ST
=0 THEN 150 :: CALL HCHAR(17
,1,32,224)
160 R=9 :: C=11 :: CH=128
170 CALL HCHAR(R,C,32):: CAL
L HCHAR(R,C,CH):: FOR D=1 TO
 10 :: NEXT D :: CALL KEY(3,
K,ST):: IF ST=0 THEN 170
180 ON POS("QWERDCXZS"&CHR$(
13),CHR$(K),1)+1 GOTO 170,31
0,230,220,210,200,190,260,25
0,240,330
190 R=R+1
200 C=C+1 :: GOTO 270
210 C=C+1
220 R=R-1 :: GOTO 270
230 R=R-1
240 C=C-1 :: GOTO 270
250 C=C-1
260 R=R+1
270 R=R-(R<9)+(R>15):: C=C-(
C<11)+(C>19):: IF CH=128 THE
N 300 :: CALL GCHAR(R,C-1,GX
):: CALL GCHAR(R,C+1,GZ):: I
F (GX<>129)*(GZ<>129)THEN 30
0
280 DISPLAY AT(22,1):"You ca
n't have two in a row":"hori
zontally!" :: FOR D=1 TO 50
:: NEXT D :: DISPLAY AT(22,1
):" ":" "
290 CH=CH-1
300 CALL HCHAR(R,C,CH):: GOT
O 170
310 CH=CH+1+(CH=129)*2 :: IF
 CH=128 THEN 320 :: CALL GCH
AR(R,C-1,GX):: CALL GCHAR(R,
C+1,GZ):: IF (GX<>129)*(GZ<>
129)THEN 320 ELSE 280
320 CALL HCHAR(R,C,CH):: GOT
O 170
330 FOR C=11 TO 19 :: X=1 ::
 FOR R=9 TO 15 :: CALL GCHAR
(R,C,G)
340 IF G=129 THEN A=A+X
350 X=X*2 :: NEXT R
360 FOR J=1 TO LEN(STR$(A)):
: CALL VCHAR(15+J,C,ASC(SEG$
(STR$(A),J,1))):: NEXT J ::
M$=M$&CHR$(A):: A=0 :: NEXT
C :: A=0
370 DISPLAY AT(20,1):"Print?
 Y/N Y" :: ACCEPT AT(20,12)V
ALIDATE("YN")SIZE(-1):Q$ ::
IF Q$="N" THEN 470
380 IF F=1 THEN 390 :: F=1 :
: DISPLAY AT(20,1):"Printer
name?" :: ACCEPT AT(20,15):P
$ :: OPEN #1:P$
390 DISPLAY AT(20,1):"ASCII
to redefine?" :: ACCEPT AT(2
0,20)VALIDATE(DIGIT)SIZE(3):
CH
400 DISPLAY AT(20,1):"Descen
der (0 or 1)? 0" :: ACCEPT A
T(20,21)VALIDATE("01")SIZE(-
1):D$ :: D=VAL(D$)
410 M$=CHR$(27)&CHR$(42)&CHR
$(1)&CHR$(CH)&CHR$(D)&M$
420 PRINT #1:M$ :: PRINT #1:
CHR$(27);CHR$(36);CHR$(1);
430 PRINT #1:RPT$(CHR$(CH),7
2):: PRINT #1:CHR$(14);RPT$(
CHR$(CH),36)
440 DISPLAY AT(20,1):"Save (
Y/N)? Y" :: ACCEPT AT(20,13)
VALIDATE("YN")SIZE(-1):Q$ ::
 IF Q$="N" THEN 470
450 IF F3=1 THEN 460 :: F3=1
 :: DISPLAY AT(20,1):"Filena
me? DSK" :: ACCEPT AT(20,14)
:F$ :: OPEN #2:"DSK"&F$
460 PRINT #2:M$
470 M$="" :: DISPLAY AT(20,1
):"Another (Y/N)? Y" :: ACCE
PT AT(20,16)VALIDATE("YN")SI
ZE(-1):Q$ :: IF Q$="Y" THEN
100
480 CLOSE #1 :: CLOSE #1 ::
END
```

```
100 REM ** FAS-CINATION **
110 REM ******************
120 REM         BY
130 REM    JAN  BARNIER
140 REM   ROZENSTRAAT 36
150 REM  8471 KM  WOLVEGA
160 REM  THE  NETHERLANDS
170 REM ******************
180 REM Runs in TI & XBasic
190 REM ******************
200 RANDOMIZE
210 CALL SCREEN(2)
220 CALL HCHAR(1,1,31,768)
230 FOR K=1 TO 12
240 READ F$
250 CALL COLOR(K,INT(RND*14+
3),2)
260 FOR R=K TO 25-K
270 CALL CHAR(33+(K-1)*8,F$)
280 CALL HCHAR(R,4+K,33+(K-1
)*8,24-X)
290 NEXT R
300 X=X+2
310 NEXT K
320 RESTORE
330 FOR X=1 TO 12
340 FOR K=4 TO 16
350 J=INT(RND*14)+3
360 IF J=B THEN 350
370 FOR S=12 TO 1 STEP -1
380 CALL COLOR(S,INT(RND*14)
+3,J)
390 NEXT S
400 B=INT(RND*14)+3
410 IF B=J THEN 400
420 FOR S=1 TO 12
430 CALL COLOR(S,1,2)
440 NEXT S
450 CALL CHAR(K*8+1,F$)
460 NEXT K
470 READ F$
480 NEXT X
490 GOTO 320
500 DATA 24247E7E7E7E2424
510 DATA 000000FFFF
520 DATA 1818181818181818
530 DATA 00003C3C3C3C
540 DATA 18183C24243C1818
550 DATA 0102040810204080
560 DATA 183C7EFFFF7E3C18
570 DATA 007E7E7E7E7E7E
580 DATA 8142241818244281
590 DATA 995A3CFFFF3C5A99
600 DATA FF818181818181FF
610 DATA 18183C3C7E7EFFFF
```

```
100 ! **************
110 ! *BRIGHT EYES*
120 ! **************
130 ! TISHUG LIBRARY 110
140 ! EXTENDED BASIC
150 ! FROM WATERSHIP DOWN
160 ! BY MIKE BATT
170 ! CONVERTED BY R DUNLOP
180 CALL CLEAR
190 AF0=779 :: A0=825 :: AS0
,BF0=875 :: C0=982 :: CS0,DF
0=1040 :: D0=1102 :: DS0,EF0
=1168 :: E0=1237 :: F0=1311
:: FS0,GF0=1389
200 CALL CLEAR :: A1=110 ::
AS1,BF1=117 :: B1=123 :: C1=
131 :: CS1,DF1=139 :: D1=147
 :: DS1,EF1=156 :: E1=165 ::
 F1=175 :: FS1,GF1=185 :: G1
=196
210 A2=220 :: AS2,BF2=233 ::
 B2=247 :: C2=262 :: CS2,DF2
=277 :: D2=294 :: DS2,EF2=31
1 :: E2=330 :: F2=349 :: FS2
,GF2=370 :: G2=392
220 A3=440 :: AS3,BF3=466 ::
 B3=494 :: C3=523 :: CS3,DF3
=554 :: D3=587 :: DS3,EF3=62
2 :: E3=659 :: F3=698 :: FS3
,GF3=740 :: G3=784
230 A4=880 :: AS4,BF4=932 ::
 B4=988 :: C4=1047 :: CS4,DF
4=1109 :: D4=1175 :: DS4,EF4
=1245 :: E4=1319 :: F4=1397
:: FS4,GF4=1480 :: G4=1568
240 G0=1471 :: GS0,AF1=1559
:: GS1,AF2=208 :: GS2,AF3=41
5 :: GS3,AF4=831 :: GS4,AF5=
1661 :: R=40000
250 C=450 :: M=C*2 :: DC=C*1
.5 :: Q=C*0.5 :: COUNT=0 ::
FL=0 :: CALL MAGNIFY(4):: CA
LL PICTURE :: GOTO 360
260 CALL SOUND(T,S1,V1):: RE
TURN !1 NOTE
270 CALL SOUND(T,S2,V2):: RE
TURN !1 NOTE(S2)
280 CALL SOUND(T,S3,V3):: RE
TURN !1 NOTE(S3)
290 CALL SOUND(T,S1,V1,S2,V2
):: RETURN !2 NOTES(S1,S2)
300 CALL SOUND(T,S1,V1,S3,V3
):: RETURN !2 NOTES(S1,S3)
310 CALL SOUND(T,S2,V2,S3,V3
):: RETURN !2 NOTES(S2,S3)
320 CALL SOUND(T,S1,V1,S2,V2
,S3,V3):: RETURN !3 NOTES
330 CALL SOUND(T,S1,V1,S2,V2
,S3,30,-4,V3):: RETURN !BASS
 +2 NOTES
340 CALL SOUND(T,S3,30,S3,30
,S3,30,-4,V3):: RETURN !BASS
350 CALL SOUND(T,S1,V1,S3,30
,S3,30,-4,V3):: RETURN !BASS
 +1 NOTE
360 V1=10 :: V2=10 :: V3=6 :
: T=Q :: S3=G0 :: GOSUB 340
:: S1=D2 :: GOSUB 350 :: S1=
G2 :: GOSUB 350 :: S1=D2 ::
GOSUB 350 :: S1=B3 :: GOSUB
350
370 S1=G2 :: GOSUB 350 :: S1
=D2 :: GOSUB 350 :: S1=G2 ::
 GOSUB 350 :: GOSUB 680 :: V
1=0 :: V2=6 :: V3=6
380 FOR COUNT=1 TO 2 :: V1=1
 :: T=Q :: S1=B3 :: S3=G0 ::
 GOSUB 350 :: S1=D3 :: S2=B3
 :: T=C :: GOSUB 330 :: T=Q
:: GOSUB 350 :: T=C :: GOSUB
 330
390 S1=B3 :: T=Q :: GOSUB 35
0 :: S1=D3 :: GOSUB 350 :: T
=M :: S1=E3 :: S2=C1 :: GOSU
B 290 :: S1=D3 :: S2=G0 :: G
OSUB 350
```

```
400 S3=G0 :: T=2*M :: GOSUB
340
410 S1=G2 :: S2=B2 :: S3=E0
:: T=C :: GOSUB 330 :: T=Q :
: S1=B3 :: GOSUB 260 :: S2=G
2 :: T=C :: GOSUB 330 :: S1=
G2 :: S2=B2 :: GOSUB 330
420 S1=B3 :: T=Q :: GOSUB 26
0 :: T=M :: S1=C3 :: S2=C1 :
: GOSUB 290 :: S1=B3 :: S3=G
0 :: GOSUB 350 :: T=1.5*M ::
 GOSUB 340 :: S3=G1 :: T=C :
: GOSUB 280
430 !@P-
440 T=Q :: S1=A3 :: S2=FS2 :
: S3=D1 :: GOSUB 320 :: GOSU
B 320 :: GOSUB 320 :: GOSUB
320 :: S3=C1 :: GOSUB 320 ::
 S1=B3 :: GOSUB 320
450 S1=C3 :: S2=A3 :: T=C ::
 GOSUB 320 :: S1=D3 :: S2=G2
 :: S3=G0 :: GOSUB 330 :: S2
=B3 :: S3=A1 :: GOSUB 320 ::
 T=M+DC :: S1=G2 :: S2=C1 ::
 GOSUB 290
460 T=Q :: S1=E2 :: S2=C1 ::
 GOSUB 290 :: S1=C3 :: S2=E2
 :: S3=A1 :: GOSUB 320 :: T=
C :: GOSUB 320 :: T=Q :: GOS
UB 320
470 S1=FS2 :: S3=D0 :: T=M*2
 :: GOSUB 350 :: S3=D1 :: T=
M+Q :: GOSUB 280 :: S1=G2 ::
 S2=E2 :: T=C :: GOSUB 320
480 T=Q :: S1=A3 :: S2=FS2 :
: GOSUB 320 :: T=C :: S1=B3
:: S2=G2 :: S3=G0 :: GOSUB 3
30 :: T=Q :: S1=D3 :: GOSUB
350 :: T=C :: S2=B3 :: GOSUB
 330
490 T=Q :: S1=B3 :: GOSUB 35
0 :: T=C :: S1=D3 :: S2=B3 :
: GOSUB 330 :: T=M :: S1=E3
:: S2=C1 :: GOSUB 290 :: S1=
D3 :: S3=G0 :: GOSUB 350
500 T=M+C+Q :: GOSUB 340 ::
S1=G2 :: T=Q :: GOSUB 350 ::
 T=M :: S3=E0 :: GOSUB 350 :
: T=Q :: S1=B3 :: S2=B2 :: G
OSUB 330
510 T=C :: S1=G2 :: GOSUB 33
0 :: T=Q :: S1=B3 :: S2=G2 :
: GOSUB 330 :: T=M :: S1=C3
:: S2=C1 :: GOSUB 290 :: S1=
B3 :: S3=G0 :: GOSUB 350
520 T=M+C+Q :: GOSUB 340 ::
T=Q :: S1=G2 :: GOSUB 350 ::
 S1=A3 :: S2=FS2 :: S3=D1 ::
 GOSUB 320 :: GOSUB 320 :: G
OSUB 320 :: T=C :: GOSUB 320
530 T=Q :: S3=C1 :: GOSUB 32
0 :: S1=B3 :: GOSUB 320 :: T
=C :: S1=C3 :: S2=A3 :: GOSU
B 320 :: S1=D3 :: S2=G2 :: S
3=G0 :: GOSUB 330
540 S2=B3 :: S3=A1 :: GOSUB
320 :: T=M+DC :: S1=G2 :: S2
=C1 :: GOSUB 290 :: T=Q :: G
OSUB 290 :: S1=C3 :: S2=E2 :
: GOSUB 320
550 T=C :: GOSUB 320 :: T=Q
:: GOSUB 320 :: T=2*M :: S1=
B3 :: S2=DS1 :: GOSUB 290 ::
 T=M :: S1=AS3 :: S2=CS1 ::
GOSUB 290
560 S1=B3 :: S2=D1 :: GOSUB
290 :: T=Q :: S1=C3 :: S2=FS
2 :: S3=D0 :: GOSUB 330 :: T
=C :: GOSUB 330 :: S1=FS2 ::
 T=Q :: GOSUB 350
570 T=M*2 :: S1=G2 :: S3=G0
:: GOSUB 350 :: T=M :: S3=C0
 :: GOSUB 340 :: T=Q :: S1=B
3 :: S2=G2 :: GOSUB 330
```

```
580 S1=D3 :: T=C :: GOSUB 33
0 :: T=2*M+Q :: S3=B1 :: GOS
UB 300 :: T=DC :: S1=G2 :: S
2=C1 :: GOSUB 290 :: S1=C3 :
: S2=G2 :: S3=C1 :: GOSUB 32
0
590 T=Q :: S1=B3 :: S2=E2 ::
 GOSUB 320 :: S1=A3 :: GOSUB
 320 :: S1=B3 :: S2=FS2 :: S
3=D1 :: GOSUB 320 :: S1=A3 :
: T=2*M+DC :: GOSUB 320
600 S1=B3 :: S2=FS2 :: S3=D0
 :: T=Q :: GOSUB 330 :: S1=D
3 :: T=C :: GOSUB 330 :: T=2
*M+Q :: S3=G0 :: GOSUB 350 :
: S1=G2 :: S2=E2 :: S3=C1 ::
 T=Q
610 GOSUB 320 :: GOSUB 320 :
: GOSUB 320 :: T=DC :: S1=C3
 :: S2=G2 :: GOSUB 320 :: T=
Q :: S1=B3 :: GOSUB 300 :: S
1=A3 :: T=2*M+Q :: S3=A1 ::
GOSUB 300
620 T=2*M :: GOSUB 280 :: T=
Q :: S1=B3 :: S2=FS2 :: S3=D
S0 :: GOSUB 330 :: GOSUB 330
 :: GOSUB 330 :: T=DC :: S3=
E0 :: GOSUB 330
630 T=Q :: S1=G2 :: GOSUB 35
0 :: S1=D3 :: S2=A3 :: S3=FS
0 :: T=DC :: GOSUB 330 :: T=
Q :: GOSUB 330 :: T=DC :: GO
SUB 330
640 S1=G2 :: S2=D2 :: S3=G0
:: GOSUB 330 :: T=Q :: S1=C3
 :: S2=G2 :: S3=C0 :: GOSUB
330 :: GOSUB 330 :: GOSUB 33
0
650 T=DC :: S1=D3 :: S2=E2 :
: S3=C1 :: GOSUB 320 :: S1=E
3 :: T=C :: S2=G2 :: GOSUB 3
20 :: T=DC :: S1=A3 :: S2=E2
 :: S3=A1 :: GOSUB 320
660 IF COUNT<>1 THEN 790
670 T=Q :: S1=C3 :: S2=FS2 :
: S3=D0 :: GOSUB 330 :: T=C
:: S1=B3 :: GOSUB 330 :: T=M
*2+Q :: S1=G2 :: S3=G0 :: GO
SUB 350
680 V1=10 :: V2=10 :: V3=6 :
: T=Q :: S1=D3 :: S2=B3 :: G
OSUB 330 :: S2=G2 :: GOSUB 3
30 :: S2=D2 :: GOSUB 330 ::
S2=G2 :: GOSUB 330
690 S2=B3 :: GOSUB 330 :: S1
=E3 :: GOSUB 330 :: S1=FS3 :
: GOSUB 330 :: S1=G3 :: GOSU
B 330 :: T=C :: S1=A4 :: S3=
E0 :: GOSUB 330
700 T=Q :: S1=G3 :: GOSUB 33
0 :: T=C :: S1=B3 :: S2=G2 :
: GOSUB 330 :: S2=E2 :: T=Q
:: GOSUB 330 :: S1=G2 :: GOS
UB 330 :: S1=E2 :: GOSUB 330
710 S2=G2 :: GOSUB 330 :: S2
=E2 :: GOSUB 330 :: S2=B2 ::
 GOSUB 330 :: S2=E2 :: GOSUB
 330 :: S1=G2 :: S2=E2 :: GO
SUB 330 :: S1=A3 :: GOSUB 33
0
720 S1=G2 :: GOSUB 330 :: S1
=A3 :: GOSUB 330 :: S1=G2 ::
 S2=R :: S3=C1 :: GOSUB 320
:: S2=C2 :: GOSUB 320 :: S2=
E2 :: GOSUB 320
730 S2=C2 :: GOSUB 320 :: S2
=E2 :: GOSUB 320 :: S2=C2 ::
 GOSUB 320 :: S2=G1 :: GOSUB
 320 :: S1=C2 :: GOSUB 320 :
: S2=E2 :: GOSUB 320
740 S2=C2 :: GOSUB 320 :: S2
=E2 :: GOSUB 320 :: S2=C2 ::
 GOSUB 320 :: S2=E2 :: GOSUB
 320 :: S1=A3 :: GOSUB 320 :
: S1=G2 :: GOSUB 320
```

↗

```
750 S1=A3 :: GOSUB 320 :: V1
=0 :: V2=6 :: V3=6 :: IF COU
NT=0 THEN RETURN
760 !@P+
770 NEXT COUNT
780 !@P-
790 T=Q :: S1=C3 :: S2=FS2 :
: S3=DO :: GOSUB 330 :: T=C
:: S1=B3 :: GOSUB 330 :: T=M
*2+Q :: S1=G2 :: S3=GO :: GO
SUB 350
800 !@P+
810 IF FL=1 THEN 850
820 FL=1
830 !@P-
840 T=M :: GOSUB 340 :: T=Q
:: S1=B3 :: S2=G2 :: GOSUB 3
30 :: GOTO 580
850 !CODA
860 T=C :: GOSUB 340 :: T=Q
:: V1=8 :: V2=10 :: V3=7 ::
S1=B2 :: GOSUB 350 :: S1=D2
:: GOSUB 350 :: S1=B2 :: GOS
UB 350 :: S1=D2 :: GOSUB 350
870 S1=G1 :: GOSUB 350 :: S1
=B2 :: GOSUB 350 :: T=C :: G
OSUB 340 :: T=Q*1.1 :: GOSUB
 350 :: T=T*1.1 :: S1=D2 ::
GOSUB 350
880 T=T*1.1 :: S1=G1 :: GOSU
B 350 :: T=T*1.1 :: S1=B2 ::
 GOSUB 350 :: T=T*1.1 :: S1=
D2 :: GOSUB 350 :: T=T*1.1 :
: S1=G2 :: GOSUB 350
890 V1=4 :: V2=4 :: V3=4 ::
T=2*M :: S1=B3 :: S2=D2 :: S
3=GO :: GOSUB 330
900 !@P+
910 FOR DEL=1 TO 1500 :: NEX
T DEL
920 CALL CLEAR :: CALL DELSP
RITE(ALL)
930 END
940 SUB PICTURE
950 CALL SCREEN(7):: DISPLAY
 AT(2,10):"BRIGHT EYES" :: D
ISPLAY AT(21,13):"FROM" :: D
ISPLAY AT(22,8):"WATERSHIP D
OWN"
960 CALL CHAR(96,"0000000107
0F3F7F",97,"FFFFFFFFFFFFFFFF
",98,"000000E0F0F8F8AC",99,"
ACFE8EFEFEFEFCFC")
970 CALL CHAR(100,"FFFFFFFFF
FFFFFFF",101,"FFFFFFFFFFFFFF
FF",102,"FCFCFCF8F8F8F0F0",1
03,"F0E0E0E0E0C0C0C0")
980 CALL CHAR(104,"FFFFFFFFF
FFFFFFE",105,"FEFEFCFCFCF8F8
F8",106,"C0C0808000000000",1
07,"0000000000000000")
990 CALL CHAR(108,"000000000
0000000",109,"0000031F7FFFFF
FF",110,"03060E3F7F7F7F7F",1
11,"7FFFFFFFFFFFFFFF")
1000 CALL CHAR(112,"00000000
00000000",113,"0000000000000
000",114,"0000030F1F3F7F7F",
115,"7F7F7F7E7C391103")
1010 CALL CHAR(116,"00000000
00000000",117,"01030307070F1
F3F",118,"070F1F3F7F7FFFFF",
119,"FFFFFFFFFFFFFFFF")
1020 CALL CHAR(120,"FFFFF7E7
CF9F3F7F",121,"FFFFFFFFFFFFF
FFF",122,"FFFFFFFFFFFFFFFF",
123,"FFFFFFFFFFFFFFFF")
1030 CALL CHAR(128,"01010101
01110909",129,"09080C0405020
000",130,"0000000000004020",
131,"108844FF42210000")
1040 CALL CHAR(132,"00301D0F
1622528A",133,"060908FF80000
000",134,"8080008080402000",
135,"088A96D421400000")
1050 CALL CHAR(136,"00000000
00000000",137,"0000000000000
000",138,"000000005000C040",
139,"5088829084A090A2")
1060 CALL SPRITE(#28,96,2,41
,145):: CALL SPRITE(#27,100,
2,73,145):: CALL SPRITE(#26,
104,2,105,145):: CALL SPRITE
(#25,108,2,57,113)
1070 CALL SPRITE(#24,112,2,8
1,81):: CALL SPRITE(#23,116,
2,113,81):: CALL SPRITE(#22,
120,2,89,113)
1080 CALL COLOR(12,2,2):: FO
R COL=15 TO 19 :: CALL VCHAR
(16,COL,124,3):: NEXT COL ::
 CALL HCHAR(18,19,124)
1090 CALL SPRITE(#10,128,13,
121,97):: CALL SPRITE(#11,13
2,13,121,129):: CALL SPRITE(
#12,136,13,89,129)
1100 SUBEND
```

```
100 REM EXTENDED BASIC
110 REM BY PETER LEES
120 REM YOUNGER SET MEMBER
130 CALL CLEAR :: CALL SCREE
N(15)
140 FOR I=0 TO 14 :: CALL CO
LOR(I,I+1,1):: NEXT I
150 CALL MAGNIFY(3)
160 INPUT "TO? ":T$
170 INPUT "FROM? ":F$
180 CALL CLEAR
190 CALL CHAR(96,"03070F0F1F
1F1F1F0F0703010001020 1C0E0F0
F0F8F8F8F8F0E0C08080000000")
200 DISPLAY AT(8,10):"HAPPY"
210 DISPLAY AT(10,13):"BIRTH
DAY"
220 DISPLAY AT(13,17):T$
230 CALL SPRITE(#1,96,2,154,
35,#2,96,12,141,32,#3,96,9,1
35,46)
240 CALL SPRITE(#4,96,4,159,
42,#5,96,14,143,26,#6,96,5,1
70,30)
250 FOR Z=1 TO 29
260 READ D,N
270 CALL SOUND(D.N.2)
280 NEXT Z
290 CALL MOTION(#1,-24,0,#2,
-22,0,#3,-25,0,#4,-23,0,#5,-
26,0,#6,-27,0)
300 FOR A=1 TO 100 :: NEXT A
310 DISPLAY AT(23,12):"LOVE,
 ";F$
320 REM DATA SOUND
330 DATA 250,262,250,262,500
,294,500,262,500,349,1000,33
0,500,17000
340 DATA 250,262,250,262,500
,294,500,262,500,392,1000,34
9,500,17000
350 DATA 250,262,250,262,500
,523,500,440,250,349,250,349
,500,330,1000,294,500,17000
360 DATA 250,466,250,466,500
,440,500,349,500,392,1500,34
9
370 GOTO 370
```

```
100 REM EXTENDED BASIC
110 REM BY PETER LEES
120 REM YOUNGER SET MEMBER
130 CALL CLEAR :: CALL SCREE
N(2)
140 FOR G=1 TO 13 :: CALL CO
LOR(G,16,1):: NEXT G
150 DISPLAY AT(11,4):"A NAME
 WE KNOW AND TRUST"
160 FOR N=1 TO 1000 :: NEXT
N
170 CALL CLEAR
180 CALL CHAR(96,"FFFFFFFFFF
FFFFFF")
190 CALL CHAR(104,"FFFFFFFFF
FFFFFFF")
200 CALL COLOR(9,1,1):: CALL
 COLOR(10,1,1)
210 FOR R=2 TO 14 :: CALL HC
HAR(R,10,96,5):: NEXT R
220 FOR R=14 TO 19 :: CALL H
CHAR(R,14,96,7):: NEXT R
230 FOR R=7 TO 14 :: CALL HC
HAR(R,23,96,3):: NEXT R
240 FOR R=8 TO 10 :: CALL HC
HAR(R,7,96,3):: CALL HCHAR(R
+1,26,96,3):: CALL HCHAR(R+1
2,R+7,96,6)
250 CALL HCHAR(R+3,15,96,2):
: CALL HCHAR(R+6,21,96,2)::
NEXT R
260 FOR R=7 TO 8 :: CALL HCH
AR(R,15,96,2):: CALL HCHAR(R
,21,96,2):: NEXT R
270 CALL HCHAR(15,11,96,14):
: CALL HCHAR(11,21,96,5):: C
ALL VCHAR(8,6,96,2)
280 CALL VCHAR(11,9,96,2)::
CALL VCHAR(11,21,96,5)
290 CALL VCHAR(12,22,96,2)::
 CALL HCHAR(7,15,96):: CALL
HCHAR(9,29,96)
300 CALL MAGNIFY(3)
310 FOR R=6 TO 11 :: CALL HC
HAR(R,18,104,2):: NEXT R
320 CALL CHAR(100,"00030F1F1
F3F3F7F7F3F3F1F1F0F030000C0F
0F8F8FCFCFEFEFCFEF8F0C000")
330 CALL COLOR(9,13,1):: CAL
L COLOR(10,9,1)
340 CALL SPRITE(#1,100,9,240
,137)
350 CALL MOTION(#1,5,0)
360 FOR I=1 TO 40 :: CALL SO
UND(I,200,1):: NEXT I
370 CALL MOTION(#1,0,0)
380 GOTO 380
```

```
100 ! ****************
110 ! *ON THE INSIDE*
120 ! ****************
130 ! TISHUG LIBRARY 110
140 ! EXTENDED BASIC
150 ! BY ALLAN CASWELL
160 ! VERSION 1
170 CALL PIC
180 CO=982 :: DO=1102 :: EO=
1237 :: GO=1471 :: FO=1311 :
: A4=880 :: B4=988 :: C4=104
7 :: D4=1175
190 A1=110 :: B1=123 :: C1=1
31 :: D1=147 :: E1=165 :: F1
=175 :: G1=196
200 A2=220 :: B2=247 :: C2=2
62 :: CS2=277 :: D2=294 :: E
2=330 :: F2=349 :: G2=392
210 A3=440 :: B3=494 :: C3=5
23 :: CS3,DF3=554 :: D3=587
:: E3=659 :: F3=698 :: G3=78
4 :: R=40000
220 M=1400 :: C=700 :: Q=350
 :: SQ=175 :: DQ=525
230 GOTO 290
240 CALL SOUND(T,S1,V1):: RE
TURN
250 CALL SOUND(T.S1,V1.S2,V2
):: RETURN
260 CALL SOUND(T,S1,V1,S2,V2
,S3,V3):: RETURN
270 CALL SOUND(T,S1,V1,S3,30
,S2,30,-4,V2):: RETURN
280 CALL SOUND(T,S1,V1,S2,V2
,S3,30,-4,V3):: RETURN
290 !
300 T=Q :: V1=6 :: V2=14 ::
V3=14 :: S2=C2 :: S3=C1
310 !@P-
320 S1=R :: GOSUB 260 :: S1=
C3 :: GOSUB 260 :: S1=F3 ::
GOSUB 260
330 S1=A3 :: GOSUB 260 :: T=
C :: S1=G2 :: S2=C2 :: S3=E1
 :: GOSUB 260 :: S1=R :: S2=
C2 :: S3=C1 :: T=Q :: GOSUB
260
340 S1=G2 :: GOSUB 260 :: GO
SUB 260 :: GOSUB 260 :: S1=F
2 :: GOSUB 260 :: S1=E2 :: G
OSUB 260 :: S1=D2 :: S2=D1 :
: S3=DO :: GOSUB 280
350 T=C+Q :: S1=F2 :: S2=D2
:: S3=A1 :: GOSUB 260 :: T=Q
 :: S3=F1 :: GOSUB 260 :: S3
=E1 :: GOSUB 260 :: S3=D1 ::
 GOSUB 260
360 S3=A1 :: GOSUB 260 :: T=
C :: S1=D2 :: S2=B2 :: S3=GO
 :: GOSUB 280 :: T=DQ :: S3=
D1 :: GOSUB 260 :: S1=G2 ::
S2=B2 :: T=SQ :: GOSUB 250
370 T=Q :: GOSUB 250 :: S1=F
2 :: S2=A2 :: GOSUB 250 :: G
OSUB 250 :: S1=E2 :: S2=G1 :
: GOSUB 250 :: T=M :: S2=G1
:: S3=CO :: GOSUB 280
380 T=Q :: S3=D1 :: GOSUB 26
0 :: S3=C1 :: GOSUB 260 :: G
OSUB 260 :: S3=B1 :: GOSUB 2
60 :: T=C+DQ :: S2=C2 :: S3=
A1 :: GOSUB 260
390 T=SQ :: S3=E1 :: GOSUB 2
60 :: S3=A2 :: T=Q :: GOSUB
260 :: S1=F2 :: GOSUB 260 ::
 S1=G2 :: GOSUB 260 :: S1=C2
 :: GOSUB 260
400 S1=D2 :: S2=F1 :: S3=DO
:: GOSUB 280 :: T=C+Q :: S1=
F2 :: S2=D2 :: S3=A1 :: GOSU
B 260 :: T=Q :: S3=F1 :: GOS
UB 260
410 S3=E1 :: GOSUB 260 :: S3
=D1 :: GOSUB 260 :: S3=A1 ::
 GOSUB 260 :: T=C :: S1=D2 :
: S2=B1 :: S3=DO :: GOSUB 28
0
420 T=DQ :: S3=D1 :: GOSUB 2
60 :: T=SQ :: S1=D2 :: S2=D2
 :: S3=F1 :: GOSUB 260 :: T=
Q :: S2=B2 :: GOSUB 260 :: S
1=E2 :: S3=G1 :: GOSUB 260
430 S1=F2 :: S2=A2 :: GOSUB
250 :: S1=G2 :: S2=B2 :: GOS
UB 250 :: T=M :: S1=E2 :: S2
=C2 :: S3=C1 :: GOSUB 260 ::
 S3=D1 :: T=Q :: GOSUB 260
440 S3=E1 :: GOSUB 260 :: S3
=D1 :: GOSUB 260 :: S3=A1 ::
 GOSUB 260 :: S1=G1 :: S2=C2
 :: S3=EO :: T=C :: GOSUB 28
0 ::
450 T=Q :: S1=R :: S2=C2 ::
S3=C1 :: GOSUB 260 :: S1=G2
:: GOSUB 260 :: GOSUB 260 ::
 GOSUB 260 :: S1=F2 :: GOSUB
 260 :: S1=E2 :: GOSUB 260
460 !@P+
470 FOR VER=1 TO 2 :: S1=D2
:: S2=D1 :: GOSUB 250 :: T=C
+Q :: S1=F2 :: S2=A2 :: S3=D
1 :: GOSUB 260 :: T=Q :: S3=
F1 :: GOSUB 260
480 !@P-
490 S3=E1 :: GOSUB 260 :: S3
=D1 :: GOSUB 260 :: S3=A1 ::
 GOSUB 260 :: T=C+Q :: S1=D2
 :: S2=B2 :: S3=GO :: GOSUB
280
500 T=SQ :: S1=G2 :: S2=GO :
: GOSUB 270 :: GOSUB 270 ::
T=Q :: S2=B2 :: GOSUB 280 ::
 S1=F2 :: GOSUB 280 :: GOSUB
 280 :: S1=E2 :: GOSUB 280
510 T=M :: S1=E2 :: S2=G1 ::
 S3=CO :: GOSUB 280 :: T=Q :
: S3=D1 :: GOSUB 260 :: S3=C
1 :: GOSUB 260 :: GOSUB 260
:: S3=B1 :: GOSUB 260
520 T=C+Q :: S2=C2 :: S3=A1
:: GOSUB 260 :: T=SQ :: S1=G
2 :: S2=A1 :: GOSUB 250 :: G
OSUB 250 :: S2=C2 :: S3=A1 :
: T=Q :: GOSUB 260
530 GOSUB 260 :: S1=F2 :: GO
SUB 260 :: S1=E2 :: GOSUB 26
0 :: S1=D2 :: S2=DO :: GOSUB
 270 :: T=C+Q :: S1=A3 :: S2
=F2 :: S3=DO :: GOSUB 280
540 T=C :: S2=D3 :: GOSUB 28
0 :: S2=A4 :: GOSUB 280 :: S
2=G3 :: S3=G1 :: T=C+DQ :: G
OSUB 260 :: T=SQ :: S1=B2 ::
 S2=G1 :: GOSUB 250
550 GOSUB 250 :: S1=C2 :: GO
SUB 250 :: T=C :: S1=D2 :: G
OSUB 250 :: T=Q :: S1=C2 ::
GOSUB 250 :: S1=G1 :: S2=E1
:: S3=CO
560 T=C+Q :: GOSUB 280 :: V1
=2 :: V2=10 :: V3=10
570 S1=C3 :: S2=G1 :: S3=CO
:: T=SQ :: GOSUB 280 :: GOSU
B 280 :: T=Q :: GOSUB 280 ::
 T=C :: S1=B3 :: GOSUB 280 :
: T=SQ :: S1=A3 :: GOSUB 280
580 S1=C3 :: GOSUB 280
590 T=C+Q :: S2=A2 :: S3=FO
:: GOSUB 280 :: S1=B3 :: T=Q
 :: GOSUB 280 :: S1=A3 :: S2
=C2 :: T=C+Q :: GOSUB 280
600 T=Q :: S1=E3 :: S2=FO ::
 GOSUB 270 :: T=C :: S1=D3 :
: S2=B2 :: S3=GO :: GOSUB 28
G :: T=Q :: S1=B3 :: S2=GO :
: GOSUB 270 :: GOSUB 270
610 S2=D2 :: GOSUB 280 :: T=
SQ :: S1=A3 :: GOSUB 280 ::
T=DQ :: S1=B3 :: GOSUB 280 :
: T=Q :: S1=A3 :: GOSUB 280
:: S1=G2 :: S2=B1 :: S3=EO
620 GOSUB 280 :: T=M+C :: GO
SUB 280 :: T=Q :: S1=E2 :: S
2=D1 :: GOSUB 280 :: S1=A3 :
: S2=CS2 :: S3=A1 :: T=C+Q :
: GOSUB 260
630 T=SQ :: S1=G2 :: S2=R ::
 GOSUB 260 :: GOSUB 260 :: T
=Q :: S2=CS2 :: GOSUB 260 ::
 GOSUB 260 :: S1=F2 :: GOSUB
 260 :: S1=E2 :: GOSUB 260
640 !@P+
650 IF FLAG=1 THEN RETURN
660 !@P-
670 S1=D2 :: S2=D1 :: GOSUB
250 :: T=C :: S1=A3 :: S2=F2
 :: S3=D1 :: GOSUB 260 :: GO
SUB 260 :: S2=F2 :: T=Q :: G
OSUB 260 :: S2=A3
680 GOSUB 260 :: S2=C3 :: GO
SUB 260 :: T=C+Q :: S1=D3 ::
 S2=B2 :: S3=GO :: GOSUB 280
 :: T=Q :: S1=B2 :: S2=GO ::
 GOSUB 270 :: GOSUB 270
690 !@P+
700 S1-C2 :: GOSUB 270 :: S1
=D2 :: GOSUB 270 :: S1=C2 ::
 GOSUB 270 :: IF VER=2 THEN
780
710 !@P-
720 T=C+Q :: S1=C2 :: S2=G1
:: S3=C1 :: GOSUB 260 :: T=Q
 :: S2=E2 :: GOSUB 260 :: S2
=F2 :: GOSUB 260 :: S2=G2 ::
 GOSUB 260 :: S2=A3
730 GOSUB 260
740 S2-C2 :: GOSUB 260 :: S1
=R :: S2=E2 :: S3=CO :: GOSU
B 280 :: V1=6 :: V2=14 :: V3
=14 :: S1=G2 :: S2=CO :: GOS
UB 270 :: GOSUB 270
750 !@P+
760 T=SQ :: GOSUB 270 :: T=D
Q :: GOSUB 270 :: T=SQ :: GO
SUB 270 :: T=DQ :: S1=F2 ::
GOSUB 270 :: T=Q :: S1=E2 ::
 GOSUB 270 :: NEXT VER
770 !@P-
780 T=C+Q :: S1=C2 :: S2=G1
:: S3=C1 :: GOSUB 260 :: V1=
2 :: V2=10 :: V3=10 :: FLAG=
1 :: T=SQ :: S1=C3 :: S2=G2
:: S3=C1 :: GOSUB 260
790 GOSUB 260 :: T=Q :: GOSU
B 260 :: T=C :: GOSUB 260 ::
 T=Q :: GOSUB 260 :: GOSUB 5
90
800 S1=D2 :: S2=D1 :: GOSUB
250 :: S1=A3 :: S2=F2 :: S3=
D1 :: T=C :: GOSUB 260 :: GO
SUB 260 :: S2=F2 :: T=Q :: G
OSUB 260 :: S2=A3 :: GOSUB 2
60
810 S2=C3 :: GOSUB 260 :: T=
C+Q :: S1=D3 :: S2=B2 :: S3=
GO :: GOSUB 280 :: T=Q :: S1
=B2 :: S2=GO :: GOSUB 270 ::
 GOSUB 270
820 S1=C2 :: GOSUB 270 :: S1
=D2 :: GOSUB 270 :: S1=C2 ::
 GOSUB 270 :: T=C+Q :: S2=G2
 :: S3=C1 :: GOSUB 260 :: T=
Q :: S2=E1 :: GOSUB 260
830 S2=F2 :: GOSUB 260 :: S2
=G2 :: GOSUB 260 :: S2=A3 ::
 GOSUB 260 :: S2=C2 :: GOSUB
 260 :: T=C :: S1=E2 :: S2=E
1 :: S3=CO :: GOSUB 280
```

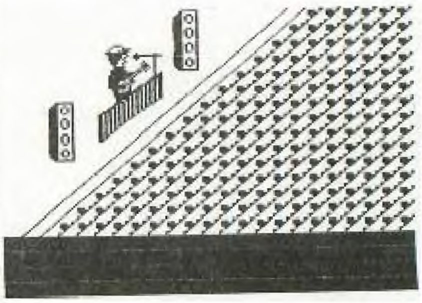
↗

```
840 S1=R :: S2=R :: T=Q :: G
OSUB 280 :: S1=G2 :: V1=6 ::
 V2=14 :: V3=14 :: GOSUB 280
 :: GOSUB 280 :: GOSUB 280 :
: S1=F2 :: GOSUB 280
850 S1=E2 :: GOSUB 280 :: S1
=D2 :: S2=D2 :: S3=D0 :: GOS
UB 280 :: T=M :: S1=F2 :: GO
SUB 280 :: S2=C4 :: T=Q :: G
OSUB 280 :: T=SQ
860 S2=F3 :: GOSUB 280 :: S2
=A4 :: GOSUB 280 :: GOSUB 28
0 :: S2=F3 :: GOSUB 280 :: T
=C+DQ :: S1=D3 :: S2=B2 :: S
3=G0 :: GOSUB 280
870 T=SQ :: S1=G2 :: S2=G0 :
: GOSUB 270 :: T=Q :: GOSUB
270 :: S1=F2 :: GOSUB 270 ::
 GOSUB 270 :: S1=E2 :: GOSUB
 270 :: T=M
880 S1=E2 :: S2=G1 :: S3=C1
:: GOSUB 260 :: T=SQ :: S2=E
3 :: GOSUB 260 :: T=DQ :: S2
=G3 :: GOSUB 260 :: T=SQ ::
S2=D3 :: GOSUB 260
890 S2=G3 :: GOSUB 260 :: S2
=DF3 :: GOSUB 260 :: S1=G3 :
: GOSUB 260 :: T=C+DQ :: S1=
C3 :: S2=E2 :: S3=A1 :: GOSU
B 260 :: T=SQ
900 S1=G2 :: S2=A1 :: GOSUB
250 :: T=Q :: GOSUB 250 :: G
OSUB 250 :: S1=F2 :: GOSUB 2
50 :: S1=E2 :: GOSUB 250 ::
S1=D2 :: S2=D0 :: GOSUB 270
910 T=M+C+Q :: S1=A3 :: S2=F
2 :: S3=D0 :: GOSUB 280 :: T
=C+Q :: S1=R :: S2=G0 :: GOS
UB 270 :: S1=B2 :: T=Q :: GO
SUB 270 :: GOSUB 270
920 S1=C2 :: GOSUB 270 :: S1
=D2 :: GOSUB 270 :: S1=B2 ::
 GOSUB 270 :: T=M :: S1=C2 :
: S2=C0 :: GOSUB 270 :: T=Q
:: S2=E1 :: GOSUB 280
930 S2=F2 :: GOSUB 280 :: S2
=G2 :: GOSUB 280 :: S2=A3 ::
 GOSUB 280 :: S2=C2 :: GOSUB
 280 :: T=2*M :: S1=G3 :: S2
=E2 :: S3=C1 :: GOSUB 260
940 !@P+
950 FOR WAIT=2000 TO 3500 ::
 NEXT WAIT
960 STOP
970 SUB PIC
980 CALL CLEAR :: CALL SCREE
N(16):: DISPLAY AT(2,10):"PR
ISONER." :: DISPLAY AT(3,7):
"'ON THE INSIDE'"
990 CALL CHAR(96,"FFFFFFFFFF
FFFFFF",97,"1818181818181818
",98,"00183C7E7E3C1800"):: C
ALL COLOR(9,2,16)
1000 FOR A=5 TO 24 :: CALL H
CHAR(A,11,96,11):: NEXT A ::
 FOR A=7 TO 10 :: CALL HCHAR
(A,14,97,5):: NEXT A :: CALL
 HCHAR(15,20,98)
1010 CALL CHAR(104,"80402010
08040201",105,"000000000000
000",106,"0000000000000000",
107,"875F3F7F7FFFFFFF")! LT
SUN
1020 CALL CHAR(108,"80808080
80808080",110,"0102040810204
080",109,"E1FAFCFEFEFFFFFF",
111,"00000000000000FF")! RT
SUN
1030 CALL CHAR(112,"FF000000
00000000",114,"FFFFFF7F7F3F5
F87",113,"0102040810204080",
115,"0101010101010101")! LB
SUN
1040 CALL CHAR(116,"FFFFFFFE
FEFCFAE1",117,"000000000000
000",118,"0000000000000000",
119,"8040201008040201")
1050 CALL MAGNIFY(4):: CALL
SPRITE(#1,104,12,32,184):: C
ALL SPRITE(#2,108,12,32,216)
:: CALL SPRITE(#3,112,12,64,
184)
1060 CALL SPRITE(#4,116,12,6
4,216)
1070 CALL CHAR(120,"00002073
7F7FFFFF",122,"0001078FCFFFF
FFF",121,"7F3F1F7F7F7F7F7F",
123,"FFFFFFFFFFFFFFFF")
1080 CALL CHAR(124,"00C0E0E0
F3FBFFFF",126,"000040E4EEFEF
EFC",125,"FFFFFFFFFFFFFFFF",
127,"F8F0C0E0FCFEFEFE")
1090 CALL CHAR(128,"FFFE7C19
01404404",130,"FFFEFCF872020
888",129,"0020280901404404",
131,"8002222808424200")
1100 CALL CHAR(132,"FFFF7F3F
1E808202",134,"FFFCF28208484
101",133,"5051010444411110",
135,"0828220208082020")
1110 CALL SPRITE(#5,120,15,3
2,5):: CALL SPRITE(#6,124,15
,32,37):: CALL SPRITE(#7,128
,15,64,5):: CALL SPRITE(#8,1
32,15,64,37)
1120 SUBEND
```

```
100 REM *************
110 REM *READ-A-DISK*
120 REM *************
130 REM TISHUG LIBRARY
140 REM TI BASIC
150 REM AUTHOR UNKNOWN
160 CALL CLEAR
170 DIM TYPE$(5)
180 TYPE$(1)="DIS/FIX"
190 TYPE$(2)="DIS/VAR"
200 TYPE$(3)="INT/FIX"
210 TYPE$(4)="INT/VAR"
220 TYPE$(5)="PROGRAM"
230 REM A=1
240 REM GOTO 200
250 INPUT "MASTER DISK[1-3]?
 ":A
260 A=INT(A)
270 IF A<1 THEN 250
280 IF A>3 THEN 250
290 OPEN #1:"DSK"&STR$(A)&".
",INPUT ,RELATIVE,INTERNAL
300 INPUT #1:A$,J,J,K
310 DISPLAY :"DSK"&STR$(A);"
 -DISKNAME= ";A$:"AVAILABLE=
";K;"USED=";J-K
320 DISPLAY :" FILENAME  SIZ
E   TYPE     P":"____________
____________":
330 FOR LOOP=1 TO 127
340 INPUT #1:A$,A,J,K
350 IF LEN(A$)=0 THEN 450
360 DISPLAY A$;TAB(12);J;TAB
(17);TYPE$(ABS(A));
370 IF ABS(A)=5 THEN 400
380 B$=" "&STR$(K)
390 DISPLAY SEG$(B$,LEN(B$)-
2,3);
400 IF A>0 THEN 430
410 DISPLAY TAB(28);"Y";
420 GOTO 440
430 DISPLAY TAB(28);" "
440 NEXT LOOP
450 CLOSE #1
460 FOR Z=1 TO 1000
470 NEXT Z
480 END
```

```
100 REM ******************
110 REM *DARLIK DESTROYER*
120 REM ******************
130 REM TISHUG LIBRARY
140 REM TI BASIC
150 REM BY JOHN CORNISH
160 CALL CLEAR
170 REM  TITLE SCREEN
180 CALL SCREEN(12)
190 PRINT TAB(4);"----------
----------":TAB(4);"-";TAB(2
3);"-"
200 PRINT TAB(4);"- DARLIK D
ESTROYER -":TAB(4);"-";TAB(2
3);"-":TAB(4);"-------------
-------": : : : : : : :
210 FOR DELAY=1 TO 500
220 NEXT DELAY
230 PRINT TAB(4);"WANT INSTR
UCTIONS (Y/N)?"
240 CALL KEY(0,K,S)
250 IF S=0 THEN 240
260 IF K<>89 THEN 280
270 GOSUB 2400
280 CALL CLEAR
290 REM  DEFINE CHARACTERS
300 CALL CHAR(105,"183C7F7C7
C7CFEFE")
310 CALL CHAR(113,"1818FE3C3
C3C2424")
320 CALL CHAR(114,"C32B341F1
F644898")
330 CALL CHAR(121,"920114B13
83CFDFE")
340 CALL CHAR(129,"00183C7E7
E7E3C18")
350 CALL CHAR(137,"0000FF00"
)
360 CALL CHAR(145,"000007070
707")
370 CALL CHAR(146,"18183C7E7
F7F3C18")
380 CALL CHAR(152,"FFFFFFFFF
FFFFFFF")
390 REM  DEFINE COLORS
400 CALL COLOR(10,14,1)
410 CALL COLOR(11,8,1)
420 CALL COLOR(12,12,7)
430 CALL COLOR(13,7,12)
440 CALL COLOR(14,16,1)
450 CALL COLOR(15,3,1)
460 CALL COLOR(16,2,1)
470 FOR COL=1 TO 9
480 CALL COLOR(COL,16,1)
490 NEXT COL
500 REM  SET VARIABLES
510 LIV=5
520 SC=0
530 SPD=2
540 GRN=32
550 BLD=30
560 RANDOMIZE
570 MR=INT((20*RND)+1)+3
580 CX$=" "&STR$(SC)
590 GO$="G A M E  O V E R"
600 SR$="SCORE= "
610 REM  START GAME
620 CALL SCREEN(2)
630 C=C+2
640 FOR RD=2 TO 23 STEP SPD
650 IF C=30 THEN 660 ELSE 67
0
660 GOTO 1750
670 CALL HCHAR(1,26,113,LIV)
680 CALL HCHAR(24,32-BLD,146
,BLD)
690 CALL HCHAR(RD,C,105)
700 CALL HCHAR(MR,30,113)
710 FOR M=1 TO LEN(CX$)
720 CALL HCHAR(1,4+M,ASC(SEG
$(CX$,M,1)))
730 NEXT M
740 IF RD=MR+1 THEN 770
750 IF RD=MR-1 THEN 770
```

```
760 IF RD=MR THEN 780 ELSE 7
90
770 GOSUB 2100
780 GOSUB 1260
790 GOSUB 2100
800 CALL HCHAR(RD,C,154)
810 NEXT RD
820 C=C+2
830 FOR RD=23 TO 2 STEP -SPD
840 CALL HCHAR(1,26,113,LIV)
850 CALL HCHAR(24,32-BLD,146
,BLD)
860 CALL HCHAR(RD,C,105)
870 CALL HCHAR(MR,30,113)
880 FOR M=1 TO LEN(CX$)
890 CALL HCHAR(1,4+M,ASC(SEG
$(CX$,M,1)))
900 NEXT M
910 IF RD=MR+1 THEN 940
920 IF RD=MR-1 THEN 940
930 IF RD=MR THEN 950 ELSE 9
60
940 GOSUB 2100
950 GOSUB 1260
960 GOSUB 2100
970 CALL HCHAR(RD,C,154)
980 NEXT RD
990 GOTO 630
1000 REM  GRENADE
1010 CALL SOUND(-100,3000,0)
1020 FOR ST=29 TO C STEP -8
1030 CALL HCHAR(MR,ST,145)
1040 CALL HCHAR(MR,ST,156)
1050 CALL HCHAR(RD,C,105)
1060 IF MR=RD THEN 1070 ELSE
 1080
1070 IF ST=FR THEN 1090
1080 NEXT ST
1090 BLD=BLD-1
1100 CALL HCHAR(24,1,152,32-
BLD)
1110 CALL HCHAR(24,32-BLD,14
6,BLD)
1120 IF BLD=1 THEN 1460
1130 IF ST=FR THEN 1150
1140 IF MR=RD THEN 1160
1150 RETURN
1160 CALL HCHAR(RD,C,121)
1170 CALL SOUND(-1000,-6,0)
1180 GOSUB 2010
1190 GOSUB 1830
1200 C=0
1210 MR=INT((10*RND)+1)*2
1220 CALL CLEAR
1230 GOTO 630
1240 REM
1250 REM  DARLIK LASER
1260 CALL SOUND(-90,110,1)
1270 FOR FR=(C+1)TO 30 STEP
19
1280 CALL SOUND(90,-1,0,4000
,4)
1290 CALL HCHAR(RD,FR,137,4)
1300 CALL HCHAR(RD,FR,154,4)
1310 CALL HCHAR(RD,C,105)
1320 CALL HCHAR(MR,30,113)
1330 IF FR=30 THEN 1360
1340 GOSUB 2100
1350 NEXT FR
1360 IF RD=MR THEN 1380
1370 RETURN
1380 CALL SOUND(1000,-6,0,30
00,5)
1390 FOR D=1 TO 50
1400 CALL HCHAR(MR,30,114)
1410 NEXT D
1420 CALL HCHAR(MR,30,129)
1430 GOSUB 1460
1440 RETURN
1450 REM  LIVES & GRENADES
1460 LIV=LIV-1
1470 IF LIV<1 THEN 2170
1480 CALL CLEAR
1490 CALL HCHAR(1,26,113,LIV
)
1500 FOR REP=30 TO 1 STEP -1
1510 GRN=GRN-1
1520 CALL HCHAR(24,GRN,146)
1530 CALL SOUND(80,800,1)
1540 NEXT REP
1550 BLD=30
1560 GRN=32
1570 RETURN
1580 REM  UP
1590 CALL HCHAR(RD,C,105)
1600 CALL HCHAR(MR,30,113)
1610 CALL HCHAR(MR,30,153)
1620 MR=MR-2
1630 IF MR<=2 THEN 1700
1640 CALL HCHAR(MR,30,113)
1650 RETURN
1660 REM  DOWN
1670 CALL HCHAR(RD,C,105)
1680 CALL HCHAR(MR,30,113)
1690 CALL HCHAR(MR,30,153)
1700 MR=MR+2
1710 IF MR>=23 THEN 1620
1720 CALL HCHAR(MR,30,113)
1730 RETURN
1740 REM  DARLIK INVADES
1750 CALL SOUND(1000,-5,0,11
0,3)
1760 CALL SOUND(100,-1,4)
1770 IF RD<>MR THEN 1790
1780 GOTO 670
1790 CALL SOUND(1000,-5,0,20
0,0)
1800 CALL HCHAR(MR,30,129)
1810 GOTO 2170
1820 REM  EXTRA MAN
1830 IF SC=200 THEN 1870
1840 IF SC=500 THEN 1910
1850 IF SC=800 THEN 1960
1860 IF SC=1000 THEN 1980 EL
SE 1990
1870 CALL SOUND(1000,500,0,7
00,1)
1880 LIV=LIV+1
1890 GOSUB 1470
1900 GOTO 1990
1910 CALL SOUND(1000,2000,0)
1920 FOR EPT=1 TO 10
1930 GOSUB 2010
1940 NEXT EPT
1950 GOTO 1990
1960 SPD=4
1970 GOTO 1990
1980 SPD=5
1990 RETURN
2000 REM  SCORE COUNT
2010 SC=SC+10
2020 CX$=" "&STR$(SC)
2030 FOR M=1 TO LEN(CX$)
2040 CALL HCHAR(1,4+M,ASC(SE
G$(CX$,M,1)))
2050 NEXT M
2060 CALL SOUND(80,900,0)
2070 C=0
2080 RETURN
2090 REM  PLAYERS INPUT
2100 CALL KEY(0,X1,Y1)
2110 IF Y1=0 THEN 2150
2120 IF X1=69 THEN 1580
2130 IF X1=32 THEN 1000
2140 IF X1=88 THEN 1660
2150 RETURN
2160 REM  GAME OVER
2170 FOR TN=1 TO 10
2180 CALL SOUND(100,110,1)
2190 CALL SOUND(100,220,1)
2200 NEXT TN
2210 FOR M=1 TO LEN(GO$)
2220 CALL HCHAR(13,6+M,ASC(S
EG$(GO$,M,1)))
2230 NEXT M
2240 CALL SOUND(1000,400,1)
2250 SR$="SCORE IS "&STR$(SC
)
2260 FOR M=1 TO LEN(SR$)
2270 CALL HCHAR(16,8+M,ASC(S
EG$(SR$,M,1)))
2280 NEXT M
2290 FOR DELAY=1 TO 100
2300 NEXT DELAY
2310 PRINT TAB(7);"ANOTHER G
AME ?"
2320 CALL KEY(0,K,S)
2330 IF S=0 THEN 2320
2340 IF K=89 THEN 2360
2350 END
2360 CALL CLEAR
2370 C=0
2380 GOTO 510
2390 REM   INSTRUCTIONS
2400 CALL CLEAR
2410 PRINT "THE DARLIKS ARE
TRYING TO":"INVADE THE EARTH
."
2420 PRINT " USING CURSOR KE
YS E & X,TO":"MOVE YOUR MAN
UP OR DOWN,":"OR SPACEBAR TO
 THROW A"
2430 PRINT "GRENADE,YOU HAVE
 TO TRY AND":"HIT THE DARLIK
 WHICH IS":"CROSSING UP AND
DOWN THE"
2440 PRINT "SCREEN.": :" YOU
 HAVE 5 MEN, AND 30 ":"GRENA
DES FOR EACH MAN."
2450 PRINT "IF THE DARLIK SH
OOTS YOU,YOU":"LOOSE 1 MAN A
ND THE GRENADES":"ARE REPLEN
ISHED. 10 POINTS"
2460 PRINT "ARE AWARDED FOR
EACH DARLIK":"HIT AND A BONU
S SCORE AT 300": : :
2470 PRINT "  PRESS ANY KEY
TO CONTINUE"
2480 CALL KEY(0,K,S)
2490 IF S=0 THEN 2480
2500 CALL CLEAR
2510 PRINT :" THE GAME START
S GETTING":"HARDER AT 800 AS
 THE DARLIK"
2520 PRINT "SPEEDS UP..."
2530 PRINT :" IF THE DARLIK
INVADES THE":"EARTH THE GAME
 IS OVER": : : :
2540 PRINT "  PRESS ANY KEY
TO BEGIN":
2550 CALL KEY(0,K,S)
2560 IF S=0 THEN 2550
2570 RETURN
```



WHAT IS A RAM DISK.

By Roger Biddle (OH-MI-TI) (NW OHIO 99'er NEWS JUNE '85)

I had heard of them before, but they were never available for the TI 99/4A, except for the hard work of Ron Gries and John Clulow.

Thats right!! Ron has developed a RAM DISK CARD that functions as a disk drive and John is currently developing software for it. At present, 75% of all the DISK MANAGER 2 functions work, and Johns SECTOR COPY PROGRAM doesn't only work, but works at a speed you can only believe when you see it.

I had the opportunity to to see a demonstration, given by Ron. In a word, it is "incredible".

If you use the SECTOR COPY program, use the RAM DISK CARD for the copy disk. It initialises the card in about 1 second and writes 1/4 of the master disk information in about 1.4 seconds including verification.

Just imagine the speed of TI-WRITER files or EDITOR ASSEMBLER files loaded into the RAM DISK. When you LF from the RAM DISK, instantly, they will be on the screen. They should be able to go between the EDITOR ---- FORMATER ---- ASSEMBLER..... BOOM !! Instantly !!

The RAM DISK CARD requires assembly by the individual. But Ron hopes to have a printed circuit board developed that will greatly reduce the task of building the card. The cost will be about $100. Just think, a second disk drive without the possibility of failing, parts that sit in an open slot in your PE BOX.

John and Ron hope to have several programs ready for the SWAPMEET in June at NEW HORIZONS. I'm sure you'll be interested in seeing it.

I'm looking foreward to having one of these RAM DISK CARDS soon. I'm hoping that the PC board will be developed. The project involves some cost that has to be researched by Ron. So as I said, that is pending. (Even if it doesn't happen, I plan to build one by the old-fashioned point to point wiring method.)

If you are interested in having a RAM DISK CARD, please contact Ron or John. The more people that are interested, the better idea Ron and John will have of developing a PC board.

=================

PERSONAL RECORD KEEPING.

By Robert Hamsher (AIRPORT AREA COMPUTER CLUB)

Some notes on the personal record keeping module (PRK)

From personal experience with the PRK, I have a few words of wisdom.

1) Believe the manual when it says not to use all available records. It says to leave about 2% of the records unused. So if there are 300 records available, only input a maximum of 294. If you do use all the 300 records, leaving 0 records available, your console can lock up and/or cause your TV screen to go crazy.

2) If you are working on cassette, and later want the data printed by printer, then leave even more records available. reduce the total file size by about 30%. If there are 300 records available, only use 200. If too many records are used, then when you try to print the file, you will get a message like -5 records available and a console lock/up.

3) If you already have a PRK file, and you want to add a new field to the file, you are out of luck. However, if you convert the PRK file to a BASIC file and back again, you can get around this.

TI WRITER SUGGESTIONS

by Bill Browning - MSP User Group.

This is the second of the articles on ways to increase the ease and efficiency of your TI WRITER word processing program.

In the first article, I talked about a way to use Replace String to make your work easier. This article covers another of those handy features.

Word processing programs can be grouped into two main categories:-

1) symbol formatted, and
2) what-you-see-is-what-you-get.

With the symbol formatted kind,the user embeds symbols in the text to tell the printer how to format the text. With the what-you-see kind, the screen displays the text in essentially the same way that it will appear in print.

TI WRITER is somewhat unusual, because you can use both methods. "Fill" (.FI) is equivalent to symbol formatting, and "no fill" (.NF) is a what-you-see method.

The what-you-see method usually has an 80-column format to accomodate the the width found on a typical 8-1/2 X 11 sheet. But if you want 80-columns on the TI WRITER, you have to look at them section by section in three overlapping windows. Most users find this to be extremely bothersome, because because when you are reviewing what you have written,you cannot see an entire block of material at one time.

To avoid windowing, you can use the .FI, or symbol formatting, mode. You simple set up the left and right margins on the TAB line to fill one screen's width. For example, for a typical TV screen, you might set the left margin at "2", and the right at "39".

This setup gives you a 38 column display. On some sets, a slightly wider or narrower setup may be optimal. (For simplicity, I will refer to this setup as a 40-column format.)

You then include formatting symbols to tell the printer to print whatever margins and indents you require. (Of course, your manual explains how to do that.)

However, the 40-column approach has a limitation that can be serious in some cases: By looking at the screen, you cannot tell where the page breaks (page endings) will occur.

therefore, without printing the document to see where the page breaks are, you have no way of eliminating widow lines.

A widow line isn't something you find at the local welfare office. It's an isolated word or line at the top or bottom of a page. It may be a sub-heading separated from the following material or it may be a line isolated from the rest of the paragraph. In any case, because widow lines look awkward, they should be avoided if possible. Therefore .....

THE CHALLENGE:

How to use the 40-column format to avoid windowing, and yet be able to preview the material on the screen to insert page breaks that will elimenate widow lines on your printout.

We are assuming that:-

1) the document is too long to be printed out in draft form, and

2) the document is important enough for you to be concerned about page breaks.

THE SOLUTION:

1) Key in your text using 40 column format for ease of typing and editing. But do not include any margin (such as .LM 12) or centering (.CE) format commands because you will be converting the format to a No Fill (.NF), what-you-see display.

At the beginning of each paragraph, key in a symbol that will not be used elsewhere in the text. A good symbol to use is the >. The reason for doing this will become clear in a moment.
Do not indent the paragraphs at this stage. Below is an example of what the beginning of a paragraph would look like:-

>Now is the time ...

2) When you are ready to print out the copy, first convert to what-you-see format. First change the left and right margins on the TAB line to the actual ones you want on the printout.
Then use the REPLACE STRING mode to automatically reformat the document. If the paragraphs are not to be indented, use />//. If they are to be indented, use />/ /, with one space symbol ( ) for each indentation space.

Press "ENTER", and "A" for all when that prompt appears. Then sit back. The file will automatically be converted from the 40-column format to the actual printing width.

3) Save the file under a different name, so that you don't lose the 40-column format, in case you want to use it later for more typing or editing. (One handy notation is to put NF,for No Fill, on the end of the file name, to distinguish it from the other one.

4) Starting from the beginning of the new file, go through and type in forced page breaks (.BP) where you want pages to end. The following table will help you to determine page breaks. The table assumes 11 inch long paper.

SPACING OF TEXT	MAXIMUM LINES
single	58
1-1/2	36
double (.LS 2)	29

For example, if you are using single spacing, you will want page breaks at or before 58 lines per page (including skipped lines). These numbers hold true whether or not you are using a header and/or footer.

When using 1-1/2 spacing, (assuming that your printer can support that spacing), set the page length (.PL) at 44 instead of the default of 66.

When you are using .LS 2 for double spacing, the printer skips 2 spaces instead of 3 when you skip a line while typing. Use the following formula when your double spaced text includes skipped lines:-

29 + (skipped lines/2) = line count

In other words, count all the skipped lines in a 29-line block. Divide that number by two, and add 29. Make the page break no later than the line following the sum.

For example, if there are 6 skipped lines in a 29 line block, divide 6 by 2, and add the result, 3, to 29. Thus you would put a forced line break following line 32, or before that if necessary to avoid a widow line.(You didn't really want to use double spacing after all, did you?)

The line numbers in TI WRITER are very helpful for locating page breaks. So is a calculator.

5) Save the document with the forced page breaks, and print it out.

On a long document, this process will take a bit of time. (Less time , however than the time required to explain it). But it does give you a way to preview the document as it will appear on the printed page.

==================

TI-PUBLIB

The TI.SHUG publication library.

It seems just the other day that I had to rush my article in order to meet Shane's publishing deadline! Well, this month it is no different!

To recap on last month. I said that I would have some definite information regarding the manner in which we would be running the library service. Unfortunately I was not able to attend the committee meeting for the RULES to be discussed and agreed. (if ever you decide to attend one of these events you will soon find out why you NEED to be present to get an OK!) However, I will ensure that by the end of September the library will be open for business.

The other matter was my request for any publications, no longer required by you, to be donated to the library. To date I am pleased to say that I have received a complete Reference manual for the TI. Professional (thanks -John), a few TI Microprocessor Data Manuals (thanks -Chris) and a promise of three other books - not directly TI. 99/4A related but nevertheless of some computing interest. Next month I will publish the details of the library holding.

Clearly, we do not have any person connected with TI.SHUG that has library experience as I have not as yet had any person come forward with the advice and guidance - much needed to set up the library in a, lets say, professional manner. Oh well, I guess I will have to press on regardless!

During the past month I have however received considerable assistance with the preparation of files for Shane's end of the business. Thanks chaps - more newsletters have just arrived so let's get into it. I would like to take this opportunity of thanking the following members:-

Chris Buttner - has prepared the files mentioned last month and is currently working on a series dealing with the PRK.

John O'Brien - I trust that you wife really appreciated the modem you purchased for her birthday? I am sure YOU will have much fun using it. John has prepared a few of the program listings and is pressing on with more.

Revell Oataway - taking a break from school work found the work mind bending but has agreed to carry on. I think he found paging through the overseas newsletters a little interesting?

Phil Roberts - now this guy is really organised! His offer of assistance comes from his wife and son! Boy, he has really got them cracking! - Thanks Family.

A few other members have just recently taken a few of the newsletters for preparation - they will receive mention next month.

Reading through this it seems like a social column! Maybe we should have one - Eh! Shane.

That's all for now. I will undertake to have the rules of the library and a listing of holdings published next month. - PROMISE!

Fred Morris.

XPERIMENTER'S XBASIC.

(Sweat (WHEW! - remember) pays off.

Previously the I/O PORT published some direct sound chip LOADing programs. First we had the BLACK BOX MEMORY LOADER (from Atlanta) and in December the symphonic WHEW6. Both created sounds that were out-of-this-world, or at the very least, the corner arcade. One wondered then if this wrinkle could be made useful - and where in the world this info would turn up.

One answer comes from the ends of the earth. Stephen Shaw, writing for the Autumn TI-TIMES in England pointed out the path. The rest was merely following 'long the different research threads:

1. I finally READ the SOUND chapter in the ED/ASM manual. AND hit the notepad with pencil and worked out TI's obtuse notation. (When in doubt, use your computer to crunch numbers.)

2. The manual won't mean a thing unless you disassemble the console ROM's sound interrupt routine. And when you do, you'll find out things TI didn't bother to document.

SIDEBAR: Long term villain, not sufficiently maligned and saved by me for the proper audience: HERB SHANZER, Manager (circa March '83) Calculator & Compact Computer Division, TI INC.
Said Herb:
"It would be impossible for us to document those (ALL features) in sufficient sanitary condition for them to be totally useful to the ... user. And it is our value judgement that, rather than restrict the flexibility of those features or hide the capabilities that we can't explain, IN VERY SIMPLE TERMS, WE WOULD PREFER ... A RELATIVELY CURSORY LEVEL OF INFORMATION AND DOCUMENTATION...." (From 99'ER HCM, April 1983. Caps are mine.) Old Herb was talking about the CC40, but he evidences a peculiar TI mind-set.

Anyway, I'm going to break with I/O PORT tradition and include a long program. But judging by other newsletters, long music listings don't seem to bother other editors, so.... But first a quick demonstration.

Stephen's article was about using the MINIMEM speech access and sound list processing and sundry approaches to using direct sound. He references a direct sound loader by Neil Lawson in SMART PROGRAMMER, something that I couldn't find.

```
100 !------------------------
--
110 !CRASH DEMO
120 !from ED/ASM manual
130 !------------------------
--
140 CHIP=-31744 :: CALL INIT
150 SRC$(0)="GROM" :: SRC$(1
)="VDP"
160 IMAGE Slist ### ###  Tri
gger ###
170 IMAGE SYSSTA  ### (####)
180 CALL DISABLE
190 CALL PEEK(-31796,A,B,C):
: PRINT USING 160:A,B,C
200 CALL LOAD(-31796,7,0,1)
210 CALL PEEK(-31747,A):: B=
1+(A=0):: PRINT USING 170:A,
SRC$(B)
220 CALL LOAD(-31747,1)
230 CALL CHAR(128,"039FE4F20
502E4F00C02E4F20A02E4F40802E
4F60602E4F60602E4F80402E4FA"
)
240 CALL CHAR(132,"0201FF00"
)
250 CALL ENABLE
260 PRINT : : : :"CRASH"
270 CALL SOUND(100,440,2)
280 PRINT "BEEP"
290 SUB ENABLE
300 CALL LOAD(-31806,0):: SU
BEND
310 SUB DISABLE
320 CALL LOAD(-31806,32):: S
UBEND
```

SUBprogram thumbnails:

1. SUB ENABLE
2. SUB DISABLE
3. SUB XOFF(VOICE)

The first two set the SOUND interrupt bit on and off. The PAD location in hex is >83C2. The high nybble (top four bits) are tested in order by the interrupt routines. The technique is simple enough, just a SLA, JOC pair. That's on a par with name dropping -- Shift Left Arithmetic sets the carry status bit and the Jump On Carry skips the appropriate routine. In other words, if the bit is equal to zero, the appropriate routine is executed.

The bits, in order and from the MSBit down are:-
- all user interrupts
- SPRITE auto motion
- SOUND list processing
- and, finally the QUIT key function.

Since these are all in the high nybble, the decimal values required to set each are: 128, 64, 32(Aha! that's DISABLE), and 16. Just add up the ones you'd like to stop.

SETTING VOICES WITH CLARITY

One would be hard pressed to make the HOW-TO of setting the sound chip's three voices plus noise any more obtuse than the ED/ASM manual makes it. Try these rules of thumb:

THE HIGH NYBBLE OF THE FIRST BYTE SENT SPECIFIES THE VOICE
Each one has a 'name': Voice1 is named >8, Voice2 is >A, Voice3 is >C and the Noise is >E

TO SET A TONE:
The first nybble will be the 'name'. Tones require two bytes.

TO SELECT A TYPE OF NOISE:
The first nybble will be >E. The second nybble ranges from 0 to 7.

TO SET ANY VOLUME:
Add one to the appropriate 'name' thus >9, >B, >D, and >F for Voice1 through Noise. Maximum volume is 0. Turn off a voice with a low nybble of >F.

Since the SOUND chip gets its values just from the high byte of the data bus, you have to pass the settings in two parts, a byte at a time. The first byte always contains the name of the voice; if the high nybble is even you'll set the frequency/noise type. Notice the volume settings pair up: 8 and 9 mean voice 1, A and B mean Voice 2 and so on.

IMPORTANT: Because only the high byte is connected, that means only even bytes on CALL LOADs get sent to the SOUND chip. To LOAD from BASIC put a dummy value for the uneven bytes: CALL LOAD(-31744,A,0,B). A more elegant method might be:

CALL LOAD(-31744,A,A,B,B,C,C,D,D) etc,
Every second instance won't do anything. The SOUND chip will respond within the range of >8400 to >87FE or -31744 to -30722, so you don't need to reset the LOAD address often.

TO CALCULATE ANY GIVEN FREQUENCY:
Divide the frequency into 111860.8, then transform into low nybble and high six bits. It's easier to show the calculation than explain them:

```
A=111860.8/FREQ :: HI=(A AND 1008)/16 :: LO=(A AND 15)
```

HI is masked by 1008, hex 3F0, then dividing by 16 drops the 0 nybble. This value gets sent second. LO must have the 'name' added: 128, 160, 193 for Voice1 through Voice3. (Go ahead and try it:

CALL LOAD(-31744,128+5,0,63,0,144).
The 144 sets Voice1 to 0 or maximum volume.

TO CALCULATE WESTERN TONAL SCALES:
A musical half-step increase can be found by multiplying by (2^(1/12)). A half-step down divides the frequency by the same value. A full-step is achieved with two multiplications or two divisions.

Frequency requires two bytes: | Volume needs one byte

by the nybbles:	BYTE ONE voice 'name'	 low 4 LLLL	BYTE TWO most signif. 00HH HHHH	decimal LOAD Values Base Maximum	BYTE ONE voice 'name'	 volume	LOAD values Base
VOICE 1	>8		00	128, 63	>9		144
VOICE 2	>A	0 TO F	00 0 TO >3F	160, 63	>B		176
VOICE 3	>C		00	192, 63	>D	0 TO F	208
Noise needs one byte:							
		OTTT					
NOISE	>E	0 TO 7	——>	224	>F		240

THE SCALE THAT SHOULD HAVE BEEN

Note/ Freq.	Hex Code (NATURALS)	Dec Load (NATURALS)	Hex Code (SHARPS)	Dec Load (SHARPS)
C	503	5,3	203	2,3
B	903	9,3	503	5,5
A 1760	004	0,4	C03	12,3
G	704	7,4	304	3,4
F	005	0,5	C04	12,4
E	505	5,5	005	0,5
D	F05	15,5	A05	10,5
C	B06	11,6	506	5,6
B	107	1,7	B06	11,6
A 880	F07	15,7	807	8,7
G	F08	15,8	708	7,8
F	00A	0,10	709	7,9
E	A0A	10,7	00A	0,10
D	E0B	14,11	40B	4,11
C	60D	6,13	A0C	10,12
B	20E	2,14	60D	6,13
A 440	E0F	14,15	00F	0,15
G	D11	13,17	D10	13,16
F	014	0,20	E12	14,18
E	315	3,21	014	0,20
D	D17	13,23	816	8,22
C	C1A	12,26	419	4,25
B	51C	5,28	C1A	12,26
A 220	C1F	12,31	01E	0,30
G	B23	11,35	B21	11,33
F	128	1,40	D25	13,37
E	72A	7,42	128	1,40
D	A2F	10,47	F2C	15,44
C	735	7,53	732	7,50
B	A38	10,56	735	7,53
A 110	93F	9,63	03C	0,60

This scale puts middle A, 440, one octave lower than usual - making entering low notes on a printed F Clef possible without transposition.

TO USE:

Find the appropriate note and add the voice nybble from the table at the top.
Thus, for Voice3 set to the D note between Clefs, one adds >C to >CE0B.

For direct BASIC LOAD, 192 is added to the first byte value, 14, giving decimal LOAD pair of 206,11. The command will look like this:-

CALL LOAD(-31744,206,0,11)

Remember to put a dummy number between each half, so that the values are passed to the SOUND chip only on even addresses.

FLATS: Take the sharp value from the next lower line, excepting F and C, which take the naturals E and B. Both B to C and E to F are legitimate half-steps. (Check E# and F.)

ABOUT BLACKBIRDS (and CRASH DEMO)

This program uses the SOUND list processing interrupt to play the song. The significant advantages include:-
- true 3 voice melodies are possible
- volume can be changed independently from the frequency.
The list is processed automatically so your program may continue doing other things. Optionally , when the song/sound list is finished, you may either stop processing, start a new list or even start over. (Important: this is determined by how the list ends - your program doesn't have to keep track. The closest TI comes to documenting this feature is in PASCAL's SOUND UNIT. Stick it in yer ear, Herb.)

CRASH DEMO.
Line 200 contains "0201FF00". Ignore the >02, letting us break down the rest; 01 - bytes to follow, FF - noise off, 00 - command duration. The 00 tells the interrupt routine to STOP processing. Change line 200 to read: CALL CHAR(132,"02000700) AND RUN. NOTICE THE DIFFERENCE?

Both BLACKBIRDS and the CRASH demo use the CALL CHAR command to load a sound list into the VDP. (This is a headache if you'd like graphics with your songs. XBASIC users may use a CALL POKEV program which is not supplied herewithin to use some other part of the VDP for their list. Ofcourse, XBASIC FORTH avoids all of that..... However, we know where any character pattern is located and can readily calculate the address - A=(C+96)*8. So, in CRASH demo, (128+96)*8=1792. Converting to hex we get >700 and splitting into bytes 7,0.

These last two are what you use to set the SOUND list pointer at -31796 (>83CC). Additionally, both programs LOAD >83CE, the SOUND list trigger, with the list's address LOAD. We get away with that because we've CALLed DISABLE; otherwise one must LOAD the SYSTEM STATUS BYTE at >83FD with 1 first which tells the interrupt we've a list in the VDP. Another undocumented feature is that a 0 in >83CE signifies that the list is in GROM. Neglecting this LOAD or given the wrong order of events, we'll process GROM as a random sound list.

Mostly harmless because the SOUND routine is smart - it saves the current GROM address before (>83CE) that specifies the list is in the VDP.
(The CRASH program can show where HONK and BEEP are located. Make a SYNTAX error - type KK <enter>, put a BREAK at line 160 and run. The GROM address shown is AFTER the sound list, so subtract 3 or 4. In short order, you'll get the correct address. Now you can zap 'em with your own HONK!)

That's about it. Remember to ENABLE the sound interrupt by resetting >83C2 (-31806.) ('Reset' means setting a bit off or equal to 0. 'Set', obviously is it's inverse of turning a bit on.) One final thing: a flowchart of the SOUND interrupt. Notice how the duration and command length bytes get handled.

FLOWCHART:

begin ?ENABLED?.. (sound bit in >83C2=0?) NO: SNDEXIT

yes.. ?TRIGGERED? (>83CE greater than 0?) NO: SNDEXIT

yes.. .DECREMENT DURATION (83CE minus byte at >83FE. This is the byte that adjusts sound, cursor flash rate - (-31746))

..... ?DURATION=0? (notice EQUAL ZERO!).. NO: SNDEXIT

yes.. .FETCH NEW COUNT (from VDP or GROM depending on bit in >83FD)

..... ?COUNT=0? (zero stops this list but not processing!) YES: MORSND

no... ?COUNT=>FF (do we want to change source of list?)... YES: SNDSRC

no... .SEND DATA TO SOUND CHIP

..... .FETCH NEW DURATION

..... ?DURATION=0? (a zero here will stop sound processing)YES: SNDOFF

no... .NEWSND

SNDSRC: Toggle the VDP/GROM bit at >83FD. Fall through MORSND.

MORSND: next word + new SOUND LIST Address. SET DURATION=1. GOTO NEWSND.

SNDOFF: DURATION=0 (Yeah, TI's code is redundant! It's already zero.

NEWSND: Save list pointer in >83CC and DURATION in >83CE (also used to TRIGGER) fall through.

SNDEXT: quit key check is next, then user interrupt routine....

This article is a follow up to the series written by Stephen Shaw and published in the last 3 issues of the SND and comes to us from the LEHIGH 99'ER COMPUTER GROUP magazine Vol. 111 no. 3 - March 1985 written by Frederick Hawkins.

```
100 DATA 03AC1AB010
110 DATA 03860D9210
120 DATA 08908F08AD11C315D01
0
130 DATA 028A0A10
140 DATA 07D291BFCB238F0720
150 DATA 03B0C31510
160 DATA 028A0A10
170 DATA 05BF8F08CC1A10
180 DATA 02860D10
190 DATA 05B1C3158A0A10
200 DATA 028F0710
210 DATA 03BFCB2310
220 DATA 028F0810
230 DATA 058A0AB2C31510
240 DATA 02860D10
250 DATA 049FBFCC1A10
260 DATA 03860D9110
270 DATA 05B1C3158F0810
280 DATA 038A0A9010
290 DATA 058F07BFC41920
300 DATA 04B1CE0F9110
310 DATA 028A0A10
320 DATA 05DF800AAD1710
330 DATA 028E0B10
340 DATA 05D1A014800A10
350 DATA 03908F0710
360 DATA 04DFAC1F9110
370 DATA 02800A10
380 DATA 05D28A0AA01410
390 DATA 028E0B11
400 DATA 039FBFDF0101B20F
410 DATA 07B08107C00A91D210
420 DATA 03AD11D20F019401
430 DATA 0290D10F0295D301
440 DATA 06AB23D1918F0720
450 DATA 0295D2010491D0AD111
0
460 DATA 049FA014DA0101D10F
470 DATA 03C20ED010
480 DATA 05D1CE0BAD1110
490 DATA 02C00A10
500 DATA 04DFB1AB2310
510 DATA 01D210
520 DATA 07AD11B08F0891DF10
530 DATA 028F0710
540 DATA 039FA01410
550 DATA 07B08107C00A91D210
560 DATA 03AD11D20F019401
570 DATA 0290D10F0295D301
580 DATA 06AB23D1918F0720
590 DATA 0295D2010491D0AD110
600 DATA 06DF8A0AAC1F9010
610 DATA 02860D10
620 DATA 078E0BAD11C315D110
630 DATA 03918A0A10
640 DATA 04BFCB239210
650 DATA 03918E0B10
660 DATA 03DF8A0A10
670 DATA 02800A10
680 DATA 03AC1AB010
690 DATA 03860D9210
700 DATA 08908F08AD11C315D01
0
710 DATA 028A0A10
720 DATA 07D291BFCB238F0720
730 DATA 03B0C31510
740 DATA 028A0A10
750 DATA 05BF8F08CC1A10
760 DATA 02860D10
770 DATA 05B1C3158A0A10
780 DATA 028F0710
790 DATA 03BFCB2310
800 DATA 028F0810
810 DATA 058A0AB2C31510
820 DATA 02860D10
830 DATA 06CC1A8B06BFD010
840 DATA 019801019000F
850 DATA 03CD179110
860 DATA 019901019000F
870 DATA 0698A007B2C31501019
01F
880 DATA 05D1CC1A98B7010291B
11F
890 DATA 0598AF07C0140101900
F
900 DATA 03800A8010
910 DATA 04CC1A8F0710
920 DATA 028B0610
930 DATA 0391C01410
940 DATA 02810710
950 DATA 03DF8B0610
960 DATA 02810710
970 DATA 098B06A40BCE2190B0D
020
980 DATA 039FBFDF010390B0D03
F
990' DATA 099FBFDF8F08AA0ACD1
120
1000 DATA 0390B0D020
1010 DATA 039FBFDF010390B0D0
3F
1020 DATA 03DF8FBF20
1030 DATA 01D210
1040 DATA 019110
1050 DATA 05B192DFAB2310
1060 DATA 02870810
1070 DATA 0790B0D1A00A8F0720
1080 DATA 06A1078008CB2320
1090 DATA 07D0CC1A8B06AA0A22
1100 DATA 039FBFDF20
1110 DATA 0390B0D020
1120 DATA 04FF9FBFDF2E0
1130 DATA 000400,
1140 CALL INIT
1145 !"fly, you blackbirds"
from Brainard's Ragtime
    Collection (1899)
1146 ! arranged by
        DENES AGAY
(from The Joy of Ragtime
  selected & edited by
        Denes Agay
1147 !Yorktown Music Press,
Inc)
1148 ! pgm by Fred Hawkins
1150 FOR VOICE=0 TO 3 :: CAL
L XOFF(VOICE):: NEXT VOICE
1160 X=32 :: CALL DISABLE
1170 CALL LOAD(-31796,4,0,1)
!set sound list and trigger
1180 CALL LOAD(-31747,1)!set
 sound source pointer
1190 READ A$ :: PRINT A$ ::
B$=B$&A$
1200 IF LEN(B$)>64 THEN GOSU
B 1240
1210 IF A$>"" THEN 1190
1220 IF LEN(B$)>1 THEN PRINT
 LEN(B$);B$ :: GOSUB 1240
1230 GOTO 1260
1240 C$=SEG$(B$,1,64):: B$=S
EG$(B$,65,64):: PRINT : :B$:
 :
```

```
1250 PRINT :"LOADING";X;LEN(
C$):: CALL CHAR(X,C$):: X=X+
4 :: RETURN
1260 ACCEPT AT(4,4):C$ :: IF
 LEN(B$)>64 THEN CALL CHAR(1
32,SEG$(B$,65,255))
1270 CALL ENABLE
1280 PRINT "PLAY"
1290 FOR A=1 TO 12 :: CALL C
OLOR(A,3,13):: NEXT A :: CAL
L MAGNIFY(4)
1300 CALL SPRITE(#1,32,16,40
,50,1,3)
1310 FOR A=0 TO 80 :: CALL P
ATTERN(#1,32+A):: CALL COLOR
(#1,13*RND+3):: NEXT A :: GO
TO 1310
1320 SUB ENABLE
1330 CALL LOAD(-31806,0):: S
UBEND
1340 SUB DISABLE
1350 CALL LOAD(-31806,32)::
SUBEND
1360 SUB XOFF(VOICE)
1370 A=(VOICE-1)*32+159
1380 CALL LOAD(-31744,A):: S
UBEND
```

REGENA

(from AIRPORT AREA COMPUTER CLUB)
(via North West Ohio 99'er news)

by REGENA

```
100 DEF R=INT(16*RND+1)
110 RANDOMIZE
120 FOR I=1 TO 16
130 CALL SOUND(-50,R*110,4)
140 CALL COLOR(I,R,R)
150 CALL SCREEN(R)
160 CALL HCHAR(R+4,R+2,R*R/2
,R*R)
170 CALL VCHAR(R+4,R+2,R*R/2
,R*R)
180 NEXT I
190 GOTO 110
200 END
```

Who else but REGENA can put so much into so few lines?

SND CLASSIFIEDS

For Sale :

TI LOGO 2 - $80 or nearest offer - Phone (02)389.7025

Welcome to another Shop column

If you are a Tunnels of Doom fanatic then please read the letter from me to Jenny's Younger Set which gives details of how you can obtain new games for this module. I haven't seen them so I cannot give an honest opinion whether they are any good or not, but the small outlay in cost probably would make them worthwhile.

Those members whom stayed until the end of the September meeting heard an exciting announcement of a new product from Navarone Industries titled PAINT 'N' PRINT. Navarone have been kind enough to send to us an advance copy to review and demonstrate to members before any orders are taken. The package will be demonstrated at the October all day workshop, so make sure you are there to see it. Just to give you an idea what it's all about here is a small review.

PAINT 'N' PRINT comes on a cartridge and in its basic form does not require memory expansion. Pictures drawn on the screen can be saved to either tape or disk for later retrieval. In its basic form PAINT 'N' PRINT will not allow you to print out your pictures unless you have either an Axiom GP-700A color printer or the Axiom GP-100 graphics printer for black and white. That's right if you have a GP-700A your pictures will be printed in living color.

While the above is a fair limitation the good news is that Navarone package an Extended Graphics Package seperately which enables pictures to be printed out on virtually any dot matrix printer. This enhancement requires 32K memory expansion but again is operative from either tape or disk.

I have fiddled with PAINT 'N' PRINT a fair bit and find it very easy and colorful to use. The documentation is of a high standard and is certainly an improvement on some of Navarone's previous efforts.

Selling cost for the basic cartridge version would be in the vicinity of $45 while the extended package - cartridge, disk or tape - would be around $70.

Both versions will be shown at the October meeting.

While on the subject of Navarone products our large order has arrived but final selling prices have yet to determined. As a guide they will be :

Cartridge Expanders	- $45
Console Writers	- $45
DBM Systems	- $75

There is still no sign of Home Computer Magazine Volume 5 No 2. I have despatched another letter to the publishers asking them to expedite our order. I have not been able to ascertain what went wrong.

As I write this I have not decided what to put on Tape 1985/10 so it will be a bit of a mystery until you buy. It will be the same good value however.

For disk users I have 2 new disks for release in October. The first has just became public domain and is called ULTRA-COPY. There are 2 versions of the program on the disk - ULTRA-SE for Minimem and ULTRA-SE/A for option 3 of Editor Assembler. This is a very good disk copying utility and was written by Bruce Caron of the Ottawa User Group. Bruce as you know also wrote another very useful disk utility program DM-1000 which is also available from the shop.

The second disk is called 2D Graphics and is a FORTH drawing program. As the program completely fills a disk it will be sold as a "flippy" with the instructions being on the reverse side.

Memory Expansion, Editor Assembler cartridge and Disk Drive are required for 2D Graphics while Memory Expansion, Editor Assembler or Mini Mem cartridges and Disk Drive are need for Ultra Copy.

IMPORTANT ANNOUNCEMENT FOR MAIL ORDER USERS - due to escalating postal charges it has been found necessary to increase postage and packing costs for mail orders. The new scale of charges to be applied for all mail orders from 1/10/85 is:

Orders up to $10 value	- $2
Between $11 and $20	- $3
Between $21 and $30	- $4
Above $31	- $5

Please ensure that the above extra charge is added to your order from 1/10/85. The above will work out cheaper in some instances than current mail order prices.

For the benefit of all members the current selling prices of popular items are - Tapes $3, Disk Software $5, Home Computer Magazine $8, Boxes of Disks $25.

Well that's about it for this month. I see that the Editor deleted one of the programs I mentioned in my last column. I hope he puts it in this time around together with a few others I have passed on to him from local members. These programs were entered in the monthly software judging and while they didn't win a prize I hope that the incentive will be there for other members to write a program and see their efforts rewarded by having it published in the SND.

→

LATE NEWS JUST RECEIVED - The latest issue of The Smart Programmer (Millers Graphics) has just arrived. Some items that took my eye are :

1. A new game called Night Mission. The game is packed with a book which walks you through the program flow line by line. This one would be very good for budding programmers. Sugested price is $US19.95.

2. Explorer, an extremely useful utility program that actually lets you see how programs and modules work. Proce of this one is $US24.95.

3. A new piece of hardware called GRAM KRACKER (TM). The unit plugs into the module port. It contains 48K of RAM and GRAM - the RAM compliment to GROM. With GRAM KRACKER (TM) you plug any module into it and save the contents to disk or cassette. Once saved they can be loaded into the battery backed up GRAM KRACKER (TM) and run. Prototypes only have been developed and test so far but Millers expect to begin shipment in mid November. Suggested price will be in the $150 to $200 range. More details on this one will be received shortly and passed on via this column.

I hope to have a copy of Explorer soon and will arrange for it to be reviewed in the SND. In the meantime anyone wanting further information on Millers Graphics products can write to the at 1475 W. Cypress Avenue San Dimas CA 91773.

Dear Jenny

TISHUG Jenny's YOUNGER SET under 18's page

In the September Sydney News Digest you quoted from a letter from Peter Lees. Peter was enquiring if there are any other games for Tunnels of Doom apart from those packaged with the module - Pennies & prizes and Quest of the King. Well the good news Jenny and Peter is that there are some new games available for the module.

Information I have just received via the Saskatoon TI Computer Club (Canada) on this very subject is reprinted hereunder:

"John Behnke of 5755 W. Grace, Chicago, Illinois 60634 has developed a new game for the Tunnels of Doom cartridge. The game is called 'Tomb of Death' and sells for $US8 on disk or cassette. Documentation and game background are included in printed or electronic format. New spells, monsters, weapons, graphics and quest items are to be found."

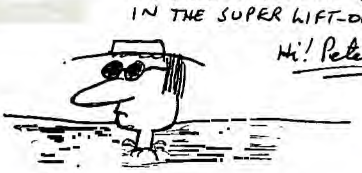
"John also has written another program which is FREE for the Tunnels of Doom module. It will be placed as FREEWARE, meaning that you give him what you think the program is worth. Other rumours have it that TI has released other Tunnels of Doom games from its archives."

Hope that Peter and other Younger Set members find this information useful.

Regards

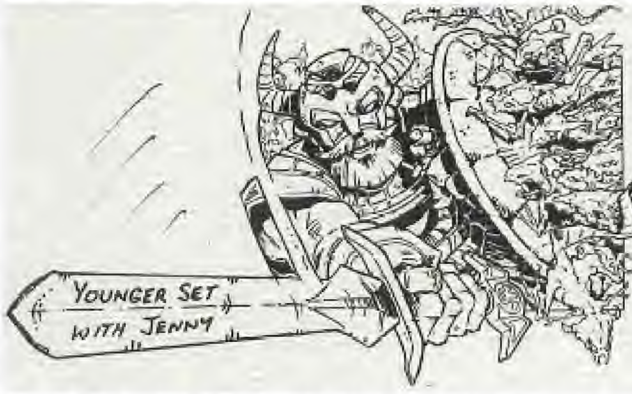
Terry Phillips
TISHUG Librarian

DRAWINGS BY PETER LEES...
CHECK OUT PETER'S PROGRAMS IN THE SUPER LIFT-OUT!



Up to his neck in it.
(PETER LEES)

(PETER LEES)



THANKS TERRY FOR THAT INFORMATION, NOW ITS UP TO US TO FOLLOW IT UP. Hi! gang, welcome to another edition of our clubs' under 18 page. Firstly, I want to thank you for your great response to the recent FRONT PAGE COMPETITION. I received so many pictures, and many of these will be shown in future issues of the SND. Due to space in this issue, the programs sent in by some of our Younger Set members have been placed into the special 8 PAGE PROGRAM LIFT-OUT. I mentioned to Shane a couple of days ago, that I wondered how many of you had bought, or had a 2 RING BINDER especially for that new idea of Shanes with the PROGRAM CENTREFOLD LIFT-OUT which he is going to continue with. Lets hear from you about that.

I have created yet another competition for you, but this one has to be done fast and neatly. I have designed a special CRACK THE CODE competition. This is a brain game in which you must de-code the following sentence and rush it to me by the 1st November so that the winner will appear in the next issue(NOVEMBER'85)of the SND. Also, For those who can write a program which re-defines the QWERTY Keyboard into NUMBERIC ones to create your very own SECRET CODE PROGRAM. You will see what I mean in a moment. I suggest that this would be a good project for the older members of Younger Set, but once again, the time limit is the same...GET IT TO ME at YOUNGER SET, P.O.Box 595, MARRICKVILLE. NSW 2204 no later than the 1st November.

This time, I am not going to tell you what the PRIZES are for this competition, but I will say that the lucky winners (one for each section) will jump up and down for joy when they get the package in the mail. By the way, the competition is only going to run as long as I receive no less than 5 entries to either section...So lets see those brains working.

Now, the CODED MESSAGE...Each number is really a letter, like this: A=12, W=19 and S=21 . There is no need to cut up your magazine, just de-code the message and write it down NEATLY and get it to me. Here goes.

11 24 13 13 8, 17 9 4 8 18 11 12 27 24 9 17 15 18 16
24 3 18 26 26 11 17 21 20 8 3 24 12 23 3 21 24 23
26 17 26 26 8 6 8 6 24, 23 17 20 24 12 23 3 26 17
3 4, 4 8 18 20 8 18 13 3 19 17 23 4 8 18 16 21 24 9
12 15 16 24 12 26 14 16 17 22 24. 26 11 17 21 24 23
26 16 4 6 18 21 26 7 24 21 24 23 26 26 8 6 24 23 8
13 12 26 24 16 26 11 12 23 26 11 24 9 17 16 21 26 8
9 23 8 27 24 6 7 24 16. 15 8 8 3 13 18 20 5!

Good luck! You're going to need it.!!!!

I hope that you will take the time to attend the next BIG FULL-DAY TUTORIAL MEETING at the fantastic WOODSTOCK MANSION in Burwood. The Committee has decided to run it on the LONG WEEK-END to give our out-of-sydney members a chance to come down for the week-end. So ask your mum or dad to bring you down to Sydney, if you live a long way away, and meet some of the other members. This will be a great opportunity for you to make new friends and then swap programs with them, as well as learning more about programming.

Bye for now,
Lets hear from you soon. Jenny

A WORD FROM THE EDITOR

Well, I'll be buttered on both sides! What a month it's been for me, but what a month it's going to be for you!

I was able to fly down to Melbourne last month, to attend Australia's largest "1st PAN PACIFIC COMPUTER CONFERENCE" at the Royal Exhibition Buildings. Man! What a show, it was so large that I had to visit it over 2 days to grasp the entire show. On the Saturday, I had the pleasure of attending the Melbourne User Group Meeting, and was able to spend some time sharing with them, the many features of the Sydney Group.

I wish to take this moment to thank them for their hospitality, with special HI! to Wayne Southwick (Secretary of the Group).

OUR NEXT MEETING

On page 2 you will see two mini reports, one from our Co-Ordinator, and the other from Fred Morris telling you all about the next meeting. We hope that you will make the effort to attend our next FULL-DAY TUTORIAL WORKSHOP, and the first one on a long-week-end so that our out-of-city and country members can come on down for this event.

IN THIS NEWSDIGEST

Last month, I introduced a new LIFT-OUT PROGRAMMING page which was ready to place into a binder. That issue was crammed full of some heavy programming information for those who could and would understand it. This month, the special LIFT-OUT Programming Pages have doubled, with much of the contents being from our own members, and in particular, some goodies from a few YOUNGER SET MEMBERS. 8 pages of programmes to type in and all ready to add to your binder.

PAGE #3: A mini review by me on the new Mini-Modem.

PAGE #4: ROADER...a program by Justin Twiss of TI-UP, plus the start of Robert's TECHO TIME.

Both the Program and Techo-time are continued through to PAGE#6 along with J.R's NOTES.

PAGE #7: PROGRAM CHALLENGE by Ed York with mini reviews of new software now available.

PAGE #8: Article by Henry Clark on CONSOLE HARDWARE.

PAGE #9: WHAT IS RAM DISK?...TI WRITER SUGGESTIONS and Rersonal Record Keeper hints.

PAGE #10: Report by Fred Morris on TI-PUBLICATIONS LIBRARY.

PAGE #11 through to #13:EXPERIMENTER'S XBASIC

PAGE #14: TI.S.H.U.G.SHOP with Terry Philips

PAGE #15: JENNY'S YOUNGER SET.

PAGE #16: This page.

For our MODEM Users ...

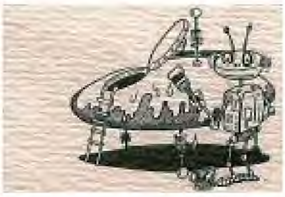
The response has been very good with interest in having our TEXPAC ELECTRONIC MAGAZINE (BBS) go to Dual Speed. With the inovation of VIATEL, and many of our members buying multi speed modems, we have desided to have our BBS SOFTWARE converted so that it will transmit in both 300 baud and 1200/75 baud speed.

We believe it is possible, and have a couple of our programmers working on that project. This is how we believe it will work...When we give the official O.K. users will be able to use any speed, and this system will be able to detect by the tone your computer is giving, which speed you are on, and will automatically swich to the same. Our modem will be able to do the work as long as a special cable is made, plus some possible change to the Machine Code software.

The result...a much faster system which will save our members heaps of dollars in STD rates for those outside of the Sydney city area, and for those living in the city, it will get them on, and off the system quicker so that others can have a chance.

These are the programmes which you are able to DOWN-LOAD from the club BBS during the month of OCTOBER:

FRENCHNOUN	20	BASIC
HIGH-NOON	27	X/B
HOUNTHOUSE	32	BASIC
KING-TUT	24	X/B
LUCKY-7	24	BASIC
SWARMI	32	BASIC
TI-99/4A	17	X/B
WORD-DICE	29	X/B
YELLOWROSE	18	X/B



GRACE has prepared the following for November...

2-FACES	19	BASIC
BULL-TERR	13	BASIC
HEBREW	7	X/B
L-PEN-GAME	30	BASIC
RIGHT-FRED	25	X/B
SKIPPY	29	X/B
TRUCKER	23	X/B
WOOD-HEART	21	X/B