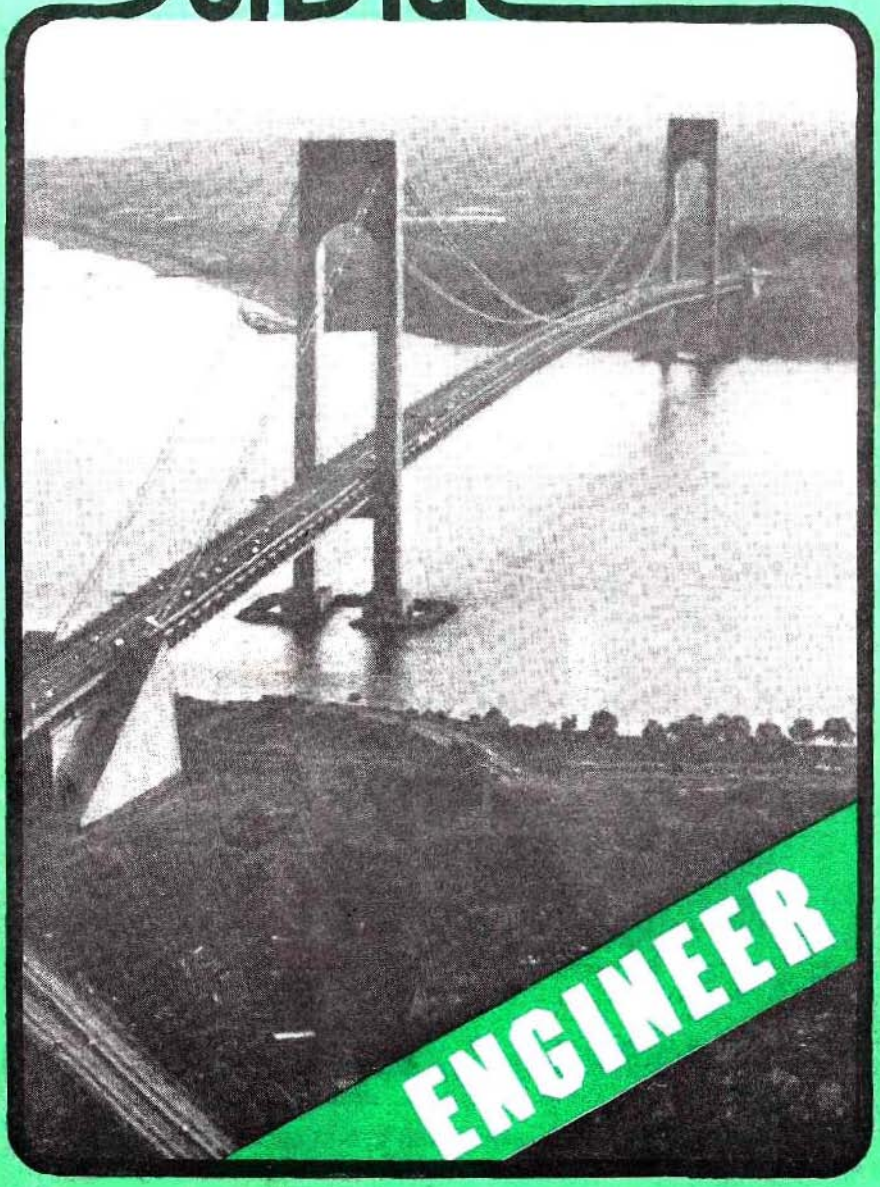
September 1979 $2.00

SoftSide™

"your BASIC software magazine"



SoftSide™

September, 1979
VOL. I, NO. 12

"your BASIC software magazine"

CONTENTS

STAFF

Publisher
Roger W. Robitaille, Sr.

Editor
George Blank

Copy Editor
Freida Day

Software Editor
Jack Moore

Advertising/Business Manager
Elizabeth Robitaille

Marketing Manager
Joseph A. Breton

Art/Production Manager
Sharon Demmerle

Layout/Composition
Alice Scofield
Laura Miller

Customer Service
Bette Keenan

Subscriptions
Diana Bishop, Mgr.

Mailing Department
Bea Kimball, Coordinator
Lester Anderson, Courier

Clerical Assistants
Jeanne Skinner
Karen Fissette
Brenda Cookingham

Accounting
Rita Ellis

Programmers
Phillip Brown
Mark Ohlund Abst

Programming Assistants
Bryan Berliebite
David K. Robitaille

Printing
Memorial Press Group

SoftSide magazine is continually seeking original articles and software for publication. Imagination and variety in concept and content are the rules at **SoftSide** — not the exceptions. Articles are purchased on a per-page basis, based on content and applicability. Our policies with respect to software purchase are highly individualized, and offer the programmer several options, including one-time publication rights, outright purchase, and royalties on sale of pre-recorded cassettes. For more information, please write: **SoftSide**, PO Box 68, Milford, NH 03055.

For uniformity, we have adopted the Radio Shack TRS-80 Level II BASIC as the BASIC dialect used within the pages of this magazine. It was chosen because it stands to become the most commonly used dialect among microcomputer users and because it shares a common heritage with the many microcomputer languages produced by Microsoft.

SoftSide is published monthly by SoftSide Publications, 17 Briar Cliff Dr. Milford, NH 03055 Telephone: 603-673-5144. Subscription rates: USA bulk rate — $18 per year. USA first class, APO, FPO, Canada, Mexico, overseas surface mail - $25 per year. Overseas airmail - $30 per year. **All remittances must be in U.S. funds.** Mail subscription inquiries to: **SoftSide Subscriptions**, PO Box 68, Milford, NH 03055.

Outgoing Mail

User of the Month

Our TRS-80 user of the month for September is FLW Associates of Alexandria, Virginia. These creative folks have developed a computer application that does not require any software (except Blackjack, Backgammon, and Level I BASIC). They use the TRS-80 to recruit "programmers, systems engineers, software engineers, and digital engineers" for placement with another company. How do they do this without a program? They give away the computer to anyone they place with the firm! So if you are in the indicated categories and don't want to pay for your TRS-80, contact them.

Here's a challenge for you!

Why not develop a program on your TRS-80 to find the largest known prime number? We can offer a few hints. First of all, you are going to have to develop something better than double precision accuracy. The most recently discovered largest prime has 13,395 digits. Naturally, I would like to print the whole thing here, but due to lack of space in the magazine, you will have to settle for scientific notation. The number is $(2^{44497})-1$. If you find a larger one, let us know! The present number took 300 hours to find on CRAY-1 computer outputting one trillion bits per second, starting with the previous largest known prime and trying all likely numbers. As another hint to help you get started; you don't need to check any even numbers.

New Hampshire Happenings

Things are nice and busy at the home office, where we are in the process of purchasing larger quarters and moving. We're also planning two new magazines; one to support the Apple computer (AppleSeed™), and one for computer chess, (MicroPawn™). Plans to reactivate the cassette version of SoftSide are also afoot. It will fill the gap between the software we publish in the magazine and that sold through the TRS-80 Software Exchange.

So many of the good programs that we are now receiving use machine language subroutines or supergraphics, or come from other publishers, that there will be few overlaps between the programs we offer in the magazine and the ones we sell. Therefore, we feel that a cassette version of the magazine may be the only way to provide our programs to those who are not willing to type them in. Incidentally, that means that if you have been waiting for one of our advertised programs to be featured in the magazine, it is unlikely that your hopes will be

realized. Future programs will largely be offered in one medium only.

Considering a disk drive?

Our June reader's poll indicated that while only a few of you have disk drives, many more are considering them. Perhaps it will be helpful if we share our own love/hate affair with you. Why would anyone in his or her right mind pay $500 for a front loading toaster that only accepts ultra thin bread? On the other hand, why is it that there is a half inch of dust on the last cassette I bought?

For many people the decision to go disk comes when you run out of things to do while a 16K program is loading into your computer from tape. For the rest, the moment comes when you realize that it didn't load correctly and you have to do it again. My own disk drive paid for itself rapidly in time spent programming instead of CSAVEing, CLOADing, and CLOAD?ing, not to mention starting from scratch because of a flaw on a tape or because I put it down on top of the power supply. I had the disk drive six months before I ever used it for anything except as a replacement for tape loading and unloading. The problem of loading SYSTEM tapes was so bad that I might have spent the $500 with a psychiatrist if I hadn't bought the disk. It is so nice to be able to reliably load a 16K program in seconds, and not even worry about a volume control!

Even without using disk files, some nice fringe benefits came with the disk system. I like the enhancements of disk BASIC, including the expanded error messages, the real time clock, and the automatic keyboard debounce loading. The ease of making backup copies of my programs meant that I did it more often, which saved my neck a number of times. I learned to use machine language with the DEBUG monitor, and would probably have never bothered with Assembly language if it were not for the ease of loading the Assembler from disk.

But the real purpose of a disk unit is its file handling capability. If you are a business user of the computer, that is probably why you bought the computer. Almost all practical commercial programs require disk files, whether for accounts, inventory, reports, or statistical analysis. If you can use a commercial software package, or have hired someone else to write one for you, you will be using this capability of your disk rapidly. However, if you are writing your own software, it may be some time before you are using your disk drive fully. It is

easy to learn Level I BASIC. It is moderately difficult to gain skill with Level II BASIC. It is downright demanding to become accomplished in Disk Input/Output Operations, especially with random files. Once you gain skill with file handling, you will discover the difference between a fancy calculator and real data processing.

Unfortunately, by that time you will also have discovered the limitations of your disk drives. First of all, you will learn that a disk does not really hold a great amount of data. When you first started programming on a 16K computer, you may have thought you had memory for any use, but when you have a mailing list or a large inventory, even the 350K bytes of a four disk system fill up fast. Even to store my programs now requires forty disks, and it is hard to find the one with the program I need.

Worst of all are the reliability problems. Radio Shack's DOS 2.2 has solved the worst nightmare, but there are still ways to spoil a complete disk. One of the most unforgivable, because it could easily be avoided, is when you try to KILL an open file, it scrambles the directory, giving you a disk full of garbage. But power failures, voltage transients, nearby radios and power supplies, and many other gremlins can aso crash a disk and bring heartache and tears.

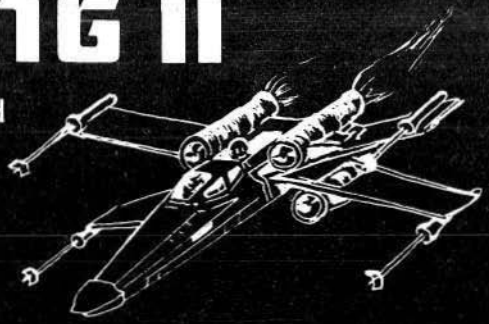
There are ways to go even beyond Radio Shack's Disk System. Percom Drives hold more data and access it faster, and NewDOS from Apparat is greatly superior to even DOS 2.2, with extra features and extra reliability and speed. Another DOS from the original author of the Radio Shack DOS has been advertised for months, but no one we know has been able to get a copy. Months ago we ordered one by phone and were promised delivery in a few days, but have not received it.

To those of you still weighing the pros and cons on disk drives, we have to say, on balance, that most of us with disk experience — despite the faults and frustrations — would rather fight than switch.

GWB

X-WING II

by Chris Freund



For the thousands who have enjoyed X-Wing Fighter, X-Wing II presents a totally new element in the game!

You are Pilot of an X-Wing fighter ...

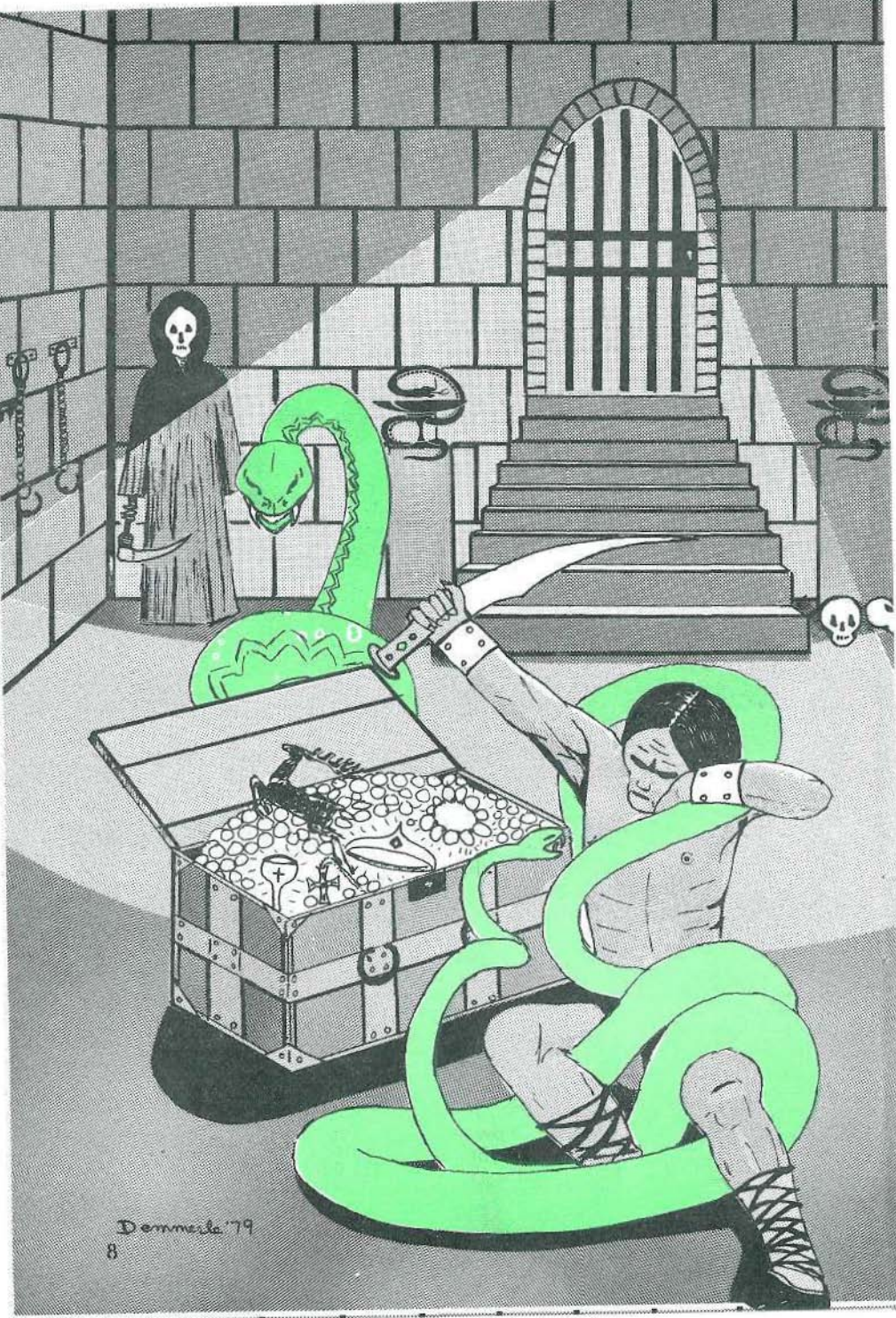
Your Mission, Destroy the Death Star!

Where X-Wing I left Death Star looming on the screen, **X-Wing II** lets you guide your fighter into the trench, find the exhaust port, aim and fire — all the while avoiding enemy fighters. Excellent graphics, 12 levels of play, and extensive INKEY$ commands make this one of our most exciting "real-time" games.

Level II, 16K — $9.95

TSE TRS-80 Software Exchange

17 BRIAR CLIFF DRIVE MILFORD, NEW HAMPSHIRE 03055


Demmerle '79

TREASURE DUNGEON 2

DAVID E. WHITE

Playing the Game

You will be exploring a dungeon of rooms and passages. In this dungeon you will find a variety of monsters. Defeat the monsters, and you will usually gain treasure. But if you are weakened, you may be killed.

The screen will display the section of the dungeon you are in as viewed from above. You are represented by a small figure in the center of the room. North is to the top, East to the right, etc. The boundaries of the room represent the type of walls around you. A solid line means a solid wall (or perhaps a secret door), a solid line with a 'D' means a closed door, on 'O' means an open door. A gap in a line mens an open passage, and no line at all means that you can see no wall in that direction.

The basic commands are: N, E, S, W, D, Q. The first four move you in the indicated direction, 'D' redraws the room, and 'Q' reports on your current status (strength, treasure, etc.). When you move in any direction, if there is a passage or no wall, you will automatically go to the next section. If the wall has a door, you may then listen or try to open it; if the door is open, you may pass through it. If there is a solid wall, you may search for secret doors and if you find one, you may listen or open as above with a regular door. Some doors may only go in one direction and not allow you to return the way you came; sometimes you may fall through a trap into a new room.

Drawing the dungeon on a piece of graph paper as you go along will make the interconnections of rooms and passages much clearer (and will also help you find your way back if you become lost).

If you find a monster, the display will show a huge monstrous figure in the center of the room. You may run or fight. If you fight, you both exchange blows until one is defeated or you retreat. If you keep on fighting against a strong monster, you may be killed. If you run away, the monster may strike a parting blow and you may drop some treasure. If you defeat the monster, you will then gain its treasure if one exists. It is always a good idea to check your strength after a fight to see if you need rest.

When you find and defeat all the monsters, the game is over. You may then rate yourself on how long it has taken you. For the first adventure, anything less than 100 hours in dungeon time is pretty good. If you wish to quit before all the monsters are defeated, you may do so by going back to the entrance and pressing 'Q'; you will then be asked if you want to quit or not.

Program Design:

The program is designed in a top-down structured fashion. The following diagram shows this more clearly. All major program blocks have clearly defined functions and are accessed as subroutines using GOSUBs. Within sub-blocks the execution flow is generally sequential with only limited use of GOTOs to control the programming flow.

All input uses the INKEY$ function which provides instantaneous response withour ever needing to press the ENTER key. The VAL function is used to strip numeric values off of the alphanumeric room pointer data.

The basic program and the particular adventure are independent entities. The program provides a structure which can run any number of dungeon adventures. The actual adventure is represented entirely by data, stored here as data statements. An entirely different adventure could be created by changing these data statements. Or a different dungeon description could be stored on tape or disk, and a short subroutine could read the data from there.

VARIABLE NAMES

Program variables:

PA	Player Attack Factor
PD	Player Defense Factor
PT	Player Treasure
MK	Monster Defeated
TM	Elapsed Time
PR	Previous Room
IM	Monster Key
IR	Current Room
IT	Treasure Key
IX	Next Room
D$	Current Command Key
K$	Current Wall Key
I$	INKEY$ Value
P$	Player Graphic
M$	Monster Graphic
WH$	Horizontal Wall
WV$	Vertical Wall

Dungeon description variables:

AD$(1-4)	Adventure Description
NR	Number of Sections
NM	Number of Monsters
NT	Number of Treasures
IR	Room Number
RD$(NR)	Room Description
RP$(NR,4)	Room Pointers
RM(NR)	Room Monster Key
IM	Monster Number
MD$(NM)	Monster Description
MA(NM)	Monster Attack Value
MD(NM)	Monster Defense Value
MT(NM)	Monster Treasure Key
IT	Treasure Number
TD$(NT)	Treasure Description
TV(NT)	Treasure Value

Creating a Dungeon:

The best way to create a dungeon is to draw it out first on graph paper. Keeping it consistent with Euclidean geometry will make it much less confusing for the adventurers. Then number each section of dungeon with a continuous series of numbers. Do the same for the monsters and treasures. Decide also what the descriptions are going to be.

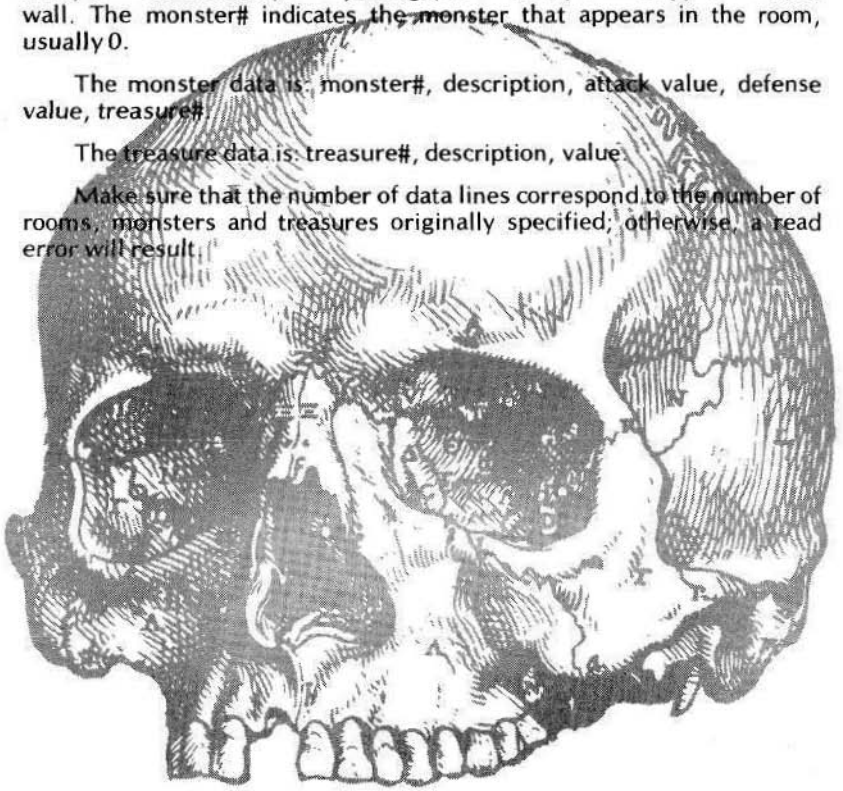
You are now ready to enter the data. The data lines in the program provide an example. First are four lines of string data describing the dungeon. Then the number of rooms, monsters, treasures. Then you are ready to enter the room, monster and treasure data in that order.

The room data is: room#, room description, room pointers 1 thru 4, monster#. The room pointers show the linkages with the other rooms. The format is 'Rm#Key', e.g. '12D' means that that wall has a door which leads to room 12. The keys are as follows: D — door, O — open door, S — secret door, P — passage, C — clear, T — trap, W — solid wall. The monster# indicates the monster that appears in the room, usually 0.

The monster data is: monster#, description, attack value, defense value, treasure#.

The treasure data is: treasure#, description, value.

Make sure that the number of data lines correspond to the number of rooms, monsters and treasures originally specified; otherwise, a read error will result.



PROGRAM COMMENTARY

ROUTINE: SETUP
LINE: 100-195
PURPOSE: This part sets up the program variables and reads the dungeon data.
PROCESS: The player starts at full strength with no treasure (PA=10, attack strength; PD=10, defense; PT=0, treasure). String variables are defined for the graphic display elements: P$, player; M$, monster; WH$, horizontal wall; WV$, vertical wall. Since the graphic displays occupy several screen lines, CHR$(26) is used to go down to the next line and CHR$(8) is used to backspace. The Subroutine at 9000 reads the descriptive dungeon data. The adventure always starts in room 1 (IR=1, current room; PR=1, previous room).

```
10 ' TREASURE DUNGEON 2
20 ' BY DAVID E WHITE
30 ' 30 APRIL 1979
35 ' REVISED 27 JUNE 1979
90 '
100 ' SETTING UP THE PROGRAM
102 CLS: PRINT@ 200,CHR$(23);"TREASURE DUNGEON 2
105 CLEAR 500: DEFINT A-Z
120 PA=10: PD=10: PT=0
139 ' GRAPHICS
140 P$=CHR$(174)+CHR$(132)+CHR$(26)+CHR$(8)+CHR$(8)+CHR$(129)+CH
R$(129)
150 M$=CHR$(138)+CHR$(191)+CHR$(133)+CHR$(26)+STRING$(5,8) + CHR
$(140)+CHR$(143)+CHR$(171)+CHR$(191)+CHR$(151)+CHR$(143)+CHR$(14
0)+CHR$(26)+STRING$(6,8) + CHR$(160)+CHR$(191)+CHR$(151)+CHR$(19
1)+CHR$(144)
160 WH$=STRING$(7,8)+STRING$(15,140)
165 V$=CHR$(191)+CHR$(26)+CHR$(8)
170 WV$=STRING$(2,27)+CHR$(188)+CHR$(26)+CHR$(8)+V$+V$+V$+CHR$(1
43)
180 DIM L(4): K$="": O$="": W$=""
189 '
190 GOSUB 9000  'READ DUNGEON DATA
195 IR=1: PR=1
199 '
```

ROUTINE: DUNGEON INTRODUCTION
LINE: 200-270
PURPOSE: This section introduces the player to the dungeon.
PROCESS: First several lines of general introduction are given. Then four unique strings of dungeon description which were previously read as part of the dungeon data are printed, AD$(1-4). The player is also told how many sections and monsters there are in this particular dungeon. The subroutine at 5000 is a commonly used one which merely waits for the player's response.

```
200 CLS
205 PRINT
210 PRINT"YOU ARE ABOUT TO ENTER ON AN ADVENTURE.
220 PRINT"IF YOU ARE WARY AND WISE, YOU MAY SURVIVE.
230 PRINT
240 FOR I=1 TO 4: PRINT AD$(I): NEXT
243 PRINT: PRINT"THERE ARE";NR;"SECTIONS AND
245 PRINT"THERE ARE";NM;"MONSTERS IN THE DUNGEON.
250 PRINT: PRINT"COMMANDS:  N-NORTH, E-EAST, S-SOUTH, W-WEST, D-
DRAW, Q-QUERY
270 GOSUB 5000
499 '
```

ROUTINE: COMMAND SECTION
LINE: 500-690
PURPOSE: Display current position and interpret basic player commands.
PROCESS: This is the primary control section which calls on other program elements and can be viewed as the top element in the program hierarchy. The subroutine at 6000 draws the graphic display of the room. If there is a monster in the room, the monster routine section at 7000 is then immediately called. Otherwise the command line is printed at the center of the screen and the INKEY$ function is used to wait for a response. If the player command indicates a direction (N, E, S, W), then the direction index I is set and control jumps to the movement block at 700. If another valid command (D,Q) is specified or an invalid command is given, then the program calls the appropriate subroutine, returns, and loops back to accept the next player command.

```
500 ' COMMAND SECTION
520 CLS: GOSUB 6000
550 IF RM(IR)<>0 GOSUB 7000
610 PRINT@ 576,"COMMAND (N,E,S,W,D,Q)
```

```
620 D$=INKEY$: IF D$="" GOTO 620
630 IF D$="N" THEN I=1: GOTO 700
640 IF D$="E" THEN I=2: GOTO 700
650 IF D$="S" THEN I=3: GOTO 700
660 IF D$="W" THEN I=4: GOTO 700
670 IF D$="D" THEN GOSUB 6000
680 IF D$="Q" THEN GOSUB 1000
690 GOTO 610
```

ROUTINE: **MOVEMENT**
LINE: 700-800
PURPOSE: Control movement in the dungeon.
PROCESS: After a movement direction I is specified, the room pointer key RP$(IR,I),is decoded using the current room number IR and direction index I to determine the type of wall on that side of the room. The room pointer keys are dimensioned string variables consisting of a number followed by a letter: the number indicates the adjacent room (if any) in that direction, and the letter indicates the type of passage or wall. By pulling off the rightmost letter using RIGHT$, the type of passage is determined as follows:

O	Open Door
W	Solid Wall
S	Secret Door
D	Closed Door
F	False Door
P	PASSAGE
C	Clear, no wall
T	Trap Door

Passage

Based on this key the appropriate subroutine is called. Then upon return, control is transferred back to the start of the Command section again.

```
700 'MOVEMENT
710 K$=RIGHT$(RP$(IR,I),1)
715 IF K$="O" THEN GOSUB 3500
720 IF K$="W" OR K$="S" THEN GOSUB 2000
730 IF K$="D" OR K$="F" THEN GOSUB 3000
740 IF K$="P" OR K$="C" THEN GOSUB 4000
750 IF K$="T" THEN PRINT"YOU FALL THRU A TRAP !": FORX=1TO500:N
EXT: GOSUB 4000
800 GOTO 500
999 '
```

ROUTINE: **PLAYER STATUS**
LINE: 1000-1190
PURPOSE: Informs the player of the current status of the adventure.
PROCESS: The dungeon time variable TM which is kept in minutes is converted to hours (by dividing by 60) and minutes (the remainder) for the display. The player attack strength variable PA and defense strength PD are converted to desscriptive adjectives to give the player a verbal indication of his current strength and wounds. The monsters defeated and treasure collected are next reported. If the pleyer's strengths are below a certain level, he may then rest to recover. The use of the INKEY$ function allows him to rest from 0 to 9 hours. If the player is in room 1 or has defeated all the monsters, he may quit the game.

```
1000 ' STATUS
1010 CLS
1015 IH=INT(TM/60): IM=TM-60*IH
1020 PRINT@64, "YOU HAVE BEEN GONE FOR";IH;"HOURS AND";IM;"MINUT
ES."
1030 S$="STRONG"
1032 IF PA<=6 THEN S$="WEARY": IF PA<=3 THEN S$="WEAK"
1035 C$="NOT"
1037 IF PD<=8 THEN C$="SLIGHTLY": IF PD<=6 THEN C$="MODERATELY":
 IF PD<=3 THEN C$="SEVERELY"
1040 PRINT"YOU FEEL ";S$;", AND ARE ";C$;" WOUNDED."
1050 PRINT: PRINT"YOU HAVE DEFEATED";MK;"MONSTER(S)."
1060 PRINT: PRINT"YOU HAVE TREASURE WORTH";PT;"GOLD PIECES."
1070 IF PA+PD>16 THEN GOTO 1100
1080   PRINT: PRINT"YOU MAY REST TO RECOVER STRENGTH"
1085   PRINT"HOW MANY HOURS WILL YOU REST ?"
1090 I$=INKEY$: IF I$="" GOTO 1090
1091   H=VAL(I$): IF H<1 GOTO 1100
1092   PRINT"RESTING ";: FOR IH=1 TO H: FORX=1TO1000:NEXTX: PRIN
T" Z";: NEXT: PRINT
1093   PRINT"YOU REST FOR";H;"HOURS.": TM=TM+60*H
1094   PA=PA+RND(H): IF PA>10 THEN PA=10
1096   PD=PD+RND(H): IF PD>10 THEN PD=10
1100 IF MK<>NM GOTO 1120
1110   PRINT: PRINT"YOU HAVE DEFEATED ALL THE MONSTERS."
1115   PRINT"THE GAME IS NOW COMPLETED.": END
```

HARDWARE

THE HARDSIDE™ OF SOFTSIDE ?

NEW! TRS-80 Computers 10% off!

Effective July 1, Radio Shack dropped the price of TRS-80 equipment. On the same date, we received authorization to sell the complete TRS-80 line, at 10% below the new low Radio Shack price. This all new equipment, with Radio Shack warranty and service, is now made more affordable than ever!

If you're looking to save even more, consider Hardside's used equipment department for prime condition, previously owned TRS-80 equipment.

Do you have something to sell? Hardside will buy any used Radio Shack TRS-80 equipment in good condition. Refer to our price list below.

So . . . if you're looking to buy or sell TRS-80 equipment, look to us first!

	NEW		USED	
UNIT	NEW R/S LIST PRICE	HARDSIDE DISCOUNT PRICE	WE'LL PAY	SELL
Level I 4K	$499	$449	$275	$375
Level II 4K	$619	$557	$370	$475
Level I 16K	$729	$656	$365	$475
Level II 16K	$849	$764	$500	$650
Level II 16K no keypad		$669	$450	$600
Expansion Interface				
.....OK	$269	$240	$175	$225
.....16K	$369	$315	$225	$300
.....32K	$469	$390	$275	$375
Disk Drives 0	$499	$449	$300	$399
1	$499	$449	$275	$375
(Percom disk drives -		$399	-	-
Printers Friction feed*	$1299	$1169	$650	$800
Tractor Feed*	$1559	$1403	$750	$900
Line PrinterII	$999	$899	New Item Not available	
Quick Printer I*	$499	$449	$250	$325
Quick Printer II	$219	$197	$125	$175
*(Requires cable	$40	$36)		
RS-232C	$99	$89	$50	$75
Telephone Int. I	Discontinued		$50	$75
Telephone Int II	$199	$179	$100	$150
16K Memory	**$149**	**$99**		

6

SPECIAL INTRODUCTORY OFFER FROM HARDSIDE™ 16K expansion interface with single disk drive **$750.**

TELEPHONE: 603-673-5144 **Prices Do Not Include Shipping**

HEADQUARTERS

EXIDY SORCERER

Until recently, you couldn't buy a microcomputer with the power and features of the Exidy Sorcerer. Now, an 8K RAM, 12K ROM Sorcerer is only $995.00. Advanced features include built-in monitor in ROM (including memory and video tests), ultra-high resolution graphics (512 x 240), upper/lower case, user-definable character set, and 1200 baud cassette I/O. Plus, 64 x 30 video (displays twice as much text as TRS-80), built in RS-232, parallel port, and the most complete keyboard of any microcomputer on the market. And, best of all, the Sorcerer is compatible with both CP/M and the S-100 bus for unlimited software and hardware expansion. Take a step up to Sorcerer power today!

PRICE LIST

Sorcerer Computer		
	16K RAM	**$995**
	32K RAM	**$1095**
	48K RAM	**$1195**
ROM Software Cartridges -		
	Word Processor	**$99**
	Editor/Assembler	**$99**
	BASIC [Included free with computer]	**$99**
Hardware Expansion		
	S-100 Expansion Unit	**$349**
	S-100 I/O Expansion Kit	**$199**
LEEDEX Video 100 monitor		**$139.95**

(The Sorcerer requires a video monitor or connection to a TV through an RF modulator.)

HARDSIDE™ Your market for new and used microcomputer equipment.

17 Briarcliff Drive **Milford, New Hampshire 03055**

```
1120 IF IR<>1 GOTO 1150
1125   PRINT"
YOU ARE BACK AT THE ENTRANCE.
DO YOU WISH TO QUIT?"
1130   K$=INKEY$: IF K$="" GOTO 1130
1135    IF K$="Y" THEN PRINT"COME BACK AGAIN.": END
1150 GOSUB 5000
1170 GOSUB 6000
1190 RETURN
1999 '
```

ROUTINE: WALL
LINE: 2000-2080
PURPOSE: Possible actions at a blank wall.
PROCESS: The player may search a solid wall. If he searches and there is a secret door, there is a 50% chance that he will discover it. If it is discovered, then the door routine at 3000 is called. If the player does not wish to search, then the subroutine returns to the command section.

```
2000 ' WALL
2010 PRINT@ 576,"YOU ARE AT THE ";D$;" WALL.": FOR X=1 TO 500: N
EXT
2020 PRINT@ 640,"DO YOU WISH TO SEARCH IT ?"
2030 I$=INKEY$: IF I$="" GOTO 2030
2040 IF I$<>"Y" THEN RETURN
2045 TM=TM+5
2050 IF K$<>"S" OR RND(3)>=2 THEN PRINT"YOU FIND NOTHING.": GOT
O 2010
2060 PRINT"YOU FIND A SECRET DOOR";
2070 GOSUB 3000
2080 RETURN
2999 '
```

ROUTINE: DOOR
LINE: 3000-3080
PURPOSE: Actions at a door.
PROCESS: This section controls the possible player actions at a closed door, where he may listen (L), attempt to open (O), or return to the promary command mode (R).

```
3000 ' DOOR
3010 IX=VAL(RP$(IR,I))
3020 PRINT@ 576,"YOU ARE AT THE ";D$;" DOOR": FOR X=1 TO 500: NE
XT
```

```
3030 IF K$="O" GOTO 3500'OPEN
3040 PRINT@ 640,"LISTEN (L), OPEN (O), OR RETURN (R) ?"
3050 I$=INKEY$: IF I$="" GOTO 3050
3060 IF I$="L" GOTO 3100
3070 IF I$="O" GOTO 3200
3075 IF I$="R" THEN RETURN
3080 GOTO 3040
3099 '
```

ROUTINE: **LISTEN**
LINE: 3100-3190
PURPOSE: Listening at a closed door.
PROCESS: When the player listens at a door there is a 33% chance that he will hear some random sound if there is a monster in the next room, and a 10% chance if there is nothing there.

```
3100 'LISTEN
3105 TM=TM+2: IF RM(IX)<>0 GOTO 3120
3110 IF RND(10)>1 THEN PRINT"YOU HEAR NOTHING";: GOTO 3190
3115 GOTO 3130
3120 IF RND(3)>=2 THEN PRINT"YOU HEAR NOTHING";: GOTO 3190
3130 ON RND(4) GOTO 3132,3134,3136,3138
3132 PRINT"THERE IS A STRANGE SOUND";: GOTO 3190
3134 PRINT"A MUFFLED SCREAM IS HEARD ..";: GOTO 3190
3136 PRINT"A CLANKING";: GOTO 3190
3138 PRINT"SOMETHING IS BEING DRAGGED";: GOTO 3190
3190 GOTO 3020
3199 '
```

ROUTINE: **OPEN DOOR**
LINE: 3200-3240
PURPOSE: Attempting to open a door.
PROCESS: The player has a 50% chance of opening a closed door on each attempt, but a false door (F) will never open. The room pointer key is then changed from "D" to "O" to indicate an open door. If the adjacent room also has a door at the appropriate spot (e.g. on the West wall if an East door is opened), then that door is opened as well. In that case J is used as the direction index for the next room by rotating clockwise 180 degrees from the current room direction index I ($J = I + \frac{1}{2}$).

```
3200 'OPEN ATTEMPT
3205 TM=TM+3
3210 IF RND(2)>=2 OR K$="F" THEN PRINT"THE DOOR DOES NOT OPEN";:
 GOTO 3020
3215 PRINT"THE DOOR OPENS";: FORX=1TO500:NEXT
3220 IF RIGHT$(RP$(IR,I),1)<>"D" THEN GOTO 3500
3225   RP$(IR,I)=STR$(IX)+"O"
3230   J=I+2: IF J>4 THEN J=J-4
3240   IF VAL(RP$(IX,J))=IR AND RIGHT$(RP$(IX,J),1)="D" THEN RP$
(IX,J)=STR$(IR)+"O"
3499 '
```

ROUTINE: **OPENED DOOR**
LINE: 3500-3550
PURPOSE: Actions at an open door.
PROCESS: If the door is open and there is a monster in the next room, then there is a 50% chance that the player will be warned. The player may decide to enter or not to enter the next room. If he enters, then the Room Movement routine is called. But in any case, control is next transferred back to the main Command section.

```
3500 'OPENED DOOR
3505 IX=VAL(RP$(IR,I))
3510 IF RM(IX)<>0 AND RND(2)=2  THEN PRINT@ 576,"THERE IS A MONS
TER THERE !"
3520 PRINT@ 640,"THE DOOR IS OPEN.  DO YOU WISH TO ENTER ?"
3530 I$=INKEY$: IF I$="" GOTO 3530
3540 IF I$="Y" THEN GOSUB 4000
3550 RETURN
3999 '
```

ROUTINE: **ROOM MOVEMENT**
LINE: 4000-4090
PURPOSE: Move player in the dungeon.
PROCESS: As the player moves from one section to another either through doors, passages, traps, etc., this subroutine is called to change the current room indicator variable IR and the previous room indicator PR. Also, open doors closed behind the player 50% of the time by changing the room pointer indicator keys from "?" to "D".

```
4000 ' MOVEMENT
4005 TM=TM+5
4010 IX=VAL(RP$(IR,I))
4015 IF IX<1 OR IX>NR THEN PRINT@ 640,"CAN NOT GO IN THAT DIRECT
ION.";:FORX=1TO500:NEXT:RETURN
4020 'CLOSING DOORS
4030 IF RND(6)<=3 OR K$<>"D" GOTO 4080
4040   RP$(IR,I)=STR$(IX)+"D"
4050   J=I+2: IF J>4 THEN J=J-4
4060   IF VAL(RP$(IX,J))=IR THEN RP$(IX,J)=STR$(IR)+"D"
4080 PR=IR: IR=IX
4090 RETURN
4999 '
```

ROUTINE: **WAIT**
LINE: 5000-5040
PURPOSE: Wait for player response.
PROCESS: This is called several times to halt all program action until the player responds.

```
5000 'WAIT
5010 X$=INKEY$
5020 PRINT@ 960,"<PRESS ANY KEY TO CONTINUE>";
5030 IF INKEY$="" GOTO 5030
5040 RETURN
5999 '
```

ROUTINE: **ROOM DESCRIPTION**
LINE: 6000-6100
PURPOSE: Graphic room display.
PROCESS: The room description data RD$(IR) is printed at the top of the screen. The screen locations for the centers of the four walls are specified by L(1-4). The four walls are displayed in sequence using the vertical and horizontal wall variables (WV$ & WH$). In the center of each wall, the O$ variable is printed to indicate the type of opening as given by the room pointer key RP$ for the appropriate direction. If there is a monster in the room, then the monster graphic is printed in the center, otherwise the player graphic is displayed.

```
6000 ' ROOM DESCRIPTION
6010 CLS: PRINT@ 0,RD$(IR);
6025 L(1)=156: L(2)=292: L(3)=412: L(4)=276
6030 FOR I=1 TO 4
6035  IF I=1 OR I=3 THEN W$=WH$: ELSE W$=WV$
6040  K$=RIGHT$(RP$(IR,I),1)
6045  O$=""
6050  IF K$="D" OR K$="F" THEN O$=" D "
6060  IF K$="O" THEN O$=" O "
6070  IF K$="P" THEN O$="   "
6080  IF K$="C" THEN W$=" "
6087  PRINT@ L(I),W$; : PRINT@ L(I)-1,O$;
6090 NEXT I
6095 IF MD(RM(IR))<=0 THEN PRINT@ 284,P$; ELSE PRINT@ 219,M$;
6100 RETURN
6999 '
```

ROUTINE: MONSTER
LINE: 7000-7080
PURPOSE: Monster interaction.
PROCESS: This procedure prints the monster description, MD$(IM), and gives a warning based on the relative strength of the player and the monster. The player has the choice of running or fighting.

```
7000 ' MONSTER
7010 IM=RM(IR): IF MD(IM)<=0 THEN RM(IR)=0: RETURN
7020 PRINT@ 512, MD$(IM)
7025 PRINT: IF MA(IM)>PD THEN PRINT"IT LOOKS PRETTY DANGEROUS.":
 ELSE PRINT"YOU MIGHT BE ABLE TO HANDLE THIS."
7040 PRINT@ 896,"RUN (R) OR FIGHT (F) ?"
7050 I$=INKEY$: IF I$="" GOTO 7050
7060 IF I$="R" THEN GOTO 7100
7070 IF I$="F" THEN GOTO 7200
7080 GOTO 7050
7099 '
```

ROUTINE: RUN AWAY
LINE: 7100-7190
PURPOSE: Escaping from the monster.
PROCESS: If the player runs, the monster has a chance (depending on the relative strengths) of striking a parting blow and wounding the player. The VAL function is used to scan through the current room

pointers to determine if there is a direct connection to the previous room from which the player entered. The VAL function returns the numeric portion of the room pointer data RP$, which thus gives the adjacent room numbers. If there is a direct return connection, then the player is returned to the previous room, otherwise the player is sent to an adjacent room at random. The player may also lose some treasure as he runs, which the monster then gains.

```
7100 'RUN AWAY
7105 TM=TM+2: CLS: PRINT@ 448,CHR$(23);
7110 IF RND(MA(IM))>PD THEN PRINT "THE MONSTER WOUNDS YOU ": PD=
PD-1: ELSE PRINT"YOU ESCAPE UNHARMED."
7120 FOR J=1 TO 4
7122  IF VAL(RP$(IR,J))=PR THEN I=J
7124 NEXT
7126 IF I>0 AND I<=4 THEN GOTO 7130
7128   I=RND(4):  IF VAL(RP$(IR,I))=0 GOTO 7128
7130 IF RND(6)>=3 OR PT=0 THEN GOTO 7190
7140   LT=PT*RND(5)/10: PT=PT-LT
7150   PRINT: PRINT"YOU DROP TREASURE WORTH";LT;"GP"
7160   TV(MT(IM))=TV(MT(IM))+LT
7180 FOR X=1 TO 2000: NEXT
7185 GOSUB 4000: GOSUB 6000
7190 RETURN
7199 '
```

ROUTINE: **FIGHT**
LINE: 7200-7300
PURPOSE: Combat with the monster.
PROCESS: Line 7204 is used to clear the previous combat lines from the screen, using CHR$(30) which erases to the end of the line and CHR$(13) which goes to the start of the next line. The combat procedure is based on the relative strength differences between an attacker's attack strength and a defender's defense strength, the greater is the chance of hitting and the greater the damage. For example, MP is the difference between the monster's attack strength, MA(IM), and the player's defense strength, PD, (with a minimum difference of 2 set if it is less than that). The amount of damage is auniformly distributed random number generated from this base minus one. e.g. if MP=3, there is an equal 33% chance of damage o, 1, 2. The same procedure is applied for the player attacking. If the monster defense strength, MD(IM), falls to 0 or

Continued on page 48

It's Back to School Time!

If you're heading back to school, or work, this fall, you probably have often thought that it would be a lot easier if you didn't have to "hunt and peck" at the typewriter or computer keyboard. If you could only double your typing speed, the job would be done in half the time! And, with increased speed and confidence comes increased accuracy.

TYPING TUTOR

by 80 US

Typing Tutor is a set of programs designed to teach you touch typing — from the basics of learning where the keys are, to practice drills at speeds that would make a Selectric shudder! Each lesson is displayed on the screen, including, at the early stages, a keyboard diagram. You are quizzed and graded, and you progress at your own pace. When you have mastered a lesson, the computer advances to the next, using CAI (computer aided instruction). Progress is fast, painless, and even fun as you keep pushing to break your own record.

So, whether you're going back to school or just want to enter programs from **SoftSide** that much faster, **Typing Tutor** can help.

Order a copy today and be a better typist by next week.

Level II, 16K — $19.95

TSE TRS-80 Software Exchange

17 Briar Cliff Drive Milford, New Hampshire 03055

HANGMAN

L_ _ '_ _L _ _ _ _N _ _ _ N!

by Russell Starkey

Do you remember how boring a rainy afternoon could be when you were a kid? How you and your brother would get into a fight, and your mother would start to yell, and finally get you both settled down and quiet by suggesting a good game of HANGMAN? Now that you're all grown up and your brother has joined the Air Force and your mother has gone back to college, you can still dispel boredom with a good game of HANGMAN — even if you're alone, although a friend helps.

The computer adds the excitement of a visual countdown, and spices the game with graphics as the hangman's victim takes shape on the screen with every missed guess. If you exceed the time limit, or miss seven times, you've had it — the rope appears around his neck, and you're HUNG!

In the two-player mode, the competitive aspect is emphasized as one player invents the word or phrase on the spot and the other attempts to guess it. You guess a letter at a time and, if you pick one that's a hit, the letter appears in every position in which it belongs in the word or phrase.

Solitary HANGMAN, on the other hand, depends on words and phrases put into the program. As it is now constructed, this program is for young and old alike, with words and phrases ranging from "see spot run" to "expansion interface". However, you can freshen the program with new words or alter its nature considerably by changing the data statement starting at line 2400 (a simple task). A whole new set of words, simpler or more difficult — some famous quotations, perhaps a foreign language, or a scientific vocabulary? With thought, many possibilities will suggest themselves. Perhaps you can ingeniously arrange to have a friend insert material you are unfamiliar with, to retain the mystery.

As you can see, the old game of HANGMAN can be lots of fun in many ways, and also a great learning aid. So, on your next rainy afternoon, draw a chair up to your computer and start keyboarding this program!

```
0 '  * * * * * * * * * * * * * * * * * *
     *      SOFTSIDE   PRESENTS       *
     *           HANGMAN              *
     *      BY RUSSELL STARKEY        *
     * * * * * * * * * * * * * * * * * *
10 REM REV ( 2 )
20 REM HANGMAN  TWO PLAYER  ALSO  SOLITARY  BY RUSSELL STARKEY
30 REM 4K SYSTEMS LOAD IN TWO PARTS :
40 REM LINES 120 TO 800 ARE FOR  TWO PLAYER HANGMAN
50 REM LINES 660 TO 3460 ARE FOR  SOLITARY HANGMAN
60 POKE 16553,255:CLEAR50:GOSUB650  :PRINT@138,"H A N G M A N "
70 PRINT@320,"ENTER  1  FOR  SOLITARY"
80 PRINT@448,"ENTER  2  FOR  TWO PLAYER"
90 S$=INKEY$:IF S$="" THEN 90
100 IF S$="1" OR S$="S" THEN RUN810
110 IF S$="2" THEN RUN120  ELSE RUN
120 CLEAR 200 :DEFSTR S : DEFINT I,H : GOSUB650  : DIM S6(30)
130 GOSUB650  :PRINT@518,"H  A  N  G  M  A  N " : FOR I2 = 1 TO
2000 : NEXT
140 S2="":GOSUB650  : PRINT@128,"ENTER YOUR WORDS TO BE GUESSED.
"
150 PRINT@768,"PRESS    /    TO START OVER."
160 S1=INKEY$ :IFS1=""THEN 160  ELSE IF LEN(S2) >=26 OR  13=ASC(
S1) THEN180  ELSE IF S1="/" THEN 140  ELSE S2=S2+S1:PRINT@384,S2
:GOTO160
170 STOP
180 IFS2=""THEN160  ELSE S2=S2+" ":GOTO190
190 '  GUESS COME POINT ....
200 I3 =832:H4=15360:H6=640:H=0:H3=0:H7=120
210 ' INIT LOOP  "-"
220 FOR I2 = 1 TO 26 : S6(I2) = "-" : NEXT
230 GOSUB650  : F1=0: H1 =-1
240 PRINT@128,"ENTER GUESSES TILL YOU'RE HUNG."
250 PRINT@514,"TIME LEFT   120  SEC. ";:FORX=0 TO92:SET(X,22):SE
T(X,27):NEXT:FOR X=23TO26:SET(0,X):SET(92,X):NEXT
260 S3=" " : GOTO 330
270 ' COME BACK POINT .....
280 F1=F1+1 : IF F1=26 F1=0 : H7=H7-1 : IF H7=-1 THEN 610
290 PRINT@536,H7;
300 S3=INKEY$
```

```
310 IFS3="" THEN 270
320 IF F1=1 THEN 270
330 ' TEST LOOP  ○  .........
340 F=1 : FOR I2 = 1 TO LEN(S2)
350 IF MID$(S2,I2,1)=S3 THEN  S6(I2) = S3 : F=2
360 NEXT
370 IF F=1 THEN PRINT@I3,S3; : IF S3 ○ S8 THEN  I3=I3+4
 : S8=S3
380 IF F=1 H=H+1:GOSUB660
390 IF F=2 H1=H1+1
400 PRINT@256,H1;"HITS",H;"MISSES";
410 ' PRINT LOOP ..............
420 S7="":FOR I2 = 1 TO LEN(S2) : S7=S7+S6(I2) : NEXT
: PRINT@384,S7;
430 IF S7=S2 THEN 460
440 IFH>=7 THEN 500
450 GOTO270
460 ' CORRECT COME POINT..
470 GOSUB650  :FOR Q=1TO10:FOR X=0TO955STEP RND(10)+24:PRINT@X,"
    WINNER !!!  ";:NEXT X,Q
480 GOSUB650
490 Q2=Q2+1:PRINT@128,"WINNER  WINNER  WINNER !!!"
500 PRINT@256,"YOU NEEDED ";120-H7;" SECONDS "
510 PRINT@384,S2
520 PRINT@512,"YOU HAD ";H;"MISSES...    "
530 FOR I2 = 1TO 2600 : NEXT
540 PRINT@704,"DO YOU WANT TO PLAY AGAIN ?  "
550 INPUT S1 : IF S1="YES" OR S1="Y"  OR S1="RUN" OR S1 = "@"  T
HEN 140
560 GOSUB650  :PRINT@320,"YOU WERE  HUNG ";Q1;" TIMES";:PRINT@19
2,"YOU WON ";Q2;" TIMES ";:FORX=1TO1800:NEXT
570 PRINT@704,"SEE YOU AGAIN SOMETIME":PRINT:PRINT:END
580 FOR X=1 TO 700 : NEXT : ' HUNG COME POINT
590 GOSUB650
600 Q1=Q1+1:PRINT@128,"HUNG !!!!!!!!  HUNG !!!!!!!!       " : GO
TO500
610 ' OUT OF TIME COME POINT
620 FOR Q=1TO7-H3:GOSUB660  :NEXT:FORQ=1 TO 2000 :NEXT
630 GOSUB650
640 Q1=Q1+1:PRINT@128,"OUT OF TIME !!! TIME IS GONE !!   " : GOT
```

```
0500
650 CLS : PRINT CHR$(23) :RETURN
660 'GRAPHICS PRINT
670 H3=H3+1:ON H3 GOTO 690 ,710 ,720 ,730 ,740 ,750 ,760
680 STOP
690 FOR H5=H4+564 TO H4+570 :POKE H5,131:POKEH5+128,131:NEXT:POK
EH4+564,151:POKEH4+570,171
700 POKEH4+628,149:POKEH4+634,170:RETURN
710 SET(100,26):SET(113,26):RETURN
720 FOR H5=30TO35:SET(112,H5):SET(109,H5):NEXT:RETURN
730 FOR H5=100TO121:SET(H5,36):NEXT:RETURN
740 FOR H5=36TO44:SET(100,H5):SET(124,H5):NEXT:RETURN
750 FOR H5=39 TO 43 STEP2 :SET(109,H5):SET(112,H5):NEXT:RETURN
760 FORH5=32TO34:SET(105,H5):SET(117,H5):NEXT:FORH5=105TO117:SET
(H5,32):SET(H5,34):NEXT
770 FOR H5=117TO123:SET(H5,33):NEXT:FORH5=33TO0STEP-1:SET(124,H5
):NEXT:FORH5=1 TO 17:H6=H6+2
780 PRINT@H6,"HUNG ! !"; :GOSUB800 :PRINT@H6,"          "; :GOSUB8
00 :NEXT H5
790 FOR H5=1 TO 600:NEXT :RETURN
800 FORX=1 TO 40 :NEXT X:RETURN
810 REM-----------------------------------------------------
820 REM DATA LINES MAY BE CHANGED TO NEW WORDS.
830 CLEAR 200 :DEFSTR S : DEFINT I,H : GOSUB3440 : DIM S6(30):DI
M I(250)
840 PRINT@320,"SOLITARY  HANGMAN"
845 READS$:IF S$<>"XXX"THEN XA=XA+1:GOTO845
850 POKE16553,255
2400 DATA RUSSELL'S COMPUTER,DIAMOND RING,PRESIDENT JIMMY CARTER
,RADIO SHACK,GOLF
2410 DATA QUEEN,GO FLY A KITE,MASTERMIND,HAWAII FIVE-O,CALENDAR,
SPACE SHIP,COMPUTER
2420 DATA WORDS,I LOVE YOU,WEEKDAY,INSTANT REPLAY,KALEIDOSCOPE,S
EASON,CORRESPONDENT
2500 DATA FIRE TOWER,VIDEO DISPLAY,VICE-PRESIDENT MONDALE,HELLO,
DAVID AND STEVEN,STARS
2510 DATA EMERGENCY,HANGMAN GAME,SEE SPOT RUN,TRS-80,LET'S GET A
 PIZZA,CLOCK,CATFISH
2520 DATA ZUCCHINI,KILOMETERS,ALPHABET,SHARK,EXCAVATION,CURRENCY
,HONEYSUCKLE,JAGUAR
```

```
2600 DATA FOOTBALL,LUNAR ORBIT,TELEPHONE,STARSKY AND HUTCH,BLACK
 AND WHITE,BASKETBALL
2610 DATA GUESS WHO?,TAPE RECORDER,BULLSEYE,MERRY CHRISTMAS,ORAN
GE JUICE,MOTHER AND DADDY
2620 DATA KIMBALL ORGAN,MORK AND MINDY,TONIGHT SHOW,DOBERMAN PIN
SCHER,MAILMAN,DC-10 AIRPLANE
2630 DATA FANTASY ISLAND,EDEN,MONKEY,MEMORY BOARDS,APPLE COMPUTE
R,SOFTWARE,BIG BIRD
2640 DATA SOFTSIDE,FORTRAN,LINE PRINTER,JACK AND JILL,DISNEYLAND
,INDIANA,INPUT OUTPUT
2650 DATA OVERTIME,EXPANSION INTERFACE,SESAME STREET,INFANT,ADD
WORDS @ LINE 2700
2999 DATA XXX
3000 RESTORE:X=RND(XA):I8=I8+1:IF I8>200 THEN 3460 ELSE IF I(X)=
1THEN3000 ELSE I(X)=1:FOR I9=1 TO X:READ S2:I8=0
3010 NEXTI9
3020 S2=S2+" "
3030 I3 =832:H4=15360:H6=640:H=0:H3=0:H7=120
3040 FOR I2 = 1 TO 26 : S6(I2) = "-" : NEXT
3050 GOSUB3440 : F1=0: H1 =-1
3060 PRINT@128,"* HANGMAN *   GUESS THE WORD."
3070 PRINT@514,"TIME LEFT   120  SEC. ";:FORX=0 TO92:SET(X,22):S
ET(X,27):NEXT:FOR X=23TO26:SET(0,X):SET(92,X):NEXT
3080 S3=" " : GOTO 3150
3090 REM COME BACK
3100 F1=F1+1 : IF F1=26 F1=0 : H7=H7-1 : IF H7=-1 THEN 3400
3110 PRINT@536,H7;
3120 S3=INKEY$
3130 IFS3="" THEN 3090
3140 IF F1=1 THEN 3090
3150 REM TEST
3160 F=1 : FOR I2 = 1 TO LEN(S2)
3170 IF MID$(S2,I2,1)=S3 THEN  S6(I2) = S3 : F=2
3180 NEXT
3190 IF F=1 THEN PRINT@I3,S3; : IF S3 <> S8 THEN  I3=I3+4
 : S8=S3
3200 IF F=1 H=H+1:GOSUB3450
3210 IF F=2 H1=H1+1
3220 PRINT@256,H1;"HITS",H;"MISSES";
3230 REM P LOOP
```

```
3240 S7="":FOR I2 = 1 TO LEN(S2) : S7=S7+S6(I2) : NEXT
: PRINT@384,S7;
3250 IF S7=S2 THEN 3280
3260 IFHD=7 THEN 3370
3270 GOTO3090
3280 REM CORR
3290 GOSUB3440
3300 IA=IA+1:PRINT@128,"CORRECT !!! "
3310 PRINT@256,"YOU NEEDED ";120-H7;" SECONDS "
3320 PRINT@384,S2
3330 PRINT@512,"YOU HAD ";H;" MISSES...    "
3340 PRINT@640,"GAME TOTALS:":PRINT"
# CORRECT  ";IA:PRINT"# HUNG     ";IB:PRINT"# TIME     ";IC
3350 FOR I2=1 TO 4000:NEXT
3360 GOTO3000
3370 FOR X=1TO700:NEXT
3380 GOSUB3440
3390 IB=IB+1:PRINT@128,"HUNG !!!!!!!!":GOTO3310
3400 REM TIME
3410 FOR Q=1TO7-H3:GOSUB3450 :NEXT:FORQ=1 TO 2000 :NEXT
3420 GOSUB3440
3430 IC=IC+1:PRINT@128,"OUT OF TIME !!! " : GOTO3310
3440 CLS : PRINT CHR$(23) :RETURN
3450 GOTO668
3460 I0=0:FOR X=1TO50:I(X)=0:NEXT:GOTO3000
```

Correction For TANK, July, 1979 SoftSide

Our thanks to Bruce and Joy Blevins of Urbana, Ohio for the following correction and useful addition to the game of **TANK** by James Garon:

Correction:
Line 71Ø should read: IF P-812 IF P-876 IF P-88Ø IF P-26Ø IF P-324 IF P-328 IF P > 191 IF P+8-64*INT((P+8)/64) 0=6Ø

Addition:
To keep the tanks from running over each other, and add the tactics of blocking the other tank's path, add Line 626: TA=ABS(P+O-Q-V): IF TA < 8 OR TA=6Ø OR TA=64 OR TA=68 THEN O=Ø:V=Ø.

ADVANCED Personal Finance

by Lance Micklus

First, we took the tape version of PERSONAL FINANCE and converted it for use under DOS. Then many new features were added such as self-verifying files which protect themselves from most common hardware faults, and the BUDGET program which collects data - automatically from the CHECKING program, and manually from the keyboard. Advanced Personal Finance will produce a 30-page report that gives you the total picture of your financial posture. To complete the package, a SAVINGS account program lets you use the one savings account as if it were ten individual accounts. This way you can set a certain amount of money aside for Christmas, save an additional amount for a rainy day, and keep track of how much is for what.

Also included are programs to convert the data file on tape from the regular personal finance program to disk.

On a 32K disk system, the package will handle about 200 checks per month and 900 checks per year. There are 33 different account names which are set up with DATA statements in each program on the disk.

The minimum system required is 32K Disk BASIC with one drive. The addition of a line printer, a second drive, and upper/lower case video display all enhance the features. A second disk (not supplied) is required to store your data, as the program disk is very full. Price, **$24.95.**

Reduced Featured Tape Edition $9.95.

TSE TRS-80 Software Exchange

17 BRIAR CLIFF DRIVE MILFORD, NEW HAMPSHIRE 03055

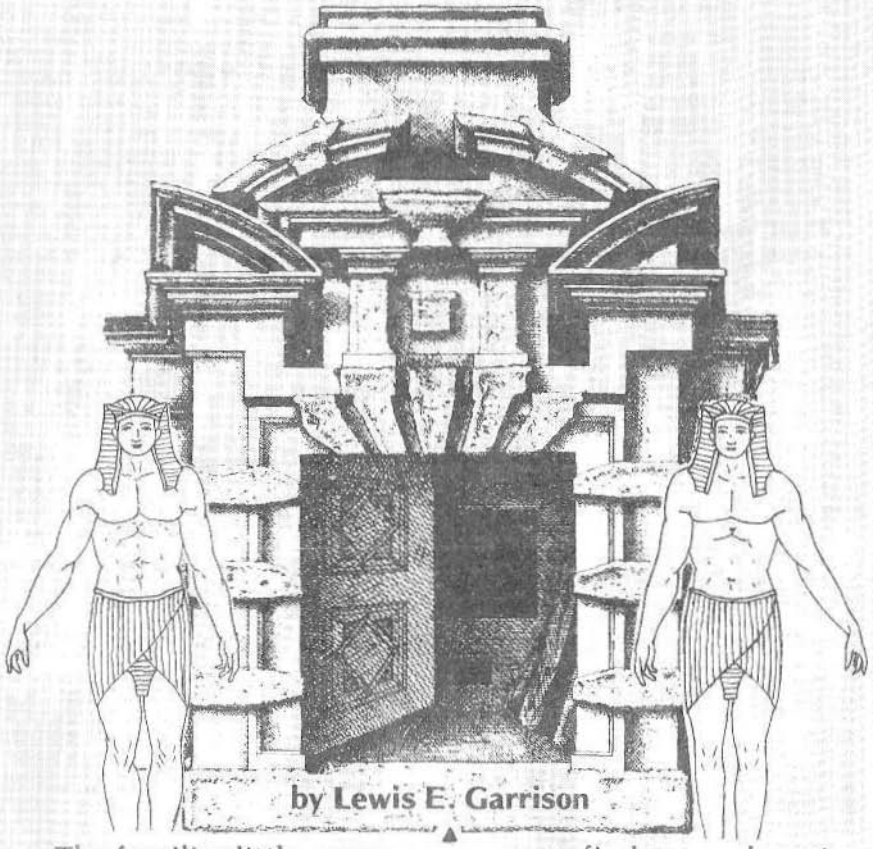
NOTE:

During normal business hours, programmers will ONLY be available from 9-10 a.m. and 5-6 p.m.

New HOTLINE Hours

Tuesday nights - 7-10 p.m. (EDT)

PYRAMIDS



by Lewis E. Garrison

The familiar little game you find everywhere in gift shops and restaurants now comes to the computer. You have 15 pyramids. You start by removing one. Then, you jump one over another one and each jumped piece is removed from the board. The object is to end with one pyramid. It sounds easy, and indeed the concept is simple. Gaining the objective is not, and can provide quite a challenge!

```
10 ' * * * * * * * * * * * * * * *
    *     SOFTSIDE PRESENTS     *
    *          PYRAMIDS         *
    *    COPYRIGHT FEB. 1979    *
    *   BY LEWIS E. GARRISON    *
    * * * * * * * * * * * * * * *
20 CLS:PRINT@25,"PYRAMIDS":PRINT:INPUT"DO YOU WANT INSTRUCTIONS
(YES/NO)";A$
30 CLS:IF LEFT$(A$,1) ="N" THEN 130
40 DEFINTA-Z
50 PRINT:PRINT"IN THIS PYRAMID GAME, ALL 15 PYRAMIDS ARE ON AT T
HE BEGINNING"
60 PRINT:PRINT"OF THE GAME. TO START, REMOVE ANY ONE PYRAMID. TH
EN JUMP"
70 PRINT:PRINT"ONE PYRAMID OVER ANOTHER INTO AN EMPTY SPACE AND
THE JUMPED"
80 PRINT:PRINT"PYRAMID WILL BE REMOVED. CONTINUE UNTIL YOU HAVE
NO JUMPS LEFT"
90 PRINT:PRINT"THE OBJECT IS TO LEAVE ONLY ONE PYRAMID ON THE BO
ARD."
100 PRINT:PRINT "TYPE 99 TO RESTART, OR 999 TO QUIT"
110 PRINT:PRINT:PRINT"GOOD LUCK!!!!"
120 INPUT"PRESS ENTER TO START";B$
130 CLS:GOSUB940:GOSUB950:GOSUB960:GOSUB970:GOSUB980:GOSUB990:GO
SUB1000:GOSUB1010:GOSUB1020:GOSUB1030:GOSUB1040:GOSUB1050:GOSUB1
060:GOSUB1070:GOSUB1080:GOSUB1090
140 PRINT@164,"REMOVE";:INPUT R
150 IF (R<1) OR (R>15) THEN 190
160 ONR GOSUB770,780,790,800,810,820,830,840,850,860,870,880,890
,900,910,920
170 PRINT@164,STRING$(20," ");:PRINT@164,"FROM";
180 INPUTA:PRINT@173,"TO";:INPUT B:PRINT@175,"";
190 IF (A=99)  OR (B=99) THEN 130
200 IF(A=999) OR (B=999) THEN 1110
210 ONA GOTO220,230,240,250,260,270,280,290,300,310,320,330,340,
350,360
220 IF(B=4)AND NOT(POINT(36,25))AND(POINT(45,16))OR(B=6)AND NOT(
POINT(77,25))AND (POINT(67,16))THEN400ELSE370
230 IF(B=7)AND NOT(POINT(26,34))AND(POINT(35,25))OR(B=9)AND NOT(
POINT(68,34))AND(POINT(59,25))THEN420ELSE370
```

```
240 IF(B=8)AND NOT(POINT(46,34))AND(POINT(58,25))OR(B=10)AND NOT
(POINT(89,34))AND(POINT(79,25))THEN440ELSE370
250 IF(B=1)AND NOT(POINT(56,7))AND(POINT(45,16))OR(B=6)AND NOT(P
OINT(78,25))AND(POINT(59,25))OR(B=11)AND NOT(POINT(16,43))AND(PO
INT(26,34))OR(B=13)AND NOT(POINT(58,43))AND(POINT(47,34))THEN460
ELSE370
260 IF(B=12)AND NOT(POINT(37,43))AND(POINT(47,34))OR(B=14)AND NO
T(POINT(88,43))AND(POINT(68,34))THEN500ELSE370
270 IF(B=1)AND NOT(POINT(56,7))AND(POINT(67,16))OR(B=4)AND NOT(P
OINT(36,25))AND(POINT(59,25))OR(B=13)AND NOT(POINT(58,43))AND(PO
INT(68,34))OR(B=15)AND NOT(POINT(99,43))AND(POINT(89,34))THEN520
ELSE370
280 IF(B=2)AND NOT(POINT(45,16))AND(POINT(36,25))OR(B=9)AND NOT(
POINT(67,34))AND(POINT(47,34)) THEN560ELSE370
290 IF(B=3)AND NOT(POINT(67,16))AND(POINT(59,25))OR(B=10)AND NOT
(POINT(89,34))AND(POINT(68,34))THEN580ELSE370
300 IF(B=2)AND NOT(POINT(45,16))AND(POINT(59,25))OR(B=7)AND NOT(
POINT(26,34))AND(POINT(47,34))THEN600ELSE370
310 IF(B=3)AND NOT(POINT(67,16))AND(POINT(79,25))OR(B=8)AND NOT(
POINT(46,34))AND(POINT(68,34))THEN620ELSE370
320 IF(B=4)AND NOT(POINT(36,25))AND(POINT(26,34))OR(B=13)AND NOT
(POINT(58,43))AND(POINT(37,43))THEN640ELSE370
330 IF(B=5)AND NOT(POINT(57,25))AND(POINT(47,34))OR(B=14)AND NOT
(POINT(79,43))AND(POINT(58,43))THEN660ELSE370
340 IF(B=4)AND NOT(POINT(36,25))AND(POINT(47,34))OR(B=6)AND NOT(
POINT(77,25))AND(POINT(68,34))OR(B=11)AND NOT(POINT(16,43))AND(P
OINT(37,43))OR(B=15)AND NOT(POINT(99,43))AND(POINT(79,43))THEN68
0ELSE370
350 IF(B=5)AND NOT(POINT(57,25))AND(POINT(68,34))OR(B=12)AND NOT
(POINT(37,43))AND(POINT(58,43))THEN720ELSE370
360 IF(B=6)AND NOT(POINT(77,25))AND(POINT(89,34))OR(B=13)AND NOT
(POINT(58,43))AND(POINT(79,43))THEN740ELSE370
370 PRINT@164,"ILLEGAL MOVE":FORT=1TO500:NEXTT
380 PRINT@164,"              ":GOTO170
390 REM MOVES
400 IFB=4THENGOSUB770:GOSUB970:GOSUB780:GOTO170
410 IFB=6THENGOSUB770:GOSUB990:GOSUB790:GOTO170
420 IFB=7THENGOSUB780:GOSUB1000:GOSUB800:GOTO170
430 IFB=9THENGOSUB780:GOSUB1020:GOSUB810:GOTO170
440 IFB=8THENGOSUB790:GOSUB1010:GOSUB810:GOTO170
```

```
450 IFB=10THENGOSUB790:GOSUB1030:GOSUB820:GOTO170
460 IFB=1THENGOSUB800:GOSUB940:GOSUB780:GOTO170
470 IFB=6THENGOSUB800:GOSUB990:GOSUB810:GOTO170
480 IFB=11THENGOSUB800:GOSUB1040:GOSUB830:GOTO170
490 IFB=13THENGOSUB800:GOSUB1060:GOSUB840:GOTO170
500 IFB=12THENGOSUB810:GOSUB1050:GOSUB840:GOTO170
510 IFB=14THENGOSUB810:GOSUB1070:GOSUB850:GOTO170
520 IFB=1THENGOSUB820:GOSUB940:GOSUB790:GOTO170
530 IFB=4THENGOSUB820:GOSUB970:GOSUB810:GOTO170
540 IFB=13THENGOSUB820:GOSUB1060:GOSUB850:GOTO170
550 IFB=15THENGOSUB820:GOSUB1080:GOSUB860:GOTO170
560 IFB=2THENGOSUB830:GOSUB950:GOSUB800:GOTO170
570 IFB=9THENGOSUB830:GOSUB1020:GOSUB840:GOTO170
580 IFB=3THENGOSUB840:GOSUB960:GOSUB810:GOTO170
590 IFB=10THENGOSUB840:GOSUB1030:GOSUB850:GOTO170
600 IFB=2THENGOSUB850:GOSUB950:GOSUB810:GOTO170
610 IFB=7THENGOSUB850:GOSUB1000:GOSUB840:GOTO170
620 IFB=3THENGOSUB860:GOSUB960:GOSUB820:GOTO170
630 IFB=8THENGOSUB860:GOSUB1010:GOSUB850:GOTO170
640 IFB=4THENGOSUB870:GOSUB970:GOSUB830:GOTO170
650 IFB=13THENGOSUB870:GOSUB1060:GOSUB880:GOTO170
660 IFB=5THENGOSUB880:GOSUB980:GOSUB840:GOTO170
670 IFB=14THENGOSUB880:GOSUB1070:GOSUB890:GOTO170
680 IFB=4THENGOSUB890:GOSUB970:GOSUB840:GOTO170
690 IFB=6THENGOSUB890:GOSUB990:GOSUB850:GOTO170
700 IFB=11THENGOSUB890:GOSUB1040:GOSUB880:GOTO170
710 IFB=15THENGOSUB890:GOSUB1080:GOSUB900:GOTO170
720 IFB=5THENGOSUB900:GOSUB980:GOSUB850:GOTO170
730 IFB=12THENGOSUB900:GOSUB1050:GOSUB890:GOTO170
740 IFB=6THENGOSUB910:GOSUB990:GOSUB860:GOTO170
750 IFB=13THENGOSUB910:GOSUB1060:GOSUB900:GOTO170
760 END
770 RESET(56,4):FORX=55TO57:RESET(X,5):NEXTX:FORX=54TO58:RESET(X
,6):NEXTX:FORX=53TO59:RESET(X,7):NEXTX:RETURN
780 RESET(46,13):FORX=45TO47:RESET(X,14):NEXTX:FORX=44TO48:RESET
(X,15):NEXTX:FORX=43TO49:RESET(X,16):NEXTX:RETURN
790 RESET(67,13):FORX=66TO68:RESET(X,14):NEXTX:FORX=65TO69:RESET
(X,15):NEXTX:FORX=64TO70:RESET(X,16):NEXTX:RETURN
800 RESET(36,22):FORX=35TO37:RESET(X,23):NEXTX:FORX=34TO38:RESET
(X,24):NEXTX:FORX=33TO39:RESET(X,25):NEXTX:RETURN
```

```
810 RESET(57,22):FORX=56TO58:RESET(X,23):NEXTX:FORX=55TO59:RESET
(X,24):NEXTX:FORX=54TO60:RESET(X,25):NEXTX:RETURN
820 RESET(78,22):FORX=77TO79:RESET(X,23):NEXTX:FORX=76TO80:RESET
(X,24):NEXTX:FORX=75TO81:RESET(X,25):NEXTX:RETURN
830 RESET(26,31):FORX=25TO27:RESET(X,32):NEXTX:FORX=24TO28:RESET
(X,33):NEXTX:FORX=23TO29:RESET(X,34):NEXTX:RETURN
840 RESET(47,31):FORX=46TO48:RESET(X,32):NEXTX:FORX=45TO49:RESET
(X,33):NEXTX:FORX=44TO50:RESET(X,34):NEXTX:RETURN
850 RESET(68,31):FORX=67TO69:RESET(X,32):NEXTX:FORX=66TO70:RESET
(X,33):NEXTX:FORX=65TO71:RESET(X,34):NEXTX:RETURN
860 RESET(89,31):FORX=88TO90:RESET(X,32):NEXTX:FORX=87TO91:RESET
(X,33):NEXTX:FORX=86TO92:RESET(X,34):NEXTX:RETURN
870 RESET(16,40):FORX=15TO17:RESET(X,41):NEXTX:FORX=14TO18:RESET
(X,42):NEXTX:FORX=13TO19:RESET(X,43):NEXTX:RETURN
880 RESET(37,40):FORX=36TO38:RESET(X,41):NEXTX:FORX=35TO39:RESET
(X,42):NEXTX:FORX=34TO40:RESET(X,43):NEXTX:RETURN
890 RESET(58,40):FORX=57TO59:RESET(X,41):NEXTX:FORX=56TO60:RESET
(X,42):NEXTX:FORX=55TO61:RESET(X,43):NEXTX:RETURN
900 RESET(79,40):FORX=78TO80:RESET(X,41):NEXTX:FORX=77TO81:RESET
(X,42):NEXTX:FORX=76TO82:RESET(X,43):NEXTX:RETURN
910 RESET(100,40):FORX=99TO101:RESET(X,41):NEXTX:FORX=98TO102:RE
SET(X,42):NEXTX:FORX=97TO103:RESET(X,43):NEXT:RETURN
920 END
930 REM BUILD PYRAMID
940 SET(56,4):FORX=55TO57:SET(X,5):NEXT:FORX=54TO58:SET(X,6):NEX
TX:FORX=53TO59:SET(X,7):NEXTX:FORX=52TO60:SET(X,8):NEXTX:RETURN
950 SET(46,13):FORX=45TO47:SET(X,14):NEXTX:FORX=44TO48:SET(X,15)
:NEXTX:FORX=43TO49:SET(X,16):NEXTX:FORX=42TO50:SET(X,17):NEXTX:R
ETURN
960 SET(67,13):FORX=66TO68:SET(X,14):NEXTX:FORX=65TO69:SET(X,15)
:NEXTX:FORX=64TO70:SET(X,16):NEXTX:FORX=63TO71:SET(X,17):NEXTX:R
ETURN
970 SET(36,22):FORX=35TO37:SET(X,23):NEXTX:FORX=34TO38:SET(X,24)
:NEXTX:FORX=33TO39:SET(X,25):NEXTX:FORX=32TO40:SET(X,26):NEXTX:R
ETURN
980 SET(57,22):FORX=56TO58:SET(X,23):NEXTX:FORX=55TO59:SET(X,24)
:NEXTX:FORX=54TO60:SET(X,25):NEXTX:FORX=53TO61:SET(X,26):NEXTX:R
ETURN
990 SET(78,22):FORX=77TO79:SET(X,23):NEXTX:FORX=76TO80:SET(X,24)
```

```
:NEXTX:FORX=75TO81:SET(X,25):NEXTX:FORX=74TO82:SET(X,26):NEXTX:R
ETURN
1000 SET(26,31):FORX=25TO27:SET(X,32):NEXTX:FORX=24TO28:SET(X,33
):NEXTX:FORX=23TO29:SET(X,34):NEXTX:FORX=22TO30:SET(X,35):NEXTX:
RETURN
1010 SET(47,31):FORX=46TO48:SET(X,32):NEXTX:FORX=45TO49:SET(X,33
):NEXTX:FORX=44TO50:SET(X,34):NEXTX:FORX=43TO51:SET(X,35):NEXTX:
RETURN
1020 SET(68,31):FORX=67TO69:SET(X,32):NEXTX:FORX=66TO70:SET(X,33
):NEXTX:FORX=65TO71:SET(X,34):NEXTX:FORX=64TO72:SET(X,35):NEXTX:
RETURN
1030 SET(89,31):FORX=88TO90:SET(X,32):NEXTX:FORX=87TO91:SET(X,33
):NEXTX:FORX=86TO92:SET(X,34):NEXTX:FORX=85TO93:SET(X,35):NEXTX:
RETURN
1040 SET(16,40):FORX=15TO17:SET(X,41):NEXTX:FORX=14TO18:SET(X,42
):NEXTX:FORX=13TO19:SET(X,43):NEXTX:FORX=12TO20:SET(X,44):NEXTX:
RETURN
1050 SET(37,40):FORX=36TO38:SET(X,41):NEXTX:FORX=35TO39:SET(X,42
):NEXTX:FORX=34TO40:SET(X,43):NEXTX:FORX=33TO41:SET(X,44):NEXTX:
RETURN
1060 SET(58,40):FORX=57TO59:SET(X,41):NEXTX:FORX=56TO60:SET(X,42
):NEXTX:FORX=55TO61:SET(X,43):NEXTX:FORX=54TO62:SET(X,44):NEXTX:
RETURN
1070 SET(79,40):FORX=78TO80:SET(X,41):NEXTX:FORX=77TO81:SET(X,42
):NEXTX:FORX=76TO82:SET(X,43):NEXTX:FORX=75TO83:SET(X,44):NEXTX:
RETURN
1080 SET(100,40):FORX=99TO101:SET(X,41):NEXTX:FORX=98TO102:SET(X
,42):NEXTX:FORX=97TO103:SET(X,43):NEXTX:FORX=96TO104:SET(X,44):N
EXTX:RETURN
1090 PRINT@159,"1":PRINT@346,"2":PRINT@356,"3":PRINT@533,"4":PRI
NT@543,"5":PRINT@554,"6":PRINT@720,"7":PRINT@730,"8"
1100 PRINT@741,"9":PRINT@751,"10":PRINT@907,"11":PRINT@917,"12":
PRINT@928,"13":PRINT@938,"14":PRINT@949,"15":RETURN
1110 CLS:PRINT:PRINT "THANKS FOR PLAYING---BYE,BYE"
```


SPACE BATTLES
By Level IV
HAVE SPACESHIP - WILL TRAVEL
You're a galactic adventurer, an old hand at calculated risks, exploring far corners of a vast universe containing aliens of every description - and they are the bounty you hunt. In defense of the empire, you coolly pit yourself against superior forces, with brilliance and daring, but with the added knowledge that every enemy ship you down means your coffers swell - and you are also a businessman!
This superb game features three levels of play; fast, machine language graphics; real-time input; and "smart" enemy ships that move and shoot! Good hunting, and watch out for the radiation belts! Available on Level II, 16K Tape or 32K Disk.
Tape - $14.95
Disk - $19.95
TSE TRS-80 Software Exchange
17 Briar Cliff Drive Milford, New Hampshire 03055

The MAGIC of Leo Christopherson

Android Nim

The newest version of TRS-80's first animated graphics game - Android NIM - now with more animation and **sound**!

Level II, 16K - **$14.95**

Snake Eggs

Here is a computerized reptilian version of **21** complete with arrogant snakes and appropriate **sound**.

Level II, 16K **$14.95**

Life Two

Two in one: Game of Life, at an astounding 100 generations a minute, plus Battle of Life with animated creatures and **sound**.

Level II, 16K **$14.95**

Cubes

Cubes gives you the solution to Instant Insanity®*, a numbered block puzzle. Drive your computer nuts trying to figure how to arrange the 4 blocks on your screens. Each side shows four different numbers. Level II, 16K **$9.95**

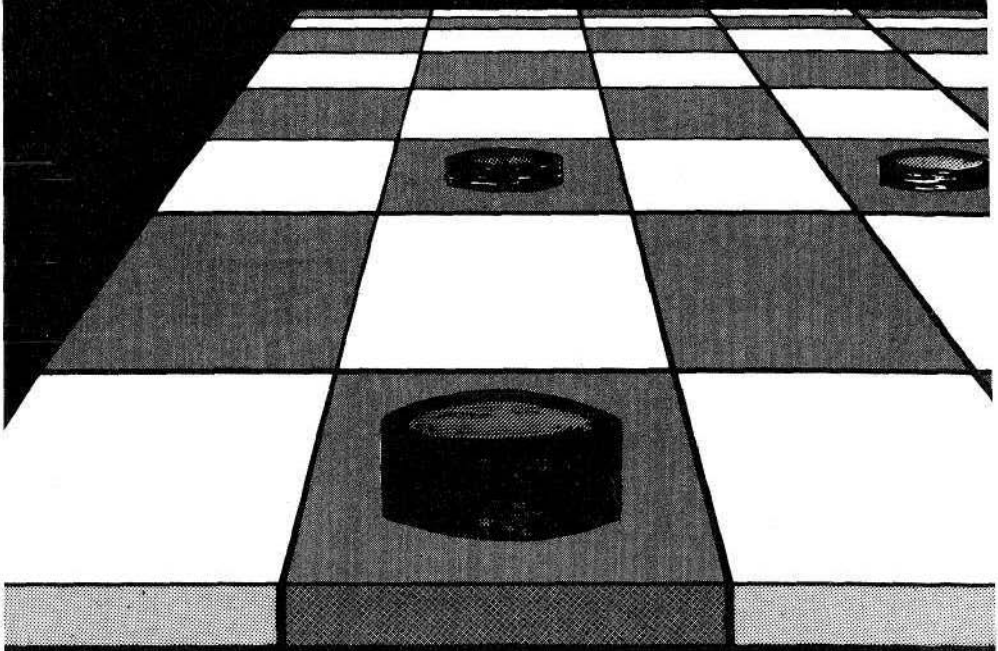
* Instant Insanity® is a Parker Brothers registered trademark

TSE TRS-80 Software Exchange

17 BRIAR CLIFF DRIVE MILFORD, NEW HAMPSHIRE 03055

THE MEAN CHECKERS MACHINE

by Lance Micklus



Our resident wizard has done it again! Designed in FORTRAN, run as machine language, this program turns your TRS-80 into an unbelievably wicked checker player! Four levels of play; at the most difficult, the machine may take ten minutes per move, as it attempts to assess all possibilities. MEAN CHECKERS MACHINE is to checkers what SARGON is to chess. (Level 4 is an exercise in humility!)

Level II, 16K	$19.95
Disk	$24.95

TSE TRS-80 Software Exchange

17 Briar Cliff Drive Milford, New Hampshire 03055

Get a LOAD of this!

If you have less time at the keyboard than you would like to, and have been missing out on some of SoftSide's feature programs, We've got a solution!

SoftSide™

Your BASIC software magazine for TRS-80

On Cassettes

That's right! Now, all out monthly programs will be available on audio cassettes, for only about a dollar apiece - **SATISFACTION GUARANTEED.**

The price for 6 months of cassettes **plus** 6 months of SoftSide magazine is only $38.00. If you're already a subscriber to the magazine, you can enroll for the cassettes only, at a cost of $30.00 for 6 months.

So, if you want to combine the best programming available for the TRS-80 with the convenience of pre-recorded program cassettes, send your order to SoftSide today, and leave the coding to us!

SoftSide™ "your BASIC software magazine"
PO Box 68 Milford, NH 03055

ALL RIGHT! Enroll me for 6 months of SoftSide on cassette. My full payment is enclosed as indicated below;

☐New Subscriber - $38.00 ☐Subscriber - $30.00
☐Check ☐Money Order
☐Charge to Master Charge/VISA

Exp. date _____ Interbank# _____
Signature _____
NAME _____
ADDRESS _____
CITY, STATE _____ ZIP _____

less, then the monster is defeated and the program goes to the Treasure section. If the player defense strength goes to 3 or less, then the player is warned that he should escape. If he continues to fight, if may fall to 0 and he is dead.

```
7200 'FIGHT
7203 TM=TM+1
7204 CL$=CHR$(30)+CHR$(13): PRINT @ 576, CL$+CL$+CL$+CL$+CL$
7205 PRINT@ 576,"THE MONSTER STRIKES AT YOU ....
7210 MP=MA(IM)-PD: IF MP<2 THEN MP=2
7215 FOR X=1 TO 300: NEXT X
7220 PX=RND(MP)-1: IF PX>0 THEN PRINT"THE MONSTER HITS YOU": ELS
E PRINT"THE MONSTER MISSES"
7230 IF PX>0 THEN PA=PA-PX: PD=PD-PX
7240 IF PD<=0 THEN PRINT"YOU ARE KILLED !": END
7250 FORX=1TO400:NEXT: PRINT"YOU STRIKE AT THE MONSTER ...."
7255 FOR X=1 TO 300: NEXT
7260 PM=PA-MD(IM): IF PM<2 THEN PM=2
7270 MX=RND(PM)-1: IF MX>0 THEN PRINT"YOU WOUND THE MONSTER": EL
SE PRINT"A MISS..."
7280 IF MX>0 THEN MA(IM)=MA(IM)-MX: MD(IM)=MD(IM)-MX
7290 IF MD(IM)<=0 THEN GOTO 7400'DEFEATED
7295 IF PD<=3 PRINT"YOU ARE SEVERELY WOUNDED.  BETTER RUN !
7300 GOTO 7040
7399 '
```

ROUTINE: **TREASURE**
LINE: 7400-7490
PURPOSE: Allocation of treasure.
PROCESS: If the monster is defeated and has a treasure, then the value of that treasure is given to the player. Where IM is the monster number, IT=MT(IM) gives the associated treasure number, and TD$(IT) is the treasure description and TV(IT) is its value. Control is returned to the main Command section.

```
7400 'TREASURE
7402 FOR X=1 TO 500: NEXT
7403 RM(IR)=0: MK=MK+1: GOSUB 6000
7405 PRINT@ 512,"THE MONSTER IS DEFEATED !": PRINT
7410 IF MT(IM)=0 PRINT"YOU FIND NO TREASURE": GOTO 7470
7420 PRINT"THERE IS A TREASURE !"
7430 IT=MT(IM)
7440 PRINT TD$(IT)
```

```
7450 PT=PT+TV(IT)
7455 PRINT"IT IS WORTH";TV(IT);"GOLD PIECES."
7460 MT(IM)=0
7470 GOSUB 5000
7480 GOSUB 6000
7490 RETURN
8999 '
```

ROUTINE: **READ ADVENTURE DATA**
LINE: 9000-9290
PURPOSE: Getting the descriptive dungeon data.
PROCESS: The dungeon data is stored in data statements which are read at the start of the program and assigned to array variables. This gives a much faster response than re-reading the data statements each time additional information is needed, and doesn't take up much more memory space because Level II BASIC establishes string variable pointers which make use of the data lines themselvess. The first items read are the general description lines AD$(1-4) and the number of rooms, NR; monsters, NM; and treasures, NT. These variables are then used to dimension the dungeon description variables which are filled by subsequent READ statements.

```
9000 ' ADVENTURE DATA
9010 POKE 16553,255
9020 RESTORE
9040 READ AD$(1),AD$(2),AD$(3),AD$(4),NR,NM,NT
9050 DIM RD$(NR), RP$(NR,4), RM(NR)
9060 FOR I=1 TO NR
9070   READ IR,RD$(IR),RP$(IR,1),RP$(IR,2),RP$(IR,3),RP$(IR,4),R
M(IR)
9080 NEXT I
9110 DIM MD$(NM), MA(NM), MD(NM), MT(NM)
9120 FOR I=1 TO NM
9130   READ IM,MD$(IM),MA(IM),MD(IM),MT(IM)
9140 NEXT I
9210 DIM TD$(NT), TV(NT)
9220 FOR I=1 TO NT
9230   READ IT, TD$(IT),TV(IT)
9240 NEXT
9290 RETURN
9299 '
```

ROUTINE: **ROOM DATA**
LINE: 9400-9475
PURPOSE: Dungeon room descriptions.
PROCESS: The format for the room data is: room number IR, room description RD$(IR), room pointers RP$(IR,1-4), and resident monster RM(IR). This is fairly self explanatory. The room pointers as mentioned before are string variables which point to adjacent rooms in clockwise order (N, E, S, W). The first part of each variable gives the adjacent room number (if any) in that direction, and the final letter gives the wall/passage type key. If there is no number, than zero is implied and movement is not permitted in that direction.

```
9310 DATA "ADVENTURE # 1"
9312 DATA PREPARE TO RISK YOUR LIFE FOR THERE ARE MANY DANGERS H
ERE.
9314 DATA ORCS AND OTHER MONSTERS HAVE BEEN REPORTED.
9315 DATA BEWARE ESPECIALLY THE FIERCE MINOTAUR ...
9317 ' ROOMS, MONSTERS, TREASURES
9318 DATA 75,13,13
9400 ' ROOM DATA
9401 DATA 1,THIS IS THE ENTRANCE TO THE DUNGEON.
LIGHT COMES INTO THE ROOM THROUGH AN OPENING IN THE CEILING., 2D
,6P,30P,4S,0
9402 DATA 2,A SMALL DRY STONE ROOM WITH SOME BONES IN THE CORNER
S., W,W,W,3D,0
9403 DATA 3,A LONG RECTANGULAR ROOM., 22D,2D,4C,14D, 1
9404 DATA 4,A LONG RECTANGULAR ROOM WITH MOLD ON THE SOUTH WALL.
, 3C,15,5D,W,1
9405 DATA 5,A SMALL SQUARE STONE BLOCK ROOM., 4D,30P,28S,W,0
9406 DATA 6,A LONG WINDING CORRIDOR GOING UPWARD., 7P,W,8S,1P, 0
9407 DATA 7,A NARROW WINDING CORRIDOR GOING WEST., W,27S,6P,22T,
 0
9408 DATA 8,A GREAT ROUND ROOM WITH A HIGH DOMED CEILING., 6S,26
D,13D,28D, 2
9409 DATA 9,THE NORTHEAST CORNER OF A LARGE L-SHAPED ROOM, 14S,W
,10C,11C, 0
9410 DATA 10,THE SOUTHERN END OF A LARGE L ROOM, 9C,W,12D,W, 0
9411 DATA 11,THE WESTERN END OF L SHAPED ROOM, W,9C,W,6SP, 10
9412 DATA 12,A LONG CURVING PASSAGE, 10D,13P,W,W, 0
9413 DATA 13,A LONG CURVING PASSAGE., 8D,W,W,12P, 0
```

```
9414 DATA 14,A DIRTY ROOM FULL OF ORC SIGNS. , W,3D,9S,W, 0
9415 DATA 15,THERE ARE SIGNS OF PASSAGE HERE. , W,22P,W,16D, 0
9416 DATA 16,THIS OPENS UP INTO A LARGE ROOM. , W,15D,W,17C, 0
9417 DATA 17,THE ROOM SHOWS SIGNS OF USE. , 18C,16C,21P,63D, 0
9418 DATA 18,THERE ARE SOME HUMAN SKELTONS IN THE NW CORNER. , W,
19C,17C,W, 0
9419 DATA 19,THERE ARE BONES SCATTERED ABOUT. , W,20D,W,18C, 0
9420 DATA 20,THE ROOM SMELLS OF ROTTING FLESH. , W,W,W,19D, 4
9421 DATA 21,THE ROOM IS DUSTY AND FULL OF SPIDER WEBS. , 17P,W,1
1S,W, 5
9422 DATA 22,A WELL USED CORRIDOR. , 23P,W,3D,15P, 0
9423 DATA 23,A LONG NORTH-SOUTH CORRIDOR. , 29P,24D,22P,49D, 0
9424 DATA 24,A LONG WINDING CORRIDOR. , W,W,25P,23D, 0
9425 DATA 25,A LONG WINDING CORRIDOR. , 24P,W,26P,27S, 0
9426 DATA 26,A LONG WINDING CORRIDOR. , 25P,W,W,8D, 0
9427 DATA 27,A NARROW FLIGHT OF STAIRS GO UPWARD TO THE WEST. , W
,25D,W,7D, 6
9428 DATA 28,A NARROW CURVING PASSAGE. , 5D,8D,W,W, 0
9429 DATA 29,A LONG CORRIDOR. , 31P,W,23P,W, 0
9430 DATA 30,THE CORRIDOR CURVES FROM NORTH TO WEST. , 1P,W,W,5P,
 0
9431 DATA 31,A LONG N-S CORRIDOR. , 32P,38D,29P,45D, 0
9432 DATA 32,THE CORRIDOR ENDS. , 33D,W,31P,W, 0
9433 DATA 33,A SMALL SQUARE ROOM. , F,34D,32D,41D, 0
9434 DATA 34,THERE ARE STRANGE SOUNDS IN THE EAST. , W,35P,W,33D,
 0
9435 DATA 35,DEBRIS IS SCATTERED ABOUT. , W,W,37D,34P, 7
9436 DATA 36,THE WALL SEALS BEHIND YOU. , W,W,W,32S, 0
9437 DATA 37,THERE ARE SOUNDS FROM BEHIND THE DOOR. , 35D,W,39C,W
, 0
9438 DATA 38,A ROOM OF STONE BLOCKS. , 36D,39D,W,31D, 0
9439 DATA 39,AN ORDINARY ROOM. , 37C,W,W,38D, 0
9440 DATA 40,ROUGH STONE WALLS DRIP MOISTURE. , W,41C,42D,W, 0
9441 DATA 41,THERE IS DAMPNESS IN THE AIR. , W,33D,W,40C, 0
9442 DATA 42,NO SIGNS OF RECENT ACTIVITY. , 40D,W,44C,W, 0
9443 DATA 43,THIS ROOM HAS BEEN A MONSTER'S LAIR. , W,W,45S,W, 8
9444 DATA 44,THERE ARE MYSTEROUS MARKINGS ON THE WEST WALL. , 42C
,45D,W,52S, 0
9445 DATA 45,A SENSE OF FOREBODING FILLS THE ROOM. , 43S,31D,W,44
D, 0
```

```
9446 DATA 46,THERE ARE ANIMAL DROPPINGS HERE., W,W,48C,W, 0
9447 DATA 47,THERE ARE BROKEN SKELTONS SCATTERED ABOUT., W,W,49D,W, 9
9448 DATA 48,THERE HAS BEEN PASSAGE THRU HERE., 46C,49P,W,58D, 0
9449 DATA 49,A WIDE PASSAGE., 47D,23D,W,48P, 0
9450 DATA 50,THE LAIR OF THE MINOTAUR., 40S,51P,53P,W, 3
9451 DATA 51,A MAZE., 50P,51P,54P,50P, 0
9452 DATA 52,A MAZE., 51P,44S,55P,51P, 0
9453 DATA 53,A MAZE., 50P,54P,53P,50P, 0
9454 DATA 54,A MAZE., 51P,54P,57P,53P, 0
9455 DATA 55,A MAZE., 52P,  W,58P,54P, 0
9456 DATA 56,A MAZE., 53P,57P,59P,53P, 0
9457 DATA 57,A MAZE., 54P,58P,59P,56P, 0
9458 DATA 58,A LABYRINTH., 55P,48D,57P,57P, 0
9459 DATA 59,A MAZE., 56P,57P,61C,56P, 0
9460 DATA 60,CLEAR, 61C,60C,60C,60C, 0
9461 DATA 61,CLEAR, 59C,63C,60C,62C, 0
9462 DATA 62,KEEP OUT
CONSTRUCTION SITE, 61C,61C,60C,64D, 0
9463 DATA 63,"STRAIGHT AHEAD LEADS THE WAY,
OTHER WAYS WILL GO ASTRAY.", 61C,17D,60C,61C, 0
9464 DATA 64,THERE ARE TOOLS AND BROKEN ROCK SCATTERED ABOUT., 6
1P,62D,60P,P, 11
9465 DATA 65,A SPRING FLOWS OUT OF THE ROCK AND DOWNHILL TO THE
WEST., W,11P,W,66P, 0
9466 DATA 66,THE STREAM DISAPPEARS UNDER THE WESTERN WALL., W,65
P,67P,68S, 0*
9467 DATA 67,A PASSAGE CARVED OUT OF THE ROCK., 66P,70D,72D,69D, 0
9468 DATA 68,THE STREAM FLOWS THRU A NARROW CRACK IN THE FLOOR.,
 W,66D,W,W, 12
9469 DATA 69,BONES AND SKELTONS ARE EVERYWHERE., W,F,71S,W, 0
9470 DATA 70,SOME EMPTY BOXES LIE ABOUT., W,W,W,67D, 0
9471 DATA 71,THE WALLS ARE ROUGH HEWEN STONE., W,72C,W,W, 0
9472 DATA 72,A LARGE ROOM., 67D,73C,74C,71C, 0
9473 DATA 73,DUST LIES THICK HERE., W,W,W,72C, 0
9474 DATA 74, ,72C,W,W,75P, 0
9475 DATA 75,BONES AND STONES ARE SCATTERED ABOUT., W,74P,W,W, 13
```

ROUTINE: **MONSTER DATA**
LINE: 9500-9513
PURPOSE: Monster descriptions.
PROCESS: The format for the monster data is: monster number !M, monster description MD$(!M), attack strength

MA(IM), defense strength MD(IM), and treasure number MT(IM).

```
9500 'MONSTERS
9501 DATA 1,FOUR ORCS ARE GNAWING ON A GNOME,7,6,1
9502 DATA 2,THREE SKELETONS RISE OUT OF COFFINS,6,5,2
9503 DATA 3,A SAVAGE MINOTAUR RUSHES AT YOU !,11,11,3
9504 DATA 4,FIERCE CARRION EATING GHOULS RUSH AT YOU.,7,5,4
9505 DATA 5,A HUGE SPIDER DROPS FROM THE CEILING.,10,5,5
9506 DATA 6,A LARGE POISONOUS CENTIPEDE COMES OUT OF A CRACK IN
THE WALL.,6,2,6
9507 DATA 7,A BAND OF TROLLS IS MUNCHING ON BONES.,8,7,7
9508 DATA 8,A LARGE GREEN CENTIPEDE LIKE CREATURE IS FEEDING OFF
 A CORPSE.,9,4,8
9509 DATA 9,A LARGE BUGBEAR RUSHES OUT TO CRUSH YOU !,9,8,9
9510 DATA 10,A GIANT RAT RUSHES OUT OF A HOLE.  IT LOOKS HUNGRY.
,5,3,10
9511 DATA 11,SOMES DWARVES WITH PICKS AND SHOVELS ARE TUNNELING.
,5,5,11
9512 DATA 12,A LARGE SNAKE COMES OUT OF THE WATER,5,3,0
9513 DATA 13,AN DELIGHTED OGRE THINKS HIS DINNER HAS ARRIVED !,
8,7,13
```

ROUTINE: **TREASURE DATA**
LINE: 9600-9613
PURPOSE: Treasure descriptions.
PROCESS: The format for the treasure data is: treasure number IT, treasure description TD$(IT), and value TV(IT).

```
9600 'TREASURE
9601 DATA 1,A SMALL WOODEN CHEST CONTAINS SOME GOLD.,150
9602 DATA 2,ONE SKELETON WEARS A SWORD WITH A JEWELED HILT.,90
9603 DATA 3,A HEAP OF SILVER AND GEMS AWAITS YOU.,230
9604 DATA 4,THERE ARE SOME COINS IN THE RUBBISH.,  40
9605 DATA 5,THERE IS A LARGE SAPHIRE IN THE SPIDER'S BELLY.,  200
9606 DATA 6,THERE IS A GOLD COIN IN A CRACK IN THE FLOOR.,  1
9607 DATA 7,THERE IS A GOOD SET OF CHAIN MAIL IN THE CORNER.,  125
9608 DATA 8,YOU FIND A SMALL PURPLE GEM.,  60
9609 DATA 9,A BOX OF TRINKETS AND COPPER PIECES.,  20
9610 DATA 10,IN THE RAT'S HOLE IS A SHINY BRACELET.,  50
9611 DATA 11,THEY RUN AWAY AND LEAVE THEIR TOOLS BEHIND.,  5
9612 DATA 12, NONE,  0
9613 DATA 13,THERE IS A BOX OF COINS AND TRINKETS.,  120
9999 END
```

immediate mode

by Dean C. Westervelt

You may never have heard it called "immediate mode" but you have used it often -- every time you tell your TRS-80 to RUN, LIST or EDIT. Did you know that nearly all of the instructions which you would normally include in a program can also be used in "immediate model"? To demonstrate, type in the following mini-program:

```
FORI = 1 TO 10: PRINT I: NEXT
```

When you hit (ENTER), the digits 1 through 10 are printed in a vertical column on the left side of your screen; the immediate mode executes this program -- even if you have a resident program in memory!

You can use this technique to help in debugging programs. After a RUN, ask your computer for the value of the variables in your program. Just type:

```
PRINT X (ENTER)
```

You will immediately see the latest value for X. This won't work if you have just performed an EDIT because editing resets all variables to zero. You can even type PRINT X; Y, Z and get the requested multiple print-out, with the usual BASIC format as specified by the semicolon and the comma.

Multiple-instruction statements like those in the FOR-NEXT loop of the first example give you a strong debugging power, especially useful if you are writing machine code or assembly language. For example, if you need to know the contents of several memory locations, just use something like:

```
FOR K = 15361 TO 15380:  PRINT K; PEEK (K),: NEXT
```

Twenty entries will be printed, each consisting of the memory location, K, followed by a one, two or three digit decimal representation of the contents of that location. In some cases, this may work even better than T-BUG. At least the values are in decimal -- not hex.

If you wish to rewind a tape or preset it at a given point, use immediate mode. You won't have to unplug the remote control cable if you use:

```
OUT 255,4: FOR K = 1 TO 15000: NEXT
```

The Out 255,4 starts the tape recorder (provided you have previously pushed the appropriate button). The recorder will continue to operate while the timing loop is running, about 30 seconds. You can increase or decrease the time by changing the 15000. You might even substitute a large number like 1 million and control the shut-off manually by pressing the (BREAK) key.

Here is a neat trick to use when debugging a loop in a program; type in the loop, but omit the final NEXT. Perform a RUN. The program will go through the loop once and then stop -- it ran out of program lines. Now, in immediate mode, you can type NEXT (ENTER). The loop will be executed again! When you are working with nested loops, and after the last performance of the innter loop (a place where you might eventually have NEXT: NEXT), the machine won't do anything when you enter NEXT. In that case, it is waiting for you to type in another NEXT so it can execute the outer loop. When you feel that the loop is completely debugged, add the NEXT statement to your program and proceed from there.

One more thing; you don't need to type RUN to start a program; you can also use something like GOTO 100. Under this command, your variables are not all reinitiated to zero as they are when you type RUN. This can be a real advantage if you wish to initiate a variable at a given value. Something like the following will do the job;

```
X = 3: GOTO 100
```

Don't forget the immediate command CONT which tells the computer to continue with the program from the point at which it was interrupted by an error, a STOP instruction or by your hitting the (BREAK) key.

Hope you are able to use the "immediate mode" and that it will help you in debugging. Let me know of any novel and interesting uses you find. My address is;

D. C. Westervelt
R.D. #2, Box 171
Acme, PA 15610

GROWING WEARY
Tracing Unstructured BASIC Programs

Introducing...
MMSFORTH

THE LATEST IN A HIGH-LEVEL LANGUAGE SOFTWARE DEVELOPMENT SYSTEM.

MMSFORTH is a new version of the powerful, fast, and structured **FORTH** language, specifically written for the TRS-80, and offers such features as:

- Double precision integer math
- Reverse polish notation
- Blinking cursor, auto-repeat keyboard
- Virtual Memory!
- Editing capabilities similar to Level II
- Supports Z-80 assembly blocks within the **FORTH** structure

MMSFORTH is available in two versions; a complete version for disk systems and a close approximation to that version which is designed for the Level II, 16K tape system. The disk version offers "virtual memory", supports one to four disk drives, does not require any other disk operating system, and also has both disk and tape input/output capabilities.

TAKE ADVANTAGE OF A POWERFUL STRUCTURED LANGUAGE WITH THE EXECUTION SPEED OF A COMPILER.

MMSFORTH is available now from TSE.

MMSFORTH

INTRODUCTORY OFFER

The **MMSFORTH** system diskette or cassette tape provides for the expansion of **FORTH** commands by the user. There are many programs and routines provided as examples of **FORTH** programming, such as:

Routines For:
- String Handling
- Graphics
- File Sorting
- Screen Printing

Programs For:
- Game of Life
- Checkbook Balancing
- String Sort
- Number Guessing Game

The **TRS-80 Software Exchange** intends to fully support the introduction of **MMSFORTH** with the development of supporting application modules. Early **MMSFORTH** projects are:

- floating-point package
- assembler/cross compiler to provide standard TRS-80 load modules
- large flexible mailing list system
- generalized data base management system
- word-processing package (**FORTHWRITE**)

MMSFORTH, by **Miller Microcomputer Services,** includes introductory documentation with further references to the **MicroFORTH** primer of **FORTH, Inc.** This manual is an invaluable reference for the **FORTH** programmer, and can be purchased separately by anyone desiring more information on the **FORTH** language structure.

30-DAY INTRODUCTORY PRICE

MMSFORTH cassette version, Level II, 16K	$45.00
MMSFORTH disk version, Level II, 16K	65.00
MicroFORTH primer	15.00

TSE TRS-80 Software Exchange
17 Briar Cliff Drive Milford, New Hampshire 03055

A UTILITY PROGRAM THAT GREATLY EXTENDS THE KEYBOARD VIDEO AND PRINTER SUBROUTINES IN YOUR LEVEL II ROM!

KVP runs under DOS or Level II BASIC. It is relocatable under your control, and so may be used simultaneously with other machine language programs.

by Lance Micklus

Here are some of the things you'll be able to do;

USE AN EXTERNAL KEYBOARD Or, use any other serial input device in place of the TRS-80 keyboard

PRACTICALLY ELIMINATE KEY BOARD BOUNCE The amount of debouncing is user-adjustable

DISPLAY UPPER AND LOWER CASE LETTERS on your video monitor screen

SIMULATE A RADIO SHACK SCREEN PRINTER using an ordinary printer

USE MOST ANY ASCII SERIAL PRINTER such as Teletype 33 or Spinterm

EXCHANGE PROGRAMS WRITTEN IN BASIC WITH OTHER COM PUTERS From the Sorcerer to the IBM 370 (and TRS-80's, too!)

At least 16K memory required
A UTILITY PROGRAM

Self-relocating for 16K, 32K, or 48k systems
$24.95 on tape $29.95 on disk

LEVEL I in LEVEL II by APPARAT

Remember the Good Old Days? . . . You can recapture them!

Level I BASIC interpreter loads in top 4K of any 16K Level II TRS-80. Allows unmodified load, run, and CSAVE of Level I programs — no conversion hassles! Great for teaching beginners, young and old, to program in BASIC.

"System" tape/Level I for Level II, 16K - $15.00.

TSE TRS-80 Software Exchange

17 Briar Cliff Drive Milford, New Hampshire 03055

ENGINEER

by David Bohlke

The object of this game is to construct a bridge connecting the two blocks at the top of the screen. To do this, you must place your beams in accordance with the inspector's specifications. You may work alone, or several players may compete to see who can construct the bridge in the fewest days.

You will discover a thing or two about construction (the right end of the beam must be supported) and inspectors (who mostly get in the way).

That bridge inspector is a pesky devil. The higher the bridge construction gets, the more he makes a nuisance of imself. No new beam can be placed above the level of his feet, and he tends to dislike heights. If you let him get you in a corner where you cant place a new beam, you have to give up and forgeit the game.

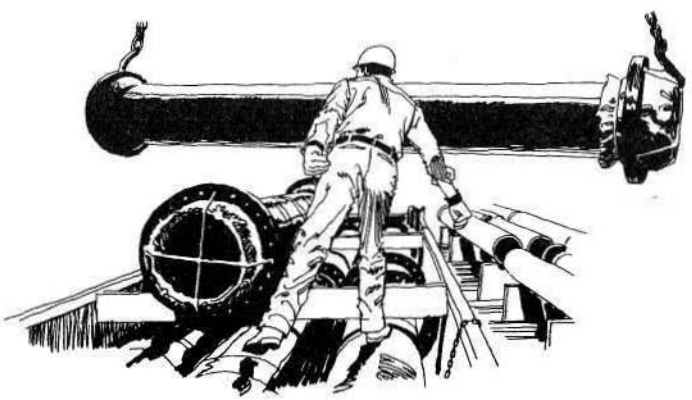
A relatively short, easy game, but still plenty of stimulation; and it's unique -- we know of nothing else that is even similar.

```
10 ' * * * * * * * * * * * * * * *
   *    SOFTSIDE PRESENTS     *
   *         ENGINEER         *
   *      BY DAVID BOHLKE     *
   * * * * * * * * * * * * * * *
20 CLS:DEFINTA-Z
30 PRINT@22,"E N G I N E E R":PRINT
40 PRINT"TO BECOME AN ENGINEER, YOU MUST CONSTRUCT A TRESTLE BRI
DGE"
50 PRINT"CONNECTING THE TWO BLOCKS AT THE TOP OF THE SCREEN.  TH
IS"
60 PRINT"IS DONE BY PLACING BEAMS IN ACCORDANCE WITH THE INSPECT
OR'S"
70 PRINT"SPECIFICATIONS.  JUST MOVE THE BLINKING GUIDE TO THE PO
SITION"
80 PRINT"YOU WISH TO PLACE THE BEAM, THEN ENTER THE DIRECTION YO
U WANT"
```

```
90 PRINT"TO SET THE BEAM.  YOUR STARTING POSITION MUST BE BELOW
THE"
100 PRINT"INSPECTORS' FEET.  SEVERAL PLAYERS MAY COMPETE TO SEE
WHO CAN"
110 PRINT"CONSTRUCT THE BRIDGE IN THE FEWEST DAYS.":PRINT
120 PRINT"DAVID J. BOHLKE    COGGON, IA    JAN 12, 1979":PRINT
130 PRINT"PRESS =ENTER= TO BEGIN . . .";:INPUTZ$:CLS
140 PRINT@128,STRING$(4,191);:PRINT@188,STRING$(4,191);
150 REM   ***   SET CONYON   ***
160 FORX=0TO127:SET(X,43):NEXT
170 Y=8:D=114:X=6
180 SET(X+2,Y):SET(X+3,Y):SET(X+D-2,Y):SET(X+D-3,Y)
190 SET(X,Y):SET(X+D,Y):SET(X+1,Y):SET(X+D-1,Y)
200 Y=Y+1:IFRND(9)>2THENX=X+1:D=D-2
210 IFY=43GOTO220ELSEGOTO180
220 X=45+RND(30):Y=42
230 GOSUB1030
240 M=RND(50)+40:N=0
250 PRINT@25,"E N G I N E E R";
260 IFPOINT(M,N+1)GOTO280
270 N=N+1:GOTO260
280 N=N-10:IFN<0THEN N=0
290 SET(M,N):SET(M+1,N)
300 REM   ***   POSITION BEAM   ***
310 PRINT@960,"MOVE GUIDE :   ";CHR$(93);" =LEFT   ";CHR$(94);"
=RIGHT   ";CHR$(92);" =DOWN";
320 C$=INKEY$:IFC$=""RESET(M,N):RESET(M+1,N):GOTO290
330 IFASC(C$)=9GOTO380
340 IFASC(C$)=8GOTO400
350 IFASC(C$)=10GOTO420
360 GOTO290
370 REM   ***   SET BEAM   ***
380 IFPOINT(M+2,N)GOTO440
390 RESET(M,N):RESET(M+1,N):M=M+2:GOTO290
400 IFPOINT(M-1,N)GOTO440
410 RESET(M,N):RESET(M+1,N):M=M-2:GOTO290
420 IFPOINT(M,N+1)GOTO440
430 RESET(M,N):RESET(M+1,N):N=N+1:GOTO290
440 PRINT@960,CHR$(31);:A$=INKEY$
```

```
450 IFNKYPRINT@960,"STICK MUST START BELOW INSPECTOR";:RESET(M,N
):RESET(M+1,N):FORI=1TO1000:NEXT.GOTO240
460 PRINT@960,"PRESS DIRECTION TO SET BEAM   ( 1-3 ) ";
470 PRINT@761,"1  2";:PRINT@825,CHR$(140);"    3";:PRINT@887,"DI
R ";
480 C$=INKEY$:IFC$=""THEN480 ELSE D=VAL(C$)
490 IF D<1 OR D>3 GOTO 440
500 PRINT@761,"    ";:PRINT@825,"       ";:PRINT@886,"       ";
510 DA=DA+1:M1=M:N1=N
520 FOR I=1 TO 18
530 IF M>122 OR N<4 OR M<2 GOTO 620
540 SET(M,N):SET(M+1,N):SET(M+2,N)
550 IFPOINT(M+3,N) THEN 770
560 ON D GOTO 570,580,590,600
570 I=I+RND(2):N=N-1:I1=I1+1:GOTO610
580 M=M+2:N=N-1:GOTO 610
590 M=M+2:GOTO 610
600 M=M+2:N=N+1
610 NEXT I
620 IFPOINT(M+2,N-1)=1 OR D=1 GOTO 770
630 PRINT@960,CHR$(31);:M=M1:N=N1
640 REM   ***   ILLEGAL PLACEMENT   ***
650 PRINT@960,"THE RIGHT END OF THE BEAM MUST BE SUPPORTED !";
660 FORI=1TO666:NEXT
670 FORI=1TO18:RESET(M,N):RESET(M+1,N):RESET(M+2,N)
680 ON D GOTO 690,700,710,720
690 N=N-1:GOTO730
700 M=M+2:N=N-1:GOTO730
710 M=M+2:GOTO730
720 M=M+2:N=N+1
730 IFN<2GOTO750
740 NEXTI
750 PRINT@832,"DAY ";DA;:PRINT@960,CHR$(31);
760 GOTO240
770 PRINT@960,CHR$(31);
780 REM   ***   MOVE INSPECTOR   ***
790 PRINT@960,"INSPECTION . . .";
800 IF W=0 THEN W=1 ELSE W=0
810 FORI=1 TO RND(40)+10
820 GOSUB1050
```

```
830 IF W=1 GOTO 880
840 IFPOINT(X-2,Y+1)=0ANDPOINT(X,Y+1)=0ANDPOINT(X+2,Y+1)=0THENY=
Y+1:GOTO910
850 IFPOINT(X-1,Y)=0THENX=X-1:GOTO910
860 IFPOINT(X,Y-1)=0THENY=Y-1:GOTO910
870 GOTO910
880 IFPOINT(X,Y+1)=0ANDPOINT(X+3,Y+1)=0THENY=Y+1:GOTO910
890 IFPOINT(X+3,Y)=0THENX=X+1:GOTO910
900 IFPOINT(X,Y-1)=0THENY=Y-1:GOTO910
910 GOSUB1030:IFY<6 OR X<8 OR X>122 GOTO 930
920 NEXT I
930 PRINT@960,CHR$(31);
940 PRINT@832,"DAY ";DA;
950 REM   ***   CHECK FOR COMPLETED BRIDGE   ***
960 FORI=15TO115STEP5:FORJ=4TO8
970 IFPOINT(I,J)GOTO990
980 NEXTJ:GOTO240
990 NEXTI
1000 PRINT@960,CHR$(31);
1010 PRINT@960,"YOU'VE  FINISHED !          PRESS =ENTER= FOR ANO
THER GAME ";:INPUTZ$:RUN
1020 REM   ***   SET INSPECTOR   ***
1030 SET(X,Y):SET(X+1,Y-1):SET(X+2,Y):SET(X+1,Y-3)
1040 SET(X,Y-2):SET(X+1,Y-2):SET(X+2,Y-2):RETURN
1050 RESET(X,Y):RESET(X+1,Y-1):RESET(X+2,Y):RESET(X+1,Y-3)
1060 RESET(X,Y-2):RESET(X+1,Y-2):RESET(X+2,Y-2):RETURN
```



ST80D

Lance Micklus' ST80-the Smart Terminal Program- just got SMARTER!!

ST80D contains extensions for disk drive systems to exchange files with a timesharing computer or another TRS-80.

USING ST80D, your TRS-80 can do all this and more:

- Gather and pre-format data, store it on disk, then transmit it to a timesharing computer for processing.
- Processed data from the timesharing computer can then be sent back to the TRS-80.
- One TRS-80 can generate a data base and share it with another TRS-80 thousands of miles away by telephone.
- Users may customize their terminal program by redefining the translation tables. Conversion from one set of tables to another takes only seconds.
- Auto logon feature sends your account name, number and password upon request.
- ST80D can transmit any type of TRS-80 ASCII file, including BASIC programs stored in ASCII format, and most BASIC data files. Binary files can also be transmitted from one TRS-80 to another, allowing even machine language programs to be sent over the phone.

ST80D is a practical, full-feature terminal program that has been used on a variety of timesharing systems. These include IBM 370, Honeywell Sigma/6, Harris/7, DECSYSTEM 20, Dartmouth Timesharing, CDC Cyber and HP 2000.

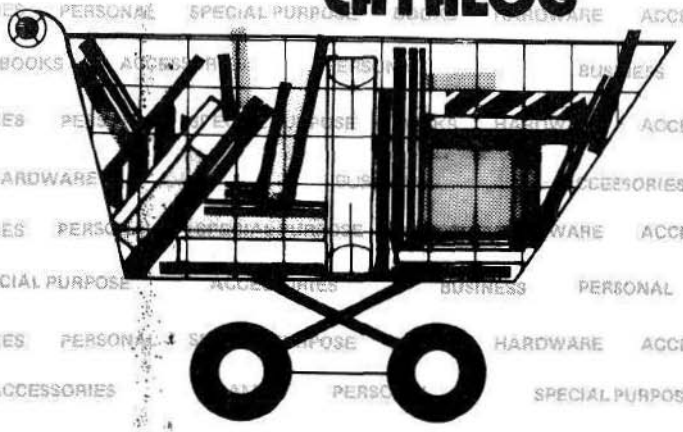
If you're looking for a professional quality product, **not an amateur program**, then order ST80D today!

For 32K disk systems - **$79.95**

TSE TRS-80 Software Exchange

17 Briar Cliff Drive Milford, New Hampshire 03055

TSE Market Basket CATALOG



GAMES

Warfare I 4-game anthology, Level II, 16K $7.95

Backgammon by Scott Adams Level II, 16K $7.95

X-Wing Fighter II by Chris Freund Level II, 16K $9.95

Taipan by Art Canfil - Level II, 16K $9.95.

Cribbage by Roger W. Robitaille, Sr. Level I or II, 16K $7.95

Sargon Chess by Dan & Kathe Spracklen Level II, 16K $19.95

Chess Companion by Michael Kelleher Level II, 16K $7.95

Three D Tic Tac Toe by Scott Adams Level I or II, 16K $7.95

Concentration by Lance Micklus Level I or II, 16K $7.95

Amazin' Mazes by Robt. Wallace Level II, 16K $7.95

Time Bomb by David Bohlke Level I or II, 16K $4.95

Balloon Race by Dean Powell, Level II, 16K, $9.95

Snake Eggs with sound by Leo Christopherson Level II, 16K $14.95

Life Two by Leo Christopherson with Sound, Level II, 16K $14.95

Android Nim with sound by Leo Christopherson Level II, 16K $14.95

Cubes by Leo Christopherson Level II, 16K $9.95

Mastermind II by Lance Micklus $7.95

Mastermind II source list $20.00

Robot/Breakaway Game duo by Lance Micklus Level II, 4K $7.95

Tycoon by David Bohlke Level II, 16K $7.95

Slalom by Denslo Hamlin Level II, 16K $7.95

9 Games for Preschool Children by George Blank Level II, 16K $9.95

Ten Pin by Frank B. Rowlett, Jr. Level II, 16K $7.95

All Star Baseball by David Bohlke, Level II, 16K $7.95

End Zone by R. W. Robitaille, Sr. Level I or II, 16K $7.95

Atlantic Balloon Crossing by Dean Powell Level II, 16K **Special price,** $9.95

Space Battles by Level IV, Level II, 16K Tape or 32K Disk, Tape - $14.95, Disk - $19.95

Star Trek III.3 by Lance Micklus Level II, 16K $14.95

Dog Star Adventure by Lance Micklus Level II, 16K $9.95

Adventures on tape by Scott Adams Level II, 16K Choose one: Adventureland, Pirate's Cove, Mission Impossible, The Count, Voodoo Castle, $14.95.

Adventures on disk by Scott Adams. Pirates Cove plus Adventureland, **or** The Count plus Voodoo Castle for 32K disk system $24.95,
Now in machine language for faster and easier response.

Safari by David Bohlke Level II, 16K $7.95

Treasure Hunt by Lance Micklus Level I or II, 16K $7.95

'Round the Horn by George Blank Level II, 16K $9.95

Pork Barrel by George Blank Level II, 16K $9.95

Kamikaze by Russell Starkey Level II, 16K $7.95

All Star Baseball by David Bohlke Level II, 16K $7.95

Air Raid by Small System Software Level I or II, 4K $14.95

Barricade by Small Systems Software. Machine language - $14.95.

Journey To The Center Of The Earth by Greg Hassett. Level II, 16K Tape $7.95.

Pentominoes by Solomon W. Golomb. Level II, 16K - $9.95.

Inventory 2.2 by M. Kelleher and R.W. Robitaille, Sr. 16K disk systems $59.95

Inventory System 2.3 by M. Kelleher. Improved version, $79.95

Inventory 'S' by BUS-80 Level II, 16K Tape - $24.95; 32K Disk - $39.95

Payroll by Stephen Hebbler for disk systems 32K $59.95

Accounts Receivable II by S. Hebbler 32K disk systems $79.95

Appointment Log by Michael Kelleher Level II, 16K $9.95

General Ledger I by M. Kelleher, requires 32K Disk, $79.95

Mail List II by BUS-80 32K disk systems $99.95

Small Business Bookkeeping by R.W. Robitaille, Sr. Level I, 4K or Level II, 16K **with journal** $22.00; **without journal** $15.00

Small Business Bookkeeping for Disk by Miller Microcomputer Services and Roger W. Robitaille, Sr., 32K Disk. With Journal $31.95; without journal $24.95

SPECIAL PURPOSE

Calculator by R.W. Robitaille, Sr. Level I or II, 4K $2.95

Moving Signboard by Circle Enterprises Level I or II, 4K $9.95

Histograph/Scattergram by Gary S. Breschini. Level II, 16K $9.95.

Simple Simon by George Blank. Level II. Written in BASIC. $4.95

Math Drill by K.L. Brown Level II, 16K $4.95

RPN Calculator by Russell Starkey Level II, 16K $9.95

Ham Radio by Michael Kelleher Level II, 16K $9.95

Ham Radio ARS 1.1 for 32K disk $24.95

Electronics Assistant by John Adamson Level II, 16K $9.95

Preflight by Stephen Hebbler Level II, 16K $20.00

Basic Statistics by Steve Reisser Level II, 16K $20.00

Drill Masters by Computer Graphics — specify title desired Level II, 16K $7.95 EACH: German, Chinese, Russian, Italian, Spanish, or Music Theory.

Keyboard-80 by John Adamson, Level II, 16K $9.95

PERSONAL

Typing Tutor by 80 US. Level II, 16K - $19.95

Secrets of the Tarot by John T. Phillipp. Level II 16K $9.95.

Biorhythms by Frank B. Rowlett, Jr. Level I or II, 4K $4.95

Personal Finance by Lance Micklus. Level II, 16K $9.95

Advanced Personal Finance by Lance Micklus For 32K disk systems $24.95

Home Financial Management by Michael Kelleher Level II, 16K $9.95

UTILITIES

NEWDOS by Apparat $49.95

NEWDOS+ by Apparat $99.95

Three Monitors for Disk by Small Systems Software Disk for 16 through 48K (all on one) $29.95

KVP Extender by Lance Micklus Tape, $24.95 Disk, $29.95

KVP 232 by Lance Micklus - KVP adapted for the TRS-232.
Tape $24.95

ST80 Smart Terminal Level II, 16K $49.95

ST80D SMARTER Terminal for disk systems $79.95

Micro Text Editor by Don Coons Level II, 4K or 16K $9.95

Text80 by Frank B. Rowlett, Jr. For 32K disk systems $59.95

8080-Z80 Conversion Level II, 16K $15.00

Renumber by Lance Micklus Level II, available in 4 through 48K (specify when ordering) $15.00

Renumber source listing $20.00
Renumber on Disk all versions on diskette $25.00

Machine Language Monitor by Small Systems Software Level I or II, 16K $26.95

Electric Pencil by Michael Shrayer. Powerful machine language word processing system. Level II, 16K tape $100; Disk version $150.

Level III BASIC by Microsoft. $49.95.

Level I in Level II by Apparat. Level II, 16K - $15.00.

For more detailed descriptions of our TRS-80 software and accessories, send for the TSE catalog — it's free! Write or call today for your copy:

TRS-80 Software Exchange
17 Briar Cliff Drive
Milford, NH 03055
603-673-5144

Fortran by Microsoft, 32K, 2 Disks — new low price $195.00

MMSForth by Miller Microcomputer Service. Powerful new compiled language — cassette version, Level II, 16K - $34.95; Disk version, Level II, 16K 44.95; MicroForth primer $15.00

System Copy by Kalman Bergen. Level II, 16K $9.95.

BOOKS

Sargon Handbook by Dan & Kathe Spracklen $14.95 plus $1.00 shipping

The BASIC Handbook by Dr. David A. Lien $14.95 plus $1.00 shipping

Z80 Instruction Handbook by Scelbi Publications $4.95

The Little Book of BASIC Style by John Nevison $5.95 plus $1.00 shipping

TRS-80 Assembly Language Programming by William Barden, Jr. $3.95 plus $1.00 shipping and handling

HARDWARE ACCESSORIES

Cassettes boxes of ten each. C-10, $6.50 plus $1.00 shipping; C-20, $7.50 plus $1.00 shipping

Diskettes Dysan, box of 5, $24.95 plus $1.00 shipping; Verbatim, box of 10, $34.95 plus $1.00 shipping

Diskette Storage Box $5.00

TRS 232 by Small Systems Hardware — $49.95

Percom Disk Drives single or dual, for TRS-80's. Single drive, $399.00; Dual drive, $799.00; Cable (required) $29.95

Floppy Armour™ Protective envelope for shipping floppy disks, 5-pack - $4.95 plus $1.00 shipping & handling

16K Memory Kits by Ithaca Audio, 16K RAM's $99.95

IMPORTANT

- **No sales tax.**
- **All C.O.D.'s or special delivery orders are a minimum of $5.00 for special handling.**
- **When ordering Percom, please add $5.00 [each] packaging and handling fee.**

WANTED: Used TRS-80 equipment! We buy and sell used equipment. Call or write for details.

SEE US AT THE NORTHEAST COMPUTER SHOW
SEPTEMBER 28-30, 1979
Hynes Auditorium / Prudential Center
Boston, MA
BOOTH CS67

TSE Order Form

Special prices in effect 60 days from mailing

DESCRIPTION	MEMORY LEVEL		PRICE
ADD HANDLING CHARGE			$1.00
ADDITIONAL CHARGES			
TOTAL ENCLOSED WITH ORDER			

☐ Check ☐ VISA ☐ Master Charge

☐ Money Order

ALL SOFTWARE GUARANTEED TO LOAD AND RUN. If you experience difficulties, simply return the tape or disk for free replacement. Send to the attention of Bette Keenan, Customer Service Representative; please enclose a brief note and your name and mailing address with the software.

Not responsible for typographical errors

TRS-80 Software Exchange

17 Briar Cliff Drive Milford, New Hampshire 03055

Order toll free: 1-800-258-1790

Level II software available on disk for a $5.00 (**per order**) medium charge. This extra fee is for any number of programs transferred to disk from tape when you order. If the order exceeds the capacity of a single disk, we absorb the extra cost.

Please state level and memory size on order form ... otherwise, we automatically ship Level II cassettes.

Be sure to include handling charge and any additional charges when figuring your total. All orders shipped within 48 hours.

Charge card account number

Signature.....

Exp. Date.....**Inter. #**.....

Charge customers: Please fill in account information above and below

Name.....

Address.....

City.....**State**.....**ZIP**.....

ALL SOFTWARE SOLD ON AN AS-IS BASIS WITHOUT WARRANTY TSE assumes no liability for loss or damage caused or alleged to be caused directly or indirectly by equipment or products sold or exchanged by them or their distributors, including but not limited to any interruption in service, loss of business or anticipatory profits or consequential damages resulting from use or operation of such equipment or software.