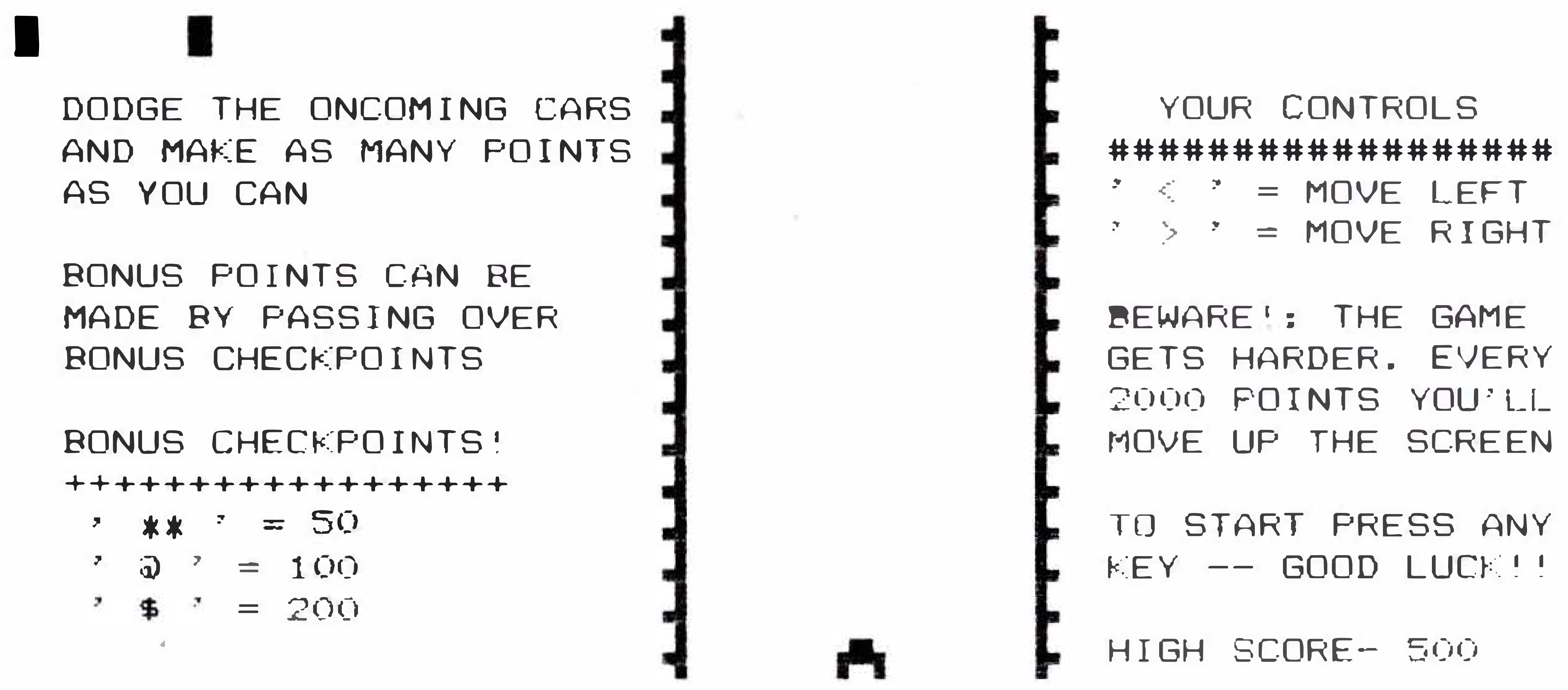
MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034. AUSTRALIA. TELEPHONE (08) 211 7244. PRICE: AUS. $3.50, N.Z. $5.00, U.K. £1.50
Registered by Australia Post — Publication No. SBQ2207

Vol. 4, Issue 8, 1984

INSIDE: PROGRAMS FOR THE VZ 200



```
DODGE THE ONCOMING CARS
AND MAKE AS MANY POINTS
AS YOU CAN

BONUS POINTS CAN BE
MADE BY PASSING OVER
BONUS CHECKPOINTS

BONUS CHECKPOINTS!
++++++++++++++++++
 ' ** ' = 50
 ' @ ' = 100
 ' $ ' = 200
```

```
  YOUR CONTROLS
##################
' < ' = MOVE LEFT
' > ' = MOVE RIGHT

BEWARE!: THE GAME
GETS HARDER. EVERY
2000 POINTS YOU'LL
MOVE UP THE SCREEN

TO START PRESS ANY
KEY -- GOOD LUCK!!

HIGH SCORE- 500
```

TRACK 80

Also in this issue:

ARTICLES:
Recreation-80
TDISIC Review
Notes from the Software Editor

SOFTWARE:
Field Finder — (Disk)
Latin Vocab Test — Level I
Lander — (Colour)
Touch Typing — Level II
Obstacle — Level II
Dog Race — VZED
Contest Log — VZED
Memory Peek — VZED

•TRS-80 •SYSTEM 80 •VIDEO GENIE
•PMC-80 •VZ 200
•TRS-80 COLOUR COMPUTER

ANNOUNCING THE '80 XT EXPANSION FOR SYSTEM 80 AND TRS-80 COMPUTERS FROM $1,199

DISK CONTROLLER, 32K RAM AND TWO DISK DRIVES ALL IN THE ONE ATTRACTIVE, COMPACT CABINET

The TRS-80/System 80 computer when equipped with additional memory and disk drives is still one of the most versatile and powerful home computer systems available. It makes a powerful word processor or data base manager which can be used in serious applications. If you would like to increase your computing power and experience economically with proven equipment and software, you should seriously consider upgrading your L2/16K machine by the addition of the appropriate '80 XT expansion.

XT stands for EXTRA and MICRO-80's '80 XT has plenty of extras. The one attractive, vinyl covered metal cabinet houses:

- ☐ Two slimline disk drives of 100K, 200K or 400K capacity each.
- ☐ A heavy duty switching power supply to give cool, reliable operation free from power glitches and random "reboots".
- ☐ DOSPLUS 3.5 disk operating system.
- ☐ MICRO-80's proven expansion interface board giving:
  - up to 32K static ram: to ensure high noise immunity and reliability
  - single density disk controller: for complete compatability with all disk operating systems
  - centronics printer port: the system 80 model has a double-decoded port to respond to both port FD and memory address 37E8H thus overcoming one of the major incompatabilities with the TRS-80.
  - RS232 communications port: for communicating by modem or direct link to other computers
  - real time clock interrupt: provides software clock facility used by most DOS's

Economical double density: an economical, high quality double density upgrade will be released shortly to enable you to increase the capacity of your disk drives by 80%.

THE INTEGRATED DESIGN OF THE '80 XT SAVES YOU MONEY TOO:

'80 XT WITH 0K RAM AND TWO SINGLE—SIDE 40 TRACK DISK DRIVES (100K byte each)	**$1,199**
'80 XT WITH 0K RAM AND TWO DOUBLE-SIDE 40 TRACK DISK DRIVES (200K byte each)	**$1,299**
'80 XT WITH 0K RAM AND TWO DOUBLE-SIDE 80 TRACK DISK DRIVES (400K byte each)	**$1,499**

ADDITIONAL 16K RAM **$99** ADDITIONAL 32K RAM **$198**

All configurations available ex stock NOW
Be sure to specify whether you have a TRS-80 MODEL 1 or a SYSTEM 80.
Add $12.00 delivery anywhere in Australia.

CONTENTS

REGULARS

EDITORIAL	2
INPUT/OUTPUT	5

DEPARTMENTS

V-ZED	2
KALEIDOSCOPE	2
FORM THREE	3
GROUP ONE	3

ARTICLES

RECREATION 80	4
NOTES FROM THE SOFTWARE EDITOR	6
T DISK REVIEW	6

SOFTWARE

LATIN VOCAB TEST (L1/4K)	7 & 17
LANDER COLOUR)	7 & 13
OBSTACLE (L2/4K)	7 & 18
TRACK 80 (L2/16K)	8 & 19
TOUCH TYPING (L2/16K)	8 & 20
FUNDAMENTAL SORT UTILITY (48K DISK)	8 & 22
DOG RACE (VZED)	9 & 16
CONTEST LOG (VZED)	9 & 16
MEMORY PEEK (VZED)	9 & 15
FIELD FINDER (48K DISK)	9 & 11

ABOUT MICRO-80

EDITOR: IAN VAGG

MICRO-80 is an international magazine devoted to the Tandy TRS-80 Model 1, Model III and Colour microcomputers, the Dick Smith System 80/Video Genie and the VZ-200. It is available at the following prices:

	12 Months	**Single Copy**
MAGAZINE ONLY	$ 36.00	$ 3.50
CASSETTE SUBSCRIPTION	$ 96.00	$ 6.00
DISK SUBSCRIPTION	$125.00	$10.00 (disk)

MICRO-80 is available in New Zealand from:
MICRO PROCESSOR SERVICES, 940A Columbo Street, CHRISTCHURCH 1 NZ. Ph. 62894

MAGAZINE ONLY	NZ$ 59.00	NZ$ 5.60
CASSETTE SUBSCRIPTION	NZ$130.00	NZ$ 7.50
DISK SUBSCRIPTION	NZ$175.00	NZ$15.00

MICRO-80 is despatched from Australia by airmail to other countries at the following rates:

(12 MONTH SUB)	**Magazine**	**Cass Sub**	**Disk Sub**
PAPUA NEW GUINEA	Aus$53.50	Aus$115.50	Aus$148.50
HONG KONG/SINGAPORE	Aus$58.00	Aus$122.00	Aus$157.50
INDIA/JAPAN	Aus$64.00	Aus$129.00	Aus$165.00
USA/MIDDLE EAST/CANADA	Aus$73.00	Aus$140.00	Aus$177.00
UNITED KINGDOM/EUROPE	Aus$75.00	Aus$150.00	Aus$180.00

Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80, System 80/Video Genie or VZ-200 and its peripherals. MICRO-80 is in no way connected with the Tandy or Dick Smith organisations.

WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS: *Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your microcomputer to earn some extra income is included in every issue.*

CONTENT: *Each month we publish at least one applications program in BASIC for each of the microcomputers we support. We also publish Utility programs in BASIC and Machine Language. We publish articles on hardware modifications, constructional articles for useful peripherals, articles on programming techniques both in Assembly Language and BASIC, new product reviews for both hardware and software and we printer letters to the Editor.*

COPYRIGHT: *All the material published in this magazine is under copyright. This means that you must not copy it, except for your own use. This applies to photocopying the magazine itself or making copies of programs on tape or disk.*

LIABILITY: *The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.*

MICRO-80 is Registered by Australia Post — Publication No. SBQ2207

AUSTRALIAN OFFICE AND EDITOR: *MICRO-80, P.O. Box 213, Goodwood, S.A. 5034. Tel. (08) 211 7244*

TYPESETTING & MAKE-UP: *Formgraphic, 117 Wright Street, Adelaide, S.A. 5000. Tel. (08) 211 7866*

PRINTED BY: *Specialty Printers, 42 Wodonga Street, Beverley, S.A. 5009*

PUBLISHED IN AUSTRALIA BY: *MICRO-80, 433 Morphett Street, Adelaide, S.A. 5000*

EDITORIAL

Some welcome news came to hand recently, the Federal Government has decided to drop customs duty on imported software. Until now, software imported from most overseas countries attracted 35% import duty and then a further 24% (usually) Sales Tax on top of the duty paid price. In all an impost of 67.4%. From now on only the media will be dutiable. i.e. the cost of the cassette or disk will be subject to duty but not the cost of the programs. The Customs department generally assesses the value of a disk at $AUS5.00 therefore, a disk program would attract a combined duty and Sales Tax of $3.37, certainly far less than previously.

The more cynical amongst us would no doubt see this as another pointer towards an early election! The decision is significant however, for several reasons of more importance to the fledgling microcomputer industry. At its crudest it shows that the industry has gained sufficient influence to be heard in the corridors of power. Perhaps more importantly, it shows the increasing awareness of Governments that this industry has considerable potential for creating new jobs and deserves to be encouraged. From the practical point of view the Government is unlikely to lose a great deal of revenue since the high volume sellers of software had managed (legally) to circumvent duty by establishing licensing arrangements with overseas suppliers which enabled them to reproduce programs in Australia from masters which were allowed in duty free. One of the most cogent reasons given for this change of heart was that the duty which was intended to foster an Australian software industry, actually had the opposite effect. Whilst the high volume programs avoided duty as described above, low volume software such as programming languages which are the tools-in-trade of the software houses, had to be imported directly thus pushing up the development costs of Australian software.

So, what will all this mean to the average computer user? The potential drop in price has been somewhat eroded by the recent weakness of the Australian dollar against the US dollar but we may look forward to price reductions of 10-15% at least and much more in some cases. We could also expect to see an even wider choice of programs as it becomes economical for importers to bring in small quantities of software. Certainly good news for all of us.

We recently had possession for a brief period of the newly released TRS-80 Model 4P, the portable version of the Model 4. Our acquaintance was too short for a full report but first impressions are good. The keyboard has a nice feel and the 9 inch display is adequate even for lengthy sessions at the machine. We will endeavour to bring you a full review in a future issue.

DEPARTMENTS

V-ZED

Last Issue we explained how to obtain three new functions from the VZ200, including a POKE which turns off the beeping keyboard. Reader Ken Hicks became concerned that this latter recommendation might actually cause some damage to the innards of the computer and possibly to the speaker itself, he writes:

I read with some interest your piece on the new functions for the V-ZED.

It was on the strength of your supporting this machine that I bought one for my young son. To date I have had no joy with the darn thing — it has twice been returned for service, and I have not yet received it or a replacement.

I purchased a copy of the Technical Reference Manual with the unit, so while waiting for the unit to turn up again, I have read the manual from cover to cover, which probably is not a bad idea, but which I almost certainly would not have done under normal circumstances. This Manual gives full circuit diagrams and reveals the very much simplified address decoding. There is also some very useful information on the System pointers, memory mapping, and particularly the details of graphics.

The addresses of a few routines in ROM are given, which will be familiar to ML programmers who use the old Microsoft ROM. For example, 28A7H and 01C9H are still message output and clear screen routines.

Evidently the writer of your article has not studied his TR Manual, as it gives details of the function of an output latch which effectively occupies all locations from 6800 to 6FFF inclusive. This is a write-only latch which services the cassette output, speaker, and video display controller. This latch is copied at 783B (30779), and its bit allocation is:

Bits 0 & 5 drive the speaker. They are normally toggled alternatively in a push-pull fashion to produce a tone. Holding one bit at '0' would therefore hold the speaker diaphragm 'pushed', while holding the other bit at '0' would keep it 'pulled', with an audible click as it went from one state to the other.

Bits 1 & 2 generate the cassette output signal. Fiddling with these could corrupt a tape if the cassette were in the RECORD position!

Bit 3 controls the VDC display mode. An '0' here sets MODE (0), while a '1' causes the VDC to operate in MODE (1). This effect is via the video controller chip.

Bit 4 controls the background colour. It it is '0' then the background will be green, while if it is '1' the background will be orange if in MODE (0) and buff if in MODE (1). Thus, its effect depends on bit 3.

The BEEP routine is at 3450H. Calling this address will produce a BEEP, but some disassembly around this area would be necessary (or perhaps around the keyboard scanning area — from 2EF4H) to find out how to silence the BEEP. It is possible that the brute force method suggested by your correspondent could damage the speaker or a chip by passing a current continuously, which is apparently what happens when '0' is POKED into 30779. I don't want to disparage your correspondent, but this just could be one instance where it is possible to cause physical damage to a computer via the keyboard!

Thank you Ken. There are two minor errors in your analysis of the situation of which one is significant to this discussion. Firstly, to correct a point of fact, bit 5 of the output latch is always held high whilst bit 0 is toggled from high to low to produce sound from the speaker. Of far more significance than that, however, is the nature of the "Speaker" itself. It is a piezo electric device. i.e. it consists of a crystalline substance with two metallised plates, one connected to bit 5 the other to bit 0. When there is a voltage difference between these two plates, the crystal actually changes shape, thus displacing the air surrounding it causing a "Click" to be heard (if the differential voltage has been applied rapidly enough). The BEEP routine you mention at 3450H alternatively sets and resets bit 0 thus applying a continually varying voltage across the crystal causing it to change shape rapidly and emit an audible tone. During this process very little energy is disippated since the piezo electric device appears electrically like a capacitor being alternatively changed and discharged. This device will not be damaged by applying a constant potential across it which is within its operating range. Nor will any IC be called on to carry excessive currents. In short, the POKE's recommended will not cause any harm to the computer. Nevertheless, thank you for raising this interesting subject. We would welcome similar contributions from our other readers.

KALEIDOSCOPE

This month's contribution comes from the "Adelaide Micro User News" and was written by Geoffrey Williamson, 18 Grevillea Cres., Stonyfell, 5066.

SPRITE GRAPHICS FOR YOUR COCO!

One of the most interesting features advertised for another brand of computer is the use of Sprite Graphics. As the accompanying program shows, the CoCo can also perform such feats with a little help from the programmer. In fact, I think you will agree that the methods used here are even easier than those which Commodore uses.

Let's run through the following program a line or two at a time . . .

```
10 DIM IV(17)
20 DIM BL(17)
30 PCLS:PMODE3,1:DRAW"BM40,
   40C6NG3F3R2NF5E2U8H2L9G2D8F
   2NG5R2"
40 PAINT(39,39),6,6
50 CIRCLE(34,28),1,8
60 CIRCLE(46,28),1,8
70 GET(27,25)-(52,50),IV,G
80 SCREEN1,1:PCLS
90 DRAW"BM0,180C7E35F20E90F45D
    12E45F40"
100 PAINT(20,180),7,7
110 CIRCLE(200,65),25,8
120 PAINT(200,60),8,8
130 Y=35:S=1
140 FOR X=0 TO 230 STEP 25
150 GET(X,Y)-(X+25,Y+25),BL,G
160 PUT(X,Y)-(X+25,Y+25),IV,OR
170 FOR Z=1 TO 80:NEXTZ
180 PUT(X,Y)-(X+25,Y+25),BL,AND
190 NEXTX
200 Y=75
210 S=S-1:IFS THEN 130
220 GOTO 140
```

Lines 10 and 20 dimension our arrays for use in the GET and PUT statements. Do NOT use the formula recommended by the Shack, unless you wish to use a great swag of memory. First of all find the elements in Get: GET(27,25)—(52,50): (52-27+1)(50−25+1) = 26*26 = 676*

Next find the divisor: as we are using the Graphics option 'G' and we are using PMode 3, then the divisor is 8.

Therefore 676/8 = 84.5 which is 85 when rounded up, Now divide this number by 5 which gives 85/5 which yields 17. Now DIM IV(17). This is an enormous saving of memory over DIM IV(25,25). RUN both and check the saving of memory if you are a disbeliever! The same applies to the BL array. Line 30 DRAW's our Invader In Color 6 and Line 40 colours it in. Lines 50 and 60 draw his 'eyes'.

Line 70 GET's this array and stores it for later use. (Note that this has not been displayed as no SCREEN statement has yet been used.)

Lines 80 to 120 DRAW the sun and mountains, while lines 130 and 140 initialise our counters.

Things now start to get interesting: Line 150 GET's the background where we are about to PUT our invader. We will use this array (BLank out array) shortly to remove the invader.

In line ,160 we PUT our invader array./Line 30 DRAW's our Invader In suffix instead of PSET or PRESET. This means that all pixels that are ON are set to the FOREGROUND colour, which in this case is orange. After the delay in line 170 we PUT the BLank out array on the same spot, this time using the suffix AND.

The use of the logical operator AND means that the only pixels that are ON are those that were in the original background, whatever that was. All this without using an array filled with blanks!

If you think that an array full of blanks is an alternative way of doing this, just try to implement it with a varying background as in this example!

Lines 190-220 just serve to loop program. (The logic in Line 210 is there to keep the Editor from falling to sleep over his proofs.)

Let's look at the use of the AND and OR operators in a little more detail — when the OR operator is used as a suffix with the PUT statement it guarantees that the area to be OR'ed is set to the foreground colour. When the invader is passing the 'sun' the invader appears to pass BEHIND the sun, as the sun itself is of the foreground colour orange. Hence we produce the illusion of the invader passing behind the sun.

In a slightly less satisfactory fashion the OR operator produces a change in colour of the invader when it passes the mountains. The illusion is of the invader passing in FRONT of the mountains, but the change in the colour of the invader does not quite keep with the illusion.

If such 'tricks' are used in PMode 4 then the illusions can be more satisfying.

*(**Ed's Note:—** I look forward to seeing your efforts at programming using the above methods.)*

FORM THREE

For those of you who are using TRSDOS 1.3 on the Model III we have some patches which should make the DOS more enjoyable to use. These patches are for the TRSDOS 1.3 ONLY!!! Also, make sure you apply them to a backup copy of the DOS, just in case something goes wrong. Patches are applied using the TRSDOS Patch utility and may be created into a BUILD file if you have more than one disk to patch. All patches appeared in 'The Alternate Source'.

The following patch will give the File Patch Utility (Model III TRSDOS) full access to all files with a protection level less than seven (no access). In effect, it will disable password protection in DEBUG (TRSDOS 1.3).

PATCH	ADD	FIND	CHG
*5	52EB	CB	36
*5	52ED	BE	ØØ

This Patch will let you bypass the DATE question.

```
PATCH *Ø (ADD=4EB5,
FIND=CD1BØ2,CHG=B72846)
```

To get long ERROR messages, install the following patch:—

```
PATCH *4 (ADD=4E28,
FIND=2Ø,CHG=18)
```

These patches will alter the stepping speed of TRSDOS 1.3 from 6 msec to 10 msec. Use 0FH & 1FH for 20 msec.

PATCH	ADD	FIND	CHG
*Ø	42EE	ØC	ØE
*Ø	4516	ØC	ØE
*Ø	4544	1C	1E
*Ø	4FE1	ØC	ØE

To set up the scenario for the following patch you might like to go into BASIC and enter CMD"&"&. This is an undocumented command which displays a TANDY copyright message. The space used for this message will be used to install a patch which will speed up the loading of BASIC programs which were saved in the compressed format. Currently they are loaded a byte at a time. This patch was devised by Jesse Bob Overholt.

ADD	FIND	CHG
5BFE	2AA44Ø	CD8754
5CØ7	FF	FE
5CØD	CD535F7723	CD9B54ØØØØ
53CC	8754	4A1E
5487	E17EFE26	2323E5DD
548B	C24A1ED7	E1ED5BA4
548F	C24A1EE5	4ØØ133ØØ
5493	219B54CD	Ø9Ø1FFØØ
5497	3F56E1C9	EDBØEBC9
549B	35Ø13D3C	DD75Ø3DD
549F	26751734	74Ø4DD36
54A3	263C3675	Ø5ØØDDCB
54A7	3C267516	Ø1EEDD34
54AB	1AØ5ØCØ7	ØA2ØØ3DD
54AF	1C121DØ1	34ØB24C3
54B3	1Ø11	535F

Once these patches are installed your programs should load about 50% faster, with less improvement being noticed on very short programs. Programs which leave less than 342 bytes of RAM after loading will cause an 'OUT OF MEMORY' error after these patches are installed. There will be no effect when loading programs which were saved in ASCII format.

Enjoy your new TRSDOS!

GROUP ONE

This month we would like to bring your attention to some bugs in the Microsoft Basic interpreter as included in the Model I. Users of the CoCo and VZ200 might like to try and see if these bugs are also present in their computers.

Firstly, there is a problem with BASIC's handling of the "raise to the power" function. Enter the following program into your computer and 'RUN' it:—

```
10 FOR X=1 TO 15
20 PRINT 2↑X
30 NEXT
```

The resultant printout will be as follows:—

```
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192.01
16384
32768
```

Whilst the above problem probably won't occur all that often, it is a good idea to be aware of it. The same applies to the following bug.

RND(X) can return a value of X + 1 when X is a power of 2. In cases where RND(0) is just under the value of one, when multiplied by X, the product is rounded and this is where the problem occurs. For instance, A = RND(16) can return a value for A of 17. To get around this, use the following:—

```
10 A=RND(16) : IF A>16
THEN 10
```

The next bug can be found if you try and use the expression PRINT VAL ("% ") in your program. Whenever you have a % sing in a string to be converted by VAL you will get a syntax error. This bug also appears in the Model III ROM. To avoid this error in Disk Basic use the following routine:—

```
1000 I=INSTR(X$,"% ")
1010 IF I THEN X=VAL
(LEFT$(X$,I-1)) ELSE
X=VAL(X$)
```

Non-disk users should use the following:—

```
1000 FOR I=1 TO LEN(X$)
1010 IF MID$(X$,I,1)="% "
THEN 1040
1020 NEXT I
1030 I=LEN(X$)+1
1040 X=VAL(LEFT$(X$,I-1))
```

This final bug also appears in all versions of the 'Level II' ROM. Enter the following program and 'RUN' it:—

```
10 INPUT A#
20 A#=INT(A#)
30 PRINT A#
```

If you were to enter −56320 in answer to the prompt, the computer would come back with a result of −56576. To explain, when taking the INT function of a double-precision number which is evenly divisible by 256 and is less than −32768 one extra bit is turned on when processing the number which is subsequently reduced by 256, 512 or some other power of 256. To avoid this add the following filter to your program:—

```
100 A#=SGN(A#)
*INT(ABS(A#))
```

The first bug was mentioned originally in '80-US'. The rest of these bugs were first mentioned in 'The Alternate Source'.

RECREATION 80

by Ed Grigonis

This month's column will be devoted to some letters which have been received containing queries about various of the 'Med Systems' Adventures.

Gavin Daniles, 46 Fossickers Way, Warrandyte, Vic. 3133, writes as follows:—

" I thank you for helping me with my problem of loading programs from cassette to disk. I also have another problem along this line and hopefully you will be able to help me again. The problem is this: every time I try to load a BASIC program from cassette while under disk BASIC when the program has finished loading, the computer doesn't turn off the cassette motor. Is the problem with the BASIC or in the computer itself. I might also add that the programs load perfectly when I use non-disk BASIC.

I also have the game 'ASYLUM' and thanks to help in earlier editions of 'MICRO-80' I have been able to finish the game. I now have a copy of 'ASYLUM II" and am having problems with that. The stage I have reached is that I have found an inmate wanting a battery, magnet and some wire. I have found the battery and know that the magnet and wire are found in the payphone. The problem is that everything I try only gets me in trouble. I have been told that you are supposed to use the axe and chop the pay phone but every time I try it tells me that violence is punished and returns me to my cell. Could somebody please help?"

(Firstly, about the cassette problem which I mention here as it is part of your letter. I have noticed this happen myself once or twice in the past but haven't worried about it. It would help if you could let us know which computer (Model I or II or System 80) and DOS you are using so we can investigate the matter. In the meantime, has anyone out there had this particular problem and, if so, how did you solve it?")

Now to "ASYLUM II"! I strongly recommend you send it back to 'Med Systems' and ask for a refund. A couple of members of the Adelaide Micro User Group went to a lot of trouble to try and get a copy of this Adventure. After a number of phone calls overseas they were eventually informed by 'Med Systems' that all copies of the program had been recalled as it contains bugs which make it unsolveable. Apparently, to get past the second level you have to complete five tasks in a specific order. Unfortunately, this doesn't have the desired effect so you can never get past this level. The company may release a working version for other computers but has no plans to re-release it for the TRS-80, etc.

John Taylor, 21 Drysdale Ave., Frankston, Vic. 3199, writes as follows:—

"I have been trying to solve 'LABYRINTH' for about two years now and it has been driving me INSANE (but I proved my sanity by solving 'ASYLUM' thanks to clues in 'MICRO-80'). PLEASE could someone tell me how to get past the UGLY LITTLE MAN and/or the CAVE GNOME. Any help you can provide would be greatly appreciated.

P.S.: Has anyone got any clues on how to get to the 5th level in 'DEATHMAZE 5000'?"

(The CAVE GNOME can be despatched if you remember some of what you may have learned at Sunday School, particularly the aftermath of Sodom & Gomorrah. You have to give him something.

As for the UGLY LITTLE MAN, he doesn't listen to reason so you will have to send him on his way with a particular weapon. Don't hold back! You will apparently need the means to effect a fast getaway (Not a car!). You will then find something but, as they say, curiosity killed the cat. Show some kindness to the Bear.

In 'DEATHMAZE 5000' you will need to find the calculator for a clue. You can get out of this location by turning right 5 times, left 4 times and right 3 times (this is from memory so it may be the other way around!).

John Dodds, 76a Karomiko Road, Wanganui, NZ, writes:—

I write seeking help with 'ASYLUM'. Having made use of your information to obtain the pass-key I have found my way into a series of offices in the guards' quarters. Firstly, how do I read the note on the desk. Secondly should I have obtained something from the "roadster" earlier on in the second maze? Such items as the voltage regulator, etc. leave me wondering.

Here's hoping your "professional" can give me some broad hints to help me on my way."

(Have you tried the most obvious method of reading the note? The reply to the following letter may help you with the second query.)

P.R. Schlesinger, 219 Ramsay Street, Haberfield, NSW, 2045 writes:—

"I bought the program 'ASYLUM' and have progressed (with the help of clues from 'Micro-80') to the third maze. On entering the Professor's office, and typing 'HELP' the message 'he needs parts' appears. I assume this refers to automobile parts. Where can these parts be obtained? I have tried stopping the roadster (in the second maze) by dropping objects in its parth, such as the nails, and using the lantern. But this was unsuccessful. Is this where the parts can be found? I would appreciate any help given."

(I am told that you are within 7-8 minutes, in game time, of solving the

Adventure. You've got the right idea but why hang around and wait for the carnage? Are you a sadist?)

Jeremy Terhoeve, P.O. Box 289, Alderly, QLD. 4051, writes:—

"I would like to submit the following for your Input/Output column in the next available issue of 'Micro-80".

Could any of your readers please give me some help, advice or clues on 'ASYLUM'?

I am in the second maze of 'ASYLUM' and have so far got the following items: matches, gold, copper, key, marbles and nails. Of course I have encountered the roadster and tried to avoid it, but I think you have to get past the roadster somehow because in the vocabulary it has some car parts listed. I have tried every logical and possible way to get past the roadster but have failed. Do you have to get past the roadster and, if so, how?

(Hopefully, the previous letter, and my reply to it, will answer your question.)

I would like to thank Mark Lively of the Adelaide Micro User Group for his assistance in providing answers to the above questions.

Next month I hope to review one of the games from the Molymerx catalogue and a CoCo game available from Software Spectrum.

INPUT/OUTPUT

P.A. Pawelski, 24 Osmond Street, Maitland, S.A., 5573 writes:—

"I have a TRS-80 Model I LII/16K cassette system. Back in Issue 10 a program called 'Lotto Prediction System' by P. Hartley was published. The data is inputted in Line 100 as:—

```
100 INPUT #- 1, L(K1,1), L(K1,2),
  . . . . . . . . . . . . . . . . . . L(K1,40)
```

As all of line 100 will not fit on one line could you please advise me of the correct way to program this line. As I see it, there are two ways:—

```
80 FOR K1 = 1 TO 40
100 INPUT#—1, L(K1,1), L(K1,2)
  . . . . . . . . . . . . . . . . . . L(K1,20)
105 INPUT#—1, L(K1,21), L(K1,21)
  . . . . . . . . . . . . . . . . . . L(K1,40)
```
or
```
80 FOR K1 = 1 TO 20
100 INPUT#—1, L(K1,1), L(K1,2),
  . . . . . . . . . . . . . . . . . . L(K1,20)
102 NEXT K1
103 FOR K1 = 21 TO 40
105 INPUT#- 1, L(K1,21), L(K1,22)
  . . . . . . . . . . . . . . . . . . L(K1,40)
```

Using the first example takes twice as long to load as the leader is written again in line 105 and the recorder switches on & off 80 times.

Although the second listing is quicker, which is correct?

In the 'Notes From the Software Editor' in Vol. 4 No. 6, it is stated that you have enough Lotto programs for the Model I but how about publishing such a program? I have got all editions of your fine magazine but the only Lotto program I have seen is the one in Issue 10 which is basically for disk users.

Also I am having problems using 'Faster' a program by J. Langsford, published in Vol. 3 No. 10. Using the statement #PRINT#- 1, . . . I get the leader printed on tape but then an SN Error. The line works OK using the statement PRINT#- 1, . . . All other statements e.g. #CLOAD, #CSAVE"A", etc. work fine. I typed the program in using ZMONL, and I am confident there are no errors as I have disassembled the program using Gregg Nott's Disassembler (Issue 18) and also 'Faster' works OK on a System 80 machine. Could there be a difference between the ROM or Reserved RAM Addresses used in the TRS-80 & the System 80?

Hoping you can help me, I thank you in anticipation."

(In reply to the first part of your letter. Have you tried entering the line using Edit Mode as this allows you to enter lines of up to 255 characters?

As to which method is correct, judging from your letter, both methods work and are therefore 'correct'. The second method is obviously preferable as you say it is faster.

See this issue's 'Notes From The Software Editor' for the mention of Lotto programs.

*Has anyone else out there had similar problems using 'Faster' on a TRS-80?—**EdG**)*

Ronald Gerstner, 26 Mount Morton Road, Belgrave South, VIC. 3160 writes:—

"I would like to congratulate you on the vast improvement in the quality of the magazine cassettes over the last two months (November/December 83 and January 84). These are the only two that I have been able to read without errors on my System 80.

I enjoyed Yahtzee but it seemed to be missing something without the added sound available to Model III users only. I made the following modifications to provide sound for Model I and System 80. I changed the count in line 1430 from 28 to 23 and changed the DATA statements as follows:

```
1460 DATA205,127,10,229,193,197,
65,16,254,62,2
1470 DATA211,255,65,16,254,62,1,
211,255,193,16,233,201
```

I found that the added sound makes the game more enjoyable. It would be nice if Model III program listings could show what modifications are necessary for them to run properly on Model I and System 80 as well.

I was very impressed with the Automatic Directory Program in the January 84 edition. It is very rare to find a utility program that is useful under NEWDOS-80 Version 2.0 which I use exclusively. I hope to see more of these. I made a couple of modifications to it which others might find useful.

The author warns that errors are not intercepted and that any errors that may occur are ignored. NEWDOS-80 Version 2.0 provides a very handy error handling routine which when called, analyzes the error return code from any DOS-CALL, displays the appropriate error message and either returns to the calling program or exits to DOS READY at the caller's discretion. I inserted the following code after every appropriate DOS-CALL ('CALL DOSCLL'):

```
JR Z,$ + 7 ;Skip if no error
OR 80H ;Set return flag on
CALL 4409H ;Display error message
```

I elected to return to the program after displaying the error message but you can exit to DOS READY by leaving out the OR 80H and changing $ + 7 to $ + 5.

I also found a hardware incompatibility in the RENAME function with my System 80 as well as some versions of the Model I. After the operator enters the new name on the screen the program adds a CR character (0DH) to the end of the new name in the video memory. The hardware converts 0DH to 4DH which is the ASCII code for the letter 'M'. This causes a BAD FILESPEC error condition and the RENAME is aborted. I modified the code to insert the CR character in the NEWNAM area after the new name is moved there from video memory. To accomplish this, I deleted lines 8250 and 8280 and added the following lines:

```
8251 RYES LD DE,PAT960 + 11
                    ;Start of name location
8252 OR A          ;Clear CARRY flag
8253 SBC HL,DE;Subtract from end
                              ;location
8254 PUSH HL;Save message length
8255 POP BC
8291 LD A,0DH
                 ;CR character in A reg
8292 LD (DE),A
                ;Insert CR after name"
```

*(Thank you very much for your amendments which I am sure will be of interest to many of our readers. Of particular value was your description of what you have done.—**EdG**)*

NOTES FROM THE SOFTWARE EDITOR

by Ed Grigonis

As at the time of writing this we have only two Level I programs on hand. If you have written any programs for Level I, I would be interested in having a look at them.

We also currently have two CoCo programs on hand for future publication. Please consider sending in any programs you may have written for the CoCo as our readers would be most grateful, not to mention the staff of 'Micro 80'. When I get my CoCo I will then have the amount of access to one needed to provide programs by myself but such efforts would probably be largely restricted to conversions of Model I, etc. programs from earlier issues. So how about it? The worst that can happen is that it may not be good enough and you will be out the cost of postage but if it's accepted you gain from the exercise and will be encouraged to try again.

For those people who submitted programs and waited some time for a reply, I have now adopted the system used by the previous Software Editor. When I receive a program I will get a copy onto disk as soon as possible and send your tape or disk back to you. That way you know it has been received.

Any documentation suplied with your program will then go into a file in the order in which it was received. When the documentation gets to the stop of the stack (a good assembly language term there!) I will have a close look at your program.

My first aim will be to try to crash it with some of the more obvious mistakes that may be made. If it survives this I will look at it even closer and decide whether any of our readers would be interested in it. An offer will then be sent out based on my assessment of the program's appeal and style.

I have a request for all of you assembly language buffs out there.

Those of you who read 'Softside' magazine will know that they used to make extensive use of a program called 'SWAT' (Strategic Weapon Against Typos). This was a Basic program which could be appended to any Basic program typed in from 'Softside', and which would then provide a series of checksums for each couple of lines in the program. These checksums would be used to help you find any typing errors and would ensure that what you typed in was exactly the same as in the magazine.

'Softside' have recently dropped 'SWAT' in favour of 'STOMP' (Stop Typos On My Programs). To quote from 'Softside':—

"SWAT has drawbacks and deficiencies. For one thing, to get a matching SWAT table, readers must type in every program line (even REM statements) exactly as it appears in the magazine. Also, SWAT can't detect simple transpositions. The numbers 32767 and 36277 are identical to SWAT. If these numbers are part of a DATA statement for a machine language routine, the computer may hang up, or important data may vanish. Furthermore, because SWAT is written entirely in BASIC, it is quite slow. In sum, SWAT, although a big help, leaves many opportunities for improvement.

STOMP is faster, easier to use, and more reliable than SWAT. In addition, STOMP ignores REM statements and insignificant spaces. If you type BASIC programs from Softside Selections, STOMP will save you many hours.

You may omit any REM statements in our programs. If you do so, be sure to remove the colon (:) immediately preceding the word REM. In addition, feel free to add REM statements (within the constraints of memory) without changing the STOMP tables."

'Rainbow' magazine has a similar program for the CoCo, known as RAINBOW CHECKPLUS.

What's the point of all this? Well, as you may have experienced yourself, it is very easy to make an error when typing in a program from a magazine. It is also very easy to convince yourself that the fault lies in the program (This is sometimes, but not usually correct!). So what we need are programs that will enable us to publish lists of checksums for each of the programs available in 'Micro 80'.

The above quote was reprinted to give you some idea of what is required. We would like programs for the Model I & II and System 80 (they should work in both Level II or any DOS), the CoCo and the VZ200. I look forward to seeing the acronyms you come up with for your efforts!

We could probably easily obtain the rights to use the 'Softside' and 'Rainbow' programs. For that matter, we have the expertise at 'Micro 80' to write such programs ourselves. However, it would be a much more valuable exercise if you, the readers, can come up with something suitable. And remember, you get paid for your efforts!

Would you believe we have only one 'X-LOTTO' program on our files? This will appear next issue. Apparently we have accepted others in the past but, as far as I can determine, the acceptances for these have lapsed. I will therefore be happy to look at anything you have to offer in this regard but it would have to be good to get accepted. If you are one of the people whose acceptances have apparently lapsed, but you would still like to see your program published, then get in touch. I would also be very interested in 'X-LOTTO' programs for the CoCo and VZ200.

We have received two 'Pengo'-type games for the Model I, etc. That's enough for now. If you have written something similar for the CoCo or VZ200 then send it in.

We also have two automobile records programs for the Model I, etc. and would prefer future offers in this category to be for the other computers we support.

Recent software received included some programs compiled with 'ZBASIC'. It would not normally be worthwhile publishing source code for such programs as the reason for them being compiled is usually that they are insufferably slow in Basic and of no use to people without compilers. Ian has suggested, however, that it may be possible to offer such programs in running versions for disk and cassette subscribers. If, therefore, you have written any programs which must be compiled for full effect, then send them in and they will be considered for possible inclusion on the disk and tape editions, with documentation appearing in the magazine. They should work with both disk and tape systems. I would expect that any offerings would be restricted to the Model I, etc. as I am not aware of any compilers available for the CoCo or VZ200.

TDISK — A MACHINE LANGUAGE UTILITY FOR THE TRS-80 I/III

a review by David Eather

Tdisk is one of two machine language programs sold under the name 'System Savers' by Acorn Software. The purpose of Tdisk is to allow a user of a disk system to save and run machine language programs that load into the same area as the DOS.

The program is supplied on tape with three copies on side B (side A has three copies of the other utility FLEXL. I have not found FLEXL a useful program and so it has not been reviewed). The quality of the recording seems to be very high and no trouble was found in loading the program.

The eight pages of documentation give a good description of what the two programs do. Only two pages are devoted to loading and operating Tdisk but the user should find this enough to get the program onto a disk and working.

In the ads for Tdisk a great deal is made of how programs such as LMOFFSET will offset a program (so that it can be saved to disk) but that they are unable to carry out any relocation. Nothing is said to tell the prospective customer that Tdisk also fails to preform any relocation!

Tdisk works by loading the program into high memory and then adding a block move that moves the program down to its original location — anyone who wants to use the program with the DOS still active be warned, this is NOT the program for you.

If however you want to play an Adventure or other game and found that holding down the BREAK key and pressing reset (as required with LMOFFSET) was just a bit tacky this may be what you want.

The program does have a few 'bugs' that don't affect the program operation but are annoying. These are:

1. The program does not clear the screen. It just draws the title block at the top and any prompts are placed at whatever the cursor positions was before you started running the program.

2. The program name must be entered in full. If you are using a lower case driver turn it off before running the program. It makes no conversation to upper case and if the file name isn't exactly what you've typed Tdisk just keeps on looking.

3. When Tdisk has finished loading and then dumping the program it jumps back to DOS Ready without asking if you want to load another file or dump the same file again. This means you must reload Tdisk each time you want to use it. Not much of a problem unless you have just got your disk system and have a lot of programs to dump.

Apart from the very messy display and the other annoying habits of this Program it does seem to work well. It will transfer most programs to disk including Scott Adams Adventures, Big Five games, Edtasm, Tape Scriptset and Tiny Pascal.

It won't however dump a workable copy of Ghost Hunter (Dubois and McNamara Software) although others from this company seem O.K. Srgon II also has problems, Tdisk will dump a working copy but you won't be able to use your printer with it.

Tdisk also won't dump any program that has been copy protected. This is only to be expected with the large number of protection schemes and the small number of protected programs.

At $34.50 the program price is a little steep but it is a useful utility for getting your programs running from the DOS Ready prompt.

SOFTWARE SECTION

LATIN VOCAB TEST L1/4K
by C. Stobert

Please note that this program is for TRS-80 Level 1 and will not run on Level 2 machines or the System-80. If you wish to use it in such machines then you would need the Level 1 in Level 2 program which was included in the Free Software Library Volume 1, sent free to subscribers for Volume 3 of Micro-80. Unfortunately, this program is no longer available from us.

It is appreciated that few people would be interested in a Form 3 Latin vocabulary test. However, the techniques used can enable L.I. to be applied for any similar 'word' comparison test, e.g. other Language tests; chemical elements; atomic numbers.

The L.I.A ( ) array is used to flag 'words' being handled.

This particular test uses 60 'words' for each part of the comparative test. These 120 elements, plus a further five which are used to hold the 'correct' indicator, require 501 bytes of memory.

The program loads in 2978 bytes, but requires 3479 to run.

The data has been set so that one part of the comparison occupies odd number elements A(1) to A(119) and its counterpart the next even element A(2) to A(120). This allows the comparative data lines to be typed in 'side by side'.

Extra data can be entered by deleting the instructions. Assuming the appropriate check on memory is made, the following lines would have to be altered:—

60, 100, 160, 410, 420, 530, 610, 660,

Remember, also, that this will slow the program down further. The current 120 'word' READ cycle is reasonable.

How it works:

10-50	Introduction and Selection. 60 sets flag to 0.
100-150	Selects a 'word' for testing. If the word selected has been used, another has to be chosen. This selection is then flagged. A 'word' flagged cannot be reused during any 20 'word' test. The counterpart is also flagged (A(Q) = 1).
160-200	Selects and flags four further options.
300-510	Presents the word being tested and 5 options, with one to be selected.
530-610	Checks if selection is correct. If initial selection is not correct a second attempt is allowed.
620-670	Adjust counters and returns program for another selection.
700-750	Conclusion sequence.
800-802	Pointer routing for wrong answer.
850-870	Approval routine for correct answer.
905-972	Data Lines.
1000-1090	Instructions.

LANDER (Colour Computer)
by Nick Cooper

This is a simulation of a 'Lunar Landing'. It runs on a TRS-80 Colour Computer with Extended Colour Basic and at least 16K Ram. Your space-craft appears in the top left corner. You have to land on one of three bases.

Right Arrow—Move Right
Left Arrow—Move Left
Up Arrow—Thrust to go up

The program uses the PEEK command to see if any keys are pressed, so to move you can just hold the key down.

At the beginning of the game, you have 1000 litres of fuel. Every time you move, you use up fuel. When you run out, you will not be able to control the space-craft, so you will fall down and crash.

You have a choice of landing on one of three bases. If you choose the first, you get one point, if you choose the second, you get five points and 200 litres of fuel, and the third, you get ten points and 300 litres of fuel.

The program runs at twice the normal speed. This is because of the POKE command in line 90 and 410. If you (BREAK) the program during execution, be sure to POKE 65494,0. If your computer will not take these POKE statements, then delete them in lines 90, 410, 210 and 1330.

Line Description

Lines	
90-210	Introduction
220-320	Instructions
330-400	Choose skill level
420	Choose graphics mode
430	Show score and fuel
440-1120	Setting up graphics
1150-1180	Checks if ship has landed or crashed
1200-1220	Checks if any keys have been pressed down
1300	Explosion
1310-1360	End of game routine

OBSTACLE (L2/4K)
by P. Brierley

Obstacle is a fast-paced arcade game for two players. The object is to control your piece with the appropriate keys so that you do not run into the walls or trails left by the pieces. Game speed is variable within a large range, and up to 50 "hazard points" may be set to increase the game difficulty. Whoever of the two players does not crash will be the one who receives the points for that game. The score is determined by the number of moves a player makes during the game.

The speed of the program is controlled by the loop at 170-260. The faster the game, the lower the number of loops made for keyboard input, and the less time for players to react. When the speed is set at one (1) and both players try to change direction at the same time, only one will succeed. For this reason you may wish to use speed 2 as the fastest speed.

Once an input is made, lines 180-250 determine the new direction of each player, and lines 270-360 set the new position. If a player does not alter his direction, the value of LD or RD (the direction variables) will remain the same, and the ON-GOTO jump will be the same as the previous circuit.

This is the basis of the program, and the rest is obvious. Note line 120, where the players initial positions are set, and the directions randomly chosen. If desired, this line could be changed to allow random positions and directions, player chosen, or preset. (e.g. LD = 1, RD = 3). This is just a matter of routine from 120 to 129 to set LX, LY, RX, RY, LD, RD.

Program analysis

10	Copyright message
20	Title, housekeeping
30	Player name input
40	Speed/hazard input
50-90	Instructions and control Inkey$ loop
100-110	Frame draw
120	Player position, direction set
130-150	Hazard point set
160	Player position set, score increment
170-260	Main Inkey$ loop
270-360	Direction test/change
370-390	Crash test
410-460	Crash display, winner/loser print, score determination
470-480	Crash position flash subroutines
490	Score print

Variable

A	General loop use
B	Main Inkey$ loop
H	No. of hazards
L	Defint
L$	Left player name
LX	Left player X co-ord
LY	Left player Y co-ord
LD	Left player direction
LS	Left player score
R	Defint
R$	Right player name
RX	Right player X co-ord
RY	Right player Y co-ord
RD	Right player direction
RS	Right player score
S	Game speed
T	Current Game score
X	Frame plot
Y	Frame plot
Z	Inkey$ (Defstr'd)

TRACK 80
by Craig MacNish

Track 80 is a racing game for one player. It involves skill and precision driving as well as quick reflexes, good judgement and quick decisions.

The game involves dodging the oncoming cars as you overtake them, as well as passing over bonus checkpoints for extra scores. This is often a dangerous risk and it must be decided whether it is worthwhile.

Instructions on how to play are given when the game is run.

The main essence of Track 80 is a machine language subroutine which moves the track, along with the opposing cars and bonus checkpoints, down the screen. This is necessary as to do this in BASIC takes much too long, would involve no skill or reflexes, and would be too easy.

The machine language subroutine has been coded into line 10 and is POKEd into higher memory by the BASIC program for ease of loading, saving and running of the program.

The program uses 'POKE' and 'PEEK' for the real time graphics as this is much faster than 'SET', 'IF POINT' or 'PRINT@'. This also allows for easy compatibility between the BASIC and machine language parts of the program.

The track swerves randomly and the opposing cars and bonus checkpoints also use a random system. This makes the game different each time and also makes it totally unpredictable.

The moving system of the car is an original BASIC sytem which allows for continuous movement and eliminates the need for a 'straight' button.

So as the game cannot just continue indefinitely, the car accelerates up the screen every time a certain score is reached. This makes the game continually harder and your reactions must be quicker.

TOUCH TYPING
by Spencer George

Touch Typing is a program to help you improve your typing skills.

There are sixteen parts to this program. The parts are graded from using only four keys to using all keys.

Part 1 uses only the	A S D F keys
Part 2 adds	J K L ;
Part 3 adds	G H
Part 4 adds	Q W E R
Part 5 adds	U I O P
Part 6 adds	T Y @
Part 7 adds	Z X C
Part 8 adds	N M , .
Part 9 adds	V B /
Part 10 adds	1 2 3 4
Part 11 adds	7 8 9 0
Part 12 adds	5 6 : —
Part 13 adds	{ } ? +
Part 14 adds	! # $
Part 15 adds	' ( )
Part 16 adds	% & * =

As you run the program it will ask you at which level you wish to try your typing skill. You will then be requested to type in a specific selection of characters. As you type, the characters you are yping will not appear on the screen. When you have completed the task the computer will display what you actually typed and will tell you if there are any errors.

A FUNDAMENTAL SORT UTILITY (48K/DISK)
by B.J.C.

It is often convenient to sort data before further processing indeed, it is frequently mandatory. The obvious course to pursue is to write a sorting routine module and call it as required. It works, and it works well. The trouble is that they take forever to run, and they chomp up RAM like it's going out of style. An attempt was made, therefore, to try to improve things. BUBBLE/BAS is the result. The heart of the program is a short (42 byte) machine language routine employing the ageless bubble-sort. Slow, of course, but it still is able to sort 100 integers before you can get your fingers off the ENTER-key! Together with the ML routine, it is necessary to provide a BASIC driver routine in order to provide data, point it in the right direction, and collect the end result. Now the program provided to demonstrate the interfacing of a BASIC program is designed for operation with a 48K multi-disk system, but there is absolutely no reason why this cannot be made to operate with a 4K system. The method is clearly explained in the May 1980 issue of MICRO-80 in the Assembly Language series by Edwin Paay. Line 90 does, however, require some clarification. DEFUSR0 = &HFEC7 is Disk-BASIC talk for POKE 16526, 199 : POKE 16527, 254. i.e. C7 = lsb and FE = msb. The comments in the listing, together with material from the Level II Reference Manual, should provide an adequate explanation of function. In an applications situation, of course, the BASIC routine would be condensed to three or four lines. You will have observed by now, no doubt, that the major constraint of this routine is the limitation to 255 data INTEGERS Not exceeding 255 in magnitude. So what can be done if an applications program calls for a sort of up to, say, 1000 records? Use a temporary array and pick them off in 250 record lots, using a BASIC logic routine to do so, whittle the value down < = 255 by the same means and after zapping them with the ML routine, put them back where they belong by the reverse route. It sounds messy, but it's really quite simple when you get down to it. But that's another story. Perhaps another time . . .

Implementation:

The following assumes the use of a 48K RAM/Disk Basic combination. Variation to suit other configurations should present no real difficulty.

1. TRSDOS command mode, DEBUG utility.
2. Enter machine language routine, using DEBUG command M, commencing at FEC6H, ending at FEEFH, entry FEC6H.
3. TRSDOS command mode, DUMP utility.
4. Executive the following: DUMPƀ BUBBLE/CMDƀ(START = X'FEC6', END = X'FEEF', TRA = 'FEC6')
5. Call Disk BASIC.
6. Protect Memory ? = 65221
7. Enter the essentials of the BASIC program at the point(s) relevant to your application. (Or as a subroutine to be called as required).

The foregoing will make the routine load up automatically each time TRSDOS is initialized (on that particular disk!). It represents only one of several options, of course. It may suit your application to contain it within the BASIC program. You know, VARPTR etc.

Machine Code:

FD 21 EF FE 06 00 OE 00 FD 7E 00 FD BE 01 DA E5 FE CA E5 FE FD 4E 01 FD 77 01 FD 71 00 OE 01 FD 23 10 E5 CB 41 C2 C6 FE C9 OO

And, if you back-up on tape, CMD"T" eh?

DOG RACE VZED
by Ron Carson

This program was published in Micro-80 some time ago for the TRS-80 and System-80. Now it has been modified to run in your VZ200.

I have only written the bare bones program. Although it runs well and is useable as is, it gives you the chance to expand the program to suit your needs.

After loading the program you are asked to do two things:
1. Press any key to continue.
2. Press SPACE TO START RACE

After the race is over the winning dog is printed in the text mode, and you are asked if you want to race again or end.

You will see there are plenty of options for you to look into to make this a really great game and a lot of fun.

CONTEST LOG VZED
by Ron Carson

This program should be of advantage to any radio amateur or shortwave listener who owns a VZ200.

As the title suggests the program is ideal for RD contests or any other type of log from which you wish to get a hard copy of call signs worked. To operate, it requires a printer to be connected to the computer.

The menu gives you 5 options:
LIST—List of all entries
SORT—Sort into alphanumeric order
PRINT—Printout to printer
END—End Program
—Enter Callsign

If you go into the sort mode all entries are placed in alphanumeric order, then you will be asked if you require a printout to printer
Printout to VDU
return to menu (cont)

After each entry you will be told if the last callsign entered is a new one or entered before. If already entered it will not be retained in data.

Do not enter END until you have your hard copy, as END or break will destroy all of your entries.

MEMORY PEEK VZED
by Ron Carson

If you are interested in finding out what your VZ200 stores in its memory enter this program and have a look.

The program will display on the screen the information you need to know to run it and asks for a start address in decimal.

After going to the start location it will print the DECIMAL address, Z80 address, CHR at that address and ASCII code.

The program runs very quickly so to slow it down press the SPACE key. Pressing the SPACE key slows down the program and also prints the HEX address of each location on the screen.

If you want to change the memory location while the program is running press the (:) colon key and you will be asked for a new start address.

FIELD FINDER
by Ken B. Smith

Our longer serving readers will remember two earlier articles by Ken B. Smith, resident in the Sultanate of Oman. We lost contact with Ken for some time but here he is again as witty and instructive as ever. This article describes a program which will be absolutely invaluable to the great majority of our readers who are frequently called upon to pilot aircraft around the more remote parts of Oman. Those few who live more pedestrian lives should nevertheless find the techniques used of value in constructing their own programs where significant amounts of data need to be manipulated and reports printed.

Well here we are again, face to face with the dreaded flashing cursor of the TRS word processor (SuperScripsit to be precise — rather good in parts, but that's hardly a review — perhaps another day). The purpose of this missive is to explain the utility FIELD FINDER. If you don't have a TRS-80 or other MICROSOFT BASIC equipped computer, a printer capable of 132 characters per line or an interest in aviation in particular or travel in general, turn the page. Unless of course you are interested in a rather neat little utility in its own right for the sake of improving your BASIC.

What is FIELD FINDER?

I presently fly and instruct on the little known aviation joke called the SHORT's SKYVAN (SC7) light twin engined transport aircraft for the Sultan of Oman's Air Force. We use this tin shoe box as the mainstay of our internal communications between otherwise unreachable villages within the Omani interior and coastline. I know from previous experience of the Australian 'bush' from my RAF C130 days, that you have a similar problem to ourselves. Viz.: some quite good fields and navigation aids surrounded by an awful lot of grotty little strips. Planning a round robin or medivac can be a nightmare of charts and rulers. Inflight diversions, even if you are very familiar with the area, can be a menace. With this type of operation in mind I developed and wrote FIELD FINDER. In essence it produces, from internal program data, an information page containing pertinent data on the strips and several, depending upon the number of strips, pages of 'anywhere' to 'anywhere' tracks and distances. This may all sound, to the non aviation minded, like a total waste of time. Believe me, it has saved my bacon (sorry no bacon out here, lamb chops?, but I digress as usual) rather more times than I would care to admit over the past three years. It is also rather nice to have the machine do something really useful and constructive once in a whole, so much nicer than Blob Chasing.

The Program

*With any luck the dear staff at Micro-80 have been so kind as to publish the program listing ((we wouldn't be game to change it, Ken—**Ed**) exactly as it was sent off by me, without modifications or additions). If so it works and is as bug free as three years of development can get it. With this assumption in mind I will discuss it by line number where appropriate and any comments in the program can be ignored and left out of your copy. The one exception is the line 600. I realise only too well that it is bad practise to have a REMark in a referenced line but you need to add some code of your own there, depending on the type of Line Printer you use. There are no regrets at the state of the line numbers, this is a mature program and has been extensively modified over the years, not only to remove problems (and insert new ones) but also to cater for changing needs and outlooks. It has also undergone considerable surgery to produce this hardware independent version for release. My personal feeling is that anyone who uses a Renumber routine needs a poke with a sharp stick as it destroys what little structure BASIC has in the first place. Those who are interested can literally read between the lines and get an insight into the program's history. (If you want to, that is.)*

On with the program

Line 10 — is just a jump to the main body as this program doesn't so much RUN as hobble. The TRS is not much of a number cruncher at the best of times and the average FIELD FINDER run calls the LAT/LONG routines an awful lot. So to keep the speed up they had to go at the top.

Lines 20 to 25 — is one of my pat routines to do a great circle track and distance from two LATitudes and LONGitudes. The entry variables are: ES & EF = Easting Start and Easting Finish: NS & NF = Northings Start and Northings Finish : VA = Variation. Output is CB & CD = Course Bearing and Course Distance from start to finish respectively. I have just realised that for the Australian continent the priority will be for Southern hemisphere operation. No real problem if you remember that this routine takes +ve numbers as North and East and −ve numbers as South and West. So you just need to prefix a — to your Latitude figures in the DATA and change the N in the PRINT USING formats in line 540 and 1120 to a S. Isn't life tough?

Line 30 — is really part of the LAT/LONG routine as this needs its figures in minutes rather than degrees and minutes. It resides directly below the calling routine as BASIC starts looking here and finds it faster. Originally this was an FuNction call, but in the interests of portability it is now a subroutine. Interestingly it seems to be a little faster in this form.

Line 100 — contains the setup parameters for string space, the variable declarations and some numeric definitions. CC and CF are the pat conversion figures for the RADIANS/DEGREES problem.

Line 105 — This puts the commonly and frequently used variables onto the top of the Variable Table. I have used the DIM method as it uses less space than the a = O type of allocation you see so often. I have seen a program advertised that claims a 50% improvement on program speed and all it does is build a line similar to this for you to add to your program. Save your money. The speed achievement is real enough, particularly in long programs with many variables, but you can achieve equally satisfactory results with commonsense.

Line 112 to 117 — these variables define the presentation of the output and before we continue let's set up a couple of definitions for the rest of this discussion. (DATA PAGE = The first page of output listing all the strips and the relevant data about them. GRID PAGE = is one of the 'anywhere' to 'anywhere' track and distances pages.) Essentially the problem is to form a square grid of places on a GRID page so that the data is readable and sensible. Although the program, as presented, is configured for 54 places it can be any multiple of 12, 13, 14, 15, 16, 17 or 18 places. Taking the square of the number of places divided by grid size gives the number of GRID pages. Add one for the DATA page and you have the total page count. For example: To get 54 places on, given a maximum grid (on A4 paper, 132 CPL, that is), the only fit is 54/18 = 3. This equals 3*3 = 9 pages of GRID and 1 page of DATA. Another example. If you had 36 places, then you could either use 4 sheets of GRID using 18 places per page (36/18 = 2 : 2*2 = 4) or 9 sheets using 12 places per GRID page (36/12 = 3 : 3*3 = 9). Obviously for ease of use you would opt for the lower page count. So set ZF for the number of places, PP to the grid size (places per page), NP to the number of GRID pages + 1 for the DATA page and unless you have some very long place names, leave Z9 and Z8 alone. Z8 and Z9 are really quite simple to change, but don't unless you know what's going on or the presentation will be spoilt. You have been warned.

Line 120 — DIMensions the array variables. SN( ) = Place Name : EL( ) = Elevation : SR( ) = Runway Direction : LE( ) = Runway Length : SU( ) = Runway Surface : SC( ) = Radio Comms Details : LA( ) = Latitude : LO( ) = Longitude : SG( ) = Grid Reference : CO$( ) = Comments : TI( ) = Titles for DATA Page : HE$( ) = Headers for GRID Pages : CH( ) = Holding array for distances. Most of the variable names conform to some logic, but some were added later and don't conform to anything except my mood at the time of modification!!!

Line 130 — RESTOREs the DATA pointer (and I realise that the initial RUN should have done so, but the statement at least identifies the beginning of a long READ operation), reading in the nine lines of the Header for the GRID pages. As you will see from the REMark on line 134, you have Z9 characters per line of Header. You do not have to have all the lines, but in that case put in a blank so that subsequent DATA reads will be correct.

Line 135 — Contains the DATA for the Header.

Line 140 — I was rather pleased, in a simple minded way, with this routine. I have been continually changing the headings on the DATA page over the years and it was getting to be a pain changing the TAB settings for each of the columns. This line reads in the Titles TI( ) and then the respective widths in ZX, accumulating into ZI( ) for each from lines 195 and 196 respectively. This may be explained rather more if you look at the two REMed lines 198 and 199 where I have shown the set up. If you follow this format and don't try to get more than 132 characters on a line, I am sure you will find it a flexible method.

Lines 200 to 415 — is all the main DATA on the strips. It must conform with the format in Line 140 and with the array details in Line 500. If not you get the confusing errors associated with DATA reading into the wrong variable type arrays. You may not want to put in all my DATA, as you probably won't need too much detail on the strips of southern Oman if you fly in Queensland!! However this DATA is the basis of a rather tricky little game called SKYTRUCK which will when I get around to finishing the damn thing, be offered up as a sacrifice to the editors of Micro 80 for your amusement.

Line 500 — reads in the data to the appropriate arrays. The final part of the line adds spaces to names that are less than Z8 characters long so that the vertical printing routine does not fall over!! Finally the current DATA item is printed in the top left hand corner of the screen so that you have a fighting chance of finding any errors in your own DATA format.

Line 502 — is something to be ashamed of. However in the search for hardware independence it was easier to KISS than to use some fancy code. (KISS = Keep It Simple, Stupid).

Line 510 — contains the initialisation code (clear buffers etc.) for an EPSON FX-80. Insert your printer start up code in here and of course, leave the EPSON's out!!!

Line 520 — Print a large title somewhere near the middle of the DATA page. Do the required for your own printer and of course use your own heading.

Lines 526 to 527 — print out two lines of joint disclaimer and an appeal for corrections. There is absolutely no way that you can get this sort of DATA right first 'go and if you have a responsive set of addressees you will get corrections and suggestions. However they will need to know the address to send them.

Line 530 — This is tied up with line 600 in that you must at both these points force your printer to 132 characters per line. If you have an EPSON or a functionally similar machine, leave these two lines alone and ignore line 600 altogether. As the comments in line 600 suggest, registration (keeping things lined up) can be a problem with this much printing in columns. So if you have a problem try and force the printer to single pass printing (non logic seeking), which should improve things dramatically.

Line 352—The complex LPRINT statement starting and ending the line is the underline sequence for an EPSON. Once again, if you have one, leave it alone. Or insert your own underline code. The FOR NEXT loop places the Titles in TI( ) at TAB position ZI( ). It is to facilitate this and the following routines that the rather involved code in line 140 is used.

Lines 535 to 540 — Looping through the number of strips, this prints the data in the arrays at the correct TAB position. Remember to change the PRINT USING format to correctly show your LAT/LONG.

Line 560 — completes the loops and STOPs. Depending on the number of lines used and the status of the printer and computer line counters, you may or may not be correctly positioned for the GRID pages and it makes sense to pause so that the paper can be realigned. Also there will be many more reprints of the DATA page than the GRIDs so this STOP saves paper, particularly if a Spooler is active.

Lines 900 to 1140 — In order to leave a little magic and mystery in your lives I will not go into any detail about these lines. Suffice to say they work and control the format of the GRID pages. Provided you have set up the variables correctly, you should be pleased with the results. Watch out for the embedded keywords if you miss out the spaces in line 1110. The correct spacing is

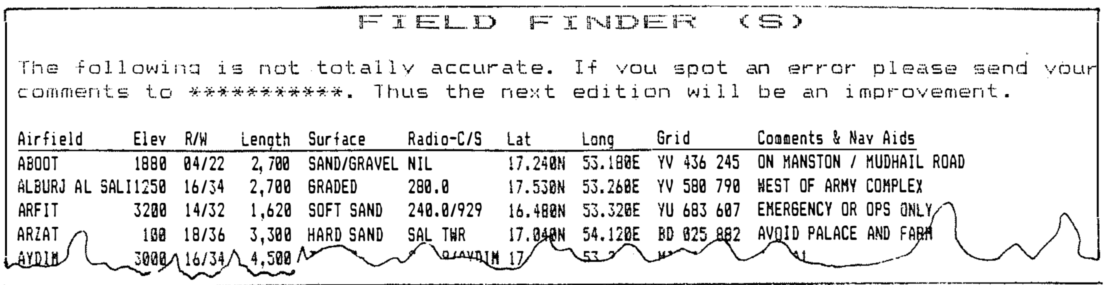
1110 XT = Z9:FORY = PS TOPF. . etc.

If the space was not there the interpreter would read a STOP. An unfortunate choice of keywords and the problem only arises in this lexicon of Microsoft where spaces are not required. It is a valid point that spaces should be left anyway, but it does speed things up to leave them out!!!

After each page has been printed there is a CHR$(12) and a pause. If you find that your printer is pageing correctly then you can omit the pause and carry straight on. It depends very much on the hardware you happen to be using. One useful tip — you may find that a better presentation is achieved by forcing a smaller line spacing on the GRID pages. I certainly use this on my setup. However, once again it depends on your kit.

Well there it is, not a particularly fancy program, but it does produce an awful lot of information from a relatively small amount of DATA. I hope that it proves as much of a time saver for your operation as it has for ours. Remember FIELD FINDER is no substitute for a good map and do check those latitudes and longitudes carefully. The real danger of a routine like this is that people believe a printed page and the output is only as good as the initial data. G.I.G.O. (Garbage In = Garbage Out). If you are tempted to avoid typing in the program, remember that to cross reference 30 strips takes 900 measurements of track and distance and worse still, 900 blocks of numbers to copy out.

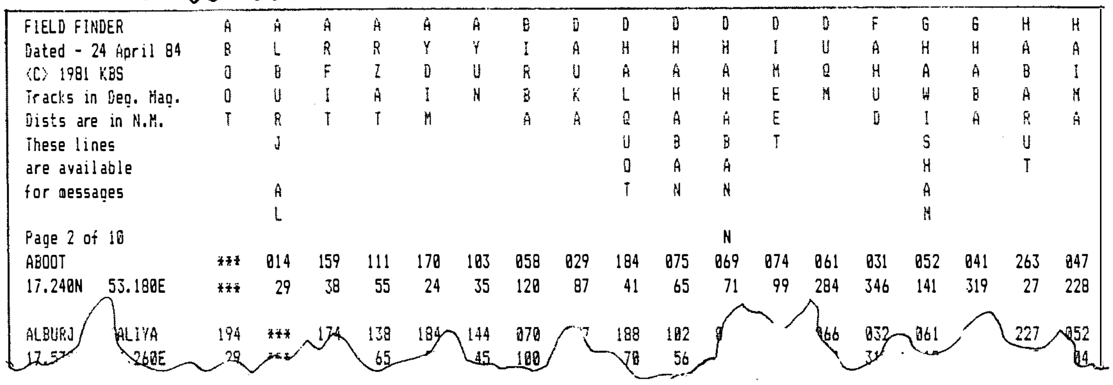
SAMPLE REPORTS



```
                              FIELD FINDER (S)

The following is not totally accurate. If you spot an error please send your
comments to **********. Thus the next edition will be an improvement.

Airfield        Elev R/W   Length Surface      Radio-C/S  Lat     Long    Grid        Comments & Nav Aids
ABOOT           1880 04/22  2,700 SAND/GRAVEL  NIL        17.240N 53.180E YV 436 245  ON MANSTON / MUDHAIL ROAD
ALBURJ AL SALI1250 16/34    2,700 GRADED       280.0      17.530N 53.260E YV 580 790  WEST OF ARMY COMPLEX
ARFIT           3200 14/32  1,620 SOFT SAND    240.0/929  16.480N 53.320E YU 683 607  EMERGENCY OR OPS ONLY
ARZAT            100 18/36  3,300 HARD SAND    SAL TWR    17.040N 54.120E BD 025 882  AVOID PALACE AND FARM
AYDIM           3000 16/34  4,500
```



```
FIELD FINDER         A   A   A   A   A   A   B   D   D   D   D   D   D   F   G   G   H   H
Dated - 24 April 84  B   L   R   R   Y   Y   I   A   H   H   H   I   U   A   H   H   A   A
<C> 1981 KBS         O   B   F   Z   D   U   R   U   A   A   A   M   Q   H   A   A   B   I
Tracks in Deg. Mag.  O   U   I   A   I   N   B   K   L   H   H   E   M   U   W   B   A   M
Dists are in N.M.    T   R   T   T   M       A   A   Q   A   A   E       D   I   A   R   A
These lines              J                           U   B   B   T           S       U
are available                                        O   A   A               H       T
for messages             A                           T   N   N               A
                         L                                                   M
Page 2 of 10                                                 N
ABOOT               *** 014 159 111 170 103 058 029 184 075 069 074 061 031 052 041 263 047
17.240N   53.180E   ***  29  38  55  24  35 120  87  41  65  71  99 284 346 141 319  27 228

ALBURJ    ALIYA     194 *** 174 138 184 144 070     188 102         066 032 061     227 052
          260E       29 ***      65      45 100      78  56                              04
```

**** LII/16K   FIELDS/BAS **** TRS-80/SYSTEM-80

```
1 '**** FIELD FINDER (southern edition) *****
2 '**** Configured for 54 strips in 18 * 18 pages *****
3 '**** giving a total of 10 pages including data header *****
4 '**** <C> 1981  - conceived and written by *****
5 '**** Ken B Smith FIAP *****
6 '**** All routines are now effectively public domain *****
7 '**** use them at your own risk !!!!. *****
8 '**** Reconfiguration is relatively straightforward *****
9 '**** Hardware independent except for a 132 CPL printer *****
10 GOTO100 'Bump over the critical subroutines and get into the
main routines
19 ' *** These are the math routines to convert LAT & LONG to be
aring and distance. (great circle)
20 CY=EF:GOSUB30:EF=CZ:CY=ES:GOSUB30:ES=CZ
21 CY=NF:GOSUB30:NF=CZ:CY=NS:GOSUB30:NS=CZ:NX=NF-NS
22 CL=((NF+NS)/2)*CC:C=COS(CL):EX=(EF-ES)*C:IFEX=0THENEX=.000001
23 CA=ATN(NX/EX)*CF:IFEX>0THENCB=90-CAELSECB=270-CA
24 CA=.5*EX*SIN(CL):IFCB=>180THENB=B-CAELSEB=B+CA
25 CD=INT(SQR(NX[2+EX[2)):CB=INT(CB)+VA:RETURN
30 CZ=FIX(CY)*60+(CY-FIX(CY))*100:RETURN 'SUB 20 needs minutes r
ather than Degrees & Minutes
99 'Various constants and variable setups. CC & CF are conversio
ns from RADIANS/DEGREES. VA is Variation
100 CLEAR2000:CLS:DEFINTX-Z:DEFSTRS-U:DEFDBLA:T1="####":T3="###.
##":T4="###":S=CHR$(30):CC=2.90888E-04:CF=57.2958:VA=0
105 DIMX,Y,CB,CA,CD,CF,CY,CZ,EF,ES,EX,NF,NS,Z9,Z8,ZF
112 Z9=19 'Inset into page for headers. See Page Header DATA for
 details
113 ZF=54 'Total Number of entries and 54 is about the limit for
 the page one data header. Although if you have LOTS of paper yo
u could go as high as you like
114 PP=18 'No of entries per page (18 is maximum. 12 is minimum)
115 NP=10 'Number of pages including data header
116 'To work out PP and NP try (ZF/PP)[2 !!!!
117 Z8=17 'The length of the longest name. Must not exceed Z9-2
120 DIMSN(ZF),EL(ZF),SR(ZF),LE(ZF),SU(ZF),SC(ZF),LA(ZF),LO(ZF),S
G(ZF),CO$(ZF),TI(10),CH(ZF)
130 RESTORE:FORX=1TO9:READHE$(X):NEXT
134 'You have 9 lines of Z9 characters. Line 10 is for the page
count
135 DATAFIELD FINDER,Dated - 24 April 84,<C> 1981 KBS,Tracks in
Deg. Mag.,Dists are in N.M.,These lines,are available,for messag
es,
140 FORX=0TO9:READTI(X):NEXT:ZB=0:FORX=0TO9:READZX:ZI(X)=ZB:ZB=Z
B+ZX:NEXT:GOTO500
195 DATAAirfield,Elev,R/W,Length,Surface,Radio-C/S,Lat,Long,Grid
,Comments & Nav Aids
```

```
196 DATA14,6,7,8,12,12,9,9,12,40
198 'DATA 17 ,   4,  5  ,  6  ,    10  , 10  ,  7 , 7  , 10 , 40
199 'DATA NAME,ELEV,RWAY,LENGTH,SURFACE,RADIO,LAT,LONG,GRID,COMM
ENTS
200 DATAABOOT,1880,04/22,2700,SAND/GRAVEL,NIL,17.240,53.180,YV 4
36 245,ON MANSTON / MUDHAIL ROAD
205 DATAALBURJ AL SALIYA,1250,16/34,2700,GRADED,280.0,17.530,53.
260,YV 580 790,WEST OF ARMY COMPLEX
210 DATAARFIT,3200,14/32,1620,SOFT SAND,240.0/929,16.480,53.320,
YU 683 607,EMERGENCY OR OPS ONLY
215 DATAARZAT,100,18/36,3300,HARD SAND,SAL TWR,17.040,54.120,BD
025 882,AVOID PALACE AND FARM
220 DATAAYDIM,3000,16/34,4500,TARMAC,257.8/AYDIM,17.000,53.220,M
ISSING,JET A1
225 DATAAYUN,2600,02/20,1200,SOFT SAND,NIL,17.160,53.540,MISSING
,ALSO 12/30 1200
227 DATABIRBA,0,**/**,0,.....,NIL,18.260,55.060,MISSING,DATA INC
OMPLETE
230 DATADAUKA,700,13/31,6600,SAND,NIL,18.400,54.040,MISSING,MONT
ASAR / FASAD ROAD JUNCTION
235 DATADHALQUOT,150,08/26,1050,EARTH,NIL,16.430,53.150,YU 408 4
88,BAD APPROACHES BOTH ENDS
240 DATADHAHABAN,1420,13/31,5248,SAND,118.8,17.405,54.244,MISSIN
G,PDO - 381/ON
242 DATADHAHABAN N,1420,**/**,6000,SAND,NIL,17.485,54.282,MISSIN
G,DATA UNRELIABLE
245 DATADIMEET,1100,18/36,1200,SAND,NIL,17.510,54.590,MISSING,RH
S & TOWER TO WEST ; 07/25 - 780
250 DATADUQM,200,01/19,1650,SAND,NIL,19.400,57.410,MISSING,ROAD
TO EAST AND CAMP TO WEST
251 DATAFAHUD,550,13/31,5900,SAND,118.8,22.210,56.286,MISSING,11
2.5/FHD ; 360/OF
252 DATAGHAWISHAM,722,17/35,6200,SAND,NIL,18.504,55.160,MISSING,
DATA INCOMPLETE
254 DATAGHABA,443,**/**,6400,SAND,118.8,21.224,57.030,MISSING,11
6.5/GBA ; 326/OL
255 DATAHABARUT,1800,07/25,1590,SAND,240.0/Z64,17.210,52.490,XV
915 180,HF STUD 2 ; PDRY 5 KS WEST
257 DATAHAIMA,400,**/**,6000,SAND,118.8,19.581,56.164,MISSING,11
3.3/HAI ; 288/HMA
260 DATAHASIK,0,18/36,990,FIRM SAND,NIL,17.260,55.160,CE 165 295
,FLOODS AFTER RAIN ; BEWARE BIRDS
265 DATAHEIRUN,2900,03/21,1800,HARD SAND,280.0?,17.030,53.210,MI
SSING,RANGE TO EAST
267 DATAIZKI,1700,01/19,5630,SAND,MUSCAT,22.533,57.453,MISSING,1
13.5/IZK ; 333/NIZ
270 DATAJARF,800,17/35,6000,GRAVEL,118.8/PDO,18.130,55.400,CF 58
0 132,C/S JARF NORTH
275 DATAJIBJAT,3000,18/36,1650,FIRM SAND,NIL,17.150,54.290,BE 34
0 086,FIRQA DWELLINGS TO EAST
280 DATAJUFFA,550,09/27,1800,STONES,NIL,17.070,55.030,BD 921 945
,DEEP WADIS BOTH ENDS ; STONE DAMAGE !
285 DATAKAHIL,70,18/36,4440,GRADED,NIL,18.340,56.330,DF 543 535,
PDO STRIP TO EAST OF MIL.
290 DATAKURIA MURIA,0,03/21,1200,SEALED,NIL,17.300,55.550,CE 898
 352,STRONG WIND FROM 210 IN MONSOON
295 DATALIQBI,50,18/36,1200,EARTH,NIL,18.140,56.320,DF 510 150,N
EW STRIP ; VILLAGERS UNRULY !!
300 DATAMADRAKA,0,06/24,1500,SAND,NIL,18.590,57.490,MISSING,STRI
P WEST OF VILLAGE AND NEAR COAST.
305 DATAMAKINAT SHIHAN,1650,04/22,1800,SAND,240.0/Z54,17.480,52.
300,XV 555 634,12/30 ALSO ; VEHICLES CROSS ON TRACKS
310 DATAMARBAT,50,07/25,2310,SAND,NIL,16.590,54.420,BD 548 798,U
SE EASTERN HALF OF R/W DUE TO ROCKS
312 DATAMARMUL,890,**/**,6000,SAND,118.8,18.083,55.103,MISSING,1
14.3/MRL ; 890/OA
315 DATAMARSAUDID,500,18/36,1380,SAND,240.0,19.240,54.270,MISSIN
G,HF STUD 2 ; DUNES TO SOUTH
317 DATAMASIRAH,62,17/35,10200,TARMAC,MAS,20.400,58.540,MISSING,
115.5/MRH(T) ; 343/MR
320 DATAMEDINAT AL HAQ,2310,09/27,1410,EARTH,NIL,17.110,54.240,B
E 215 004,03/21 ; 09 & 21 SLOPE DOWN
325 DATAMEDINAT AL SAAN,1990,09/27,1800,EARTH,NIL,17.110,54.160,
MISSING,VILLAGE TO NORTH ; 04/22 ALSO
330 DATAMUDHAIL,1700,17/35,2700,HARD SAND,NIL,17.280,53.210,YV 5
00 335,TV MAST TO E. ; USE E. SIDE OF STRIP
335 DATARAKHUT,0,10/28,1410,EARTH,NIL,17.000,53.340,YU 580 530,2
8 WHEN WIND FROM N.;DATA INCOMPL.
340 DATARAMLAT MITAN,1100,13/31,1800,SAND,NIL,18.360,52.270,MISS
ING,USE 1 to 100000 MAP
345 DATARAMLAT SHUWAIT,710,05/23,2400,SAND,NIL,18.543,52.058,MIS
SING,1 to 100000 MAP SHOWS DETAIL
350 DATARAYSUT,30,18/36,3300,SAND,SAL TWR,16.560,54.000,ZU 188 7
66,BEWARE HELI
352 DATASALALAH,73,07/25,8967,TARMAC,119.1/SAL,17.020,54.040,MIS
SING,112.3/SLL ; 310/SAN
355 DATASARFAIT,4100,05/23,1300,FLINTS,240.0/9C,16.420,53.050,YU
 240 478,SEE SOP 9 ; 280.0
357 DATASEEB,43,08/26,11762,TARMAC,SEEB,23.356,58.168,MISSING,11
4.5/SBB ; 318 NSB
360 DATASHALEEM,900,07/25,1350,SAND,118.8,18.080,55.390,CF 575 0
60,CALL JARF FOR PDO TRAFFIC
365 DATASHARBITHAT,180,01/19,1800,SAND,NIL,17.580,56.160,DE 225
900,PARKING TO EAST
370 DATASHWAYHIYAH,120,01/19,1680,SAND,NIL,17.540,55.370,CE 525
790,PARKING TO WEST
375 DATASUGORAH,50,17/35,1500,EARTH,NIL,18.080,56.230,DF 525 015
,ROTOR DURING MONSOON
380 DATATAWI ATTAIR,1990,10/28,1320,DIRT/GRASS,240.0,17.050,54.3
30,BD 403 921,LAND 10 ; T/O 28
382 DATATHUMRAIT,1500,17/35,13123,TARMAC,THUM,17.401,54.013,MISS
ING,BEWARE FIGHTERS
385 DATATUDHO,1650,18/36,1800,SAND,NIL,17.410,53.050,YV 722 958,
VERY HARD TO SEE
390 DATA WADI KINITH,2500,14/32,1800,SAND,240.0,17.060,52.496,MI
SSING,DEFENDER EMERG ONLY
400 DATAWADI MITAN,1140,11/29,1650,SAND,NIL,18.210,52.170,MISSIN
G,1 to 100000 MAP SHOWS DETAIL ; WHITE STONES
410 DATAWATIF,1000,17/35,1650,SAND,NIL,18.260,56.270,DF 448 380,
UNRELIABLE
415 DATAZAWAL,0,17/35,1500,SAND,NIL,19.000,57.300,MISSING,TO NW
OF VAGUE RED AREA ;TRACK AT NORTH END
500 FORX=1TOZF:READSN(X),EL(X),SR(X),LE(X),SU(X),SC(X),LA(X),LO(
X),SG(X),CO$(X):SN(X)=SN(X)+STRING$(28-LEN(SN(X)),32):PRINT@0,SN
```

```
(X):NEXT
502 PRINT"PAGE ONE OR THE REST : <1> OR <R> - ";:INPUTA$
504 IFA$="R"THEN600
509 '**** Printout page one only
510 LPRINTCHR$(27);"@";' *** This is EPSON FX 80 initialisation
520 LPRINTCHR$(14);"                FIELD FINDER (S)":LPRINT 'CHR$(1
4) = LARGE TEXT (40 CPL)
525 '**** Always need a disclaimer on the data !!!!!
526 LPRINT"The following is not totally accurate. If you spot an
 error please send your"
527 LPRINT"comments to ***********. Thus the next edition will b
e an improvement."
530 LPRINTCHR$(27);CHR$(15);'set condensed mode
532 LPRINTCHR$(27)"-"CHR$(49):FORX=0TO9:LPRINTTAB(ZI(X))TI(X);:N
EXT:LPRINTCHR$(27);"-";CHR$(48) 'The CHR$(27);"-";CHR$() sequenc
e is the UNDERLINE ON/OFF for EPSON
535 FORX=1TOZF:LPRINTTAB(ZI(0))LEFT$(SN(X),14);:LPRINTTAB(ZI(1))
USING"####";EL(X);:LPRINTTAB(ZI(2))SR(X);:LPRINTTAB(ZI(3))USING"
##.###";LE(X);:LPRINTTAB(ZI(4))SU(X);
540 LPRINTTAB(ZI(5))SC(X);:LPRINTTAB(ZI(6))USING"##.###N";LA(X);
:LPRINTTAB(ZI(7))USING"##.###E";LO(X);:LPRINTTAB(ZI(8))SG(X);:LP
RINTTAB(ZI(9))CU$(X)
560 NEXT:LPRINTCHR$(12):STOP 'Page out and STOP. Due to very hig
h paper useage the CHR$(12) may dump you past TOF so halts are t
he in thing.
561 'Anyhow it is likely that more runs on the page one data wil
l be required than on the T/D sheets and it saves a BREAK, parti
cularly if a spooler is in.
600 '**** Insert any particular printer initialisation codes for
 your hardware here. You will need to go to 132 CPL and if you h
ave registration problems, single pass printing (or buy an EPSON
)
700 LPRINTCHR$(27);CHR$(15)'132 CPL MODE
900 PG=2:FORMS=1TOZFSTEPPP:MF=MS+PP-1:FORPS=1TOZFSTEPPP:PF=PS+PP
-1 '*** Tricky bit to set up pages
910 HE$(10)="Page"+STR$(PG)+" of"+STR$(NP)
990 FORX=1TO10:XT=Z9:LPRINTHE$(X);'Print first 10 characters of
name vertically
1000 FORY=PS TO PF:XT=XT+6:LPRINTTAB(XT-1)MID$(SN(Y),X,1);:NEXTY
:LPRINT:NEXTX
1100 FORX=MSTOMF:LPRINTSN(X); 'The name of the strip or airfield
1110 XT=Z9:FORY=PS TOPF:NS=LA(X):ES=LO(X):XT=XT+6:NF=LA(Y):EF=LO
(Y):GOSUB20:LPRINTTAB(XT-3)"";
1115 CH(Y)=CD:IFCD=0THENLPRINT" *** ";:GOTO1120
1116 IFCB<=0THENLPRINT" 000";:GOTO1120
1117 IFCB=>100THENLPRINTUSINGT1;CB;:GOTO1120
1118 IFCB=>10THENLPRINT" 0";:LPRINTUSING"##";CB;ELSELPRINT" 00";
:LPRINTUSING"#";CB;
1120 NEXTY:LPRINT:LPRINTUSING"##.###N   ##.###E";LA(X);LO(X);:XT
=Z9:FORY=PS TOPF:XT=XT+6:LPRINTTAB(XT-3)"";:IFCH(Y)=0THENLPRINT"
 *** ";ELSELPRINTUSINGT1;CH(Y);
1130 NEXTY:LPRINT:LPRINT
1140 NEXTX
1145 CLS:PRINT:PRINT"RESET PAPER FOR NEXT PAGE AND PRESS <ENTER>
":INPUTA$ '*** Grotty patch for hardware independance
1150 PG=PG+1:NEXTPS:NEXTMS:END
1200 '*** All the bells and whistles have been removed for hardw
are independance.
1201 '*** Hopefully a full explaination of the program will appe
ar in PCN shortly.
1202 'If you really want to modify it further and can't get into
 the code enough, drop me a line.
1203 'Ken B Smith,  Officers Mess, SOAF SALALAH, PO Box 897, MUS
CAT, OMAN
1204 ' or 12, Larch Way, HAXBY, YORK. Tel 760351
1300 '**** The End ****
1301 '**** MICROSOFT BASIC 5.1 lexicon (No spaces required)
1302 'Program size including REMarks 9965 bytes
1303 'Simple variables 77 bytes
1304 'Array variables 2290 bytes
1305 'Reserved string space 2000 bytes. Used string space 1082 b
ytes
```

**** Lunar Lander ****

COLOUR COMPUTER

```
10 '   *****************
20 '   *               *
30 '   *  NICK COOPER  *
40 '   * 80 SWAINE AVE. *
50 '   * TOORAK GARDENS *
60 '   *    S.A. 5065   *
70 '   *               *
80 '   *****************
90 LINE-(241,96),PSET:LINE-(241,
106),PSET:LINE-(239,109),PSET
100 LINE(116,107)-(116,92),PSET:
LINE(127,92)-(127,107),PSET
110 LINE(84,92)-(84,107),PSET:LI
NE-(95,107),PSET
120 DRAW"BM100,107U11E4R3F4D4L11
R11D7"
130 DRAW"BM132,92D15R7E4U7H4L7"
140 DRAW"BM159,92L11D8R7L7D7R11"

150 DRAW"BM164,107U7R2L2U8R7F3D1
G3L5F8"
160 FORY=20TO78STEP2:X=126
170 COLOR2,1:DRAW"BM"+STR$(X)+",
"+STR$(Y)+"D7G4E4F4":COLOR1,1:DR
AW"BM"+STR$(X)+","+STR$(Y)+"D7G4
E4F4":NEXTY
180 COLOR2:DRAW"BM"+STR$(X)+","+
STR$(Y)+"D7G4E4F4"
190 FORT=0TO500:NEXTT
200 COLOR1:DRAW"BM"+STR$(X)+","+
STR$(Y)+"D7G4E4F4"
210 POKE65494,0:SCREEN1,1:I=1:GO
SUB1380:SCREEN1,0:FORT=0TO500:NE
XTT
220 CLS:PRINT@9,"*** LANDER ***"
:PRINT
230 PRINT" THE OBJECT OF THIS GA
ME IS TO  LAND YOUR CRAFT ON ONE
 OF THE   LANDING BASES ON THE A
LIEN"
240 PRINT"PLANET. YOU MUST MANOE
UVER       AROUND THE MOUNTAINS B
EING       CAREFUL NOT TO HIT ANY
. IF YOU   RUN OUT OF FUEL, YOU W
ILL NOT BEABLE TO MOVE SO YOU WI
LL CRASH."
250 M$="    HIT ENTER    "
260 M$=RIGHT$(M$,13)+LEFT$(M$,1)
:FORC=1TO50:NEXT:PRINT@392,M$;
270 A$=INKEY$:IFA$=""THEN260ELSE
```

```
IFASC(A$)=13THEN280ELSE260
280 CLS:PRINT@9,"*** LANDER ***"
:PRINT
290 PRINT" IF YOU LAND ON THE FI
RST BASE, YOU GET 1 POINT. IF YO
U LAND ON THE SECOND, YOU GET 5
POINTS AND";
300 PRINT"THE THIRD YOU GET 10 P
OINTS. TO STEER, USE THE RIGHT A
ND LEFT   ARROW KEYS. FOR THRUST
, YOU MUSTPRESS THE UP ARROW. WA
RNING- DO NOT GO TOO FAR TO THE
RIGHT OR  YOU WILL APPEAR ON THE
 LEFT."
310 M$=RIGHT$(M$,13)+LEFT$(M$,1)
:FORC=1TO50:NEXT:PRINT@424,M$;
320 A$=INKEY$:IFA$=""THEN310ELSE
IFASC(A$)=13THEN330ELSE310
330 CLS:PRINT@64,"WHICH SKILL LE
VEL:":PRINT:PRINT
340 PRINT@198,"(1) BEGINNER":PRI
NT@230,"(2) AMATEUR":PRINT@262,"
(3) EXPERT":PRINT@294,"(4) CHAMP
ION":SCREEN0,1
350 A$=INKEY$
360 IFA$="1"THENSK=1:GOTO410
370 IFA$="2"THENSK=2:GOTO410
380 IFA$="3"THENSK=3:GOTO410
390 IFA$="4"THENSK=4:GOTO410
400 GOTO350
410 F=1000:SC=0:POKE65495,0
420 PMODE3,1:PCLS:COLOR2,1:CLS
430 PRINT@200,"FUEL:"F"LITRES":P
RINT@268,"SCORE:"SC:SCREEN0,1
440 LINE(0,67)-(7,61),PSET:LINE-
(16,61),PSET
450 LINE-(20,67),PSET:LINE-(20,7
6),PSET:LINE-(18,80),PSET
460 LINE-(18,82),PSET:LINE-(12,9
2),PSET:LINE-(12,99),PSET
470 LINE-(8,103),PSET:LINE-(4,10
3),PSET:LINE-(4,107),PSET
480 LINE-(7,111),PSET:LINE-(7,11
9),PSET:LINE-(11,123),PSET
490 LINE-(11,131),PSET:LINE-(15,
133),PSET:LINE-(24,133),PSET
500 LINE-(27,137),PSET:LINE-(27,
143),PSET:LINE-(31,152),PSET
510 LINE-(31,156),PSET:LINE-(34,
160),PSET:LINE-(34,165),PSET
520 LINE-(30,168),PSET:LINE-(30,
172),PSET:LINE-(32,176),PSET
530 LINE-(52,176),PSET:LINE-(55,
172),PSET:LINE-(55,168),PSET
540 LINE-(50,165),PSET:LINE-(50,
160),PSET:LINE-(55,156),PSET
550 LINE-(55,152),PSET:LINE-(60,
148),PSET:LINE-(60,138),PSET
560 LINE-(56,132),PSET:LINE-(56,
126),PSET:LINE-(48,119),PSET
570 LINE-(48,108),PSET:LINE-(46,
104),PSET:LINE-(40,104),PSET
580 LINE-(40,97),PSET:LINE-(35,9
2),PSET:LINE-(35,88),PSET
590 LINE-(39,84),PSET:LINE-(47,8
4),PSET:LINE-(51,80),PSET
600 LINE-(59,80),PSET:LINE-(68,7
2),PSET:LINE-(75,72),PSET
610 LINE-(80,64),PSET:LINE-(85,6
4),PSET:LINE-(88,61),PSET
620 LINE-(91,61),PSET:LINE-(91,5
6),PSET:LINE-(105,40),PSET
630 LINE-(105,36),PSET:LINE-(99,
31),PSET:LINE-(99,28),PSET
640 LINE-(104,25),PSET:LINE-(112
,25),PSET:LINE-(117,30),PSET
650 LINE-(120,30),PSET:LINE-(124
,27),PSET:LINE-(130,27),PSET
660 LINE-(130,34),PSET:LINE-(134
,42),PSET:LINE-(134,45),PSET
670 LINE-(124,55),PSET:LINE-(124
,60),PSET:LINE-(126,62),PSET
680 LINE-(126,65),PSET:LINE-(124
,71),PSET:LINE-(124,76),PSET
690 LINE-(128,76),PSET:LINE-(131
,80),PSET:LINE-(140,80),PSET
700 LINE-(143,84),PSET:LINE-(143
,88),PSET:LINE-(139,92),PSET
710 LINE-(139,95),PSET:LINE-(136
,99),PSET:LINE-(136,102),PSET
720 LINE-(132,107),PSET:LINE-(12
8,107),PSET:LINE-(125,111),PSET
730 LINE-(125,115),PSET:LINE-(12
0,123),PSET:LINE-(120,147),PSET
740 LINE-(135,161),PSET:LINE-(13
5,164),PSET:LINE-(131,164),PSET
750 LINE-(128,167),PSET:LINE-(12
8,171),PSET:LINE-(131,175),PSET
760 LINE-(159,175),PSET:LINE-(16
3,172),PSET:LINE-(163,168),PSET
770 LINE-(160,164),PSET:LINE-(15
6,164),PSET:LINE-(156,161),PSET
780 LINE-(163,156),PSET:LINE-(16
3,152),PSET:LINE-(162,149),PSET
790 LINE-(162,146),PSET:LINE-(15
7,142),PSET:LINE-(157,140),PSET
800 LINE-(155,136),PSET:LINE-(15
5,132),PSET:LINE-(145,130),PSET
810 LINE-(145,129),PSET:LINE-(15
5,116),PSET:LINE-(155,112),PSET
820 LINE-(163,112),PSET:LINE-(16
3,108),PSET:LINE-(167,104),PSET
830 LINE-(167,100),PSET:LINE-(17
5,92),PSET:LINE-(175,88),PSET
840 LINE-(172,84),PSET:LINE-(168
,84),PSET:LINE-(168,80),PSET
850 LINE-(160,76),PSET:LINE-(160
,72),PSET:LINE-(156,64),PSET
860 LINE-(156,60),PSET:LINE-(154
,55),PSET:LINE-(154,48),PSET
870 LINE-(159,42),PSET:LINE-(159
,40),PSET:LINE-(156,36),PSET
880 LINE-(156,32),PSET:LINE-(163
,28),PSET:LINE-(168,28),PSET
890 LINE-(183,43),PSET:LINE-(191
,43),PSET:LINE-(195,47),PSET
900 LINE-(199,47),PSET:LINE-(203
,44),PSET:LINE-(208,44),PSET
910 LINE-(216,51),PSET:LINE-(216
,56),PSET:LINE-(212,65),PSET
920 LINE-(212,70),PSET:LINE-(215
,72),PSET:LINE-(219,72),PSET
930 LINE-(224,68),PSET:LINE-(228
,68),PSET:LINE-(232,72),PSET
940 LINE-(232,76),PSET:LINE-(224
,82),PSET:LINE-(224,84),PSET
950 LINE-(228,84),PSET:LINE-(234
,94),PSET:LINE-(239,94),PSET
970 LINE-(236,109),PSET:LINE-(23
3,106),PSET:LINE-(233,103),PSET
980 LINE-(229,103),PSET:LINE-(22
9,106),PSET:LINE-(223,112),PSET
990 LINE-(223,114),PSET:LINE-(21
6,120),PSET:LINE-(216,125),PSET
1010 LINE-(230,144),PSET:LINE-(2
27,146),PSET:LINE-(218,146),PSET

1020 LINE-(216,148),PSET:LINE-(2
16,158),PSET:LINE-(219,160),PSET

1030 LINE-(219,164),PSET:LINE-(2
16,166),PSET:LINE-(216,171),PSET

1040 LINE-(223,175),PSET:LINE-(2
44,175),PSET:LINE-(252,171),PSET

1050 LINE-(252,166),PSET:LINE-(2
48,164),PSET:LINE-(248,160),PSET

1060 LINE-(251,158),PSET:LINE-(2
51,140),PSET:LINE-(248,136),PSET

1070 LINE-(248,132),PSET:LINE-(2
51,128),PSET:LINE-(255,128),PSET

1080 PAINT(128,191),2,2:COLOR3,1

1090 LINE(32,177)-(51,177),PSET:
LINE(32,176)-(51,176),PSET:LINE(
32,175)-(51,175),PSET:LINE(32,17
4)-(51,174),PSET
```

```
1100 LINE(132,177)-(159,177),PSE
T:LINE(132,176)-(159,176),PSET:L
INE(132,175)-(159,175),PSET:LINE
(132,174-(159,174),PSET
1110 LINE(224,177)-(243,177),PSE
T:LINE(224,176)-(243,176),PSET:L
INE(224,175)-(243,175),PSET:LINE
(224,174)-(243,174),PSET
1120 DRAW"BM39,182E3D11L3R7":DRA
W"BM151,180L7D3F1R4F2D2G2L5":DRA
W"BM224,182E3D11L3R6":LINE(234,1
79)-(240,190),PSET,B:SIsACREEN1,
0
1130 X=5:Y=0
1140 COLOR4,1:DRAW"BM"+STR$(X)+"
,"+STR$(Y)+"D7G4E4F4"
1150 IFPPOINT(X+1,Y)=2THEN1300
1160 IFPPOINT(X-4,Y+9)=2THEN1300

1170 IFPPOINT(X+4,Y+9)=2THEN1300

1180 IFPPOINT(X,Y+12)=3THEN1370
1190 COLOR1,1:DRAW"BM"+STR$(X)+"
,"+STR$(Y)+"D7G4E4F4"
1200 IFPEEK(344)=247GOSUB1240
1210 IFPEEK(343)=247GOSUB1260
1220 IFPEEK(341)=247GOSUB1280
1230 Y=Y+SK:GOTO1140
1240 IFX=>249THENX=5:Y=0:RETURNE
LSEIFF=<0THEN1410
1250 X=X+2:F=F-1:RETURN
1260 IFX=<5THENRETURNELSEIFF<=0T
HEN1410
1270 X=X-2:F=F-1:RETURN
1280 IFY=<(SK+2)THENRETURNELSEIF
F=<0THEN1410
1290 Y=Y-(SK+2):F=F-5:RETURN
1300 PLAY"V31L255AG3AG":FORT=0TO
30:RD=RND(8):CIRCLE(X,Y),J,RD:J=
J+1:NEXTT:J=0
1310 FORT=0TO500:NEXT:CLS
1320 PRINT@107,"YOU CRASHED":PRI
NT@230,"BUT YOU GOT"SC"POINTS":P
RINT@391,"ANOTHER IsAGAME (Y/N)"
:SCREEN0,1
1330 POKE65494,0:IFO=1THENO=0:GO
TO1340ELSESOUND90,8:SOUND90,8:SO
UND90,4:SOUND90,8:SOUND118,8:SOU
ND109,4:SOUND109,6:SOUND90,4:SOU
ND90,8:SOUND79,4:SOUND90,8
1340 A$=INKEY$:IFA$="Y"THEN330
1350 IFA$="N"THENCLS:END
1360 GOTO1340
1370 IFX<80THENSC=SC+1ELSEIFX>20
0THENSC=SC+10:F=F+300:ELSESC=SC+
5:F=F+200
1380 PLAY"V3102L70FGACFGACFGACFG
ACFGACFGACFGACFCACFCAC":IFI=1THE
NI=0:RETURN
1390 CLS:FORT=0TO500:NEXTT:IFF=<
0THEN1400ELSE420
1400 PRINT@103,"YOU RAN OUT OF F
UEL":PRINT@230,"BUT YOU GOT"SC"P
OINTS":PRINT@359,"ANOTHER GAME (
Y/N)":SCREEN0,1:O=1:GOTO1330
1410 COLOR4,1:DRAW"BM"+STR$(X)+"
,"+STR$(Y)+"D7G4E4F4"
1420 IFPPOINT(X,Y+12)=3THEN1370
1430 IFPPOINT(X,Y+12)=2THEN1300
1440 COLOR1,1:DRAW"BM"+STR$(X)+"
,"+STR$(Y)+"D7G4E4F4"
1450 Y=Y+SK:GOTO1410
```

```
1 REM    *** MEMORY PEEK ***
2 REM        FOR  VZ 200
3 REM        BY R.CARSON
4 REM    ********************
5 CLS
6 PRINT"[illegible]";
7 PRINT"[illegible]";
8 PRINT"[illegible]";' TO SLOW DOWN PRINTING
9 PRINT"[illegible]";
10 PRINT"[illegible]";
11 PRINT"[illegible]";
12 PRINT"[illegible]";
20 INPUT"MEMORY LOCATION DECIMAL=";X1
22 PRINT"[illegible]";
23 FORD=0TO499:NEXTD
24 GOTO20000
25 X=ABS(X1)+ABS(A1)
26 IFX>65535THENGOTO20100
30 A2=X/4096:B2=A2-INT(A2):C2=INT(A2-B2):Z=65
40 FORY=10TO15
50 IFC2=YTHENQ$=CHR$(Z):GOTO80
60 Z=Z+1:NEXT
80 D2=B2*4096:E2=D2/256:F2=E2-INT(E2):G=INT(E2-F2):Z=65
90 FORY=10TO15
100 IFG=YTHENR$=CHR$(Z):GOTO130
110 Z=Z+1:NEXT
130 H=F2*256:I=H/16:J=I-INT(I):K=INT(I-J):Z=65
140 FORY=10TO15
150 IFK=YTHENS$=CHR$(Z):GOTO180
160 Z=Z+1:NEXT
180 L=J*16:M=L-INT(L):P=INT(L-M):Z=65
190 FORY=10TO15
200 IFP=YTHENT$=CHR$(Z):GOTO230
210 Z=Z+1:NEXT
230 IFC2>9THEN240ELSE250
240 PRINTTAB(2)Q$;:GOTO260
250 PRINTC2;
260 IFG>9THEN270ELSE280
270 PRINTTAB(4)R$;:GOTO290
280 PRINTG;
290 IFK>9THEN300ELSE310
300 PRINTTAB(6)S$;:GOTO320
310 PRINTK;
320 IFP>9THEN330ELSE340
330 PRINTTAB(8)T$;:GOTO350
340 PRINTP;
350 GOTO5055
5030 FORA1=0TO65535
5032 X2=A1+X1
5035 IFX2>65535THENGOTO20100
5037 IFX2>32767THENX2=X2-65536
5040 B1=PEEK(X2)
5045 L$=INKEY$:IFL$="  "THEN25
5047 GOTO5055
5052 PRINT"[illegible]";
5053 FORD=0TO499:NEXTD
5055 PRINTTAB(12)X1+A1;
5060 PRINTTAB(20)X2;
```

```
5070 PRINTTAB(26)CHR$(B1);
5080 PRINTTAB(28)B1
5085 K$=INKEY$:IFK$=" "THEN20
5100 NEXTA1
20000 IFX1<-32768THENGOTO20100
20020 GOTO5030
20100 PRINT"[illegible]";
20110 PRINT"[illegible]"
20115 K$=INKEY$
20116 I$=INKEY$:IFI$=""THEN20116
20117 IFI$="Y"CLS:GOTO20
20118 IFI$="N"CLS:END
20120 I$=INKEY$:IFI$<>"Y"ANDI$<>"N"THEN20116
```

CONTEST LOG (VZED)

```
170 CLEAR 2000
180 DIM C1$(2000)
190 CLS
200 REM
210 CLS:PRINT:PRINT" NEXT CALL SIGN, SEE BELOW":PRINT
211 PRINT:PRINTTAB(3);" LIST  :- LIST WITHOUT SORT"
212 PRINT:PRINTTAB(3);" SORT  :- SORT CALL SIGNS"
213 PRINT:PRINTTAB(3);" PRINT:- LIST ON PRINTER"
214 PRINT:PRINTTAB(3);" END   :- END PROG."
215 PRINT:PRINTTAB(3);"       :- ENTER CALLSIGN"
216 PRINT:PRINT" ENTER :- ";:INPUT A1$
220 IF A1$="SORT" THEN 500
230 IF A1$="LIST" THEN 700
235 IF A1$="END" THEN CLS:END
236 IF A1$="PRINT" THEN 950
240 FOR I=1TO LEN(A1$)
245 NEXT I
260 CLS
270 REM
280 FOR I=1 TO N
290 IF A1$=C1$(I) THEN 400
300 NEXT I
310 REM
320 N=N+1
325 C1$(N)=A1$
330 PRINT:PRINT:PRINT TAB(3)" ";A1$;" IS NEW CALL SIGN"
340 PRINT:PRINT:PRINT TAB(5)" ";N;" CALLS LOGGED "
345 FOR X = 1 TO 1000
350 NEXT X
360 GOTO 200
400 REM
410 PRINT:PRINT:PRINT TAB(4)" ";A1$;" ALREADY LOGGED"
420 GOTO 340
500 REM
510 CLS:PRINT:PRINT TAB(12);"SORTING":PRINT
520 FOR I=1 TO N
530 A1$=C1$(I)
540 PRINT "*";
550 FOR J=I TO N
560 IF A1$<=C1$(J) THEN 580
570 B1$=C1$(J)
572 C1$(J)=A1$
574 A1$=B1$
580 NEXT J
590 C1$(I)=A1$
600 NEXT I
610 PRINT:PRINT:PRINTTAB(9);"SORT COMPLETE"
620 PRINT:PRINT:PRINT:PRINT" DO YOU WANT A PRINTOUT?":PRINT
621 PRINT"  [PRINT] = PRINTOUT TO PRINTER"
622 PRINT"  [YES]   = PRINTOUT TO VDU"
623 PRINT"  [NO]    = RETURN TO MENU"
625 K$=INKEY$
626 I$=INKEY$:IFI$=""THEN625
627 IFI$<>"Y"ANDI$<>"P"ANDI$<>"N"THEN625
630 IF I$="N" THEN 190
635 IF I$="P" THEN 950
640 IF I$="Y" THEN 700
700 PRINT
702 CLS:PRINT:PRINT TAB(7);"CALL SIGNS LOGGED":PRINT
710 FOR I=1 TO N
750 PRINT C1$(I),
760 NEXT I
764 PRINT:PRINT"  PRESS >>SPACE<< TO CONTINUE"
765 K$=INKEY$
770 I$=INKEY$:IFI$<>" "THEN765
780 GOTO 900
900 REM
910 CLS:PRINT:PRINT:PRINT:PRINT"   DO YOU WANT TO STOP NOW?"
912 PRINT:PRINT"          Y=YES     N=NO "
913 PRINT:PRINT:PRINT:PRINT:PRINT TAB(5)" ";N;" CALLS LOGGED "
915 K$=INKEY$
917 X$=INKEY$:IF X$=""THEN 917
920 IF X$="Y" THEN CLS:END
925 IF X$="N" THEN 200
927 IFX$<>"Y"ANDX$<>"N"THEN917
950 LPRINT TAB(15);"CALL SIGNS LOGGED"
955 LPRINT
960 FOR I=1 TO N+1
970 LPRINT C1$(I),
980 NEXT I
990 GOTO 900
```

```
100 REM  ***** DOG RACE *****
101 REM  AS PRINTED IN MICRO-80          FOR TRS80 - SYS80
102 REM  MODIFIED BY R. CARSON           FOR VZ 200
103 REM
104 REM
105 REM
109 CLS:PRINT:PRINT
110 PRINT"       **** DOG RACE ****"
115 PRINT:PRINT:PRINT:PRINT"   PRESS ANY KEY TO CONTINUE"
117 PRINT:PRINT:PRINT:PRINT"  PRESS <SPACE> TO START RACE"
120 I$=INKEY$
```

```
125 A3$=INKEY$:IFA3$=""THEN120
130 CLS:MODE(1)
131 COLOR4:FORX=0TO127:SET(X,0):NEXT:FORX=0TO127:SET(X,1):NEXT
134 FORX=0TO127:SET(X,2):NEXT
135 FORX=0TO127:SET(X,42):NEXT:FORX=0TO127:SET(X,43):NEXT
136 FORX=0TO127:SET(X,44):NEXT:COLOR3
137 FORX=0TO123:SET(X,12):NEXT
138 FORX=0TO123:SET(X,22):NEXT
139 FORX=0TO123:SET(X,32):NEXT
140 A=22:B=5:C=22:D=15:G=22:H=25:I=22:J=35
145 COLOR2
150 REM DRAW STAT DOG
160 X=A:Y=B:GOSUB370
170 X=C:Y=D:GOSUB370
180 X=G:Y=H:GOSUB370
190 X=I:Y=J:GOSUB370
210 COLOR2:FORY=4TO40STEP5:SET(124,Y):NEXTY
220 I$=INKEY$
225 K$=INKEY$:IFK$<>" "THEN225
230 Z=RND(4)
235 P=RND(5)
240 IFZ=1THENX=A:Y=B:GOSUB410:A=X:GOTO280
250 IFZ=2THENX=C:Y=D:GOSUB410:C=X:GOTO280
260 IFZ=3THENX=G:Y=H:GOSUB410:G=X:GOTO280
270 IFZ=4THENX=I:Y=J:GOSUB410:I=X:GOTO280
280 IFX<130THENGOTO230
285 FORW=1TO1000:NEXTW
290 IFA>=130THENPRINT"NO. 1 IS THE WINNER PAY";O$;P*15;"CENTS".
300 IFC>=130THENPRINT"NO. 2 IS THE WINNER PAY";O$;P*15;"CENTS"
310 IFG>=130THENPRINT"NO. 3 IS THE WINNER PAY";O$;P*15;"CENTS"
320 IFI>=130THENPRINT"NO. 4 IS THE WINNER PAY";O$;P*15;"CENTS"
330 FORF=1TO1000:NEXTF
340 INPUT"WOULD YOU LIKE ANOTHER RACE (Y/N)";A2$
350 IFA2$="Y"THEN100
360 IFA2$="N"THENCLS:END
370 SET(X-9,Y):SET(X-20,Y):SET(X-6,Y+1):SET(X-7,Y+1)
380 SET(X-8,Y+1):SET(X-19,Y+1):SET(X-10,Y+4):SET(X-17,Y+4)
390 SET(X-11,Y+5):SET(X-16,Y+5)
400 FORU=9TO18:FORV=2TO3:SET(X-U,Y+V):NEXTV:NEXTU:RETURN
410 RESET(X-20,Y):RESET(X-19,Y+1):SET(X-17,Y+1):SET(X-16,Y)
420 SET(X-5,Y+1):SET(X-4,Y+1):RESET(X-9,Y):SET(X-6,Y)
430 RESET(X-18,Y+2):RESET(X-17,Y+2):SET(X-8,Y+2):SET(X-7,Y+2)
440 RESET(X-8,Y+1):RESET(X-7,Y+1):RESET(X-11,Y+5):RESET(X-10,Y+4)
450 SET(X-8,Y+4):SET(X-7,Y+5):RESET(X-18,Y+3):RESET(X-17,Y+3)
460 SET(X-8,Y+3):SET(X-7,Y+3):RESET(X-17,Y+4):SET(X-15,Y+4)
470 RESET(X-17,Y+1):SET(X-15,Y+1):RESET(X-16,Y+2):RESET(X-16,Y+3)
480 RESET(X-15,Y+2):RESET(X-15,Y+3):RESET(X-16,Y+5)
490 RESET(X-15,Y+5):RESET(X-15,Y+4)
500 SET(X-13,Y+4):SET(X-12,Y+5):RESET(X-8,Y+4):SET(X-6,Y+4)
510 SET(X-6,Y+2):SET(X-5,Y+2):SET(X-6,Y+3):SET(X-5,Y+3)
520 SET(X-3,Y+1):SET(X-2,Y+1):RESET(X-6,Y):SET(X-5,Y)
530 RESET(X-6,Y+1):RESET(X-5,Y+1):X=X+4:RETURN
```

****** (LI/4K) Latin Vocabulary Test ******

```
1 REM C.STOBERT C/O P.O.BOX 14100 KILBIRNIE WELLINGTON N.Z.
2 G.1000
10 C.:C=0:E=0:A=0:P.:P."TRY OUR LATIN TEST"
30 P.:P."DO YOU WISH TO TEST 'LATIN/ENGLISH'-<1>"
40 I."OR 'ENGLISH/LATIN'-<2> ";S
50 IF(S<>1)*(S<>2)P."1 OR 2 PLEASE":G.30
60 F.N=1TO125:A(N)=0:N.N
100 N=R.(60-1):N=N*2+1
110 IFS=2N=N+1
120 IFA(N)=2G.100
130 A(N)=2:W=1:IFS=2Q=N-1
150 IFS=1Q=N+1
160 A(Q)=1:M=R.(60)-1:M=2*M+1
170 IFS=1M=M+1
180 IFA(M)=0A(M)=1:W=W+1
190 IFW=5G.300
200 G.160
300 C.:REST.:IFS=1B$=LATIN
320 IFS=2B$=ENGLISH
330 F.R=1TON:READA$:N.R
340 C.:P.AT64,"YOUR ";B$;" WORD IS :-";A$
350 IFS=1B$=ENGLISH
360 IFS=2B$=LATIN
370 P.:P."YOUR ";B$;" EQUIVALENT IS :-"
400 Z=1:REST.
410 F.R=1TO120:READA$
420 IFR=QA(120+Z)=1
430 IFA(R)=1P.T.(5);Z;")- ";A$:Z=Z+1
440 N.R:I=0
510 P.:I."SELECT YOUR OPTION -<1> TO <5> ";O:A=A+1
520 O=I.(O):IF(O<1)+(O>5)P.:P."1 TO 5 PLEASE":G.550
530 IFA(120+O)=1GOS.850:P.:G.630
535 I=I+1:IFI=2G.600
540 P.:P."SORRY!";
550 I."PRESS <ENTER> TO TRY AGAIN ";B$
560 GOS.770:G.510
600 P.:P."NO! THE CORRECT RESULT WAS";
610 F.D=1TO5:IFA(120+D)=1P.D:Z=D
620 N.D:E=E+1:GOS.800:G.640
630 C=C+1
640 GOS.770:IFC+E=20G.700
650 P.:I."PRESS <ENTER> FOR NEXT WORD ";B$
660 F.R=1TO125:IFA(R)=1A(R)=0
670 N.R:G.100
700 C.:P.AT64,"YOU HAVE HAD 20 WORDS"
720 P.:P."YOU HAD";A;"ATTEMPTS"
730 P."FOR";C;"CORRECT."
740 P.:P."OR";I.(C/A*100);"%"
750 P.:I."PRESS <ENTER> FOR ANOTHER SELECTION ";B$:G.10
770 P.AT640:P.:P.:P.:P.AT576,:RET.
800 L=195+64*Z:F.X=1TO50
810 P.ATL," ";:F.T=1TO15:N.T
820 P.ATL,"^";:F.T=1TO20:N.T:N.X:RET.
850 ONR.(3)G.852,854,856
852 B$="** CORRECT **":G.860
854 B$="## WELL DONE ##":G.860
```

```
856 B$=%% GOOD WORK %%
860 F.X=1TO30:P.AT778,"                           ";:F.T=1TO10:N.T
870 P.AT778,B$;:F.T=1TO20:N.T:N.X:RET.
900 D.AGRICOLA,FARMER,AMITA,AUNT,ARA,ALTAR,BESTIA,BEAST
905 D.DEA,GODDESS,FABULA,FABLE,FIGURA,FIGURE,FLAMMA,FLAME
910 D.INCOLA,RESIDENT,INSULA,ISLAND,LINGUA,TONGUE,NAUTA,SAILOR
915 D.SCHOLA,SCHOOL,SILVA,FOREST,ALBA,WHITE,AMICA,FRIENDLY
920 D.ANGUSTA,NARROW,BEATA,HAPPY,BENIGNA,KIND,MARITIMA,OF THE SE
A
925 D.MEA,MY/MINE,MIRA,STRANGE,NOSTRA,OUR,NOTA,FAMOUS
930 D.OBSCURA,DARK/DIM,PERICULOSA,DANGEROUS,PROPINQUA,NEARBY
932 D.PULCHRA,BEAUTIFUL
935 D.TUA,YOUR,DAT,GIVES,DICEBANT,SAID,HABITAT,LIVES
940 D.MONSTRAT,SHOWS,NARRAT,TELLS,NECABANT,KILLED,TIMEBANT,FEARE
D
945 D.VASTABANT,DESTROYED,VIDEBANT,SAW,VIDEMUS,WE SEE,VOCAT,CALL
S
950 D.MOX,SOON,PROCUL,FAR AWAY,SUM,I AM,ES,YOU ARE(S)
955 D.EST,IT IS,SUMUS,WE ARE,ESTIS,YOU ARE(P),SUNT,THEY ARE
960 D.ERAM,I WAS,ERAS,YOU WERE(S),ERAT,IT WAS,ERAMUS,WE WERE
965 D.ERATIS,YOU WERE(P),ERANT,THEY WERE,ERO,I WILL BE
966 D.ERIS,YOU WILL BE(S)
970 D.ERIT,IT WILL BE,ERIMUS,WE WILL BE,ERITIS,YOU WILL BE(P)
972 D.ERUNT,THEY WILL BE
1000 C.:P."THIS PROGRAMME IS A SIMPLE LATIN-ENGLISH VOCABULARY T
EST":P.
1020 P."YOU SELECT EITHER LATIN OR ENGLISH & FIVE OPTIONS OF THE
 OTHER"
1030 P."ARE OFFERED":P.:P."IF YOUR SELECTION IS CORRECT"
1040 P."YOU GET ANOTHER WORD."
1050 P.:P."OTHERWISE YOU GET ONE MORE TRY.":P.:I."PRESS <ENTER>"
;B$
1060 C.:P."THERE ARE 20 WORDS POSED"
1070 P.:P."AFTER THIS YOUR 'SCORE' IS POSTED"
1080 P.:P."YOU CAN THEN PROCEED TO A NEW GROUP.":P.
1090 I."PRESS <ENTER>";B$:G.10
-----------------------------------------------------------------
```

**** (LII/4K) Obstacle ****

TRS-80/SYSTEM-80

```
5 '                 OBSTACLE (L2/4K) BY P.BRIERLEY
7 '                 P.O.BOX 158, LINDFIELD, N.S.W., 2070
8 '                 COPYRIGHT (C) JANUARY, 1981
9 CLS:PRINTCHR$(23):PRINT@448,"*********** OBSTACLE ***********"
:FORC=1TO1500:NEXT
10 CLS:PRINTCHR$(23):PRINT"NAME OF PLAYER ON LEFT ":INPUTL$:PRIN
T"NAME OF PLAYER ON RIGHT":INPUTR$
11 CLS:PRINTCHR$(23):INPUT"SPEED 1-20 (FAST-SLOW";A:INPUT"NUMBER
 OF HAZARDS (0-50)";H:GOTO350
12 S=0:RANDOM:CLS:FORX=0TO127:SET(X,0):SET(X,47):NEXT:FORY=0TO47
:SET(0,Y):SET(1,Y):SET(126,Y):SET(127,Y):NEXT:GOTO360
15 L=RND(4):R=RND(4):LX=42:LY=23:RX=84:RY=23
20 SET(LX,LY):SET(LX+1,LY):SET(RX,RY):SET(RX+1,RY):S=S+1
30 FORB=1TOA:A$=INKEY$
40 IFA$="W"THENL=1
45 IFA$="O"THENR=1
50 IFA$="X"THENL=2
55 IFA$="."THENR=2
60 IFA$="A"THENL=3
65 IFA$="K"THENR=3
70 IFA$="D"THENL=4
75 IFA$=";"THENR=4
77 NEXT
80 ONLGOTO90,100,110,120
90 LY=LY-1:GOTO140
100 LY=LY+1:GOTO140
110 LX=LX-2:GOTO140
120 LX=LX+2:GOTO140
140 ONRGOTO150,160,170,180
150 RY=RY-1:GOTO190
160 RY=RY+1:GOTO190
170 RX=RX-2:GOTO190
180 RX=RX+2:GOTO190
190 IF(POINT(LX,LY)=-1)AND(POINT(RX,RY)=-1)THEN225
200 IFPOINT(LX,LY)=-1THEN235
210 IFPOINT(RX,RY)=-1THEN245
220 GOTO20
225 PRINT@0,"YOU BOTH LOSE!";
230 FORB=1TO10:GOSUB260:GOSUB270:NEXT:GOTO275
235 PRINT@0,L$;" LOSES, ";R$;" WINS";:RS=RS+S
240 FORB=1TO20:GOSUB260:NEXT:GOTO275
245 PRINT@0,R$;" LOSES, ";L$;" WINS";:LS=LS+S
250 FORB=1TO20:GOSUB270:NEXT:GOTO275
260 FORC=1TO10:NEXTC:RESET(LX,LY):RESET(LX+1,LY):FORC=1TO10:NEXT
C:SET(LX,LY):SET(LX+1,LY):RETURN
270 FORC=1TO10:NEXTC:RESET(RX,RY):RESET(RX+1,RY):FORC=1TO10:NEXT
C:SET(RX,RY):SET(RX+1,RY):RETURN
275 CLS:PRINTCHR$(23):GOSUB300:PRINTL$;"'S SCORE IS";LS:PRINTR$;
"'S SCORE IS";RS:FORC=1TO1500:NEXT:CLS
280 PRINT"PRESS ANY KEY TO START":PRINT"PRESS 'N' TO RESTART":PR
INT"PRESS 'H' TO RESET SPEED/HAZARDS";:A$=INKEY$
290 A$="":A$=INKEY$:IFA$=""THEN290ELSEIFA$="N"THENRUNELSEIFA$="H
"THEN11ELSE12
300 PRINT"GAME SCORE IS";S:RETURN
320 CLS:PRINT@10,"LEFT PLAYER":PRINT@41,"RIGHT PLAYER":PRINT@143
,"W":PRINT@207,CHR$(91):PRINT@267,"A ";CHR$(93);" S ";CHR$(94);"
 D":PRINT@335,CHR$(92):PRINT@399,"X"
330 PRINT@175,"O";:PRINT@239,CHR$(91);:PRINT@299,"K ";CHR$(93);"
 L ";CHR$(94);" +";:PRINT@367,CHR$(92);:PRINT@431,">":PRINT:PRIN
T"USE KEYS SURROUNDING CENTRAL LETTER TO CONTROL DIRECTION":PRIN
T:PRINT"IF YOU COLLIDE WITH ANY WALL OR TRAIL YOU WILL LOSE"
340 PRINT"IF YOU GO BACK ON YOUR OWN TRAIL YOU WILL ALSO LOSE":P
RINT"IF YOU BOTH CRASH AT THE SAME TIME, YOU BOTH LOSE":GOTO280
350 IFA>20ORH>50THEN11ELSE320
360 IFH=0THEN15
365 FORC=1TOH
370 X=2*RND(62):Y=RND(46):IFX=42ORX=84ORY=23THEN370ELSESET(X,Y):
SET(X+1,Y):NEXT:GOTO15
-----------------------------------------------------------------
```

```
**** (LII/16K)   Track 80 ****

TRS-80/SYSTEM-80

1 REM CREATED BY C. MAC NISH
2 REM  19 GRANGE ST. CLAREMONT, W.A. 6010
3 REM PHONE: (09) 384 5408
4 REM
5 CP$="COMPUTER":HS=500:FORI=32000TO32127:READX:POKEI,X:NEXTI
10 CLS:M=3:SC=0:TA$="":MP=0::POKE(32700),0
20 TL=15360+26:TR=TL+15
30 A=16353
100 POKE16526,0:POKE16527,125
190 GOSUB10000
196 POKE15367,191
198 POKEA,158:POKEA+1,173
200 FORI=15360+26TO15360+26+15*64STEP64
210 POKE I,174:POKEI+15,157
220 NEXTI
230 GOSUB 11000
300 GOTO500
320 POKE TL+E,32:POKETL+E+1,32
400 POKETL,32:POKETL+15,32:RT=RND(3)
410 TL=TL-2+RT:IF TL<15368 THENTL=TL+1ELSE IFTL>15406THENTL=TL-1

420 POKE TL,174:POKETL+15,157
430 R=RND(12)
440 ON R  GOSUB 2400,2400,2400,2400,2400,2900,2800,2600,2700,250
0,2500,2500
450 IF PEEK(14368)=16 THENPOKEA,32:POKEA+1,32:A=A-2:POKEA,158:PO
KEA+1,173:GOTO500
460 IF PEEK(14368)=64 THENPOKEA,32:POKEA+1,32:A=A+2:POKEA,158:PO
KEA+1,173
500 X=USR(0)
510 IFPEEK(32700)=1THEN3000
530 SC=SC+10+PEEK(32702):PRINT@0,SC;
540 MU=SC-MP:IF MU>2000THENPOKEA,32:POKEA+1,32:A=A-64:POKEA,158:
POKEA+1,173:MP=MP+2000
600 GOTO320
2000 POKEA,32:POKEA+1,32:A=A-2:POKEA,158:POKEA+1,173:RETURN
2100 POKEA,32:POKEA+1,32:A=A+2:POKEA,158:POKEA+1,173:RETURN
2400 RETURN
2500 E=RND(13)
2510 POKETL+E,187:POKETL+E+1,183
2520 RETURN
2600 E=RND(13)
2610 POKE TL+E,157:POKETL+E+1,174
2620 RETURN
2700 E=RND(13)
2710 POKE TL+E,154:POKETL+E+1,165
2720 RETURN
2800 E=RND(13)
2810 POKETL+E,153:POKE TL+E+1,166
2820 RETURN
2900  E=RND(13)
2910 EP=RND(3)
2920 IFEP=1THEN POKETL+E,42:POKETL+E+1,42:RETURN
2930 IFEP=2THEN POKETL+E,64ELSE POKETL+E,36
2940 RETURN
3000 POKE A,158:POKE A+1,173
3010 FOREX=1TO20:NEXTEX
3020 POKEA,152:POKEA+1,164:FOREX=1TO30:NEXTEX
3030 POKEA,160:POKEA+1,144:FOREX=1TO30:NEXTEX
3040 POKEA,146:POKEA+1,161:FOREX=1TO20:NEXTEX
3050 POKEA,182:POKEA+1,184:POKEA-1,32:POKEA+2,32
3060 FORI=1TO50:NEXTI
3070 IFA<16320 THENPOKEA,32:POKEA+1,32:A=A+64:POKEA,182:POKEA+1,
184:FORJ=1TO20:NEXTJ:GOTO3070
3100 FORI=16322 TOA-1
3110 POKEI,181:POKEI-1,172:POKEI-2,191:POKEI-3,32
3120 FORJ=1TO12:NEXTJ
3130 NEXTI
3150 FORI=ATO16381
3160 POKEI+2,184:POKEI+1,182:POKEI.181:POKEI-1,172:POKEI-2,191:P
OKEI-3,32
3170 FORJ=1TO30:NEXTJ
3180 NEXTI
3200 POKEI-3,32:POKEI-2,32:POKEI-1,32:POKEI,32:POKEI+1,32
3220 FORJ=1TO100:NEXTJ
3500 CLS
3510 PRINT:PRINT"YOUR SCORE  IS"SC
3515 IFSC>HS THEN HS=SC:PRINT:PRINT"CONGRATULATIONS!!!   YOU  MA
DE THE HIGH SCORE":GOSUB3600
3517 PRINT:PRINT"HIGH SCORE---<";HS">---  BY  ";CP$
3520 FORGG=1TO600:NEXTGG:GOTO10PRINT:PRINT:PRINT"WOULD YOU LIKE
TO TRY AGAIN";:INPUTTA$
3600 PRINT:PRINT"WHAT IS YOUR NAME";:INPUT CP$:CLS:RETURN
10000 CLS:POKE15360,191:PRINT@92,"#######";
10010 X=USR(0):FORI=1TO60:NEXTI
10020 PRINT@92,"#     #";
10030 X=USR(0):FORI=1TO40:NEXTI
10040 PRINT@92,"#  0  #";
10050 X=USR(0):FORI=1TO40:NEXTI
10060 PRINT@92,"#  8  #";
10070 X=USR(0):FORI=1TO40:NEXTI
10080 PRINT@92,"#     #";
10090 X=USR(0):FORI=1TO40:NEXTI
10100 PRINT@92,"#  K  #";
10110 X=USR(0):FORI=1TO40:NEXTI
10120 PRINT@92,"#  C  #";
10130 X=USR(0):FORI=1TO40:NEXTI
10140 PRINT@92,"#  A  #";
10150 X=USR(0):FORI=1TO40:NEXTI
10160 PRINT@92,"#  R  #";
10170 X=USR(0):FORI=1TO40:NEXTI
10180 PRINT@92,"#  T  #";
10190 X=USR(0):FORI=1TO40:NEXTI
10200 PRINT@92,"#     #";
10210 X=USR(0):FORI=1TO40:NEXTI
10220 PRINT@92,"#######";
10230 X=USR(0):FORI=1TO40:NEXTI
```

```
10250 FORJ=1TO18:X=USR(0):FORI=1TO50:NEXTI:NEXTJ
10260 RETURN
11000 PRINT@130,"DODGE THE ONCOMING CARS";
11010 PRINT@194,"AND MAKE AS MANY POINTS";
11020 PRINT@258,"AS YOU CAN";
11030 PRINT@386,"BONUS POINTS CAN BE";
11040 PRINT@450,"MADE BY PASSING OVER";
11050 PRINT@514,"BONUS CHECKPOINTS";
11060 PRINT@642,"BONUS CHECKPOINTS!";
11070 PRINT@706,"+++++++++++++++++";
11080 PRINT@770," ' ** ' = 50";
11090 PRINT@834," ' @ ' = 100";
11100 PRINT@898," ' $ ' = 200";
11110 PRINT@172,"  YOUR CONTROLS";
11120 PRINT@236,"##################";
11130 PRINT@300,"' < ' = MOVE LEFT";
11140 PRINT@364,"' > ' = MOVE RIGHT";
11150 PRINT@492,"BEWARE!: THE GAME";
11160 PRINT@556,"GETS HARDER. EVERY";
11170 PRINT@620,"2000 POINTS YOU'LL";
11180 PRINT@684,"MOVE UP THE SCREEN";
11190 PRINT@812,"TO START PRESS ANY";
11200 PRINT@876,"KEY -- GOOD LUCK!!";
11205 PRINT@1004,"HIGH SCORE-";HS;
11210 ST$=INKEY$:IF ST$<> ""THENRETURN ELSE 11210
12000 DATA 33,255,63,1,191,63,10,119,43,11
12010 DATA 10,254,191,40,112,254,158,40,102,254
12020 DATA 173,40,98,126,254,158,40,7,254,173
12030 DATA 40,3,195,6,125,10,254,42,40,27
12040 DATA 254,64,40,41,254,36,40,55,17,0
12050 DATA 0,237,83,190,127,254,32,40,205,17
12060 DATA 1,0,237,83,188,127,201,17,40,0
12070 DATA 237,83,190,127,62,32,2,62,42,50
12080 DATA 129,60,195,8,125,17,90,0,62,32
12090 DATA 2,62,64,50,129,60,237,83,190,127
12100 DATA 195,8,125,17,190,0,237,83,190,127
12110 DATA 62,32,2,62,36,50,129,60,195,8
12120 DATA 125,62,32,119,195,8,125,201
```

**** (LII/16K) Touch Typing ****

TRS-80/SYSTEM-80

```
5 CLEAR 1000
10 '      T Y P E / B A S :1

Written By Spencer George
           14 / 47 Yerrin Street
           Balwyn, 3103

       (03) 836 4225
20 ' FIRST VERSION  29/4/80
IMPROVED  7/7/80
IMPROVED  28/7/80
IMPROVED  8/10/80

30 PRINT

              COPYRIGHT   C      SPENCER GEORGE   1980
80 CLS
82 DIM HOW$(12)
85 PRINT@520, CHR$(23); "TYPING PRACTICE"
90 DIM CHARACTERS$(59),  WHERE(59),KEY(16) , W2(59), W3$(59)
95 BOARD$ ="ASDFJKL;GHQWERUIOPTY@ZXCVNM,.B/1234789056:-<>?+! #$%
&'() *="
96 GOSUB 1000
97 GOSUB 1100
140 FOR J = 0 TO 16
150 READ KEY(J)
151 NEXT J
152 FOR J = 1 TO 59
153 READ CHARACTERS$(J), WHERE(J)
154 NEXT J
155 GOSUB 9000
161 CLS
162 PRINT"THERE ARE SIXTEEN PARTS TO THIS PROGRAM.
THE PARTS ARE GRADED FROM USING ONLY FOUR KEYS TO USING ALL KEYS

PART 1 USES  A S D F ONLY " : Z$= INKEY$ : GOSUB 9000
163 CLS:PRINT"PART 2 USES ALSO   JKL;
PART 3 USES ALSO   GH
164 PRINT"PART 4 USES ALSO   QWER
PART 5 USES ALSO   UIOP              PART 1 USES ONLY    ASDF
165 PRINT"PART 6 USES ALSO  TY@
PART 7 USES ALSO   ZXC
PART 8 USES ALSO   NM,.
PART 9 USES ALSO   VB/
PART 10 USES ALSO  1234
PART 11 USES ALSO  7890
PART 12 USES ALSO  56:-
PART 13 USES ALSO  <>?+
PART 14 USES ALSO  !#$
PART 15 USES ALSO  '()
166 PRINT"PART 16 USES ALSO   %&*=
170 PRINT"HOW MANY PARTS DO YOU WANT TO TRY " ;: INPUT PARTS
171  W = 0
180 DATA  0, 4,8,10,14,18,21,24,27,31,35,39,43,47,51,55,59
181 FOR J = 1 TO 12
182 READ HOW$(J)
183 NEXT J
200 UNIT = 3 + RND(5)
201 NUMBER = 5
205 CLS
210 FOR K = 1 TO PARTS
212 CLS
213 PRINT@525,"PART  "; K
214 PRINT@ 704,  HOW$(K)   : GOSUB 9000
215 FOR EASE = 1 TO 2  : CLS  : GOSUB 1010  : IF K > 12 GOSUB 11
05
216 PRINT@0,"PLEASE TYPE THESE CHARACTERS
```

```
"
220 FOR J = 1 TO NUMBER
230 FOR L = 1 TO UNIT
235 ON EASE GOSUB 640, 600
245 PRINT CHARACTERS$(  VARIED (L)   ) ;
250 NEXT L
255 PRINT TAB(12);"=            ";
259 Z$ = INKEY$
260 FOR L = 1 TO UNIT
270 ANSWER$(L) = INKEY$
275 IF ANSWER$(L) = "" THEN 270
280 NEXT L
290 FOR L = 1 TO UNIT
295 PRINT ANSWER$(L);
300 IF ANSWER$(L) <> CHARACTERS$( VARIED(L)  )   PRINT, "     ER
ROR  " :  GOSUB 2999 : CLS :GOSUB 1010 : W = W + 1 : PRINT@60,W;
 :   PRINT"
TRY AGAIN
PLEASE TYPE THESE CHARACTERS
" :   FOR LL = 1 TO UNIT : PRINT CHARACTERS$( VARIED(LL)  )  ; :
 NEXT LL
301 IF ANSWER$(L) <> CHARACTERS$( VARIED(L)  )  PRINT "    =
"; GOTO259
305 NEXT L
306 PRINT
308 UNIT = RND(3) + RND(6)
310 NEXT J
312 NEXT EASE
320 NEXT K
390 CLS
400 PRINT"YOU HAVE MADE"; W; "ERROR";: IF W > 1 PRINT "S. "  ELS
E PRINT    ". "
405 IF PEEK(&H37E8) <> 63 THEN 451
410 LPRINT"YOU HAVE MADE"; W ; "ERROR";:  IF W>1 LPRINT "S. " EL
SE LPRINT     ". "
430 FOR J = 1 TO 59
440 IF W2(J) <> 0 LPRINT CHARACTERS$(J) , W2(J); " TIME";: IF W2
(J) = 1 LPRINT "      "; ELSE LPRINT "S     ";
441 IF W2(J) <> 0 LPRINT "INCORRECT KEY";
442 IF W2(J) = 1 LPRINT " WAS   "; W3$(J)
443 IF W2(J) > 1 LPRINT "S WERE  "; W3$(J)
450 NEXT J
451 FOR J = 1 TO 59
452 W3$(J) = "":
    W2(J) = 0
453 NEXT J
460 GOTO 170
599 END
600 VARIED(L) = RND( KEY (K) )
610 RETURN
640 VARIED (L) = RND (    KEY(K) - KEY (K-1)   )   + KEY (K -1)
650 RETURN
1000 VIEW$ = "   1   2   3   4   5   6   7   8   9   0   :   -
     Q   W   E   R   T   Y   U   I   O   P   @
       A   S   D   F   G   H   J   K   L   ;
SHIFT   Z   X   C   V   B   N   M   ,   .   /   SHIFT
1001 RETURN
```

```
1010 PRINT@768,VIEW$;
1020 RETURN
1100 V21EW$ ="    !       #    $    %    &    '    (    )       *    =

                                                  +
                                             <      >    ?"
1101 RETURN
1105 PRINT@776,CHR$(34)
1110 RETURN
2999 PRINT@768,VIEW$;
3000 FOR Z4 = 1 TO 59
3010 IF CHARACTERS$( Z4 ) = CHARACTERS$ ( VARIED ( L )   ) THEN H
ERE =15360 + 768 + WHERE ( Z4 ) : GOTO 3025
3020 NEXT Z4
3021 PRINT@768,V2IEW$;
3022 PRINT@776,CHR$(34)
3023 GOTO3000
3025 FOR J8 = 1 TO 10
3030 POKE HERE,32
3040 FOR J9 = 1 TO 40
3041 IF INKEY$ <>"" THEN 3091
3050 NEXT J9
3060 POKE HERE , ASC( CHARACTERS$(  VARIED  (L)  )   )
3070 FOR J9 = 1 TO 40
3071 IF INKEY$<>"" THEN 3091
3080 NEXT J9
3090 NEXT J8
3091 W2( VARIED (L) ) = W2 (VARIED (L) ) + 1
3092 W3$( VARIED (L)) = W3$( VARIED (L) ) + " " + ANSWER$(L)
3100 RETURN
8000 DATA A,135,          S,139,  D,143,  F,147,  J,159,  K,163,
          L,167,  ;,171,  G,151,  H,155,  Q,69,   W,73,   E,77,
          R,81,   U,93,   I,97,   O,101,  P,105,  T,85,   Y,89,
          @,109,  Z,200,  X,204,  C,208,  V,212,  N,220
8010 DATA m,224,          ",",228,        .,232,  b,216,  /,236,
          1,3,    2,7,    3,11,   4,15,   7,27,   8,31,   9,35,
          0,39,   5,19,   6,23,   ":",43,  -,47,   <,227,  >,232,
            "?",236,  +,169,  !,4,    Q,7,    #,11,   $,15
8020 DATA %,19,   &,23,   ',27,   (,31,   ),35,   B,1,    *,43,
          =,47
9000 FOR J = 1 TO 2000
9010 IF INKEY$ <> "" THEN 9030
9020 NEXT J
9030 RETURN
10000 DATA THE FOUR FINGERS OF THE LEFT HAND SHOULD REST ABOVE T
HE KEYS
A   S  D   F.
THE FOUR FINGERS OF THE LEFT HAND ARE USED FOR KEYS
A  S  D  F.
10010 DATA THE FOUR FINGERS OF THE RIGHT HAND SHOULD REST ABOVE
THE KEYS
                                       J   K   L   ;.
THE FOUR FINGERS OF THE RIGHT HAND ARE USED FOR KEYS
                                       J  K  L  ;.
10020 DATA THE RIGHT FINGER OF THE LEFT HAND MOVES ACROSS TO THE

G   KEY.
```

```
THE LEFT FINGER OF THE RIGHT HAND MOVES ACROSS TO THE
                                      H   KEY.
10030 DATA THE FINGERS OF THE LEFT HAND MOVE UP TO THE KEYS
Q  W  E  R
10040 DATA  THE FINGERS OF THE RIGHT HAND MOVE UP TO THE KEYS
                                          U   I  O  P.
10045 DATA THE LEFT HAND FINGER MOVES UP TO THE
T   KEY.
THE RIGHT HAND FINGERS MOVES UP TO THE  KEYS
                                      Y       @.
10050 DATA THE FINGERS OF THE LEFT HAND MOVE DOWN TO THE KEYS
SHIFT  Z   X  C
10060 DATA THE FINGERS OF THE RIGHT HAND MOVE DOWN TO THE KEYS
                                       N  M  ,  .
10070 DATA THE FINGERS MOVE DOWN TO THE KEYS
V      B                                  /
10080 DATA THE LEFT HAND FINGERS MOVE UP TO
1  2  3  4
10090 DATA THE RIGHT HAND FINGERS MOVE UP TO
                                  7   8  9  0
10100 DATA THE FINGERS MOVE UP TO
                         5     6                  :   -
65000 '
```

****** (48K Disk Basic)   Sort Demo ******

TRS-80/SYSTEM-80

```
1 REM.              A FUNDAMENTAL SORT UTILITY

      B.J.C. 1980                           (C) MICRO-80 1980
10 DEFINT A,X,Y,Z: A=0: X=0: Y=0: Z=0: ZZ=0:
                       REM.  INTEGER DECL. AND INITIALIZATION
20 ZZ=-273+1:          REM.  M/L "END" POINTER INITIALIZATION
30 FOR X=-273 TO -224:
                       REM.  LOCATION IN RAM OF 50 SAMPLE BYTES
                             (UP TO -19 :  I.E. 255 DATA ITEMS)
40   Z=Z+1:            REM.  BYTE COUNTER
50   Y=RND(250):       REM.  DATA SOURCE FOR DEMONSTRATION
60   POKE X,Y:         REM.  LOADS DATA INTO RAM DEFINED IN 30
70   PRINT Y;:         REM.  PRINTOUT OF DATA INPUT TO ROUTINE
80 NEXT
90 DEFUSR0=&HFEC7:     REM:  DEFINE ENTRY POINT TO M/L ROUTINE
100 POKE -309,Z:       REM.  POKES NO. DATA ITEMS INTO "B"
                             REG IN M/L PROGRAM
110 POKE (Z+ZZ),198: POKE (Z+ZZ+1),254:
                       REM.  POKES LSB & MSB FEC6H AS END
                             STATEMENT AFTER LOCATION OF LAST
                             DATA ITEM LOADED BY LINE 60
120 Z8=USR0(Z9):       REM.  "GOSUB" M/L ROUTINE.  Z8 & Z9
                              ARE DUMMY ARGUMENTS
130 PRINT:PRINT
140 FOR X=-273 TO -224
150  A=PEEK(X):        REM.  THESE LOCATIONS NOW CONTAIN THE
                             DATA RE-ORDERED INTO SEQUENCE
160  PRINT A;:         REM.  AND HERE THEY ARE ! ! !
170 NEXT
180 DEFSNG X,Y,Z,A: A=0: X=0: Y=0: Z=0: ZZ=0:
                       REM.  LEAVE THINGS THE WAY ONE WOULD
                             WISH TO FIND THEM !!!
190 END
200 REM.          V A R I A B L E S

     X  :  COUNTER               Y  :  DATA - INPUT
     Z  :  BYTE COUNTER          A  :  DATA - OUTPUT
     ZZ :  END POINTER           Z8 :  DUMMY
                                 Z9 :  DUMMY
```

```
00100 ;         * * *   BUBBLE   * * *
00110 ;
00120 ;                          ...  SINGLE BYTE SORT ROUTINE
00130 ;
00140           ORG     65222
00150 LOOPA     LD      IY,DATA
00160           LD      B,0
00170           LD      C,0
00180 LOOPB     LD      A,(IY)
00190           CP      (IY+1)
00200           JP      C,ALPHA
00210           JP      Z,ALPHA
00220           LD      C,(IY+1)
00230           LD      (IY+1),A
00240           LD      (IY),C
00250           LD      C,1
00260 ALPHA     INC     IY
00270           DJNZ    LOOPB
00280           BIT     0,C
00290           JP      NZ,LOOPA
00300           RET
00310 DATA      DEFB    0
00320           END     65222
```

****** (48K ml)   Bubble Sort ******

TRS-80/SYSTEM-80

```
Start=FEC6  End=FEEF  Entry=FEC6


FEC6:  FD 21 EF FE 06 00 0E 00 FD 7E 00 FD BE 01 DA E5
FED6:  FE CA E5 FE FD 4E 01 FD 77 01 FD 71 00 0E 01 FD
FEE6:  23 10 E5 CB 41 C2 C6 FE C9 00
```

NEXT MONTH'S ISSUE

LOTTO OR POOLS — LII/16K

Here's your second chance to try to get rich quick! (Although we certainly aren't giving any guarantees.)

BACKGAMMON 2 — LII/16K

Can't find a partner for backgammon? Hone your skills with this BASIC version of the classic board game.

EDITOR — LII/16K — ML

This program enhances the Level II "Edit" function and also includes a lowercase driver with flashing cursor and key-repeat.

FROGLET — CoCo

Get your frog safely across the busy highway. If you can manage this, you then have to help him across the river by jumping onto logs and turtles. Another arcade classic makes it on to your CoCo.

SIMON — VZ200

Test your response time and musical ear with this simulation of the popular electronic game. Very good for young children and lots of fun.

MAILING LIST — VZ200

This simple mailing list program stores names and addresses on tape and prints them out. Useful for Club Secretaries or anyone who needs to keep a list of names and addresses.

APPLICATION FOR PUBLICATION OF A PROGRAM IN MICRO-80

Date ....

To **MICRO-80**
SOFTWARE DEPT.,
P.O. BOX 213,
GOODWOOD, S.A. 5034

Please consider the enclosed program for publication in MICRO-80.

Name ....

Adress ....

.... Postcode ....

*** CHECK LIST ***

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes separated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it, etc.

For system tapes, the start, end, and entry points, etc.

The changes or improvements that you think may improve it.

Please package securely — padabags are suggested — and enclose stamps or postage if you want your cassette or disk returned.

CASSETTE/DISK EDITION INDEX

The cassette edition of MICRO-80 contains all the applicable software listed each month, on cassette. For machine language programs copies of both the source and object file are provided. All programs are recorded twice. Level 1 programs can only be loaded into a Level 2 machine if the 'Level 1 in Level 2' program from the MICRO-80 Software Library — Vol. 1 is loaded first.

***Note:** System 80/Video Genie computers have had different tape-counters fitted at different times. The approximate start positions shown are correct for the very early System 80 without the volume control or level meter. They are probably incorrect for later machines. The rates for a cassette subscription are printed on the inside front cover of each issue of the magazine.*

The disk edition contains all applicable programs which can be executed from disk. Level 1 disk programs are saved in NEWDOS format. Users require the Level 1/CMD utility supplied with NEWDOS+ or NEWDOS 80 version 1.0 to run them.

VZ200 programs are not currently available on cassette or disk.

				Approx. Start Position		
Side 1	**Type**	**I.D.**	**Disk Filespec**	**CTR-41**	**CTR-80**	**System 80**
Touch Typing	LII/16K	"T"	TYPE/BAS	10	6	4
Touch Typing	LII/16K	"T"	TYPE/BAS	75	42	28
Obstacle	LII/16K	"O"	OBSTACLE/BAS	140	79	53
Obstacle	LII/16K	"O"	OBSTACLE/BAS	165	93	62
Track 80	LII/16K	"T"	TRACK/BAS	190	107	72
Track 80	LII/16K	"T"	TRACK/BAS	230	130	87
Sort Demo	LII/16K	"B"	BUBBLE 16/BAS	270	152	102
Sort Demo	LII/16K	"B"	BUBBLE 16/BAS	290	164	110
Sort Demo	48K Disk	"B"	BUBBLE/BAS	310	175	117
Sort Demo	48K Disk	"B"	BUBBLE/BAS	330	186	125
Sort Demo Mod	LII/16K	"M"	BUBBLEMD/BAS	350	198	132
Sort Demo Mod	LII/16K	"M"	BUBBLEMD/BAS	360	203	136
Sort Demo	16K ml	BUBBLE	BUBBLE 16/CMD	370	209	140
Sort Demo	16K ml	BUBBLE	BUBBLE 16/CMD	375	212	142
(Addresses — Start 7ED7, End 7F00, Entry 7ED7)						
Sort Demo	48K ml	BUBBLE	BUBBLE/CMD	380	215	144
Sort Demo	48K ml	BUBBLE	BUBBLE/CMD	385	218	146
(Addresses — Start FEC6, End FEEF, Entry FEC6)						
Sort Demo Src	EDTASM	BUBBLE	BUBBLE/EDT	390	220	148
Sort Demo Src	EDTASM	BUBBLE	BUBBLE/EDT	400	226	151
Side 2						
Lunar Lander	CoCo	LUNAR	—	10	6	3
Lunar Lander	CoCo	LUNAR	—	40	22	15
Latin Vocab	LI/4K	—	LATIN/LVI	70	39	26
Latin Vocab	LI/4K	—	LATIN/LV1	130	73	49

TO:
MICRO-80, P.O. BOX 213, GOODWOOD,
SOUTH AUSTRALIA. 5034.
Please RUSH to me the items shown below:

$ enclosed Date ....

.... 12 month subscription to MICRO-80
.... 12 month subs. to MICRO-80, plus the cassette edition
.... 12 month subs. to MICRO-80, plus the disc edition
.... The latest issue of MICRO-80 (see inside front cover for prices)

FOR TRS-80 ☐ 1 ☐ 2/16 ☐ 3 — KRAM ☐ TAPE ☐ DISK
SYSTEM 80 MARK ☐ 1 ☐ 11 — KRAM

DESCRIPTION	QTY	PRICE

TOTAL ENCLOSED WITH ORDER
☐ Cheque ☐ Bankcard ☐ Money Order
Bankcard Account Number

P/H	
Total	

Signature .... Exp. End ....

NAME ....

ADDRESS .... Postcode ....

*Post/Handling charge on all Software ordered — $4.00.

MOLYMERX

Australia's broadest range of software for TRS-80's and SYSTEM 80's

MOLYMERX has the Australian distribution rights for literally hundreds of top grade programs from American, Canadian and British publishers. From games to utilities, from DOS's to Databases, if it's top quality then MOLYMERX almost certainly has it.

Now, MOLYMERX is being distributed in Australia by MICRO-80. To help you chose from the incredibly wide range of programs available, you may purchase a MOLYMERX catalogue. For only $5.0(you receive over 80 pages of what is virtually an encyclopedia of '80 software plus regular updates for 12 months. The useful information contained in this catalogue is worth many times its cost.

There are now generous BULK BUYING DISCOUNTS of 10% off list price for single orders in excess of $500 or 15% for single orders in excess of $1,000. So get together with your friends or User Group members to place a combined order and save yourselves real $$$.

EXPANSION INTERFACES FOR SYSTEM 80 and TRS-80 COMPUTERS

MICRO-80's new family of expansion interfaces for the System 80 and TRS-80 offer unprecendented features and reliability including:

Up to 32K STATIC RAM : to ensure high noise immunity and reliability

Centronics Printer Port: The Systems 80 Expansion Interface has a double-decoded port to respond to both port FD and memory address 37E8H, thus overcoming one of the major incompatabilities with the TRS-80.

RS232 Communications Port: for communicating via modem or direct link to other computers

Single Density Disk Controller: for complete compatability with all Disk Operating Systems

Supports double-sided Disk Drives up to 80 tracks: with a suitable disk operating system such as DOSPLUS, NEWDOS 80 or LDOS, the interface will support single or double sided drives of 35-80 track capacity.

Economical double density: an economical, high quality double-density upgrade will be released shortly to enable you to increase the capacity of your disk drives by 80%.

Real time clock interrupt: provides software clock facility used by most DOS's.

SYSTEM-80 EXPANSION IN/FACE		TRS-80 EXPANSION INTERFACE	
WITH ØK RAM	$450.00	WITH ØK RAM	$450.00
ADDITIONAL 16K RAM	99.00	ADDITIONAL 16K RAM	99.00
ADDITIONAL 32K RAM	198.00	ADDITIONAL 32K RAM	198.00

SYSTEM 80 AND TRS-80 PRINTER INTERFACES $99 + $3.00 p&p

For those who wish to add a printer to their SYSTEM 80. MICRO-80's new printer interface provides the ideal solution. Double-decoded to both port FD and address 37E8H, this interface overcomes one of the major incompatabilities between the SYSTEM 80 and the TRS-80. Price includes a Centronics printer cable. Operates with Centronics compatible printers including GP-80 and GP-100.

MICRO-80

LEVEL 2 ROM

ASSEMBLY LANGUAGE TOOLKIT

by Edwin Paay

FOR TRS-80 MODEL 1, MODEL 3 AND SYSTEM 80/VIDEO GENIE

This is a new package consisting of two invaluable components:

- **A ROM REFERENCE** Manual which catalogues, describes and cross-references the useful and usable ROM routines which you can incorporate into your own machine language or BASIC programs.
- **DBUG**, a machine language disassembling debugging program to speed up the development of your own machine language programs. DBUG is distributed on a cassette and may used from disk or cassette.

Part 1 of the ROM REFERENCE manual gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements etc. It also describes the various formats used for BASIC, System and Editor/Assembly tapes. There is a special section devoted to those additional routines in the TRS-80 Model 3 ROM. This is the first time this information has been made available, anywhere. Differences between the System 80/Video Genie are also described. Part 1 is organised into subject specific tables so that you can quickly locate all the routines to carry out a given function and then choose the one which meets your requirements.

Part 2 gives detailed information about each of the routines in the order in which they appear in the ROM. It describes their functions, explains how to use them in your own machine language programs and notes the effect of each on the various Z80 registers.

Part 2 also details the contents of system RAM and shows you how to intercept BASIC routines. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory — the only restriction is your own imagination!

The Appendices contain sample programmes which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

DBUG: Eddy Paay was not satisfied with any of the commercially available debugging programs, so he developed his own. DBUG: allows you to single-step through your program; has a disassembler which disassembles the next instruction before executing it or allows you to bypass execution and pass on through the program, disassembling as you go; displays/edits memory in Hex or ASCII; allows Register editing; has the ability to read and write System tapes and all this on the bottom 3 lines of your screen, thus freeing the rest of the screen for program displays. Four versions of DBUG are included in the package to cope with different memory sizes.

The best news of all is the price. The complete Level 2 ROM ASSEMBLY LANGUAGE TOOLKIT is only:

— Aus. $29.95 + $2.00 p&p
— UK £18.00 + £1.00 p&p

SPECIAL OFFER TO OWNERS OF THE LEVEL II ROM REFERENCE MANUAL ...
UPGRADE TO THIS ASSEMBLY LANGUAGE TOOKIT FOR ONLY $19.95!
Send back your original Level II ROM Reference Manual plus a cheque, money order or Bankcard authorisation for $19.95 plus $2.00 p&p and we will send you the new ASSEMBLY LANGUAGE TOOLKIT

MICRO-80