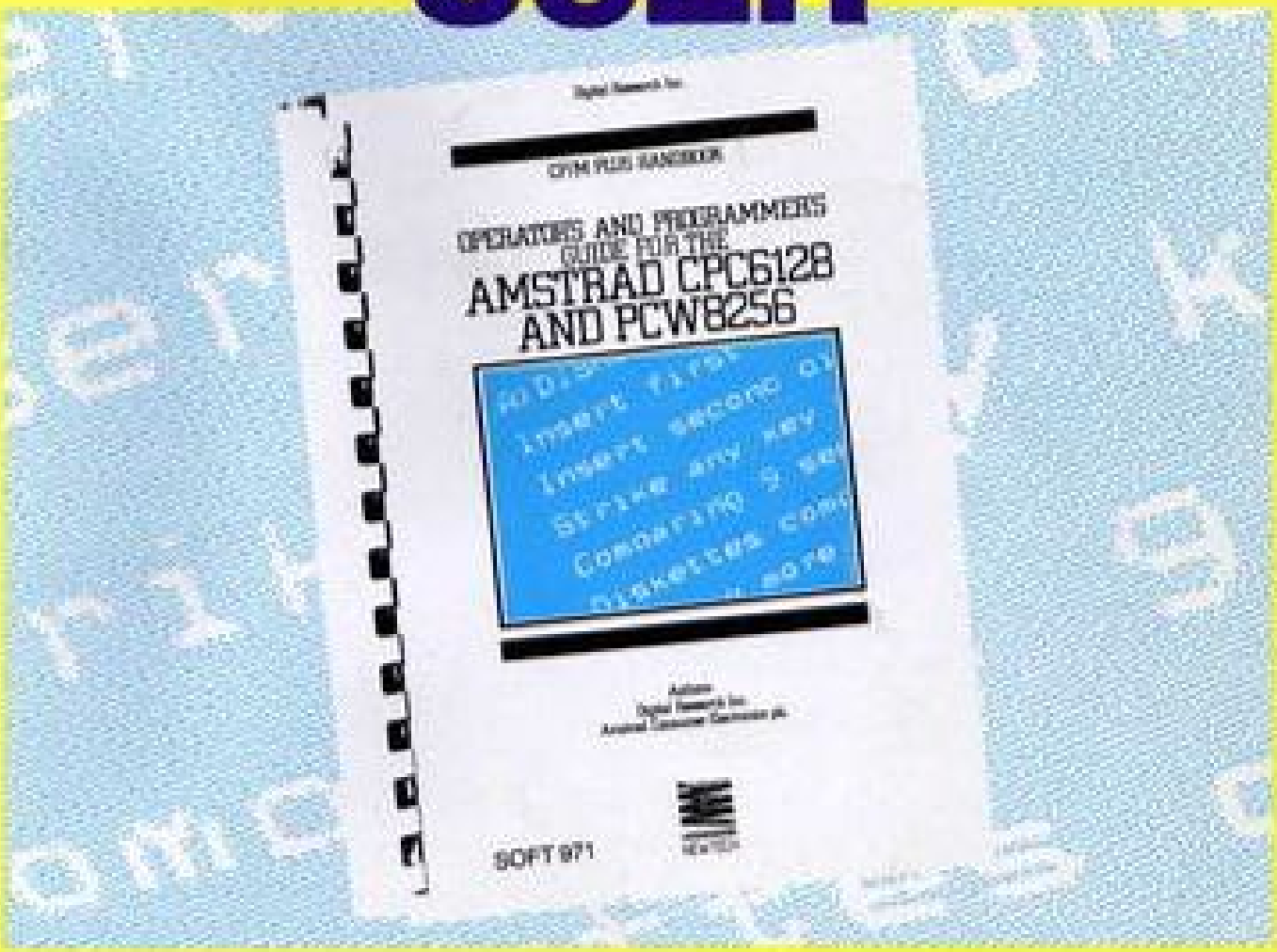
The Aussie Mag for Amstrad owners

THE AMSTRAD USER

Issue No. 21 $3.50 October 1986



- *REVIEWS on the new Digital Research CP/M Plus Manual, two PASCAL packages and CAMBASE*
- *A sneak preview of the new Amstrad AIRO*
- *Three listings for 464/664/6128 hackers*
- *USER GROUP INFORMATION*

FOR THE NOVICE & EXPERIENCED USER

Registered by Australia Post – Publication No VBP7017

THE AMSTRAD USER

Serious Side

Issue No. 21
October 1986

For Tape subscribers, the programs can be found at the following approximate positions:
Side 1: PETRLOAD - 14, PETRCALC - 25 Side 2: FIVEROW - 14, TWISTCUB - 50

All enquiries and contacts concerning this Publication should be made in the first instance by writing to The Amstrad User, Suite 1, 245 Springvale Road, Glen Waverley, Victoria 3150, Australia. Urgent matters can be phoned through on (03) 233 9661.
The Amstrad User is normally published on the first working day of each month.

The single copy price of $3.50 is the recommended retail price only.
The subscription rate (for Australia) is $35.00 for 12 issues of the magazine only, or $75.00 for 12 issues of the magazine plus tape containing programs appearing in that issue. Postage is included in the above prices. For subscriptions to New Zealand, PNG, Solomon Islands or Vanuatu please add $21 airmail. Other overseas prices available upon application.

Please note that whilst every effort is made to ensure the accuracy of all features and listings herein, we cannot accept any liability whatsoever for any mistakes or misprints.
Contributions are welcome from readers or other interested parties. In most circumstances the following payments will apply to published material: Letters $5.00, Cartoon $5.00 and a rate of $10.00 per page for programs, articles etc. Contibutions will not be returned unless specifically requested coupled with a suitable stamped and return addressed padded bag (for tapes or discs).

The Amstrad User is an independent Australian magazine and is not affiliated in any way with Amstrad or their Australian distributors Mitsubishi Electric AWA Pty Ltd., or any other dealer in either software or hardware.

THE AMSTRAD USER

G'day,

The new Amstrad AIRO was released in the UK at the beginning of September, in fact as I write, only a few days ago, which will give you some idea of how far ahead we have to finalise your magazine. It is an exciting step into IBM compatibility and quite different from the previous range of machines to come off the Amstrad production line. An overview is provided on Page 18 and naturally we will be giving it a more critical look when the unit actually reaches this country.

You'll also find details of the next two Year Discs (Page 50) which brings the software available on disc up to Issue 20. We have included some Public Domain software for you to try and if the response is good and the disc space permits, we will endeavour to provide more on future discs. Please remember, we didn't produce the software so it carries no guarantees and we cannot provide any support on problems you may encounter.

On average, about $100 a day is spent at the Post Office mailing back copies, tapes, books and other items available through the offices of The Amstrad User. Of course, if we were in the conglomerated press league this amount would probably not cover our daily coffee and biscuits bill - but we're not - and so it's 'big bickies' to us (or rather to the Post Office). It will come as no surprise that the latest increases from the big P have hurt!

Let me give you some examples. The cost of mailing a 150 gms item interstate is now $1 and to send a 300 gms item is $1.90. (It used to be 90¢ and $1.55 respectively). How the Post Office justify a 37.5% increase on the 300 gms item and get away with it is beyond me. This mainly effects the back copies where up to now we have only asked for a nominal 50¢ towards the postage costs. Unfortunately we must now charge a more realistic amount. I am sure everyone will be delighted with this news and will be rushing to send a letter of thanks to the Post Office - but not many, I suspect, will be putting stamps on the envelopes.

See you next month,

Ed

Letters

All correspondence published in this section earns a payment of five dollars.
Letters should be addressed to The Editor, The Amstrad User, Suite 1, 245 Springvale Road, Glen Waverley, Victoria 3150.
We regret that we cannot enter into any personal correspondence.

When I reviewed the DR DRAW package on my PCW8256 (TAU September 1986) I had a fairly tight time constraint, and I did not have time to explore the possibilities of overcoming some of the difficulties mentioned in my review. Since then I have found a way of avoiding the wrist-fatiguing disc-swapping that I was so critical of.

First let me say that the instruction manual did say that picture files should be stored on Side B of the working disc, and later transferred to another storage disc. The reason I chose not to do this was that there is only 35k left on Side B, and I was able to quickly fill this with a simple diagram and a sprinkling of text. The program would then crash requiring a re-load of DR DRAW. As there is 103k available on Side A of the working disc it seemed appropriate to use this side for picture file storage, and it certainly overcame the crashing problem due to a full disc. It did, however, introduce the problem of having to swap the disc over after every dozen letters, each time the picture was redrawn.

I have now found a way around the first problem without causing the second problem. The solution requires relocation of files on the working disc and an alteration to the DRAW.SUB file on Side A of the working disc. The procedure is generally as follows:

1. Having made your working disc as per instructions, place the CP/M system disc in the computer and reset by pressing SHIFT + EXTRA + EXIT.

2. Type *pip m:=a:pip.com* - this transfers the PIP utility into the computer. Type *pip m:=a:erase .com* - this transfers the ERASE utility into the computer for use in step 6 later on.

3. Place Side B of the working disc in the drive and type *m:* - this moves you into drive M.

4. Type *pip m:=a:draw.** - this transfers all the DRAW files into the computer.

5. Place Side A of the working disc in the drive and type *pip a:=m:draw.** - this transfers the DRAW files onto Side A of the working disc.

6. Place Side B of the working disc in the drive and type *erase a: draw.** - this should clear all the DRAW files from Side B of the working disc.

7. Place the system disc back in the drive and restore the system as in step 1. Now type *submit rped* and when the computer had loaded this editing utility, select *existing file* and when asked for the filename, type *DRAW.SUB*, place Side A of the working disc in the drive and press RETURN.

8. You are now able to edit the DRAW.SUB file. The line which needs changing is the line which reads *<m:=b:dr*.** - this should be changed to read *<m:a:dr*.00**. Now press EXIT and you have completed the modification.

When running DR DRAW you now enter your picture filename as *b:filename* and you will have 108k to store them in. It is still

advisable to transfer them to another file or storage, and back again when you want to do further work on them.

It should also be possible to achieve the same end result by editing two of the files on the DR DRAW release disc, or rather your *copy* of the release disc. The files that need editing are MAKE8256.SUB and DRAW8256.SUB. The alterations required are very simple. On MAKE8256.SUB the change is in the text line after *pip m:=a:draw.** where the instruction 'now insert SIDE B .. .etc.' should be changed to 'now insert SIDE A ...etc.'. The changes to DRAW8256.SUB are exactly as described in item 8 above.

Arnold Goldman, Dandenong, Vic

I would like to know if anybody has sorted out the bugs in the game "AIRWOLF" or has Mr. Gallagher mastered the game (March 1986), or if there are any more instructions, or where I can exchange it for one that can be played, or should I just say goodbye to my $29.95?

I own a 6128 now (I started with a 464) and the tape has been tried on other Amstrads just to make sure that Arnold didn't have a headache or something. I have no trouble loading tapes now that I have connected a Sanyo data recorder DR201, and so far every program I have bought loads without the problems I first had after the salesman sold me just an ordinary cassette player.

M.J. Viner, Nambour, QLD

I recently purchased from you the disc called "The Amstrad User - Issues 1 to 12", and I am very disappointed by the number of errors contained within. Eleven of the programs will not run for errors, there are three duplicates, five contain no instructions on how to use them and one has a spelling error on the paper catalogue. I hope you will at least tell me how to correct the errors and possibly assist in understanding the other five. The following are the erroneous programs:

ANJUMBLE	Syntax error in 170
SUPPRESS	Ready
CLOCKPT2	Unexpected WEND in 2780
MUSLST2	Line does not exist in 10
FINDUTIL	Memory full in 30 or Bad command
MAP-DRAW	Syntax error in 260
MCODE1	Break in 20
MCODE2	Break in 99
DIYTYPER	Does not run
OWLFABET	Syntax error in 30
SORTS	Blank screen

The following are the programs with no explanation of their function:

SHORTWP	No instructions
RSXGEN	What is meant by "Start address of Version 1"?
MAP-CODE	No instructions
MATHTABL	Not printing and does not list
BUZZLINE	Cannot operate with keyboard

The duplicates are as follows:

GRAFPLOT and GRAFLOAD
SPACELDR and SPACEXP
(Misspelt as SPACEEXP)
FFLOADER and FFORTS

Hoping that you will be able to supply the corrections and explanations for these faulty programs.

W.J. Tuffs, Epping, NSW

There is nothing wrong with the programs and there are certainly no duplicates. The problem is your 6128, or put another way, up to Issue 10 the programs in The Amstrad User and on the cassettes that accompanied each issue were written specifically for the CPC464 - after all, that was the only machine around at the time. Then followed in quick succession the CPC664 and CPC6128. Despite what Amstrad said about upwards compatibility, most 6128 owners quickly realised that there were some small modifications to be made to the older programs. (Since Issue 11, published programs have been checked on both the 464 and 6128).

The Year Disc (1 - 12) contains almost exact copies from the original tapes. (I say 'almost' because some obvious bugs have been fixed). Most programs have instructions within them (try LISTing a few), but some, especially utilities and machine code routines, will not make sense to a beginner. Explanations have appeared in the relevant issues of The Amstrad User and it is sensible to refer to them for more details, but to produce a compilation for the few recent 6128 owners who are having problems would most certainly be time and cost prohibitive.

For the benefit of other beginners, a brief list of answers follows showing the program name and the issue number in brackets:

ANJUMBLE(3) — *The 6128 does not like the text in brackets. Edit it out and repeat with the next four lines.*

SUPPRESS(2) — *Routine which resides in High Memory to suppress line feeds during printing.*

CLOCKPT2(11) — *This should be merged with CLOCKPT1.*

MUSLST2(10) — *This should be*

merged with MUSLST1.

FINDUTIL(4) *464 only*

MAP-DRAW(5) *Remove offending REM*

MCODE1(6) *Series of machine code routines to incorporate in your own programs.*

MCODE2(7) *Same as MCODE1.*

DIYTYPER(1) *Your machine should be reset before using.*

OWLFABET(4) *Remove text following instructions.*

SORTS(9) *Requires to output to printer.*

SHORTWP(1) *Instructions in the first two REM lines.*

RSXGEN(9) *Top of memory addresses used with machine code assembly program.*

MAP-CODE(5) *Loads machine code routine and calls MAP-DRAW.*

MATHTABL(4) *Requires "Y" to print, not "y".*

BUZZLINE(6) *Not designed to operate through keyboard.*

GRAFLOAD(8) *RUN this to automatically load GRAFPLOT.*

SPACELDR(12) *RUN this to automatically load SPACEXP.*

FFLOADER(11) *RUN this to automatically load FFORTS.*

To all Year Disc (1 - 12) purchasers only, try to get hold of a back copy if you get stuck, but in the last resort send us a SAE (100 x 230 mm approx.) and we will send you back a photocopy of the relevant article. Please be sensible though, we certainly will not entertain protracted copying exercises!

I would like some infinite lives, time and energy and, if possible, 'no drown' pokes for Sorcery (cassette). To anyone who has these pokes I would give a copy of Ghostbusters, Roland in Space, Rally 2 or Finders Keepers.

Also, if anyone has a copy of Yie Ar Kung Fu I would gladly give a copy of any of the above games for it. Please send a Blank cassette (longer than 15 minutes) for me to copy it onto and I will send you one to copy Yie Ar Kung Fu on to.

P.J. Schmidt, Loxton, SA

Obviously, Mr. Schmidt doesn't mind risking a two year jail sentence and/or a $50000 fine. He is also encouraging other people to risk the same penalties. When will people learn that copying is not only illegal but is seriously damaging the software market in Australia and making it more difficult for the genuine purchaser to obtain the software he/she wants? We are happy to publish the pokes if anyone has them, but don't bother asking for Mr. Schmidt's address.

After purchasing an Amstrad CPC464 in April '86, I suppose I could be regarded as a new "convert".

This is my first "hands-on" experience with a computer although I must confess to an ever present longing to get involved with this fabulous field of learning.

Being presently unemployed, I did a lot of homework as to the best value for money. The Amstrad range proved by far the clear choice.

I have to say that for a beginner or experienced user the Amstrad is a superb computer. The decision to invest in this machine is one I'm sure I'll never regret.

Hopefully my skill will grow sufficiently to do justice to this very versatile machine. Thank you Amstrad!

K. Rigby, Para-Hills, SA

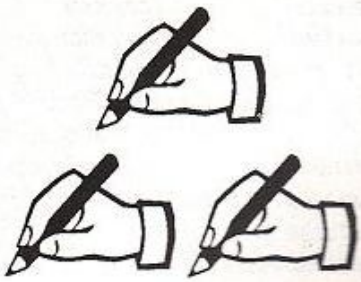
A warning about Amsoft's Cyrus Chess - it cheats! On a number of occasions now when using the technique of forcing the computer's king onto the edge of the board with, for example, two rooks, the computer has cheated. At the 'mate in one' position it takes its move, then replaces the king to its original position, and swaps sides. I mean how underhanded can you get? Has anyone else had this problem?

Now I will sing the praises of Arnor for their Protext word processor, on which this letter was produced. The word processor is fast, easy to use, and very well thought out. I would recommend it to anybody, from beginner to expert. I use the ROM-based version, which also has the advantage of making the disc drive on my 464 easier to use; in fact I now switch on the word processor to do most of my disc editing since you don't need to use the awful procedure of defining a string for the file name, you just type it straight out!

Paul Tansom, Portsmouth

Could you enquire through your pages if any reader has got hold of a Printed Circuit Design CAD program for Amstrad computers (or CP/M). If so, would they please contact me by phone on (02) 883143 or write to me at Dept. of Psychology, A19 Sydney University, NSW 2006.

Raja Vijayenthiran, President - Sydney Amstrad Computer Club



User Group Contact List

Please note that the following names are listed as contacts for new user groups and should NOT be viewed as a problem solving service.

See also Nationwide User Groups list on previous pages.

NSW		
Chris Craven	Canowindra	(063) 44 1150
Trevor Farrell	Coolah/Mudgee area	(063) 77 1374
T.J. Webb	Glossodia	(045) 76 5291
David Higgins	Inverell	(067) 22 1867
John Patterson	Lismore	(066) 21 3345
Paul Wilson	Moruya	(044) 74 3160
Frank Humphreys	Mummulgum	(066) 84 7290
Martin Clift	Narrabri	(067) 92 3077
Bob Hall	Newcastle	(049) 52 6915
Stephen Gribben	Singleton	(065) 72 2732
Ken Needs	St. Ives	(02) 449 5416
Chas Fletcher	Toongabbie	(02) 631 5037
Nick Bruin Snr.	Tweed Valley	(066) 79 3280
Vic		
Stuart McLean	4/304 Albert St. Sebastopol, 3356	
David Carbone	Burwood	(03) 29 4135
Rod Anderson	Camperdown	(055) 93 2262
Paul Walker	Heathmont	(03) 729 8657
Terry Dovey	Horsham	(053) 82 3353
Andrew Portbury	Leongatha	(056) 62 3694
Sue Kelly	Manangatang	(050) 35 1402
M.G. Donaldson	Morwell	(051) 34 5711
Angela Evans	Mt. Evelyn	(03) 736 1852
Keith McFadden	Numurkah	(058) 62 2069
Lindsay Parker	Wandin North	(059) 64 4837
Maureen Morgan	Warnambool	(055) 67 1140
QLD		
Debbie Topp	Bribie Island	(075) 48 1688
Steven Doyle	Caloundra	(071) 91 3147
Mick O'Regan	Gladstone	(079) 79 2548
Kylie Telford	Goondiwindi	(076) 76 1746 (weekendsonly)
D.F. Read	Ingham	(077) 77 8576
Tim Takken	Ipswich	(07) 202 4039
Alan Laird	Maryborough	(071) 22 1982
R.C. Watterton	Toowoomba	(076) 35 4305
SA		
Lindsay Allen	Murray Bridge	(085) 32 2340
Michael Spurrier	Murray Bridge	(085) 32 6984
Mrs. S. Engler	Penola	(087) 36 6029
Rita Bascombe	Port Lincoln	(086) 82 1633
WA		
Dave Andersen	6 Kitchener Rd Merredin, 6415	
Graeme Worth	Scarborough	(09) 341 5211
P.M. Nuyens	Warcona	(095) 33 1179
TAS		
Conal McClure	Scottsdale	(003) 52 2514
NT		
G.P. Heron	Tiwi	(089) 27 8814

CHEAT MOD

Some Tips, Pokes and Game busting ploys to improve your scores. If you've got any - let's have 'em.

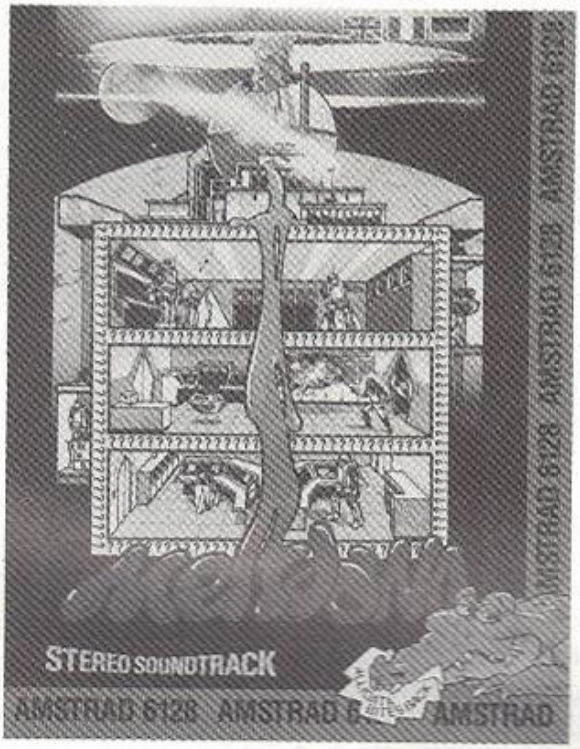
To input the majority of pokes, there are two methods.

Method 1: Type in the listing. Rewind the game tape. Type RUN followed by pressing enter. Follow on screen prompts to load the game.
Method 2: The tape header/loader/title screen, comprising usually of one or two data blocks has to be skipped. Rewind the game tape. Type in CAT followed by enter and play the tape. Watch the screen and a message will come up:
Found FILENAME block 1 OK
where FILENAME will be replaced by the name of the loading section. The next message that appears will be the main program appearing in the same style:
Found MAINFILE block 1 OK
where MAINFILE will be different for each game. Note the point on the tape counter at which this second file appears. Stop the tape and rewind to just before that point. Type in the program and RUN it. Then press play on the tape deck.

RADZONE

This poke is entered using Method 1 and gives you infinite lives.

```
10   FOR f=&BB48 TO &BB4D
20   READ a:POKE f,a:NEXT
30   DATA &21,&ee,&09,&36,&67,&c9
40   RUN"
```



ZOIDS

A tip for anyone trying to get into Martech's game of these plastic dinosaurs.
When you come to a city complex, call up the base and tell them to launch a missile on the power station. The zoid's heart beat will go up and you should now fire a short range missile at either the mine or the beacon. Repeat this procedure but go for the city domes.
To pick up extra power cells, destroy an installation and move to where it was standing. Turn on the scanner and any object around will show up on it and be automatically picked up.
When you're moving, the best tip is to go to the missile icon once you begin to move. That way there is a red zoid creeping towards you. You can despatch a short range missile to turn it into a pile of zoidar scrap.
It is easy to be destroyed by incoming red missiles. You should really spend a few games practising shooting against them. You can do this by switching to the gun icon and pressing fire. You're then in the gunnery practice mode where no missiles will harm you if they hit you.

Here are some general tips followed by some advice on how to complete the first six mini games on the disc version of Meltdown.

1. Always make sure you are standing still when firing or you will do a somersault. If you land on a cyberman you will die.
2. Don't stand still and let a cyberman come to you. Move into its line of fire and shoot it, because it will have less time to shoot you.
3. After releasing a bullet, sidestep to avoid incoming bullets.
4. In room which are cut in half by two storey computer stacks you can always shoot one of the stacks to make it explode. Only one will explode, but it's always the same one for a particular screen.
5. In the room with a row of cybermen, try to stand behind them because as they turn round you can shoot then easily and quickly.
6. In any room before the gamma radiation room you will find a stationary cyberman in the top right hand corner of the room. Touch it and the cyberman suit will transfer to you so that you can pass through the gamma radiation. On the other side you will be able to remove the suit in the same way you acquired it so that you can leave.

MINI GAMES

Terminal 1: Move the joystick diagonally for a second, then just fire until all nasties are killed.
Terminal 2: Give top priority to side missiles as they move twice as fast.
Terminal 3: Fire, then move slightly diagonally, then fire and repeat the process. This must be done to avoid shooting yourself.
Terminal 4: Line up with the top target and shoot when the gap is a few squares away, then move to avoid ricochets.
Terminal 5: Move to the top then come down in horizontal rows. By doing this you don't overlap or leave the screen. When you've got all the numbers the password is obtained.
Terminal 6: When you press the button to start the game, fire instantly and you will get the A. Then it just takes practice to get the word "ALIVE". Fire when the letter required is about 4 or 5 squares away.

Here are some general tips followed by some details on levels two and three for Imagine's 'stab-em-up' which should get you a lot further in the game.
1. Never stand at the very edge of a screen.
2. If you see a Commandant (blue uniform), position yourself at the bottom of the first ladder he will come to. If he is not already on the bottom level he will climb down and you can stab him to get his weapon.
3. Only Commandants and Rifle Carriers follow you around. To beat them off just climb a ladder, lie down and knife them as they come after you.
4. If the enemy is right behind you, run right. As the screen scrolls jump left over their heads. Face right and, if you have a weapon, blow them away.
5. If you have a weapon don't be miserly with it. If you are close to the right hand edge of the screen when you fire it, it will scroll with the screen and you get twice the killing power.
6. If you don't have a flamethrower at the end of the first level then lie down and stab the enemy.
7. Karate troops are a real pain. They are dressed in green and brown with black hats. Jump up or lie down and stab to kill them.
8. If you are on the 2nd, 3rd or 4th levels try to get on one of the higher platforms. Get two enemy troops running along below you (one will do, three is a gift). When you drop down follow them along since only three enemy troops appear at a time except at the end of a level. This way you will easily get to the end of a level.

Level 2: At the end of this you will come to a submarine. Lie down in the centre of the screen, or slightly left of centre, facing left. Four dogs will come after you. You must lie down and stab these because the rocket launcher is useless against them. When these four are dispatched turn right and, still lying down, kill another four dogs. Repeat this twice more.
Level 3: This is very tough. Take a step or two right, jump left over the mine and climb on the wall. If you see a Rifle Carrier throwing a grenade (you'll hear a warbling noise), jump at him. On this level it is vital that you use your bazooka when two men are ahead of you. When you come to the fence get up onto it as soon as possible because it's much easier up there. At the end you come up against a small helicopter that I've yet to beat.

CHUCKIE EGG

This tip speeds up the game (line 30) and removes the monster detection (line 40). You may find it possible to get 'stuck' in a falling loop at times, but you may be able to get out of this, or just abort the game currently in progress. Enter using Method 2 to get past the first block of the program.

```
10  OPENOUT"d":MEMORY
    1000
20  LOAD"!"
30  POKE 39557,201
40  POKE 39698,0:POKE
    39699,0
50  CALL 39575
```

TURBO ESPRIT

This provides infinite lives for the drug chasing, racing game. Enter using Method 2 to get past the TITLE block 1 (be careful since there is only a small gap between this and the section you want to load).

```
10  MODE 1:MEMORY 2015
20  FOR f=2016 TO 2027
30  READ a:POKE f,a:NEXT
40  DATA &21,0,&40,&11,&68,
    &42,&3e,&16,&cd,&a1,
    &bc,&c9
50  CALL 2016:CALL 32768
60  a=2017:POKE a,0:POKE
    a+1,8
70  POKE a+3,&54:POKE
    a+4,&a1
80  CALL 2016:POKE 34409,0:
    CALL 6496
```

JET SET WILLY

For the TSAM disc version of JSW, enter the following lines, type SAVE "CHEAT" and enter, then type RUN "CHEAT" and it will automatically load JSW with infinite lives.

```
10  MEMORY &2000
20  LOAD"jetset.sbf",&1260
30  FOR n=&9C40 TO &9C4B
40  READ m
50  POKE n,m
60  NEXT
70  DATA 33,96,18,17,96,2
80  DATA 1,134,128,237,
    176,201
90  CALL &9C40
100 POKE &57B7,0
110 CALL &4D65
```

Some tips for this Amsoft arcade adventure.

1. After collecting the grey key from the upstairs bathroom, collect the book from the adjoining master bedroom.
2. Take the book to the Library and then walk to the far left of the room, avoiding the phantom.
3. When safe, walk towards the right and jump, and you will enter the Secret Passageway.
4. Follow the passageway and you will enter the Armoury.
5. Collect the blue page, then climb the barrels to collect the shield at the top.
6. With the shield and the grey key go to the end of the battlements outside the Castle of Tombstowne. You will now be able to pass the flying arrow and enter the Hut to retrieve another blue page for the book of spells.

"GET DEXTER"

Here is a poke which gives Dexter infinite energy. It's entered using Method 1 and you should ignore the odd colours on the loading screen and any funny noises near the end of the load - they're unavoidable.

```
10  DATA 21,22,b2,06,c3,e5,c5,21,97,c2,e5
20  DATA c5,21,5e,02,06,cf,e5,c5,21,ff,bb
30  DATA 06,08,e5,c5,21,55,7b,3e,04,11,05
40  DATA 7c,c1,fd,e1,f5,cd,13,79,38,fb,f1
50  DATA 3d,20,f0,21,5e,7b,36,fa,21,4e,40
60  DATA 11,fb,7b,01,13,00,ed,b0,f3,21,55
70  DATA 7b,11,31,b9,01,c0,00,ed,b0,c3,31
80  DATA b9,21,16,b3,36,e2,23,36,b9,c3,12
90  DATA b3,21,d1,7e,36,00,c3,da,6b
100 MEMORY &3000
110 FOR x=&4000 TO &4060
120 READ a$
130 POKE x,VAL("&"+a$)
140 NEXT
150 LOAD""
160 POKE &7924,&c9
170 MODE 0
180 CALL &4000
```

See next page for map of "Get Dexter"

ALIEN BREAK IN

Using Method 1, these pokes will provide infinite lives.

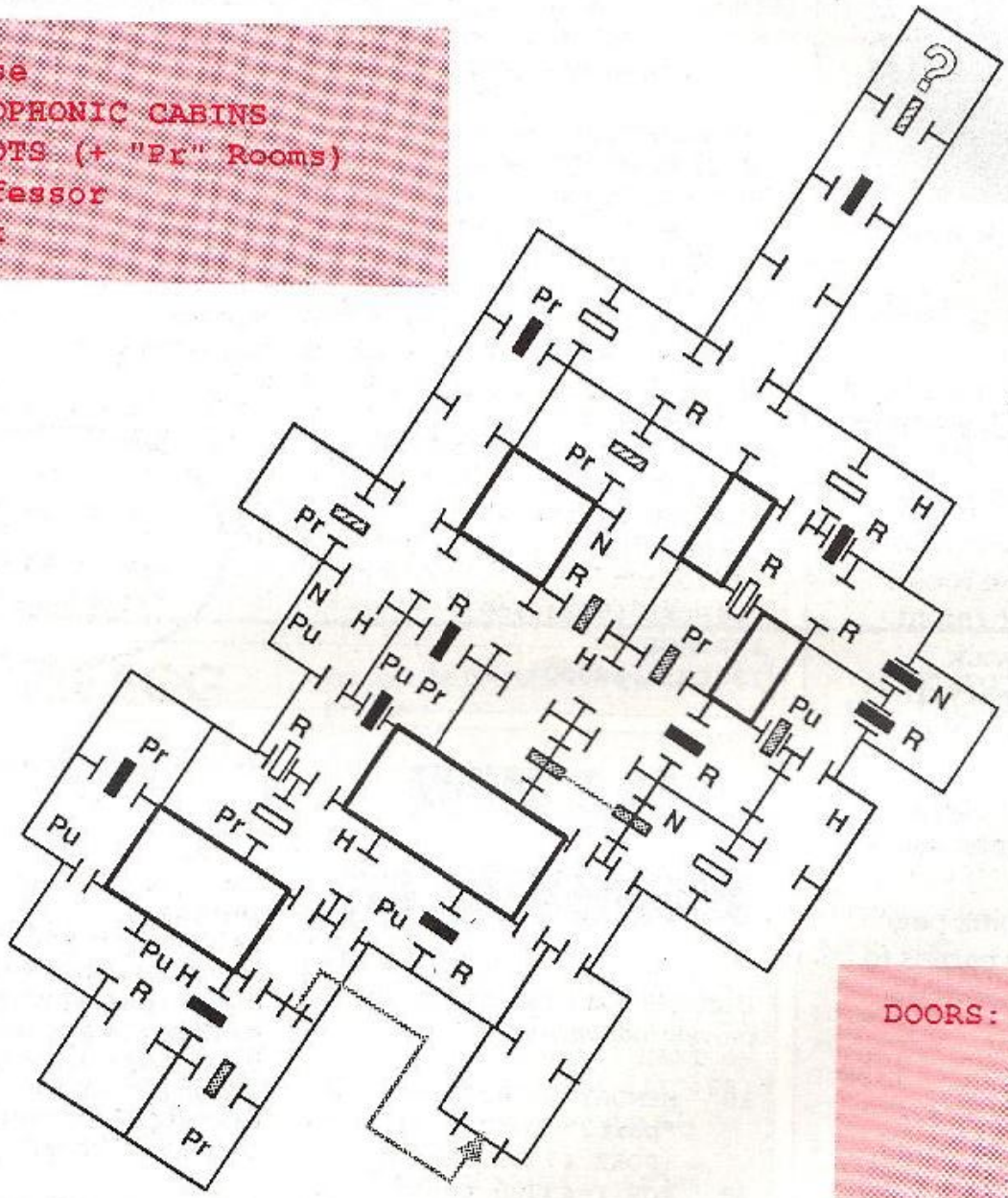
```
10   MEMORY &5500:LOAD
     "part2":POKE &740C,0
     :POKE &70C8,60
20   FOR f=&9000 to
     &9003:READ a:POKE
     f,a:NEXT:READ
     a,b:POKE a,b
30   CALL 29658
40   DATA 71,63,21,47,
```

We will endeavour to publish "Cheat Mode" for as long as possible, that means for as long as we are supplied with the tips and pokes.
This is where you come in, and to tempt you a little to pass your own game busting secrets to other players/readers of The Amstrad User, we are offering $5 for every published "cheat".

Send your revelations to:
Cheat Mode
The Amstrad User
Suite 1,
245 Springvale Road
Glen Waverley, Vic 3150

"GET DEXTER"

N Nurse
H HOLOPHONIC CABINS
R ROBOTS (+ "Pr" Rooms)
Pr Professor
Pu Punk



MAPPED BY MIKE SIMPSON

Amstrad's first IBM compatible

by Simon Anthony

Amstrad's newest computer, named the AIRO, is due for release in Australia in October (with supplies available in November) through Mitsubishi Electric AWA. It will be sold as a 'complete package' which means that the price tag will include the 512k computer with dual 360k floppy disc drive, keyboard, high resolution grey level or colour monitor, two button mouse and a number of exclusive Amstrad enhancements. Commenting on the new PC, Product Manager John Chandler said he is confident that with these enhancements, such as multi-tasking and multi-media flexibility and the inclusion as standard of a number of hardware features which are extras on all other PC's, the AIRO will quickly establish a significant position in the IBM compatible market. He added that the AIRO will set new benchmarks by which all IBM compatibles will be judged.

The 512k system, expandable to 640k includes serial and parallel interfaces with standard connectors. The 8086 processor can execute programs at more than twice the speed of the IBM PC. There will be a choice of one or two 360k 5.25" floppy discs. There will also be three 8-bit peripheral expansion slots for applications such as internal modem cards, hard discs and networking. Like other Amstrads, the power supply is connected to the monitor.

It has a full-size keyboard with illuminated Numlock and Capslock, extra Delete and Enter keys, dedicated ports for the joystick and the mouse which leaves the serial port free for printers or communicators, light pen connectors, socket for 8087 maths processor, or loud speaker with volume control.

The high resolution monitor (on a swivel base) is in either monochrome grey level or colour. In addition to the IBM compatible Alpha and graphics modes, a special high resolution colour graphics mode is provided as standard and allows 16 colours in 640 x 200 pixels.

Extensive operating software will include MS DOS 3.2, the latest version of the widely used operating system with enhanced PC DOS compatibility; DOS PLUS, which runs MS DOS and CP/M-86 applications. The AIRO DOS PLUS multi-tasking facility should be an invaluable time saver that allows two programs to be run at once. Either applications or utilities can run in the background. The mouse driven Window and Icon operating system is provided by Digital Research's GEM Desktop and GEM Paint.

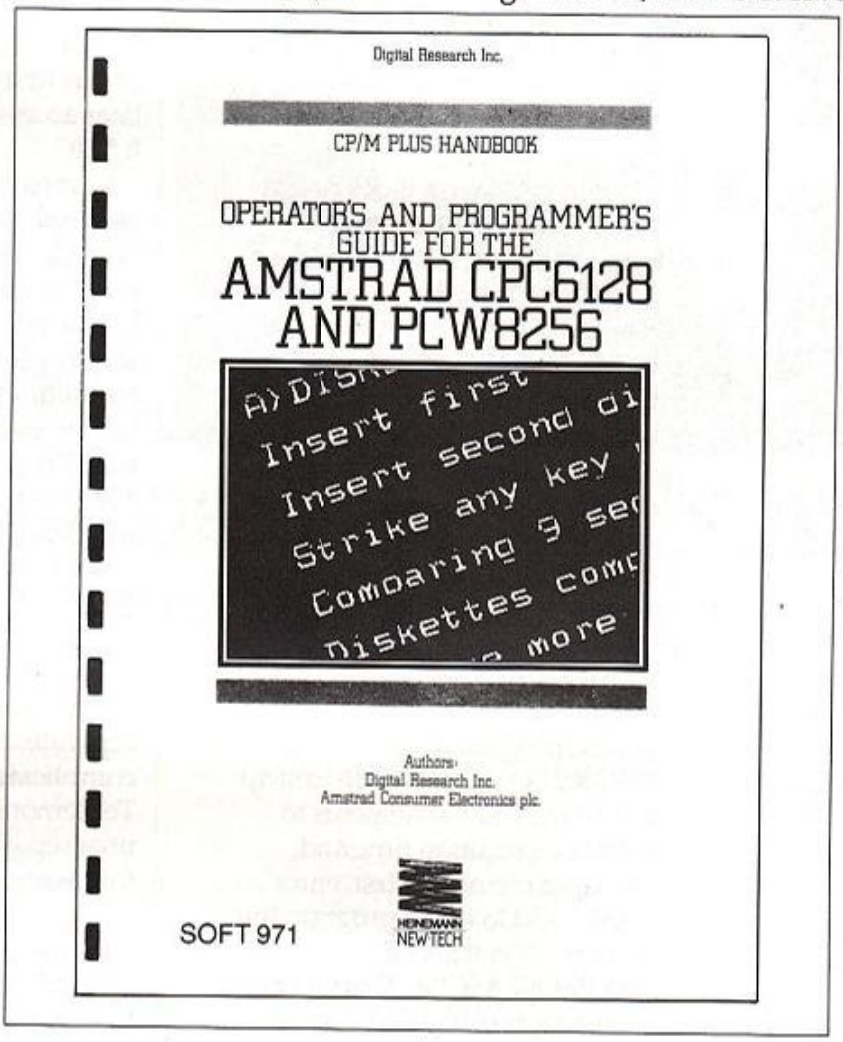
It runs Locomotive Basic 2, the fastest 16-bit Basic known, which provides the display of up to four 'windows' at one time, thus allowing the programmer to simultaneously keep track of the program listing and the effect the instructions are having in another part of the PC.

So there it is - a brief insight into what promises to be a major influence in the IBM compatible market.Naturally, we will be bringing you a much fuller review when we get hold of an AIRO, which reminds me, I would have thought with a name like AIRO that it should have had bubble memory!

The Digital Research CP/M Plus Manual

For CPC6128 and PCW machines

Let's get one thing clear before we start: this book is not for beginners. If you shy away from memory maps and assembly language you're unlikely to find it comprehensible, never mind actually useful. Quite simply, it's the only book you'll ever need on the technical side of CP/M Plus - and that means it's very technical indeed.



If you're a bit of a machine code buff and want to start delving in CP/M 2.2, there are some books available. If it's CP/M Plus you're out to gen up on, you've got much less choice. In fact, the only really helpful book up till now has been MML's The Amstrad CP/M Plus by Andrew Clarke and David Powys-Lybbe, which unfortunately has not been freely available in Oz, and one or two American imports. So how does Digital Research's official effort measure up?

In terms of presentation, the DR manual is a lot slicker. The typesetting is certainly much easier to read and that's a major advantage to my mind. In practical terms it loses out however, as its chunky ring-binding weakens its pages and the thin covers give it inadequate protection for a reference work.

As far as content goes, the Digital Research manual is really quite narrow. It concentrates on the structure of CP/M Plus in its Amstrad implementations, the workings of the DR/Amstrad utilities and the techniques used in writing applications of your own. This is all very theoretical stuff, and needs to be read slowly. MML's guide quite definitely set out (amongst other things) to teach beginners how to do useful things with CP/M Plus. It covered a wide variety of programming and hobbyist topics in a friendly, even chatty way.

In contrast, the DR book is formal and precise. Though it starts from basics, the emphasis is on the technicalities of CP/M Plus on the Amstrad machines rather than on how to do what you want to. Each section starts off with a brief summary of what lies ahead, and terms are carefully defined before they are used. The authors waste no time getting stuck into the nitty-gritty of BDOS entries. RSXs (which they explain extremely well) and the two Amstrad CP/M Plus implemen-tations. For the beginner this is of no use at all, but for the seasoned machine code programmer intent on learning to hack in CP/M Plus it's just about ideal.

It's partially a question of the sort of approach you like, but I found the writing style of the Digital Research book much more straightforward and easy to read than Messrs Clarke and Powys-Lybbe. To be sure, the subject matter is pretty complex stuff. All the more reason, to my mind, to set things out plainly as

Using ASCII files with Locoscript

by R.J. Webster

In all the reviews of the PCW, critics enthuse over the power of Locoscript, but lament over the fact that it uses a different data format which means that ASCII files cannot be loaded into the word processor. However, do not despair, there is no need to rush out and purchase a copy of Wordstar as ASCII files can be read and created using Locoscript quite simply.

The document READ.ME supplied on the Locoscript system disc gives details on how to convert Locoscript files to standard ASCII format or into a Page Image file suitable for printing on an alternative printer connected to the optional interface. It is quite a simple matter and is done by selecting f7=Modes from the disc manager screen.

When you save a block of text from a Locoscript document. Locoscript saves it as a standard ASCII file. Therefore it is easy to merge an existing ASCII file into a Locoscript document. Pressing f7=Modes in the text editing mode allows this.

Selecting the Insert text option returns you to the disc manager screen, you select the file to insert, and it will be inserted at the cursor position.

EXAMPLES

Suppose you have completed a BASIC program called MYPROG.BAS and wish to strip out all the REM statements to enhance execution time and conserve memory. First, enter BASIC and load the program into memory. Type SAVE "MYPROG.ASC",a. This will save your program in ASCII format. Now, boot your Locoscript disc and create an empty document MYPROG.LOC. Set up the file header for 12 pitch, width of 90 characters and no right justification. This will ensure that the program listing is easily printed, and utilises the full screen width. Now select f7=Modes and insert the file MYPROG.ASC. The listing can now be edited using the find and cut facilities of Locoscript. The insert operation may take some time for a lengthy file and can be speeded up if the source file (MYPROG.ASC) is first copied to drive M. When completed, exit to the disc manager. Now, select f7=Modes, make ASCII file and create a simple text file called MYPROG1.ASC on group0 (first group) of your disc. This file can later be loaded and executed from BASIC in the usual way by typing LOAD "MYPROG1.ASC",r. It is important that the file is created on the first group to enable it to be later accessed from CP/M or BASIC.

Alternatively, if you develop assembly language programs, and are sick and tired of using RPED or ED, you can write them in Locoscript after setting the tab stops appropriately. When you have completed your program, save it and create the simple text file MYPROG.ASM on group0. This can then be assembled by MAC or RMAC from CP/M in the usual fashion.

NOTE

Saving the ASCII file in a group other than 0 will save the file in a separate user group and complicates the retrieval process. To demonstrate how Locoscript uses separate user groups, try the following:

1. From CP/M, copy DIR.COM onto a Locoscript disc which has a number of files in different groups.
2. Type in DIR[FULL]. This will display all the files in Group0 (User 0).
3. Now type in DIR[FULL USER=ALL]. You will see a directory for each user group which corresponds with the separate groups used by Locoscript.

(SOFT 971 continued)
the DR manual does. There's no chat or waffle to get in the way; just detailed information. That said, you might well find it a little too clinical and prefer the MML manual's style - it's a matter of taste.
Good Points: All the details you need; Clear and concise style; Thorough explanation of techniques;
If DR don't know the technical ins and outs, who does?
Bad Points: $89.00 is certainly on the pricey side; Definitely not for beginners.

Programming Power with Pascal

Let's face it, BASIC isn't the most powerful computer language in the world. It might be friendly and docile, but it's very slow. If you're looking for a second language which is both useful and easy to learn, you could do a lot worse than PASCAL.

The difference between BASIC and PASCAL goes a lot deeper than the commands used. They are different kinds of languages: BASIC is an INTERPRETED language whereas PASCAL is COMPILED. To understand the difference we need to take a look at how your computer understands program commands.

All the computing that goes on in either Arnold or Joyce is done by a chip called the Z80. This is the 'brains' of your machine. When you want to program your machine - to tell it what to do, in other words - you have to speak to the Z80. Unfortunately the Z80 only understands a language called MACHINE CODE, and machine code is very difficult to learn. Machine code commands are just numbers, and it takes an awful lot of them to do anything impressive.

The alternative most people choose is to use a high-level language like BASIC or PASCAL. High-level language commands are much easier to understand than their machine-code equivalents, and tend to be based on English words or mathematical symbols. That's fine as far as we're concerned, but now our programs don't make any sense to the Z80 - because it can only cope with machine code commands.

This is where the high-level language earns its keep. It translates the high-level commands (which make sense to you) into machine code commands (which makes sense to the Z80). If this sounds like a pretty good way of keeping everybody happy, you'll see why high-level languages have caught on in such a big way.

COMPILERS AND INTERPRETERS

That's not quite all there is to it, though. This translation business can be done in one of two ways. Some languages are translated as they go along: the computer translates an instruction, performs it, then translates the next instruction and so on. This is called INTERPRETING, and the section of the language that does the translation is called the INTERPRETER. As I said earlier on BASIC is interpreted. This makes it a very friendly language, but it also makes it very slow.

The alternative system works like this: you translate the entire high-level program in one go, so that you end up with a pure machine code program. This process is called COMPILING, and the machine code program produced by it is called the COMPILED CODE. Because the compiled program is in machine code the Z80 can understand it without the need for an interpreter. PASCAL is a compiled language, and this makes it (potentially) very powerful and very fast.

USING PASCAL

If you're programming in PASCAL, the first thing you're going to want to create is your source code. That's the proper name for the actual text of your program - the PASCAL commands. For this you need an EDITOR, and I'm not talking about the magazine variety here.

Since BASIC generally comes with a built-in editor, it tends to get taken for granted. It's the thing that lets you type in, list and delete lines of your program. With PASCAL you normally use a separate text-editor - if you have WordStar or NewWord, for example, either of them would be fine if you use 'non-document mode'.

Once you've written your file using the editor and saved it to disk as, say, PROGRAM.PAS, you'll then want to compile it. To do this you have to run the compiler. If the compiler was called, say, COMPILE.COM then you type something like COMPILE PROGRAM as the A>prompt. Assuming you haven't made any mistakes the compiler then produces a new file called PROGRAM.COM. This is the compiled version of your program, and behaves just like any other .COM file. That is, you can run it simply by typing PROGRAM at the A> prompt.

The chances are that things won't go anywhere near so smoothly at first. PASCAL is very fussy, particularly about punctuation. If you make a mistake, it'll probably be spotted during compilation. The compiler will report

the kind of mistake you've made, but it's not always very precise. Often you'll get several different error messages all set off by one mistake. There's also a tendency for the messages to crop up some way after the actual error, which only adds to the general confusion.

P-CODE

Some implementations of PASCAL don't work in quite the way that's described above. Instead of converting your commands into Z80 machine code, they translate it into a special language called P-code. Then an interpreter translates this P-code once the program is running. There are theoretical advantages to this. In particular, the same program is usually much smaller in P-code than it would have been in machine code.

There are also several drawbacks to P-code compilation. A major problem is that P-code programs need the interpreter to be present when they run. If you want to turn a P-code program into a stand-alone program - something that can run without an interpreter - you'll need to put it through an extra processing stage. Turning a P-code program into a stand-alone .COM file usually makes it a great deal bigger. It could easily end up being larger than the file a normal compiler would have produced.

JENSEN & WIRTH

PASCAL is a highly standardised language - much more so than BASIC. A program written for one version of PASCAL should, in theory, need very little modification to compile on another.

For the purposes of standardisation, PASCAL implementors refer to a specification drawn up by JENSEN and WIRTH back in 1975. PASCAL packages usually boast about how accurately they conform to the appropriate standard, but it doesn't actually mean that much these days. From what it's worth, neither of these packages could compile the other's demonstration files without considerable modification.

OXFORD PASCAL
System Software (Oxford) Ltd
CP/M Plus (6128 and PCW's)

This is the cheapest of the packages covered here and also the most beginner-orientated; two facts that may not be unrelated. It comes in two different forms, a standard disk-resident compiler PAS.COM and a memory-resident compiler/editor/trace utility PASCAL.COM. The first of these works in pretty much the usual way - you type in PAS PROGRAM and it compiles the file PROGRAM.PAS into the P-code file PROGRAM.OBJ. You can run this as is with the interpreter RUN.COM, or turn it into a standard-alone program with LOCATE.COM. So much for PAS.COM, at least for the moment. PASCAL.COM is rather more unusual.

It's major point in favour of most BASIC systems that the editor and interpreter are MEMORY-RESIDENT - that is, you load them into memory at the start of a session and they're both on call there until you've finished programming. (I'm talking here about CP/M BASICs like Mallard and MBASIC - Locomotive BASIC doesn't even need to be loaded in.) If you write a BASIC program you can run it, find an error, edit it till it's correct and run it again, all within BASIC.COM.

In contrast, with most PASCAL systems you'd have to write your program with the editor, exit to CP/M, run the compiler and find the errors, exit to CP/M, enter the editor again and so on until you get it right. For beginners this is arduous to say the least - especially if you don't really know what the compiler's having problems with. What Oxford Pascal offers you in the form of PASCAL.COM is the BASIC-like ability to switch between editing, compiling and actually running your program.

It must be pointed out that the system has its drawbacks. The compiler is a cut-down version of PAS.COM, and still needs to access the disk for its error message file. It actually falls through into the system if you've not got that on the disk, but there is a handy utility called RECOVER.COM which lets you salvage any source code in memory.

More seriously, the PASCAL.COM editor is very unfriendly and quite poorly thought out. To edit a line you need to refer to it by number. But unlike BASIC, lines in PASCAL programs aren't given numbers by the programmer: they're automatically numbered as lines 1,2,3 and so on. When you list your source code by the clumsy '1,$p' command the line numbers aren't shown. You just have to count down till you reach the line you're interested in. This unfriendliness is a real shame. The editor has some quite powerful features, but it's such a hard slog getting to them I'd sooner use WordStar and forget about the memory-resident system altogether.

Turning to PAS.COM reveals other problems. The system as a whole uses a system of colour protection to avoid piracy. If you're using PASCAL.COM you only have to go through this rigmarole once in each session, but PAS.COM insists on checking your credentials every time.

If works like this: you're given a 40x26 grid with numbers up the side and letters along the bottom. In each cell of the grid is a coloured dot. Every time you run the compiler, it asks you for the colours of the dots in each of four cells of the grid. Get one wrong and it unceremoniously dumps you back into the operating

```
1
2 program bounce (input,output)
3 const thecowscomehome = false
4       DELAY = 30;
```

system.

The theory is that you can't reproduce the grid on a photocopier, so a pirate copy is unusable. In practice honest users get extremely annoyed at having to waste their time like this, get locked out of the system when they lose their grid or are stranded from the word go thanks to colour-blindness; while pirates get out their felt pens or trot off to the nearest copyshop that has a colour photocopier.

This really puts me right off using the package in a serious way - I mean, it's just too much effort to go through all that every time you want to compile something. And some enterprising hacker is bound to work out a poke to get around it....

Good Points: Nice price; Optional resident system; P-code system makes for short programs; Manual caters well for beginners.

Bad Points: Rotten built-in editor; Aggravating colour protection scheme; Needs an extra locating stage to get stand-alone programs.

PASCAL 80

Hisoft, CP/M 2.2 & Plus (All CPC's and PCW's)

Hisoft's Pascal 80 has long been the standard package to recommend. It's solid, dependable and very short on gimmickry. It also costs more than Oxford Pascal.

For your money you get the standard Hisoft editor ED80.COM and a disk-resident compiler HP80.COM. You also get a fair few other files, but we can take a look at these later on.

As usual, the first thing you'll be doing is writing your source code. You could use your favourite CP/M word processor/text editor for this, but ED80 will meet most people's needs handsomely. ED80's a full screen editor, and it uses almost the full set of WordStar control key combinations. This does strike me as a little strange - after all, if you know WordStar the chances are you've already got a copy of it, and won't need ED80. That said, ED80 only takes up 12K of disk and so is a lot more convenient on single-drive systems.

In use ED80 is straightforward and, for a CP/M editor, quite fast. It doesn't behave exactly like WordStar, but that's not going to bother too many people. It can be configured for the larger 8256/8512 screen without too much difficulty, and it's size makes it natural for the M:drive.

As for HP80, it's just a good practical compiler. You call it up from disk in the usual way, so that typing HP80 PROGRAM will compile the file PROGRAM.PAS into PROGRAM.COM. It's pretty fast, and quite fussy. A missed semi-colon, for example, will produce an avalanche of errors on the next line.

Pascal 80 would be the grand old man of Amstrad PASCALs if it wasn't for the way Hisoft keep adding to it. Since it was first launched it's gained not only random access filing but also a fascinating library of GSX graphics routines. Both of these come as PASCAL source code and are thoroughly documented. You can use them as procedures within your own programs, or read through them to study the techniques used - the latter being well worth the effort.

The smart ring-bound manual also contains support material for these files, plus a great deal of information on the Pascal 80 implementation. It doesn't cover PASCAL for beginners, as the authors are at pains to point out. You may be able to pick up what you need to know from the manual and some experimentation, but you're probably going to need a book of some sort to help you over the more difficult points. To this end, the manual's bibliography lists a few suitable introductory books.

This is all starting to add to the price of the system as far as beginners are concerned, but it does mean that the manual can give more experienced users the information they need. That's very important, and not just for hardened PASCAL fanatics. If you buy a package you're going to learn with, you don't want one you're going to outgrow too soon; nor do you want one that restricts you.

It may not be aimed at beginners, but I'd still recommend PASCAL 80 to anyone starting out. Beginners need the best and, for the price, I'd say that's PASCAL 80.

Good Points: Nice editor and good solid compiler; Thorough manual; Nice, well documented GSX routines; Compiles to true Z80 code; Runs under both CP/M systems; Takes up very little disk space.

Bad Point: Not too cheap; Manual doesn't cater for beginners.

Public Domain Software

from Peter Campbell

First 'Hardcat' and now 'Copycat', it is obvious that Amstrad users want to be able to catalogue their discs for quick reference, but, in some extent, they are re-inventing the wheel, for there is already a very useful program for cataloguing discs and it is a piece of public domain software.

So what is 'public domain software' you ask. Public domain software is software which you may use freely. Copyright is usually retained by the author, but often he/she makes the source code available so that you can improve upon his original effort, the one stipulation being that the resultant program cannot be sold, only given away.

In the early days of computing there was little commercial software, and public domain software grew out of the sharing of software written by those pioneering enthusiasts. To that software has been added early versions of programs destined for the commercial market. After the program has spent some time in the public domain, its faults ('bugs') come to light and users have either rectified them or suggested other improvements. The software can then be released in an enhanced version as a commercial product.

Some public domain software was donated by commercial organisations going into voluntary liquidation, some by individuals proud of their first programming efforts and some by universities in the United States. This last has become a legal requirement and has brought such items as 'E-Basic' (from the Micro-computer Laboratory, Naval Post-Graduate School, Monterery, California) and the 'Yale catalogue of Bright Stars' (!) into the public domain. Some is machine specific and some requires greater program area than the humble CPC464 (with DDI-1 disc drive and interface) and CPC664 computers can provide.

Many public domain programs come complete with documentation. This has not always been updated to reflect changes made in the latest version of the program. In other instances, documentation is virtually non-existent, or the user is referred to an obscure magazine reference, which means a visit to the library.

Public domain software varies enormously in quality and the would-be user must first sift the wheat from the chaff. He/she must then find a source which can supply the software in Amstrad's 3" format. Sources include bulletin boards, clubs and various commercial firms in the United Kingdom. Commercial firms in Australia also offer public domain software, but I do not know of any who offer the 3" format, but I should be happy to be proven wrong!

Given the problems that have to be overcome before you can obtain just the right program for your needs, you might well ask is it worth the effort. Yes, it is, particularly if someone else has done the necessary sifting! As I suggested at the beginning of this article, an excellent disc cataloguing program is just one of the many useful programs that are available. It is called 'Master catalogue'.

(Continued on Page 28)

To use Master catalogue, you place on a disc the following files:

MCAT41.COM CAT3
MAST.CAT

MCAT41.COM is the CP/M utility which does the cataloguing. It occupies 7k. CAT3 is a CP/M program which gives a tidy printout of the catalogue. It occupies 1k. MAST.CAT is an ASCII text file, which is produced by the cataloguing program. As it is updated a MAST.BAK file is produced. However, as 1100 entries will fit into 20k (40k with the .BAK file) a very large catalogue can be fitted into the remaining 161k of your system-formatted disc. Data formatted discs can be used for storing programs.

If you are not supplied with a configured MAST.CAT file, you must install your own. This is quite simple. All that is required is an ASCII text file containing a pair of brackets (), followed by a couple of line feeds. This can be produced on a word processor or you could even use the CP/M system's ED.COM. Optionally you can list between the brackets any programs which you intentionally have on several discs. If you do not so list them, the program will catalogue each occurrence and CAT3 will associate the duplications on one line. This allows you to decide if you need to erase any of the duplications, rename them, or include them within the brackets.

The remaining setting-up operation involves numbering (or naming) all your discs. On system discs, this is done using the CP/M SAVE command. SAVE 0 -1.001 will produce a file of zero length, having the catalogue name, 1.001 (the hyphen triggers the cataloguing process). On data-formatted discs you can create a Basic file consisting of a single remark or create a 0k file on a system disc and copy it across to the data disc. The latter is preferable as the former will create a 1k file. Don't forget to erase the 0k file on the system disc after you have copied, or you will have two numbers on the one disc!

The advantage of Master catalogue over some other disc cataloguing systems is that it produces the catalogue on disc, rather than paper. It is thus not necessary to print the catalogue out every time that you save a program, or erase one, on a particular disc. When you do require a hard copy, just type CAT3 <CTRL P>. (Fig.1 shows the screen output during the cataloguing process and Fig.2 shows a portion of the CAT3 printout.)

Other public domain utilities that will work on the CPC464 include file managers (such as NewSweep), directory sorters (D and NI), file comparers (Compar and CRCK44) and the 'disc surgeon' (DU). There are even utilities to find and isolate bad sectors on a disc (Findbad), to cou words in a text file (Wcount), to check the spelling (Spell-11 with Spell-Ed and Master.Lex) and to protect your files from

```
CATALOG PGM v4.0 - 11/20/83
for help,  reboot then type: MCAT40 ?

The DISK TO BE CATALOGED and the CATALOG SYSTEM DISK
will alternately be placed in Drive A:

Mount DISK TO BE CATALOGED in Drive A: Ready? (Y/N): y

                    Processing disk: 1.029

Mount CATALOG SYSTEM DISK  in Drive A: Ready? (Y/N): y
ADD: +++45K.FRE
ADD: ACUG.
ADD: COMPARE.COM
ADD: COMPARE.DOC
ADD: CPMII.
ADD: D.COM
ADD: DISPLAY.COM
ADD: FILEMAN.COM
ADD: FILEMAN.DOC
ADD: FIND.COM
ADD: FINDBAD.COM
ADD: IDUMP.COM
ADD: LISTT.COM
ADD: LOG.
ADD: SECTORED.COM
ADD: SECTORED.DOC
ADD: TYPWRITE.COM
ADD: UNERASE.COM

MAST.CAT has   24 entries,  with disk: 1.029

Mount DISK TO BE CATALOGED in Drive A: Ready? (Y/N): y

                    Processing disk: 2.029

Mount CATALOG SYSTEM DISK  in Drive A: Ready? (Y/N):
ADD: +++46K.FRE
ADD: ACE.COM
ADD: ACE.DOC
ADD: ACE3.COM
ADD: D.COM
ADD: KEYDEF.DAT
ADD: KEYSA128.ACE
ADD: KEYS8256.ACE
ADD: LASM.COM
ADD: LASM.DOC
ADD: LOMC.COM
ADD: LOOK.COM
ADD: LOOKAT.COM
```

Figure 1

unauthorised access (Password and Scramble). Assemblers and a disassembler are also available.

Amstrad disc space is allocate in 1k blocks. If a file does not completely fill a block, the remaining sectors of the block are wasted. LU, a library utility pushes the files in the library closer together by starting the next file in the next available sector. Another space saver, which works best on text files is Squeeze. Savings of around 40% are possible, which reduces the amount of disc space copies of correspondence, and the like, need. Naturally, there is Unsqueeze to reverse the process. There are also utilities which can produce a screen or printer copy of library and squeezed files without the necessity of taking them out of the library and unsqueezing them first.

Ever ERAse a file and immediately afterwards realise that you have made a mistake? You need Unera190, which can undo the damage at the touch of a key. Want to set up your keyboard so that you can summon up CP/M commands at the touch of a couple of keys? Have a look at Qwikkey.

On the lighter side there are banner makers, a calendar generator, a bio-rhythm chartmaker, chess, golf and Othello. I particularly like Othello because you can handicap the computer!

There are many Basic programs available in the public domain. However, they are mostly written in versions of Basic that will need adapting before they run on the CPC's, although some may run on the PCW's under Mallard Basic. Make sure you get an ASCII text of the source code (to edit into Amstrad Basic) before investing you hard earned cash, for 'free' public domain programs do cost money.

Although the programs themselves are free, discs, postage and copying are not and you will have to pay any costs involved. If you download programs from a bulletin board, you need an interface, a modem, a telephone and the ability to pay large telephone bills. However, the software for doing the job does not cost anything if you get it from the public domain!

If any readers are interested in getting any of the programs mentioned, they should write to me C/- Southern Tasmania Amstrad Club, PO Box 247, North Hobart, TAS 7002 and I will send them a list of what is available and what costs need to be met.

```
++++0K.FRE - 1        .026,
+++46K  .FRE - 2        .029,
+++65K  .FRE - 1        .029,
++145K  .FRE - 1        .009,
ACE     .COM - 2        .029,
ACE     .DOC - 2        .029,
ACE2    .COM - 2        .029,
ACUG    .    - 1        .029,
CAT3    .COM - 1        .009,
COMPARE .COM - 1        .029,
COMPARE .DOC - 1        .029,
CPMII   .    - 1        .029,
D       .COM - 1        .026,  1        .029,  2        .029,
DISK76  .COM - 1        .026,
DISPLAY .COM - 1        .026,  1        .029,
ED      .COM - 1        .009,
FILEMAN .COM - 1        .029,
FILEMAN .DOC - 1        .029,
FIND    .COM - 1        .029,
FINDBAD .COM - 1        .026,  1        .029,
GOTHIC  .COM - 1        .026,
HELP    .COM - 1        .026,
I/O-CAP .COM - 1        .026,
IDUMP   .COM - 1        .026,  1        .029,
KEYDEF  .DAT - 2        .029,
KEYS6128.ACE - 2        .029,
KEYS8256.ACE - 2        .029,
LASM    .COM - 2        .029,
LASM    .DOC - 2        .029,
LDIR    .COM - 1        .026,
LISTT   .COM - 1        .026,  1        .029,
LOG     .    - 1        .029,
LOHD    .COM - 2        .029,
LOOK    .COM - 2        .029,
LOOKAT  .COM - 2        .029,
LRUN    .COM - 1        .026,
LU      .COM - 1        .026,
LU      .DOC - 1        .026,
M       .COM - 1        .009,
MAKE    .COM - 1        .026,
MAST    .BAK - 1        .009,
MAST    .CAT - 1        .009,  1        .026,
MAZE    .COM - 1        .026,
MCAT41  .COM - 1        .026,
MCAT41  .DOC - 1        .026,
NDDT    .COM - 2        .029,
NDDT    .DOC - 2        .029,
NI      .COM - 1        .026,
NSWP205 .COM - 1        .026,
NSWP205 .DOC - 1        .026,
```

Figure 2

To get a 'taste' of Public Domain software, the following free programs appear on Year Discs 2 and 3 (see Page 50):

SD.COM
SD.DOC
SQ.COM
SQUEEZE.COM
USQ.COM
UNSPOOL.COM
UNSPOOL.DOC
WCOUNT.COM
WCOUNT.DOC

Petrol Consumption and Log Program

by Mike Perr

Like many programs, this one started as a simple idea (a ten liner) and just grew from that. And with Mr. Keating's demands upon the motorist to keep strict records, the program may well help, apart from identifying how much money your car consumes in fuel and servicing!

You'll find the instructions within the program, but basically it calculates petrol used in miles per gallon or litres per 100 kilometres. The number of entries is determined when the program is first run and is only limited by the available memory. Service information may be entered at any stage without effecting the calculations. A blank form may be printed to keep in the car to record information for later entry into the computer.

The program itself is preceded by an optional loader.

HOW IT WORKS

10 - 60	Dimension Variables
80 - 270	Menu
280 - 560	Enter Data
570 - 980	Discrepancy correction
990 - 1120	Calculate
1130 - 1310	Display result of last entry
1320 - 1620	Screen Display
1630 - 1960	Data Alteration
1970 - 2190	Save to cassette or Disc
2200 - 2410	Retrieve from cassette or Disc
2420 - 2610	Instructions
2620 - 2770	Print blank form
2780 - 2900	Service information

```
10 'Petrol calculator
    by: Mike Perry (1986)
20 'for lines 40 to 160:- key def 84,0,2
07:key def 85,0,220:key def 57,0,221:key
 def 56,0,222:key def 49,0,223
30 MODE 1:BORDER 0:PAPER 0:INK 0,0:PEN 1
:INK 1,0:CLS
40 PRINT:PRINT
50 PRINT"      00_  000  000  00_  ^0
0"
60 PRINT"      0 0  0    0    0 0  0 0
0"
70 PRINT"      00\  000  0    000  0 0
0"
80 PRINT"      0    0    0    0 0  0 0
0"
90 PRINT"      0    000  0    0 0  10\
000"
100 PRINT:PRINT
110 PRINT"^00 ^0_ 0   ^00 0 0 0   ^0_ 0
0 ^0  00_"
120 PRINT"0   0 0 0   0   0 0 0   0 0
  0 0 0 0"
130 PRINT"0   000 0   0   0 0 0   000
  0 0 000"
140 PRINT"100 0 0 000 100 10\ 000 0 0
  10\ 0 0"
150 PRINT:PRINT"       . AND AUTOMOTIVE
OG BOOK"
160 PEN 2:INK 2,0
170 PRINT:PRINT:PRINT" For:   Miles/Gal
on  &  Litres/100 Kms"
180 PRINT:PRINT:PRINT"              By Mike
Perry (1986)"
190 INK 1,5:FOR i=0 TO 1000:NEXT:INK 2,
5
200 FOR i=0 TO 2000:NEXT:PEN 3:INK 3,26
LOCATE 8,25:PRINT"Loading":z=16:GOSUB 2
0:PRINT"the":z=20:GOSUB 210:PRINT"progr
m":z=28:GOSUB 210:PRINT"now....":RUN"!p
trcalc"
210 FOR i=0 TO 400:NEXT:LOCATE z,25:RET
RN
```

```
10 '++++++++++++++++++++++++++
20 '+  Dimension variables  +
30 '++++++++++++++++++++++++++
40 '
50 MODE 1:LOCATE 1,17:PRINT"      Press
ENTER> for 100 entries              or en
er  number  required.":LOCATE 5,12:INP
T" How many entries do you wish
   to have available";e:IF e=0 THEN e=1
0
60 DIM d$(e),p$(e),o(e),k(e),kb(e),y$(e
),l(e),lb(e),c(e),t(e),s$(e),ca(e),oa(e)
,da(e),la(e),ta(e),lk(e),mg(e)
70 '
80 '++++++++++
90 '+  Menu  +
100 '++++++++++
110 '
```

```
120 MODE 1:PAPER 0:INK 0,11:PEN 1:INK 1,
11:BORDER 11:CLS
130 LOCATE 15,3:PRINT"_M__E__N__U_":PRIN
T
140 PRINT:PRINT TAB(11)"nstructions",TAB
(11),TAB(11)"nter  Data",TAB(11),TAB(11)
"hange Data",TAB(11),TAB(11)"ave to Cass
ette/Disc"
150 PRINT TAB(11),TAB(11)"etrieve from C
assette/Disc",TAB(11),TAB(11)"iew  Data"
,TAB(11),TAB(11)"rint Data",TAB(11),TAB(
11)"lank Form"
160 LOCATE 16,23:PRINT"Which ?"
170 INK 1,0:WINDOW#1,10,10,6,22:PAPER#1,
0:PEN#1,2:INK 2,15:PRINT#1,"I E C S R V
P B"
180 m$=INKEY$:IF m$=""THEN 180
190 IF UPPER$(m$)="I" THEN 2470
200 IF UPPER$(m$)="E" THEN 330
210 IF UPPER$(m$)="C" THEN 1680
220 IF UPPER$(m$)="S" THEN 2020
230 IF UPPER$(m$)="R" THEN 2250
240 IF UPPER$(m$)="V" THEN 1370
250 IF UPPER$(m$)="P" THEN 1370
260 IF UPPER$(m$)="B" THEN ch=0:ci=1:GOT
0 2670
270 GOTO 180
280 '
290 '++++++++++++++++++
300 '+   Enter data   +
310 '++++++++++++++++++
320 '
330 b=0
340 IF n>=0 THEN n=n+1 ELSE n=1
350 IF i>3000 THEN n=n-1:i=0
360 WINDOW 1,40,1,24:BORDER 9:PAPER 0:IN
K 0,9:PEN 1:INK 1,0:CLS
370 WINDOW#1,1,40,25,25:PAPER#1,2:INK 2,
0:PEN#1,3:INK 3,2:CLS#1
380 m=FRE("")
390 PRINT#1," ENTRY NUMBER:";n;"  MEMORY
 LEFT:";m
400 IF n=1 THEN INPUT"Purchase date (eg
07.03.86)";d$(n):GOTO 420
410 PRINT"Purchase date (eg 07.03.86)":P
RINT"(enter 'S' to enter service informa
tion)":LOCATE 30,1:INPUT"";d$(n):PRINT:I
F UPPER$(d$(n))="S" THEN 2830
420 PRINT:INPUT "Place of Purchase ";p$(
n)
430 IF n>1 GOTO 490
440 PRINT:INPUT "First Odometer Reading"
;o(n):IF o(n)=0 THEN 760
450 PRINT:INPUT "Did you fill the tank (
Y/N)";y$(n)
460 IF UPPER$(y$(n))="N" THEN PRINT:PRIN
T"You must commence calculations on a fu
lltank":GOTO 440
470 IF UPPER$(y$(n))="Y" THEN y$(n)="Y":
GOTO 340
480 GOTO 450
490 PRINT:INPUT "Next  Odometer Reading"
;o(n)
500 IF o(n)=0 THEN 540
510 IF o(n)<o(n-1) THEN PRINT:PRINT"INVA
LID ENTRY !":GOTO 490
520 PRINT:INPUT "Did you fill the tank (
Y/N)";y$(n)
530 IF UPPER$(y$(n))="N" OR UPPER$(y$(n)
)="Y"THEN 540 ELSE 520
540 PRINT:INPUT "Litres purchased";l(n)
550 PRINT:INPUT "cost in cents per litre
 (c/l)";c(n)
560 PRINT:INPUT "Total cost in dollars &
 cents,          (eg. 6.00)";t(n)
570 '
580 '++++++++++++++++++++++++++
590 '+ Discrepency correction +
600 '++++++++++++++++++++++++++
610 '
620 a=n:GOTO 650
630 FOR a=1 TO n:mg(a)=0:lk(a)=0:k(a)=0:
IF a=1 THEN l(a)=1
640 IF s$(a)>""THEN o(a)=o(a-1):y$(a)=y$
(a-1):c(a)=c(a-1):l(a)=l(a-1):GOTO 1210
650 IF UPPER$(y$(a))="N" THEN y$(a)="N"
660 IF UPPER$(y$(a))="Y" THEN y$(a)="Y"
670 IF b=0 AND l(a)=0 AND c(a)=0 AND t(a
)=0 THEN 760
680 IF t(a)*100=c(a)*l(a) THEN 1040
690 IF b=0 AND l(a)=0 AND c(a)=0 THEN 76
0
700 IF b=0 AND l(a)=0 AND t(a)=0 THEN 76
0
710 IF l(a)=0 THEN l(a)=t(a)*100/c(a):GO
TO 1040
720 IF t(a)=0 THEN t(a)=(c(a)*l(a))/100:
GOTO 1040
730 IF c(a)=0 THEN c(a)=t(a)*100/l(a):GO
TO 1040
740 IF l(a)>0 AND c(a)=0 AND t(a)=0 THEN
 1040
750 GOTO 770
760 INK 3,0,2:PRINT#1," Invalid entry,..
... Insufficient Data.":FOR i=0 TO 4000:
NEXT i:GOTO 120
770 i=2:ca(a)=t(a)*100/l(a):ca(a)=ROUND(
ca(a),i):la(a)=t(a)*100/c(a):la(a)=ROUND
(la(a),i):ta(a)=(c(a)*l(a))/100:ta(a)=RO
UND(ta(a),i)
780 t(a)=ROUND(t(a),i)
790 IF ta(a)*100=ca(a)*la(a) THEN t(a)=t
a(a):c(a)=ca(a):l(a)=la(a):GOTO 1040
800 IF la(a)MOD l(a)=0 AND ca(a)MOD c(a)
=0 AND ta(a)MOD t(a)=0 THEN 1040
810 IF b=1 THEN 1040
820 INK 1,9:CLS
830 PRINT"-Cents  per litre multiplied
by  litres purchased =$";ta(a);"(total c
ost)"
840 PRINT"-Total cost divided by litres
 purchased =";ca(a);"c/l"
850 PRINT"-Total cost  divided by  cents
 per litre =";la(a);"lt."
860 LOCATE 1,9:PRINT"Do you wish to alte
r:-"
870 LOCATE 1,11:PRINT"1 Cents per litre
 from";c(a);"to";ca(a)"?"
880 LOCATE 1,13:PRINT"2 Litres purchased
 from";l(a);"to";la(a)"?"
```

```
890 LOCATE 1,15:PRINT"3 Total cost from"
;t(a);"to";ta(a);"?"
900 LOCATE 1,17:PRINT"4 Leave the figure
s as they are ?"
910 LOCATE 10,21:PRINT"( Select 1, 2, 3.
 or 4.)"
920 INK 1,0
930 clt$=INKEY$:IF clt$=""THEN 930
940 IF clt$="1" THEN c(a)=ca(a):GOTO 104
0
950 IF clt$="2" THEN l(a)=la(a):GOTO 104
0
960 IF clt$="3" THEN t(a)=ta(a):GOTO 104
0
970 IF clt$="4" GOTO 1040
980 GOTO 930
990 '
1000 '+++++++++++++++
1010 '+   Calculate  +
1020 '+++++++++++++++
1030 '
1040 IF o(a)>0 AND o(a-1)>0 THEN k(a)=o(
a)-o(a-1)
1050 IF o(a)=0 AND o(a-1)>0 THEN oa(a)=o
(a-1)
1060 IF o(a)=0 AND o(a-1)=0 THEN oa(a)=o
a(a-1)
1070 IF o(a)>0 AND o(a-1)=0 THEN k(a)=o(
a)-oa(a-1)
1080 IF o(a)=0 THEN y$(a)="N"
1090 IF y$(a)="Y"AND y$(a-1)="Y" THEN lk
(a)=100*(l(a)/k(a)):mg(a)=2.8248928*(k(a
)/l(a))
1100 IF y$(a)="N"AND y$(a-1)="Y" THEN lb
(a)=l(a):kb(a)=k(a):IF b=0 THEN CLS:GOTO
 1220
1110 IF y$(a)="N"AND y$(a-1)="N" THEN lb
(a)=lb(a)+lb(a-1):kb(a)=kb(a)+kb(a-1):IF
 b=0 THEN CLS:GOTO 1220
1120 IF y$(a)="Y"AND y$(a-1)="N" THEN lk
(a)=100*(l(a)+lb(a-1))/(k(a)-kb(a-1)):mg
(a)=2.8248928*(k(a)+kb(a-1))/(l(a)+lb(a-
1))
1130 '
1140 '++++++++++++++++++++++++++++++++
1150 '+ Display result of last entry +
1160 '++++++++++++++++++++++++++++++++
1170 '
1180 i=2:lk(a)=ROUND(lk(a),i):mg(a)=ROUN
D(mg(a),i):IF b=1 THEN 1210
1190 CLS:PRINT:PRINT"Miles per gallon=";
mg(a)
1200 PRINT:PRINT"Litres per 100 Kms=";lk
(a)
1210 IF b=1 THEN NEXT a
1220 IF y$(n)="N"THEN PRINT:PRINT"Entrie
s must finish with a full tank."
1230 IF b=1 THEN FOR i=0 TO 2000:NEXT i:
GOTO 120
1240 LOCATE 10,15:PRINT"Do you wish to:"
1250 PRINT:PRINT TAB(10)"1. Enter more d
ata.":PRINT:PRINT TAB(10)"2. Enter servi
ce details.":PRINT:PRINT TAB(10)"OR"
1260 PRINT:PRINT TAB(10)"3. Return to me
nu."
1270 a$=INKEY$:IF a$="" THEN 1270
1280 IF a$="1" THEN CLS:GOTO 340
1290 IF a$="2" THEN 2820
1300 IF a$="3" THEN 120
1310 GOTO 1270
1320 '
1330 '++++++++++++++++++++
1340 '+  Screen display  +
1350 '++++++++++++++++++++
1360 '
1370 ch=0:ci=1:lt=0:tt=0:lka=0:mga=0:q=0
1380 IF n<1 THEN CLS:PRINT:PRINT"No data
 to display or print!":FOR i=0 TO 2000:N
EXT i:GOTO 120
1390 MODE 2:BORDER 9:INK 0,12:INK 1,0:CL
S
1400 WINDOW 1,80,2,24:CLS:WINDOW#1,1,80,
1,1:PAPER#1,3:PEN#1,0:CLS#1:WINDOW#2,1,8
0,25,25:PAPER#2,3:PEN#2,0:CLS#2
1410 PRINT#ci,"No     DATE     PLACE     ME
TER     KMS    FULL  C/L     L        $      L
/100km  M/G"
1420 IF ch=8 THEN PRINT#ch,"____________
____________________________________
_____________________"
1430 PRINT#ch," 1"TAB(5)d$(1);TAB(15)p$(
1);TAB(23)o(1);TAB(41)y$(1)
1440 FOR a=2 TO n:IF y$(a)="Y" AND s$(a)
="" THEN q=q+1
1450 i=2:c(a)=ROUND(c(a),(i/2)):l(a)=ROU
ND(l(a),i):t(a)=ROUND(t(a),i):lk(a)=ROUN
D(lk(a),(i/2)):mg(a)=ROUND(mg(a),(i/2)):
d$(a)=LEFT$(d$(a),8):p$(a)=LEFT$(p$(a),6
):IF o(a)=0 THEN y$(a)=" "
1460 IF s$(a)>"" THEN PRINT#ch,a;TAB(5)d
$(a);TAB(15)p$(a);TAB(23)ob(a);TAB(33)s$
(a);TAB(58)t(a):GOTO 1500
1470 PRINT#ch,a;TAB(5)d$(a);TAB(15)p$(a)
;TAB(23)o(a);TAB(32)k(a);TAB(41)y$(a);TA
B(44)c(a);TAB(50)l(a);TAB(58)t(a);TAB(66
)lk(a);TAB(72)mg(a):lt=lt+l(a):tt=tt+t(a
):lka=lka+lk(a):mga=mga+mg(a)
1480 IF UPPER$(m$)="P"THEN 1500 ELSE IF
ch=0 AND a MOD 20=0 THEN CLS#2:LOCATE#2,
26,1:PRINT#2,"Press <Space> for next pag
e."ELSE 1500
1490 IF ch=0 THEN spac$=INKEY$:IF spac$<
>" "THEN 1490 ELSE CLS
1500 NEXT a
1510 IF q=0 THEN q=1
1520 lka=ROUND((lka/q),(i/2)):mga=ROUND(
(mga/q),(i/2))
1530 PRINT#ch,TAB(49)"_____
__________":PRINT#ch,TAB(50)lt;TAB(58)tt
;TAB(68)lka;TAB(72)mga
1540 PRINT#ch,TAB(51)"Total";TAB(59)"Tot
al";TAB(67)"Ave";TAB(73)"Ave"
1550 lt=0:tt=0:lka=0:mga=0:q=0
1560 IF ch=8 THEN 120
1570 IF ch=0 AND UPPER$(m$)="V"THEN PRIN
T#2," PRESS:  'M' to return to main menu
   or  'P' to print a copy of these resu
lts"
1580 IF UPPER$(m$)="P"THEN ch=8:ci=8:PRI
NT#2,TAB(35)"PRINTING .....":GOTO 1410
```

```
1590 a$=INKEY$:IF a$="" GOTO 1590
1600 IF UPPER$(a$)="M"THEN 120
1610 IF UPPER$(a$)="P"THEN ch=8:ci=8:PRI
NT#2,TAB(35)"PRINTING .....":GOTO 1410
1620 GOTO 1590
1630 '
1640 '++++++++++++++++++++
1650 '+  Data alteration  +
1660 '++++++++++++++++++++
1670 '
1680 IF n<1 THEN CLS:PRINT:PRINT"No data
 available to alter!":FOR i=0 TO 2000:NE
XT i:GOTO 120
1690 BORDER 15:INK 0,15:CLS
1700 INPUT"Which entry number are you al
tering";a:IF a>n THEN PRINT"Invalid entr
y!":FOR i=0 TO 2000:NEXT i:GOTO 1700
1710 CLS:PEN 2:INK 2,0:PRINT TAB(10)"- A
lteration Menu -":PEN 1:INK 1,14:LOCATE
12,3:PRINT"Altering item:";a
1720 LOCATE 13,5:PRINT"ate.";TAB(13),TAB
(13)"lace.";TAB(13),TAB(13)"eter.";TAB(1
3),TAB(13)"ull tank (y/n).";TAB(13),TAB(
13)"itres purchased.";TAB(13),TAB(13)"en
ts per litre.";TAB(13),TAB(13)"otal cost
."
1730 LOCATE 13,19:PRINT"ervice.":LOCATE
13,21:PRINT"eturn to menu."
1740 PEN 2:LOCATE 12,24:PRINT"    Which
?"
1750 WINDOW#1,12,12,5,22:PEN#1,3:INK 3,6
:PRINT#1,"D P M F L C T S R"
1760 IF s$(a)>""THEN WINDOW#3,1,40,11,16
:CLS#3
1770 alt$=INKEY$:IF alt$="" THEN 1770
1780 GOTO 1850
1790 LOCATE 1,2:PRINT"You have entered:"
:PRINT:PRINT"Is this correct (y/n) ?"
1800 md$=INKEY$:IF md$="" THEN 1800
1810 IF UPPER$(md$)="Y"THEN md$="Y":RETU
RN
1820 IF UPPER$(md$)="N"THEN 1840
1830 GOTO 1800
1840 PRINT:PRINT"Enter correct item:":RE
TURN
1850 IF UPPER$(alt$)="D"THEN CLS:LOCATE
19,2:PRINT d$(a):GOSUB 1790:IF md$="Y"TH
EN 1960 ELSE LOCATE 21,6:INPUT"",d$(a):G
OTO 1710
1860 IF UPPER$(alt$)="P"THEN CLS:LOCATE
19,2:PRINT p$(a):GOSUB 1790:IF md$="Y"TH
EN 1960 ELSE LOCATE 21,6:INPUT"",p$(a):G
OTO 1710
1870 IF UPPER$(alt$)="M"AND s$(a)=""THEN
 CLS:LOCATE 19,2:PRINT o(a):GOSUB 1790:I
F md$="Y"THEN 1960 ELSE LOCATE 21,6:INPU
T"",o(a):GOTO 1710
1880 IF UPPER$(alt$)="M"AND s$(a)>""THEN
 CLS:LOCATE 19,2:PRINT ob(a):GOSUB 1790:
IF md$="Y"THEN 1960 ELSE LOCATE 21,6:INP
UT"",ob(a):GOTO 1710
1890 IF UPPER$(alt$)="F"AND s$(a)=""THEN
 CLS:LOCATE 19,2:PRINT y$(a):GOSUB 1790:
IF md$="Y"THEN 1960 ELSE LOCATE 21,6:INP
UT"",y$(a):GOTO 1710
1900 IF UPPER$(alt$)="L"AND s$(a)=""THEN
 CLS:LOCATE 19,2:PRINT l(a):GOSUB 1790:I
F md$="Y"THEN 1960 ELSE LOCATE 21,6:INPU
T"",l(a):GOTO 1710
1910 IF UPPER$(alt$)="C"AND s$(a)=""THEN
 CLS:LOCATE 19,2:PRINT c(a):GOSUB 1790:I
F md$="Y"THEN 1960 ELSE LOCATE 21,6:INPU
T"",c(a):GOTO 1710
1920 IF UPPER$(alt$)="T"THEN CLS:LOCATE
19,2:PRINT t(a):GOSUB 1790:IF md$="Y"THE
N 1960 ELSE LOCATE 21,6:INPUT"",t(a):GOT
O 1710
1930 IF UPPER$(alt$)="S"THEN CLS:LOCATE
19,2:PRINT s$(a):GOSUB 1790:IF md$="Y"TH
EN 1960 ELSE LOCATE 21,6:INPUT"",s$(a):G
OTO 1710
1940 IF UPPER$(alt$)="R"THEN 1960
1950 GOTO 1770
1960 LOCATE 12,24:PRINT"Calculating...."
:b=1:GOTO 630
1970 '
1980 '++++++++++++++++++++++++++
1990 '+  Save to cassette/disc  +
2000 '++++++++++++++++++++++++++
2010 '
2020 IF n<1 THEN CLS:PRINT:PRINT"No data
 to save!":FOR i=0 TO 2000:NEXT i:GOTO 1
20
2030 CLS:PRINT"SAVE DATA":PRINT:INPUT"Ar
e you saving to tape or disc (T/D) ";s$
2040 IF UPPER$(s$)="T" THEN |TAPE:GOTO 2
070
2050 IF UPPER$(s$)="D" THEN |DISC:GOTO 2
070
2060 GOTO 2030
2070 PRINT:PRINT"Press 'M' to return to
menu or enter,":PRINT:INPUT"File name ",
f$:PRINT:IF UPPER$(f$)="M" THEN 120 ELSE
 OPENOUT f$
2080 PRINT#9,n
2090 FOR a=1 TO n
2100 WRITE#9,d$(a)
2110 WRITE#9,p$(a)
2120 WRITE#9,o(a)
2130 WRITE#9,y$(a)
2140 WRITE#9,l(a)
2150 WRITE#9,c(a)
2160 WRITE#9,t(a)
2170 WRITE#9,s$(a)
2180 NEXT a
2190 CLOSEOUT:GOTO 120
2200 '
2210 '++++++++++++++++++++++++++++++++
2220 '+ Retrieve from cassette/disc +
2230 '++++++++++++++++++++++++++++++++
2240 '
2250 CLS:PRINT"RETRIEVE DATA":PRINT:INPU
T"Are you retrieving from tape or disc
  (T/D) ";s$
2260 IF UPPER$(s$)="T" THEN |TAPE:GOTO 2
290
2270 IF UPPER$(s$)="D" THEN |DISC:GOTO 2
290
2280 GOTO 2250
2290 PRINT:PRINT"Press 'M' to return to
```

```
menu or enter,":PRINT:INPUT"File name ",
f$:PRINT:IF UPPER$(f$)="M" THEN 120 ELSE
 OPENIN f$
2300 INPUT#9,n
2310 FOR a=1 TO n
2320 INPUT#9,d$(a)
2330 INPUT#9,p$(a)
2340 INPUT#9,o(a)
2350 INPUT#9,y$(a)
2360 INPUT#9,l(a)
2370 INPUT#9,c(a)
2380 INPUT#9,t(a)
2390 INPUT#9,s$(a)
2400 NEXT a
2410 CLOSEIN:b=1:LOCATE 12,12:PRINT"Calc
ulating......":GOTO 630
2420 '
2430 '++++++++++++++++++
2440 '+  Instructions  +
2450 '++++++++++++++++++
2460 '
2470 '
2480 BORDER 0:INK 0,0:INK 1,13:CLS
2490 PRINT"   This  program  will  calcu
late  milesper gallon  as well  as litre
s  per  100kilometres.  The  calculatio
n  will  bemade  from  one time the  tan
k  has beenfilled untill the next."
2500 PRINT:PRINT"   The number of  entri
es  available  isdetermined when the pro
gram is first runand is only limited by
available memory.
2510 PRINT"   To bypass a data entry, pr
ess  ENTER.However  the minimum  data re
quired  is;litres purchased  (only)  OR
any two of:1. litres purchased 2. cost i
n cents perlitre 3. total purchase cost.
   If  only"
2520 LOCATE 1,16:PRINT"two  are  entered
  the  third   will  becalculated.   If
all three  are  enteredbut don't agree,
 the option is given toadjust one item."
2530 PEN 2:INK 2,26:LOCATE 7,24:PRINT"Pr
ess  <space>  to continue."
2540 spac$=INKEY$:IF spac$<>" " THEN 254
0
2550 PEN 1:INK 1,13:CLS:PRINT:PRINT"   I
f the meter reading is bypassed, theprog
ram  will  assume,  for  calculationpurp
oses  that  the  tank  has not  beenfill
ed."
2560 PRINT:PRINT"   When  prompted,  ser
vice  informationmay  be  entered  witho
ut  effecting thecalculations."
2570 PRINT:PRINT"   On  access from  the
 menu,  data  maybe  changed  at  any
 stage  and  anycalculated data will be
 adjusted."
2580 PRINT:PRINT"   A blank form may be
 printed to  keepin the  car  to  record
  information forlater entry into the co
mputer."
2590 PEN 2:INK 2,26:LOCATE 3,24:PRINT"Pr
ess  <space>  to return to Menu."
2600 spac$=INKEY$:IF spac$<>" "THEN 2600
2610 GOTO 120
2620 '
2630 '++++++++++++++++++++++
2640 '+  Print blank form  +
2650 '++++++++++++++++++++++
2660 '
2670 MODE 2:BORDER 9:INK 0,12:INK 1,0:ch
=0:ci=1:CLS
2680 WINDOW 1,80,2,24:CLS:WINDOW#1,1,80,
1,1:PAPER#1,3:PEN#1,0:CLS#1:WINDOW#2,1,8
0,25,25:PAPER#2,3:PEN#2,0:CLS#2
2690 IF ch=0 THEN b=10 ELSE INPUT#2,"How
 many lines do you wish to print";b
2700 PRINT#ci,"No    DATE    PLACE    ME
TER    KMS   FULL  C/L     L      $    L/1
00km  M/G"
2710 PRINT#ch,"_____________________________
___________________________________________
________"
2720 FOR a=1 TO b
2730 IF a<=9 THEN PRINT#ch,a;" 1
1       1            1        1  1      1
  1       1      1    1"
2740 IF a>9 THEN PRINT#ch,a;"1           1
      1           1          1  1     1
1      1      1"
2750 PRINT#ch,"____1__________1___1____1___
_____1_______1_1______1______1______1___
___1___1"
2760 NEXT a:IF ch=8 THEN 120
2770 ch=8:ci=8:GOTO 2690
2780 '
2790 '+++++++++++++++++++++++++
2800 '+  Service Information  +
2810 '+++++++++++++++++++++++++
2820 n=n+1
2830 o(n)=o(n-1):y$(n)=y$(n-1):c(n)=c(n-
1):l(n)=l(n-1):CLS
2840 PRINT#1," ENTRY NUMBER:";n;"   MEMO
RY LEFT:";m
2850 PRINT:INPUT "Date of Service (eg 07
.03.86)";d$(n)
2860 PRINT:INPUT "Place of Service ";p$(
n)
2870 PRINT:INPUT "Odometer Reading at se
rvice";ob(n)
2880 PRINT:PRINT"Enter your service info
rmation          (25 characters maximum)
.":INPUT"",s$(n):s$(n)=LEFT$(s$(n),25):I
F s$(n)=""THEN s$(n)=" "
2890 PRINT:INPUT "Cost of service in dol
lars & cents,     (eg. 6.00)";t(n)
2900 GOTO 1220
```

Adventurer's Attic

by Philip Riley

So far in this column we have looked at writing adventures, so this month we will take a look at solving adventures, and going by the amount of people who have sent us their problems to worry about, this article is not a moment too soon.

To start with I will quote another one of my golden rules 'examine everything'. Just about every adventure column you read in the various magazines tell you to do this so half a million column writers can't be wrong.
But seriously it is the major rule it is amazing how much help you can get by examining various things.
Another good idea is to examine a thing more than once - at least one of the Arnold Blackwood adventures has three different objects hidden in the one location.
You can also use other words besides 'look' and 'examine'. Try 'search' or 'sort' as well.
Next we look at books. When you find a book you naturally type 'read book' and you may get a clue or some witty little comment. Next time try 'turn page' and then 'read book' rather obvious really.
When in a new location always try every direction. Even those that the game does not mention, and don't just try North, South, East and West. Also try NE, NW, SE, SW, UP and DOWN.
If you are confronted by some sort of barrier e.g. bubbling hot mud. Try to jump or leap it you might make it across. (But then again you might not).
If you are really, really stuck, you have gone around in circles and can find no way past a certain problem always go back through the adventure and look at everything again to see if you can find some other useful object hidden away.
If this fails list out the objects that you are carrying to see if you can make something with two or more of them. In my previous adventures I have made everything from grappling hooks to blowpipes and bombs.
Single word commands can be useful as well. 'Help' is always popular although it very rarely helps you in any way. You can also try the following 'look', 'list', 'jump', 'shout', 'listen' and 'run' the last one might help you out of a sticky situation.
If a command does not work first time try it again. It may work a second or third time. An example of this is in King Solomon's mines. When stuck in the mud it takes several pulls on the rope to free yourself.
If in doubt kill it, rather drastic but if you are confronted by some rather vicious slobbering monster and you are armed it may be wise to kill it. Once killed don't forget to 'search body' you may find something useful.
One thing that most people do which can normally be a fatal mistake is 'eat food' as soon as they find it. This is not a good idea. It is a well known fact that ninety per cent of all food in adventures is poisoned in some way. You may also need to give the food to someone or something to get past them. For instance you could give the food to the rather vicious slobbering monster mentioned earlier. It may just distract him long enough for you to slip past. (Either that or the poison in it will kill him). But only eat food as a last resort when you are getting desperate.
If all else fails sleep on it (things always look better in the morning) and if that doesn't work write to us and we will worry about it as well as you.
In closing this month I would like to thank Steve Alatakis for his letter. We don't normally mention individuals but because of the quality and quantity (approximately 10 pages of questions, answers and maps) we think Steve deserves a mention. Thanks Steve. I showed the Editor the section telling him to leave this column alone. I think (or rather hope) he will take notice.
See you next month and keep adventuring.

Please note that line 20 of the coding last month to read two different types of data should have been:

```
20  FOR T=1 TO 2:READ
    IT$(T),IT(T):NEXT
```

Brainteaser Duo

from J.J. Vinopal

FIVE-IN-ROW is the first program listed. It is based upon the board game "Gomoku" written by G. Charlton some years ago. Although the program is only 6k and has no built-in "intelligence, winning the game is not easy at first. The idea is simple, merely to get five of your pieces in a row either horizontally, vertically or diagonally. Alternatively, you can try to stop the computer from achieving the same thing.

The second game is TWIST-CUBE. Here you are presented with four each of cars, trains, tanks and bicycles which the computer will jumble before play commences. You have to re-arrange by moving a selected block of four (including the centre blocks) to duplicate the picture on the right of the screen in the minimum number of moves. It's not too hard to start with, but it gets progressively difficult as you try to reduce the number of moves it takes.

```
10 'Program must be copied exactly with
exception
20 'of rems.If you prefer to make some c
hanges,
30 'type program as written,if it works
(it should)
40 'save it and then you may be able to
do some
50 'small changes.
60 REM********** 5 IN ROW **********
70 CALL &BC02:SYMBOL AFTER 32
80 fla=0:GOSUB 1340
90 '********** JJ VINOPAL **********
100 '********** 22 GROONGAL ST *******
110 '********** MAYFIELD WEST *******
120 '********** N.S.W.  2304 ********
130 MODE 1:INK 0,0:INK 1,24:INK 2,6:INK
3,11
140 :BORDER 0:PAPER 0:PEN 1:a$=CHR$(24)
150 ON BREAK GOSUB 1300:PEN 1:GOSUB 1290
160 GOSUB 860:GOSUB 1290:GOSUB 1650:GOSU
B 290
170 :GOSUB 460:GOSUB 1290:GOSUB 290:GOSU
B 1290
180 GOSUB 590:GOSUB 1290:GOSUB 290:GOSUB
 1290
190 '
200 REM****** C win ******
210 IF posi>3 THEN GOSUB 440:PRINT " I w
in ! ":GOTO 1170
220 GOTO 170
230 '
240 REM****** select ******
250 :spot=hor
260 :spot=spot+set:IF hor(spot)<>y THEN
RETURN
270 calk=calk+1:GOTO 260
280 '
290 :REM****** board ******
300 LOCATE 1,1:PEN 1:ZONE 6
310 PRINT,"1 2 3 4 5 6 7 8";:PEN  3
320 PRINT"        FIVE IN":PEN 1:PRINT
330 FOR hor=1 TO 8:PRINT TAB(4);hor;
340 FOR ver=2 TO 9:mark=hor(hor*10+ver)
350 IF mark=comp THEN GOSUB 1500
360 SOUND 1,200+(hor*10),1,15
370 IF mark=hum THEN PEN 2:GOSUB 1550
380 IF mark=207 THEN PEN 3:GOSUB 1600
390 PEN 1:NEXT:PRINT CHR$(8);hor:PRINT:N
EXT
400 PRINT,"1 2 3 4 5 6 7 8";:PEN 3
410 PRINT "        A ROW !!":RETURN
420 '
430 REM****** ink ******
440 :LOCATE 28,10:INK 2,8,24:PEN 2:RETUR
N
450 '
460 :REM******* input ******
470 GOSUB 1290
480 :PEN 2:LOCATE 2,21:PRINT base$:LOCAT
E 2,22:PRINT base$
490 LOCATE 2,23:PRINT base$:PEN 1:PRINT
CHR$(22);CHR$(1)
500 LOCATE 5,22:PRINT"Please enter your
move ";
510 LOCATE 29,22:INPUT MOV:PRINT CHR$(22
);CHR$(0):MOV=MOV+1
520 IF MOV<12 OR MOV>89 THEN 480
530 IF hor(MOV)<>207 THEN 480
540 y=hum:PEN 2:LOCATE 2,21:PRINT base$:
LOCATE 2,22:PRINT base$
550 LOCATE 2,23:PRINT base$:PRINT CHR$(2
2);CHR$(1)
560 PEN 1:LOCATE 5,22:PRINT"Wait for me
or else!";
570 PRINT CHR$(22);CHR$(0):hor(MOV)=y:RE
TURN
580 '
590 :REM****** main ******
600 REM****** H win ******
610 GOSUB 1220
620 IF posi>3 THEN GOSUB 440:PRINT " You
```

```
win!":GOTO 1170
630 stim=res
640 :IF stim<>2 THEN y=comp
650 IF stim=2 THEN y=hum
660 MOV=0:hum1=0:posi=0
670 FOR hor=12 TO 89:mark=0
680 IF hor(hor)<>207 THEN 720
690 GOSUB 1050
700 IF mark<=hum1 THEN 720
710 hum1=mark:MOV=hor
720 :NEXT
730 IF hum1<>0 THEN 820
740 stim=stim+1:IF stim<>4 THEN 640
750 hor=res
760 :MOV=CINT(RND*77)+12
770 IF hor(MOV)=207 THEN 820
780 hor=hor+1:IF hor<100 THEN 760
790 '
800 REM****** draw ******
810 GOSUB 440:PRINT "It's draw!":GOTO 11
70
820 :hor(MOV)=comp:y=comp
830 GOSUB 1220
840 RETURN
850 '
860 :REM****** title ******
870 CLS:base$=STRING$(37,227):wal$=STRIN
G$(9,227)
880 IF fla=1 THEN 890 ELSE DIM hor(100),
x(4)
890 x(1)=1:x(2)=9:x(3)=10:x(4)=11
900 FOR comp=1 TO 8:FOR ver=2 TO 9
910 hor(comp*10+ver)=207:NEXT:NEXT
920 hum=249:comp=225:GOSUB 1770
930 INK 0,0:INK 1,24:INK 2,6:INK 3,11
940 PEN#1,0:PRINT#1:PRINT#1:PRINT#1
950 PRINT#1,"   *[   F I V E   I N   A
 R O W   ]*":PRINT#1
960 PRINT#1:PEN#1,1:PRINT#1,"          By
  J.J."t$" @ 1986":PEN#1,2
970 LOCATE#1,1,12:PRINT#1,"  "i$"   Ente
r 'Y' if you want the        "i$
980 PRINT#1:PRINT#1,"  "i$"   first move
..'N' if you don't!!   ";i$
990 res=ASC(b$)-72:set=0:set=set+1
1000 :a$=INKEY$:IF a$=""THEN 1000
1010 CLS:IF a$="y"OR a$="Y"THEN RETURN
1020 FOR j=1 TO CINT(RND*12):READ y:NEXT
:y=y+30:hor(y)=comp:RETURN
1030 '
1040 REM****** H calcul ******
1050 :FOR x=res TO 4:calk=0:set=x(x)
1060 GOSUB 250:set=-set:GOSUB 250
1070 IF calk>posi THEN hum1=0:posi=calk
1080 IF posi<>calk THEN 1110
1090 IF stim=res AND posi<4 OR (stim=2 O
R stim=3) AND posi<2 THEN 1110
1100 mark=mark+1
1110 :NEXT:RETURN
1120 '
1130 REM****** data ******
1140 DATA &19,&E,&5,&24,&10,&11,&4,&18,&
36,&1A,&1B,&2D
1150 '
1160 REM****** end ******
1170 :FOR s=1 TO 21:SOUND 1,140+15*s,22-
s,14
1180 SOUND 1,930,22-s,14:SOUND 1,240,22-
s,14
1190 NEXT:CALL &BC02:LOCATE 29,22:PRINT"
NEW GAME?"
1200 CALL &BB18:fla=1:GOTO 130
1210 '
1220 :REM****** CALCUL ******
1230 hor=MOV:posi=0:FOR x=res TO 4:calk=
0:set=x(x)
1240 GOSUB 250:set=-set:GOSUB 250
1250 IF calk>posi THEN posi=calk
1260 NEXT:RETURN
1270 '
1280 REM****** sound ******
1290 :SOUND 1,11,5,15:SOUND 2,21,5,15:RE
TURN
1300 :INK 0,0:INK 1,13:MODE 2:PAPER 1:PE
N 0:CLS:LIST
1310 :RUN 140
1320 '
1330 REM****** symbols ******
1340 :SYMBOL 225,60,36,231,165,189,90,24
,102
1350 SYMBOL 216,128,124,32,16,8,5,5,5
1360 SYMBOL 217,8,16,32,64,128,4,2,1
1370 SYMBOL 218,0,0,0,16,0,18,127,128
1380 SYMBOL 219,0,0,0,1,1,36,251,4
1390 SYMBOL 220,0,0,0,192,52,138,133,64
1400 SYMBOL 221,16,8,4,4,2,2,225,30
1410 SYMBOL 222,21,133,68,56,0,0,0,0
1420 SYMBOL 223,33,1,130,124,0,0,0,0
1430 SYMBOL 224,64,32,32,16,16,8,8,0
1440 SYMBOL 226,0,24,24,24,66
1450 SYMBOL 249,28,28,72,62,9,24,164,70
1460 SYMBOL 250,12,0,64,8,9,8,128,130
1470 SYMBOL 227,254,254,254,0,239,239,23
9,0
1480 RETURN
1490 '
1500 :REM****** picture1 ******
1510 PRINT CHR$(128);:PRINT CHR$(22);CHR
$(1);
1520 PRINT CHR$(8);CHR$(225);:PEN 2:PRIN
T CHR$(8);CHR$(226);
1530 PRINT" ";CHR$(22);CHR$(0);:RETURN
1540 '
1550 :REM****** picture2 ******
1560 PEN 1:PRINT CHR$(128);:PRINT CHR$(2
2);CHR$(1);
1570 PRINT CHR$(8);CHR$(249);:PEN 3:PRIN
T CHR$(8);CHR$(250);
1580 PRINT" ";CHR$(22);CHR$(0);:RETURN
1590 '
1600 :REM****** lot ******
1610 PRINT CHR$(128);:PRINT CHR$(22);CHR
$(1);
1620 PRINT CHR$(8);CHR$(207);:PEN 2:PRIN
T CHR$(8);CHR$(144);
1630 PRINT" ";CHR$(22);CHR$(0);:RETURN
1640 '
1650 :REM****** sign ******
1660 PEN 3:LOCATE 28,4:PRINT i$"Newcastl
e":LOCATE 28,5:PRINT" Amstrad ";i$
```

```
1670 LOCATE 28,6:PRINT i$"  Users   ":LOC
ATE 28,7:PRINT"  Group   ";i$
1680 LOCATE 28,9:PRINT wal$:LOCATE 28,11
:PRINT wal$
1690 LOCATE 28,8:PRINT i$"              ":LOC
ATE 28,10:PRINT i$wal$
1700 LOCATE 28,3:PRINT i$"              ":LOC
ATE 28,15:PRINT" "t$" ";i$
1710 LOCATE 28,13:PRINT i$" "CHR$(164)"
1986  "
1720 LOCATE 28,14:PRINT"   J.J.   ";i$
1730 LOCATE 28,12:PRINT i$"             ":LO
CATE 28,17:PRINT"              ";i$
1740 LOCATE 26,16:PRINT i$"Australia";i$
:RETURN
1750 '
1760 REM"****** title a ******
1770 :LOCATE 18,8:FOR w=216 TO 221:PRINT
 CHR$(w);:NEXT
1780 LOCATE 18,9:PRINT CHR$(222)CHR$(223
)"  "CHR$(224)" "
1790 t$="VINOPAL":ORIGIN 0,0:TAG:x=142:y
=380
1800 FOR t=1 TO 16
1810 PLOT -10,-10,VAL(MID$("132123133212
312123",t,1)):ty=y-t*6
1820 IF t<6 THEN MOVE x-t*24,ty:PRINT"JJ
V";:MOVE x+310+t*24,ty:PRINT"JJV";
1830 MOVE x+96-t*12,ty:PRINT"1986 JJ";:M
OVE x+182+t*12,ty:PRINT t$;
1840 SOUND 1,14,1,9:NEXT:b$=MID$(t$,2,1)
:TAGOFF
1850 WINDOW#1,1,40,10,25:PAPER#1,3:PEN#1
,1:CLS#1
1860 RETURN
```

```
10 'Program must be copied exactly with
exception
20 'of rems.If you prefer to make some c
hanges,
30 'type program as written,if it works
(it should)
40 'save it and then you may be able to
do some
50 'small changes.
60 REM ****** TWIST CUBE  *********
70 REM ****** copyrights (c) ******
80 REM ****** J.J.VINOPAL     ******
90 REM ****** ++++++++++++++ ******
100 REM ****** MAYFIELD WEST  ******
110 REM ****** NSW       2304  ******
120 REM
130 CALL-&43FE:MODE 1:INK 0,0:INK 1,6:IN
K 2,9:INK 3,26
140 ON BREAK GOSUB 1780
150 REM
160 REM ********* dimensions **********
170 DIM b(16),c(4):tw=0:b=1:c=0:x=0:d$="
J.J.VINOPAL":m$=CHR$(11)
180 f$=STRING$(38,32):e$=MID$(d$,6,1):k$
=CHR$(8):j$=CHR$(10)
190 z1$=(CHR$(22)+CHR$(1)):z0$=(CHR$(22)
+CHR$(0)):o1$=CHR$(15)
200 i1$=CHR$(24):GOSUB 1240:GOSUB 1920
210 CALL-&43FE:MODE 1:INK 0,0:INK 1,6:IN
K 2,9:INK 3,26
220 WINDOW#1,9,28,4,24:WINDOW#2,30,39,12
,18:WINDOW#3,6,9,4,24
230 PEN#1,1:PAPER#1,3:CLS#1:PAPER#3,3:CL
S#3
240 u$="M.J.LINHART":d=ASC(e$)-72:GOSUB
1100
250 REM
260 REM ************ made array ********
*
270 FOR b=1 TO 4:b(b)=CINT(b/2):b(b+4)=C
INT(b/2)
280 b(b+8)=CINT(b/2)+2:b(b+12)=CINT(b/2)
+2
290 NEXT
300 IF d=1 THEN GOSUB 420 ELSE CALL &400
6
310 REM
320 REM ********** twistigs ***********
330 PEN 1:IF tw<11 THEN LOCATE 21,2:PRIN
T"randomly twisting":tw=tw+1
340 SOUND 1,12,1,15:IF tw<11 THEN x=CINT
(RND*12)
350 IF tw>10 THEN GOSUB 500
360 IF x<2 OR x=5 OR x=9 OR x>12 THEN 34
0
370 GOSUB 640
380 IF tw<10 THEN LOCATE 23,3:PRINT"twis
t "tw" times"
390 GOSUB 420:tw=tw+1
400 GOTO 340
410 REM
420 REM ********** print board *********
**
430 BORDER 0:LOCATE #1,1,3
440 FOR b=1 TO 16:GOSUB 800:REM ? b(b)
450 SOUND 1,999,1,15
460 IF b/4=INT(b/4) THEN GOSUB 1160:REM
1-9
470 NEXT
480 RETURN
490 REM
500 REM ********** input ***********
510 PEN 3
520 IF tw>10 THEN GOSUB 690:CALL &BB9C:L
OCATE 15,2:PRINT"Your twist it "tw-11" t
imes":CALL &BB9C
530 IF tw<11 THEN RETURN
540 PRINT" "f$
550 LOCATE 31,5:PRINT"CHOICE?":LOCATE 31
,8:PRINT"< 1-9 >"
560 PEN 1:LOCATE 29,20:PRINT d$:LOCATE 2
9,22:PRINT u$
570 a$=INKEY$:IF a$="" THEN 570
580 x=VAL(a$):IF x<1 OR x>9 THEN 570
590 SOUND 1,800,10,15
600 IF x<4 THEN x=x+1:RETURN
610 IF x<7 THEN x=x+2:RETURN
620 x=x+3:RETURN
630 REM
640 REM ********** twist **********
650 c(1)=b(x):c(2)=b(x+4):c(3)=b(x+3):c(
```

```
4)=b(x-1)
660 b(x)=c(4):b(x+4)=c(1):b(x+3)=c(2):b(
x-1)=c(3)
670 RETURN
680 REM
690 REM ********** check ***********
700 sc=0:IF b(1)=1 AND b(2)=1 AND b(5)=1
 AND b(6)=1 THEN sc=sc+1
710 IF b(3)=2 AND b(4)=2 AND b(7)=2 AND
b(7)=2 THEN sc=sc+1
720 IF b(9)=3 AND b(10)=3 AND b(13)=3 AN
D b(14)=3 THEN sc=sc+1
730 IF b(11)=4 AND b(12)=4 AND b(15)=4 A
ND b(16)=4 THEN sc=sc+1
740 IF sc<>4 THEN RETURN
750 REM
760 REM ********** end ***********
770 INK 3,1,3:PEN 3:PRINT"good":PRINT"by
e!"
780 GOSUB 2230:CALL-17640:RUN
790 REM
800 REM ******** picture **********
810 wi=1:IF b(b)=1 THEN GOSUB 880
820 IF b(b)=2 THEN GOSUB 930
830 IF b(b)=3 THEN GOSUB 980
840 IF b(b)=4 THEN GOSUB 1030
850 RETURN
860 REM
870 REM ******** draw 4 signs *********
880 PRINT#wi,"  "j$k$k$"  "m$k$k$;
890 PRINT#wi,z1$o1$"1"CHR$(129)+CHR$(130
)j$k$k$CHR$(132)+CHR$(135)m$;
900 PRINT#wi,k$o1$"2"CHR$(131)j$k$k$CHR$
(133)+CHR$(136)k$k$;
910 PRINT#wi,o1$"0"CHR$(134)+CHR$(137);
920 PRINT#wi,z0$m$"    ";:RETURN
930 PRINT#wi,"  "j$k$k$"  "m$k$k$;
940 PRINT#wi,z1$o1$"0"CHR$(140)+CHR$(141
)j$k$k$CHR$(143)+CHR$(145)m$;
950 PRINT#wi,k$k$o1$"1"CHR$(139)k$j$CHR$
(144)+CHR$(138)m$k$o1$"2";
960 PRINT#wi,CHR$(142)j$k$k$CHR$(137)+CH
R$(137);
970 PRINT#wi,z0$m$"    ";:RETURN
980 PRINT#wi,"  "j$k$k$"  "m$k$k$;
990 PRINT#wi,z1$o1$"2"CHR$(149)+CHR$(151
)j$k$k$CHR$(153)+CHR$(156)m$;
1000 PRINT#wi,o1$"0"k$k$CHR$(148)+CHR$(1
50)j$k$k$CHR$(155)+CHR$(158);
1010 PRINT#wi,m$o1$"1"k$k$CHR$(147)+CHR$
(152)j$k$k$CHR$(154)+CHR$(157);
1020 PRINT#wi,z0$m$"    ";:RETURN
1030 PRINT#wi,"  "j$k$k$"  "m$k$k$;
1040 PRINT#wi,z1$o1$"1"CHR$(160)+CHR$(16
3)j$k$k$CHR$(166)+CHR$(170)m$;
1050 PRINT#wi,k$k$o1$"2"CHR$(161)+CHR$(1
64)k$j$k$CHR$(168)+CHR$(169)m$k$;
1060 PRINT#wi,o1$"0"k$CHR$(162)+CHR$(165
)k$j$k$CHR$(167)+CHR$(171);
1070 PRINT#wi,z0$m$"    ";:RETURN
1080 REM
1090 REM ******** side signs **********
1100 PAPER#2,3:CLS#2
1110 PEN#2,2:LOCATE#2,2,2:wi=2:GOSUB 890
:LOCATE#2,6,2:GOSUB 940
1120 LOCATE#2,2,5:GOSUB 990:LOCATE#2,6,5
:GOSUB 1040
1130 PEN#wi,0:FOR dd=1 TO 7:LOCATE#2,9,d
d:PRINT#wi,i1$"  "i1$;:NEXT
1140 RETURN
1150 REM
1160 REM ********** heading 1-9 ******
*****
1170 PEN#1,2:IF b=4 THEN PRINT#1,j$j$"
"b-3"  "b-2"  "b-1
1180 IF b=8 THEN PRINT#1,j$j$"  "b-4"  "
b-3"  "b-2
1190 IF b=12 THEN PRINT#1,j$j$"  "b-5"
"b-4"  "b-3
1200 PEN#1,1:PRINT#1
1210 RETURN
1220 REM
1230 REM ********* define symbols ******
***
1240 SYMBOL AFTER 128
1250 SYMBOL 246,128,124,32,16,8,5,5,5
1260 SYMBOL 247,8,16,32,64,128,4,2,1
1270 SYMBOL 248,0,0,0,16,0,18,127,128
1280 SYMBOL 249,0,0,0,1,1,36,251,4
1290 SYMBOL 250,0,0,0,192,52,138,133,64
1300 SYMBOL 251,16,8,4,4,2,2,225,30
1310 SYMBOL 252,21,133,68,56,0,0,0,0
1320 SYMBOL 253,33,1,130,124,0,0,0,0
1330 SYMBOL 254,64,32,32,16,16,8,8,0
1340 SYMBOL 129,0,1,3,2,2,2,2,2'g
1350 SYMBOL 130,0,2,255,17,17,17,17,17'g
1360 SYMBOL 131,252,252'r
1370 SYMBOL 132,255,127,127,199,131'g
1380 SYMBOL 133,0,0,128,0,0,16'r
1390 SYMBOL 134,0,0,0,56,68,68,68,56'b
1400 SYMBOL 135,255,255,255,226,193'g
1410 SYMBOL 136,0,0,0,1,0,8'r
1420 SYMBOL 137,0,0,0,28,34,34,34,28'g-b
-g
1430 SYMBOL 139,10,16'r
1440 SYMBOL 140,0,0,120,120,48,50,50,255
'b
1450 SYMBOL 141,0,255,34,34,34,34,34,255
'b
1460 SYMBOL 142,62'g
1470 SYMBOL 143,170,255,255,99,221'b
1480 SYMBOL 144,85,0,0,128,0,8'r
1490 SYMBOL 145,183,193,255,227,221'b
1500 SYMBOL 138,72,54,0,0,0,8'r
1510 SYMBOL 147,2,0,0,0,1'r
1520 SYMBOL 148,0,0,0,0,240'b
1530 SYMBOL 149,0,7,15,240,6,15,15,8'g
1540 SYMBOL 150,0,0,0,0,0,132'b
1550 SYMBOL 151,0,240,248,96,48,120,248,
8'g
1560 SYMBOL 152,0,0,0,128,64,128'r
1570 SYMBOL 153,127,191,223,58,42,57'g
1580 SYMBOL 154,0,64,0,0,16'r
1590 SYMBOL 155,0,0,0,129,129,128,200,12
6'b
1600 SYMBOL 156,254,253,251,92,84,156'g
1610 SYMBOL 157,0,2,0,0,8'r
1620 SYMBOL 158,0,0,0,129,129,1,35,254'b
1630 SYMBOL 160,147,224,128,64,64,64,64,
64'r
```

```
1640 SYMBOL 161,0,22,96,1'g
1650 SYMBOL 162,0,0,2,4,8,10,17,16'b
1660 SYMBOL 163,248,0,0,64,0,64,0,64'r
1670 SYMBOL 164,0,12,0,18'g
1680 SYMBOL 165,0,240,72,4,66,10,81,33'b
1690 SYMBOL 166,106,64,32,0,32'r
1700 SYMBOL 167,21,16,17,122,140,136,142
,113'b
1710 SYMBOL 168,0,0,0,0,0,33'g
1720 SYMBOL 169,64,0,0,0,0,16'g
1730 SYMBOL 170,128'r
1740 SYMBOL 171,21,161,81,10,66,4,72,240
'b
1750 RETURN
1760 REM
1770 REM ************ inks ************
*
1780 CALL &BC02:MODE 2:INK 2,13:INK 1,26
:PEN 1:LIST
1790 REM
1800 REM ******** title a *********
1810 :LOCATE 18,8:FOR w=246 TO 251:PRINT
 CHR$(w);:NEXT
1820 LOCATE 18,9:PRINT CHR$(252)CHR$(253
)"  "CHR$(254)" "
1830 t$=RIGHT$(d$,7):w$="LINHART":ORIGIN
 0,0:TAG:x=142:y=380
1840 FOR t=1 TO 16
1850 PLOT -10,-10,VAL(MID$("132123133212
312123",t,1)):ty=y-t*8
1860 IF t<6 THEN MOVE x-t*24,ty:PRINT"JJ
V";:MOVE x+310+t*24,ty:PRINT"MJL";
1870 MOVE x+96-t*12,ty:PRINT t$;:MOVE x+
182+t*12,ty:PRINT w$;
1880 SOUND 1,14,1,9:NEXT:TAGOFF
1890 WINDOW#1,1,40,10,25:PAPER#1,3:PEN#1
,1:CLS#1
1900 RETURN
1910 REM
1920 REM ********* title b **********
1930 CLS:i$=CHR$(24)
1940 GOSUB 1800
1950 INK 0,0:INK 1,24:INK 2,6:INK 3,11
1960 PEN#1,0:PRINT#1:PRINT#1
1970 PRINT#1," *[   T W I S T   I N   A
  C U B E  ]*":PRINT#1
1980 PRINT#1:PEN#1,1:PRINT#1,"          B
y  J.J."t$" @ 1986":PEN#1,2
1990 LOCATE#1,1,12:PRINT#1,"  "i$"  Ent
er 'Y' for instructions      "i$
2000 PRINT#1:PRINT#1,"  "i$"  Enter 'N'
 if you don't need it!  "i$
2010 :a$=INKEY$:IF a$=""THEN 2010
2020 CLS:IF a$="y"OR a$="Y"THEN GOTO 206
0
2030 RETURN
2040 REM
2050 REM ******** instructions board ***
******
2060 MODE 1:CLS:PRINT"Instructions"
2070 PRINT:PRINT CHR$(238)STRING$(38,198
)CHR$(238)
2080 FOR s=4 TO 23:LOCATE 1,s:PRINT CHR$
(197):LOCATE 40,s:PRINT CHR$(199):NEXT
2090 LOCATE 1,24:PRINT CHR$(238)STRING$(
38,196)CHR$(238)
2100 LOCATE 12,25:PRINT i$"J. J. V i n o
 p a l"i$
2110 WINDOW #4,3,38,5,22:PAPER #4,1:PEN
#4,2:CLS#4
2120 REM
2130 REM ********instructions ********
2140 PRINT#4:PRINT#4
2150 PRINT#4,"   YOU MUST TWIST CLOCKWIS
E NEST 0F":PRINT#4
2160 PRINT#4," 4 SIGNS SO AS TO FINALLY
ARRIVE AT":PRINT#4
2170 PRINT#4," THE SMALL PICTURE ON THE
RIGHT. IT":PRINT#4
2180 PRINT#4," IS LIKE A 2D RUBIC CUBE.
 THE TOP ":PRINT#4
2190 PRINT#4," LEFT MUST BE 4 CARS, TOP
 RIGHT  4":PRINT#4
2200 PRINT#4," TRAINS, BOTTOM  LEFT 4 TA
NKS  AND ":PRINT#4
2210 PRINT#4," BOTTOM RIGHT 4 BICYCLES.
 JJV.     "
2220 GOSUB 2230:CALL &BB18:RETURN
2230 REM
2240 REM ****** sound ******
2250 :ENT -1,1,-7,8,1,7,8,8,1,1,8,1,7:SO
UND 1,60,770,9,0,1
2260 ENT -5,4,1,1,4,-1,1
2270 FOR w=1 TO 2:RESTORE 2300:FOR n=0 T
O 14
2280 READ p:SOUND 2,p*2,30,14-w*3,,5:NEX
T:NEXT
2290 p=0:w=0:n=0:RETURN
2300 DATA 119,119,126,134,134,142,150,17
9,189,213,239,239,179,189,213
```

New slant on Software Marketing

A different approach to marketing software in Australia has just been launched, and may well benefit Amstrad users, especially User Groups. Amsnet International Pty. Ltd. is a new company but has already been given the go-ahead by major UK software houses to import and distribute their software in Australia. These include Sandpiper, Quest and Advanced Memory Systems with more to be finalised shortly.
Naturally, any moves to give a greater selection of software to Amstrad Users can only be in the right direction - but what prompted Amsnet to enter the Amstrad software market through direct marketing?
Neville Wright, Managing Director of Amsnet, explained "my own experience with retail stores has been that while the sales staff are courteous to you, my opinion is that they either speak computer jargon which makes them to appear to be talking down to the client, or they only wish to spend time with clients who are buying an expensive program or system. In discussions with wholesalers and suppliers in the UK, the same opinion was expressed by them to me. They are not happy with the present retail situation. Mind you, there are some stores that are exceptional in their dealings with clients and these certainly deserve praise for their efforts."
Having a background of Direct Marketing and been involved with different network marketing programmes in the past, Neville feels that the computer industry is in need of a marketing system that will get more software and hardware into the marketplace than the traditional way.
User Groups may be interested in this incentive marketing plan by obtaining upto 25% off retail prices (plus bonuses) in becoming distributors themselves. It is not necessary to hold any stock so would not be a drain on resources, and sales could provide an income to help buy equipment, books or software for the group. All this will depend upon "pirates" being put out of business for if anyone is caught illegally copying, they will be prosecuted and cause the group to which they belong to lose all its benefits.
For further information, write to
Amsnet International Pty Ltd, 49 Riversdale Road, Oxenford, QLD 4210.

Make static a blast in the past

The Static Buster products can dissipate static from its two major sources, the computer system and the computer user. The static potential equaliser Model 201 removes negative and positive charges from the video display screen and the user and diverts these charges to a grounding point. It features two coiled leads with easy to connect conductive foam pick up heads, one for the display screen, and one for the keyboard, that plug into the static collector unit with attached ground wire.
The Static Buster Anti-Stat cleaner and wipe are specially developed for one step cleaning and treating of video display screens and peripherals. Anti-Stat cleaner is a concentrated static dissipative solution and large size Anti-Stat wipes are clean room quality, non abrasive, lint free cloths for safe and convenient use.
For more information ring:
Zenology Pty. Ltd. - (03) 233 5764

More Forth

from Petr Lukes

It is heresy, but I prefer my FORTH in lower case. (I never remember to push CAPS LOCK on boot-up). The following routines allow case conversion of the dictionary names and the RAM-disc memory of the ams-FORTH. The dictionary routines are, except for the two indicated lines, fairly general, and should work with other FORTHs with little alteration (as long as they are in RAM, not ROM). They change the case of the dictionary names, by starting at the last compiled word and going up the dictionary ladder until the link-field indicates the end. LATEST will usually point to the FORTH vocabulary. To convert all vocabularies, start with the last one: in ams-FORTH enter 'EDITOR DEFINITIONS' before invoking the conversion. The names and their addresses are displayed as they are converted.
The two lines which deal with conversion of digits (letters serve as digits for bases higher than 10 decimal) are specific to ams-FORTH 1.1; other FORTHs will have different offsets into the two words which convert binary to ASCII and vice versa. They adjust the conversion factor for digits greater than nine: in hex, the decimal value of 10 will be represented by 'a', not 'A', after conversion to lower case. This will limit the highest useable base to 36 decimal, as against 70 if the full ASCII set is used.
The STANDARD requires the word names in upper case, so changing the dictionary names to lower case will produce a non-standard FORTH. The routines should be tested on an extra back-up copy. Any mistake will corrupt the memory and most likely cause a crash sooner or later.
(See program on next page).

```
Block 1
  0 ( 1 CASEFLIP LKS 860710)     HEX  FORTH DEFINITIONS
  1 ( P. LUKES, 26 NOLL ST., TOOWOOMBA, Q 4350)
  2   41 VARIABLE LOR  5A VARIABLE UPR  ( LIMITS FOR WITHIN)
  3 : LOWER    (  -- )     41 LOR !  5A UPR ! ;
  4 : UPPER    (  -- )     61 LOR !  7A UPR ! ;
  5 : WITHIN    ( N -- F)   ( TF:  LOR @ <= N <= UPR @)
  6    DUP LOR @ < SWAP UPR @ > OR 0= ;
  7 : CASE    ( C1 -- C2)    ( RESET/SET BIT 5 OF C1 IF WITHIN)
  8    DUP WITHIN LOR @ 41 -
  9     IF  IF DF AND THEN    ELSE  IF 20 OR THEN   THEN ;
 10    ( BIT MASKS: DF HEX=11011111 BIN   20 HEX=00100000 BIN)
 11 : SCRFLIP  ( FLIP CASE IN RAM-DISC)
 12    CR ." CONVERTING RAM-DISC FROM SCR# 1 TO " LOR @ 41 -
 13    IF ." UPPER" ELSE ." LOWER" THEN ."  CASE" 7 EMIT CR
 14    HI @ LO @ 400 + DO I DUP C@ CASE DUP EMIT SWAP C! LOOP CR ;
 15 -->

Block 2
  0 ( 2 CASEFLIP LKS 860710)      HEX
  1 : SCRU>L    ( FLIP RAM-DISC TO LOWER CASE)  LOWER  SCRFLIP ;
  2 : SCRL>U    ( FLIP RAM-DISC TO UPPER CASE)  UPPER  SCRFLIP ;
  3 : DICNAMES    (  -- )    ( FLIP CASE OF ALL DICTIONARY NAMES)
  4    LATEST ( NFA OF LATEST WORD)
  5    BEGIN CR
  6     DUP 2DUP 2DUP 2DUP              ( 8 COPIES OF NFA)
  7     0E .R 2 SPACES ID. CR           ( ADDR AND ORIGINAL NAME)
  8     C@ 1F AND + DUP ROT 1+ DO I DUP C@ CASE SWAP C! LOOP
  9        ( FLIP FIRST TO PENULTIMATE CHARACTERS OF NAME)
 10     DUP C@ 7F AND CASE 80 OR SWAP C!    ( FLIP LAST CHAR)
 11     10 SPACES ID. CR                ( CONVERTED NAME)
 12     C@ 1F AND + 1+ @ DUP            ( 2 COPIES OF NEXT NFA)
 13    0= UNTIL DROP                    ( TEST FOR END OF DICTIONARY)
 14 -->
 15

Block 3
  0 ( 3 CASEFLIP LKS 860710)      HEX
  1    ." DICTIONARY IS NOW IN " LOR @ 41 -
  2    IF ." UPPER" ELSE ." LOWER" THEN ."  CASE" 7 EMIT CR
  3 ;
  4 : DICU>L    (  -- )     ( FLIP DICTIONARY NAMES TO LOWER CASE)
  5 ( NEXT 2 LINES FOR AMS-FORTH ONLY, TO ENABLE LOWER CASE DIGITS)
  6 (       ' DIGIT 0E + DUP C@  7 = IF 27 SWAP C! ELSE DROP THEN  )
  7 (       ' #     16 + DUP C@  7 = IF 27 SWAP C! ELSE DROP THEN  )
  8    LOWER  DICNAMES
  9 ;
 10 : DICL>U    (  -- )     ( FLIP DICTIONARY NAMES TO UPPER CASE)
 11 ( NEXT 2 LINES FOR AMS-FORTH ONLY, TO ENABLE UPPER CASE DIGITS)
 12 (       ' DIGIT 0E + DUP C@ 27 = IF  7 SWAP C! ELSE DROP THEN  )
 13 (       ' #     16 + DUP C@ 27 = IF  7 SWAP C! ELSE DROP THEN  )
 14    UPPER  DICNAMES
 15 ;
```