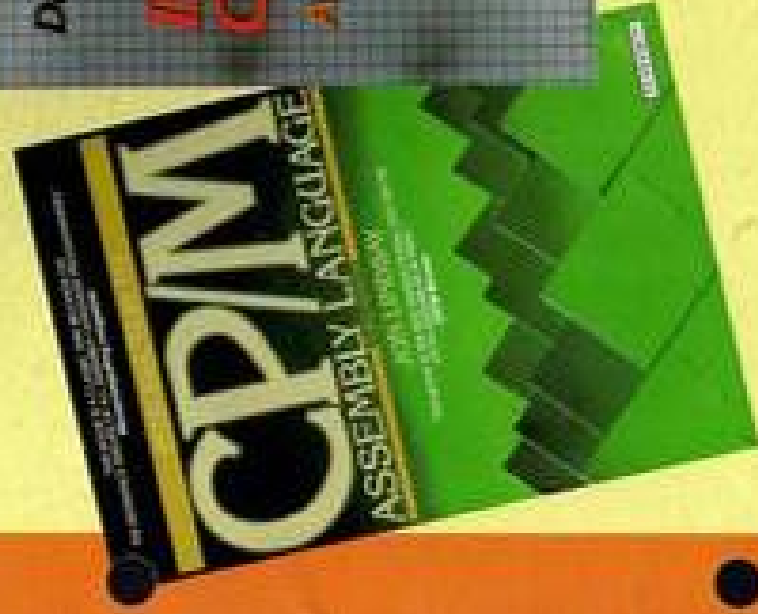


THE AMSTRAD USER

Issue No. 19 \$3.50

August 1986

PCW
Software
and CP/M Books
Reviewed



- 14 PAGES OF PROGRAMS INCLUDING 2 UTILITIES
- SOFTWARE REVIEWS FOR COMPLETE RANGE
- REVIEWS ON BOOKS FOR CP/M and CPC6128
- USER GROUP INFORMATION and CONTACTS

FOR THE NOVICE & EXPERIENCED USER

THE AMSTRAD USER

Issue No. 19
August 1986

Editorial	2
Letters	3
Nationwide User Groups	6
<i>The final hand of Black Jack</i> from Tibor Gyore and Tim Baldock	10
Software Games Reviews by Darren Robinson & Nathan Yim	17
<i>The Amstrad User Hall of Fame</i> plus <i>The Amstrad Top 25</i>	20
Hedge-Hog <i>A prickly game from Ian Barnes</i>	21
Copycat <i>A disc labelling utility</i> from Dennis Shanahan	26
Screen Editor for ams-Forth by Petr Lukes	28
Book Reviews <i>Three CP/M titles plus one for the 6128</i> by Simon Anthony	32
<i>Poke a Monster</i> by P.T. Crowe (SCAUG)	35
<i>Gossip from the UK</i> by our "Spy" in London	36
Progedit <i>A program editing utility</i> Adapted by C.J. Collins	37

For Tape Subscribers, the programs can be found at these approximate positions:

Side 1: BLACKJAK-11 (18 blocks), MONSTERP-205

Side 2: HEDGEHOG-14, COPYCAT-81, PROGEDIT - 101

All enquiries and contacts concerning this Publication should be made to The Amstrad User, Suite 1, 245 Springvale Road, Glen Waverley, Victoria 3150, Australia. [Telephone: (03) 233 9661].

The Amstrad User is published each month by Strategy Publications. Reprinting of articles published in The Amstrad User is strictly forbidden without written permission. All rights reserved. Copyright 1986 by Strategy Publications.

The single copy price of \$3.50 is the recommended retail price only.

The subscription rate (for Australia only) is \$35.00 for 12 issues of the magazine only, or \$75.00 for 12 issues of the magazine plus tape containing all programs appearing in that issue.

Postage is included in the above prices. Overseas prices available upon application. (New Zealand, PNG, Solomon Is, Vanuatu add \$12 airmail)

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, Cartoons-\$5.00 and a rate of \$10.00 per page for programs, articles etc.

Contributions will not be returned unless specifically requested coupled with suitable stamped and return addressed padded bag (for tapes or discs).

THE AMSTRAD USER

G'day,

Here's yet another bumper edition to keep your minds and fingers occupied over the next four weeks or so.

The good news for us is that sales of The Amstrad User have sky-rocketed in the news agents and consequently the magazine has sold out in many places within the first few days of publication. This has left rather too many users without a copy. So we have now increased the print run (again!) to correct the situation. Informed sources tell us that The Amstrad User now outsells most other machine specific magazines in Australia. Not bad in just 18 months and, of course, mainly due to your continued support.

Continuing the success story, the Year Disc has proved exceptionally popular - probably because it saves so much time in keying in and debugging. I just hope that there isn't another 3" disc shortage!

We've also had good vibes from PCW owners who are pleased to see independent assessments of software for their machines. We continue the reviews this month and expect to double the number of pages devoted to this range next month. As I mentioned in the last Editorial, feedback from PCW owners will be slow to start with until they master the word processing capabilities of Locoscript and manage to put printer head to paper and send in their discoveries/hints/tips/programs/et al to share with other users in the same position.

This last paragraph was going to tell you of a special offer which will be running for two months starting from September (no - it's not software), but I have changed my mind. Instead, you will have to wait until next month to see how much you can save on a.....?

See you next month,

Ed

Letters



Having purchased Amsoft's Screen Designer some time ago, I was disappointed that there was no function for a screen dump. Part of my problem was solved when I stumbled upon a dump routine by Roy Eiberg in the December '85 issue of your magazine. When loading the file it is necessary to enter the following program:

```
10 MEMORY 29999: CLEAR
20 LOAD "", 30000
30 CALL 30000
```

I then decided to add the following lines:

```
40 LOAD "SCREENDUMP"
900 to 1380 - the dump routine
2990 RUN
3000 ISCRDMP
```

Lines 40 and 2990 are executed but line 3000 does not run. The computer gives the Ready message at the top of the screen and then I have to type ISCRDMP which has now ruined my graphics. I have also tried loading the dump routine beforehand to get the command into memory, but the memory is being set too low or high as I am returned with a Memory Full error.

The screen dump function has operated in a line number in other programs to produce fantastic dumps.

I would like to know if anyone can help me create unadulterated Screen Designer dumps?

Paul Mezzavia, Geelong, Vic

I wonder whether you could tell me if there is a CP/M call which will return the cursor position, either relative to the start of the screen, or absolute.

I have a 464, and I know how to get it by a system call, but that makes it too system-specific.

I am going to try to bring up MVP Forth from scratch on the 464 and later use it on the 8512, so it will have to be fairly portable.

P. Lukes, Toowoomba, QLD

Shane Kelly says - No! To expand on that a bit, CP/M is built to be as device independent as is possible. To know where the cursor is at any given time would require a knowledge of the hardware that the CP/M system is running on at that time.

Since there are literally hundreds of terminals out there, this would be clearly impossible. You may have seen Wordstar which comes with a program called Install which has a long list of terminals with which it can communicate, accurately positioning the cursor anywhere on that terminal. The necessary codes are supplied in Install, as CP/M has no way of knowing what they could be.

Regarding MVP (Mountain Valley Press?) Forth - you may also be interested in Forth-83 which is the public domain version and is available from the Forth interest group, PO Box 103, Camberwell, Vic 3124 (03) 29 2600.

All letters published in this section attract a payment of five dollars. Any correspondence should be addressed to The Editor, The Amstrad User, Suite 1, 245 Springvale Road, Glen Waverley, Victoria 3150.

For those people who bothered to complete the Galaxian Revenge (UK User Feb '86), the following data changes will transfer controls to the joystick.

```
1420 DATA 3E,4A,CD,1E,BB,
      CA,68,51,945
1450 DATA 3E,4B,CD,1E,BB,
      CA,A0,51, 1002
1740 DATA 3E,4C,CD,1E,BB,
      C8,C3,E0, 1179
8060 DATA 42725,44117,42128,
      44066,43436,40597,37961,
      33526,328556
```

P.T. Crowe, Mulgrave, Vic

I have valued my subscription of The Amstrad User since acquiring my 6128 in February this year. It has been very informative as I expand my knowledge in Amstrad programming. Since gaining my 6128, I have been given quite a few games by other Amstrad users.

Having the system operating onboard my ship, it has been the source of entertainment for a number of the ship's company. Two personnel in particular have attained noteworthy scores in some games. At this moment we have no way of recording our particular scores on each games honour list. Therefore I was wondering if any of your readers could tell me a way/program to enable us to be recorded on the different honour lists.

R.G. Stenhouse, HMAS
Parramatta

I have a problem with Spitfire 40 (Mirrorsoft), the flight simulation game. This program cannot be saved regardless of what I do. The program, when requiring to save, lists the following instruction: "INPUT FILENAME" - which

requires 8 or less characters, then "Prepare the Tape/Disc then press FIRE".

This starts the cassette going, but no "Loading program" is screened to indicate the program is being saved.

J.P. Malusa, Hope Valley, SA

We are not sure we understand the question. If you are trying to load the program then save it again using the SAVE option in the program, it will not work. The SAVE option is there to preserve your flight log only, which can be recalled later.

A program which is being saved will not show a "Loading" message, only a program which is being loaded will do that. A careful read of your user manual is in order!

I was particularly glad to see the program of Yahtzee that Alf Azzopardi had adapted to the Amstrad. I also own one of the so-called 'redundant' computers and as it was a game I enjoyed playing, I was about to attempt the conversion myself before Alf saved me the trouble.

I notice, however, that he has changed the rules slightly in regard to the scoring of straights. For those who are used to playing, any four numbers in a run is a small straight and five in a run is a large straight. I would like to offer the following changes:

```
3460 IF YES>1 THEN 3650
3530 ON COUNT(I) GOTO
      3570,3600,3580,3580,
      3610,3260
3570 IF ADD>14 THEN 3650
3600 IF ADD>19 THEN 3650
      ELSE 3580
3610 IF ADD<16 THEN 3650
      ELSE 3580
3620 IF ADD<21 THEN 3650
      ELSE 3580
```

3690 IF ADD=15 THEN 3710

If anyone knows of a way to make the program score second yahtzees with a bonus, I would be glad to hear from them.

J.B. Murphy, Holesworthy, NSW

I am having trouble with saving a program on disc that has now exceeded 38k in length. The computer returns the same message "Memory Full" each time. When I check the memory content still remaining in the computer, it claims to have over 4k left yet I cannot utilise it to hold a program. A colleague of mine is having the same problem with the recently published Eliza program. He claims his computer will not accept the full program. We both have the CPC6128 model.

Could you please explain why we cannot utilize the supposed 42k+ available to the user or what we must do to be able to do so.

Barrie M. Eaton, Keysboro', Vic

Some people have reported experiencing problems when entering large programs. When a program utilises variables this effects the VLT (Variable List Table) which stores the pointers for the strings in 3 bytes. The more variables, the more space required.

In the first place, if you are entering a large program, SAVE it as you go along before RUNNING it until the whole program is entered. You should then SAVE and reLOAD the program for testing.

If you have got stuck with "Memory full" when saving, type CLEAR and try again. If that doesn't work enter PRINT FRE(1) which will give you the amount of free space left in your machine. If it is less than 4096 you will not be able to save your program. The

only way out of this is to delete lines of your program in order to reduce it to a size capable of being saved. You can then reLOAD and put the lines back in.

An article which will provide more information on this kind of problem is planned for publication in the near future.

Your 6128 contains 2 x 64k sections. Whilst CP/M Plus uses both, Basic is confined to just 64k unless you use the extra commands provided with 'Bank Manager'. Eliza just fits into the first 64k after Basic and the VLT have taken their slice of the action.

I wish to advise you that I do not intend to renew my subscriptions for The Amstrad User (Aust) or The Amstrad Computer User (UK) since I am about to dispose of my Amstrad CPC664 and obtain an IBM compatible.

It is perhaps unfortunate that the Amstrad CPC664 was promoted as suitable for running a small business. I have found Tasword to be an excellent word processor and I am happy enough with it. However, I need a Database package for the business to store up to a thousand or more addresses and phone numbers of clients and about six months ago I purchased Micropen. I've had so many troubles with the software locking me out as far as updating the file is concerned that I have decided it worth the expense of going for the industry standard which is extensively used by business (and our accountant - which is handy). I hope you continue to provide the high level of support for continuing Amstrad Users despite the probability that Amstrad will release a 16 bit machine with a megabyte of RAM shortly. Trying to provide support for an ever increasing number of models must be rather daunting.

Bernard Grainger, Mt Evelyn, Vic

I wish to pen a reply to N. McMartin's letter in The Amstrad User, July 1986.

Mr Kelly is indeed correct when he details how to connect to Viatel/Videotex services supporting the split 1200/75 baud rate, however this is only half the story when considering communications on the Amstrad.

Due to the fact that the Amstrad is a European computer, it supports communication protocols for that part of the world. However, here in Australia we have adopted the CCITT (read European) standard of data transmission over the telephone lines whereas the actual protocols of the data we transmit are mostly Christensen protocols (also called XMODEM, MODEM7 and YAM) which emanate from the United States, since that is where Remote Bulletin Board Services (RBBS) first started. This means that for a communications system to be useful in Australia your modem has to "speak" CCITT and your computer has to "speak" Christensen. Consequently the Amstrad can talk to RBBS via the TERMINAL command (after setting the appropriate baud rates) but many of the other goodies associated with RBBS usage are unavailable to the ordinary 'Amstrad-er'. Luckily help is at hand!

There exists a number of public domain (read free) terminal programs which will run via CP/M 2.2 and CP/M Plus. Unfortunately in the past these types of programs have required some work using a macro assembler, or a C compiler in order to run, but now there are a number of good programs custom made for the Amstrad CPCs. Unfortunately you generally need a terminal program to download the terminal programs. The terminal programs are used to upload and download programs to RBBS and other machines using the same

protocols and also save the terminal session you are engaged in.

Thus CP/M comes into its own as it is the common format between differing machines, meaning that you can write programs which can be run on other machines, and other people's efforts will run on the Amstrad (assuming that the program is not too large). So RBBS becomes giant swap meets to exchange software, ideas and views (not necessarily related to computing).

With respect to the question of downloading software from Viatel, unfortunately at the moment there seem to be no real ideas primarily because no one is interested in providing the software on Viatel, and also because the standards differ between Australia and the UK which is where we originally obtained the Viatel system.

One word of warning should come with every modem: TIME COSTS MONEY, which means that the more you talk to other computers, the more it costs, particularly if it's ISD as it generally is in my case!

I hope that this may be of assistance.

D. MacKinnon, Unanderra, NSW

If you have any views or comments to make (and they are fit for publication!) we are happy to consider them.

But please remember that we cannot enter into any personal correspondence.

NATIONWIDE USER GROUPS

It's quite amazing that since December last year the number of groups has grown from seventeen to thirty-two!

We have another six to welcome this month. In Queensland, the **Western Suburbs Amstrad Users Group** has been established with much assistance from Keith Jarrot and Peter Wighton. They meet in Jindalee whilst further north the **Mackay Amstrad Users Group** have advised us of their formation.

In NSW the **Central Coast Amstrad Users Club** which was formed a few months ago has some 25 people attending meetings in Long Jetty, and the **Port Macquarie Amstrad Users Group** has just registered with us. Also in NSW, we are pleased to welcome **Blue Mountain Amstrad Users** (with our apologies for not getting you in the list sooner).

Finally, in WA, **AMSWEST (Blackwood)**, who are affiliated with the main **AMSWEST** (Perth) group, are "only a small group but growing at a consistent rate". Details of these and all other groups can be found in the next few pages.

WESTERN AUSTRALIA

AMSWEST (Perth)

President: Tony Clitheroe (09 275 1257)
Vice President: Steve Cushnahan (09 445 2062)
Secretary: Mrs. P. T. Ardron (09 361 8975)
Treasurer: John Firth

Regular meetings take place at a venue in Shenton Park on the first and third Tuesdays of each month starting at 7.30p.m.

AMSWEST (Blackwood) USERS GROUP

This small group is affiliated to **AMSWEST** (Perth). For further details contact George Muscat on (097) 61 1488.

SOUTHSIDE AMSTRAD USER CLUB

President: John Marshall (09 390 7335)
Secretary: Debbie Hoffman (09 459 8702)
Treasurer: Eric Tytherleigh (09 390 8865)
Librarian: Roy Depurouzel (09 457 9026)

SAUC meets from 7.00 p.m. every 2nd and 4th Tuesday of each month at Gosnells Scout Hall on the corner of Verra and Corfield Streets, Gosnells. All meetings are socially orientated with a minimum of business matters and can include software and hardware demonstrations, discussions or lectures which all prove to be helpful for beginners or advanced users alike. The club has an extensive library of tapes, discs, magazines etc. and discounts have been obtained from most local dealers and are available to financial members. Contact can be made with any of the above officers or by writing to The Secretary, Southside Amstrad Users Club, PO Box 324, Gosnells, WA 6110.

ROCKINGHAM-KWINANA AMSTRAD USER GROUP

President: Bob Harwood
Vice-President: Keith Gaisford
Treasurer: Rob MacIlroy
Secretary: Keith Saw (095 27 6519)

This new group meets at 7.30 p.m. at the Cooeloongup Primary School, Westerly Way, Cooeloongup (Rockingham), every second Wednesday. Further details can be obtained from Keith Saw on the above number or by writing to 29 Millgrove Avenue, Cooeloongup, WA 6168.

SOUTH AUSTRALIA

AMSTRAD COMPUTER CLUB INC. (SA)

President: Chris Sowden (08 295 5923)
Vice Pres: Frank Matzka (08 382 2101)
Treasurer: Les Jamieson (08 356 9612)

The group now meets each Tuesday at the Church Hall, 15 Clayton Avenue, Plympton between 6.30 p.m. and 9.00 p.m. Any of the above officers can be contacted for further details and correspondence can be addressed to PO Box 210, Parkholme, 5043.

PORT PIRIE AMSTRAD USER GROUP

President: Rick Cable (086 32 5967)
Treasurer: Dave Green (086 32 6834)
Secretary: Keith Partridge (086 32 3919)

The group meets at 7.30 p.m. every first and third Monday of each month at the Way Inn Coffee Lounge, Ellen Street, Port Pirie City Centre. Meetings are well attended with members from Pt. Broughton, Warnertown and even Burra. For further details contact Rick Cable who will advise on the benefits of belonging to this group.

API COMPUTER GROUP

Contact: Mike Denieuwe (08 225 5995)

The Australian Post-Tel Institute has a number of computer groups, almost entirely dedicated to Amstrads. Monthly meetings are held in:

Blair Athol - 2nd Tuesday of the month at 5.30 p.m.
Elizabeth - last Tuesday of the month at 5.30 p.m.
St. Marys - 3rd Tuesday of the month at 5.15 p.m.

with a central meeting place in the City at various times during the month. Membership is \$15 per year. For more details, contact Mike Denieuwe during office hours on the above telephone number, or on 08-297 8500 after hours.

The final hand of Black-Jack

from Tibor Gyore and Tim Baldock

... or Pontoon, or Twenty-one. To a hacker at three o'clock in the morning it really doesn't matter what the name of the program is!

This is the last part of the mega-listing which hopefully won't, but probably will, send more new members to the "RSI Club". A tip when keying in - save the program frequently, and if you want to test it, load a saved version. This should avoid the dreaded 'Memory Full' message.

```
7530 ;
7530 ; --BETTING MENU DRIVER--
7540 ;
7610 FOR DELAY=1 TO 50:NEXT
7620 IF Y1<1 THEN Y1=5
7630 IF Y1>6 THEN Y1=2
7640 IF Y1=2 THEN PEN 1 ELSE PEN 0
7650 IF Y1=2 THEN PAPER 0 ELSE PAPER 1
7660 LOCATE X4+1, Y4-1:PRINT SPC(1);"ONE"
;SPC(2)
7670 PEN 0:PAPER 1
7680 IF Y1=4 THEN PEN 1 ELSE PEN 0
7690 IF Y1=4 THEN PAPER 0 ELSE PAPER 1
7700 LOCATE X4-1, Y4+3:PRINT SPC(1);"TWO"
;SPC(2)
7710 PEN 0:PAPER 1
7720 IF Y1=6 THEN PEN 1 ELSE PEN 0
7730 IF Y1=6 THEN PAPER 0 ELSE PAPER 1
7740 LOCATE X4+1, Y4+5:PRINT SPC(1);"FIVE"
;SPC(1)
7750 PEN 0:PAPER 1
7760 IF TOGGLE=1 THEN RETURN
7770 IF (INKEY(0)=0 OR JOY(0)=1) AND Y1<1
THEN FOR DELAY=1 TO 50:NEXT:Y1=Y1-2
7780 IF (INKEY(2)=0 OR JOY(0)=2) AND Y1<7
THEN FOR DELAY=1 TO 50:NEXT:Y1=Y1+2
7790 IF (INKEY(9)=0 OR JOY(0)=16) THEN FOR
DELAY=1 TO 300:NEXT:GOTO 7610
7800 GOTO 7610
7810 IF T=1 THEN T=0:GOTO 7640
7820 Y2=9:TOGGLE=1:GOSUB 6730:TOGGLE=0:G
OSUB 8200
7830 IF CANCEL=1 THEN CANCEL=0:Y1=4:GOTO
7180
7840 IF Y1=2 THEN LEFT=1
7850 IF Y1=4 THEN BUT=2
```

```
7860 IF Y1=6 THEN BET=5
7870 BETTOT=BETTOT+BET
7880 MONEY=MONEY-BET
7890 IF MONEY<0 THEN MONEY=MONEY+BETTOT
:GOTO 12800
7900 LOCATE 26,21:PRINT"CHIPS";MONEY
7910 RETURN
7920 ;
7930 ; --ACE MENU DRIVER--
7940 ;
7950 PAPER 1:PEN 0
7960 Y1=4:LOCATE X4+1, Y4+1:PEN 3:PRINT S
PC(1);"ACE";SPC(2)
7970 IF Y1<3 THEN Y1=6
7980 IF Y1>6 THEN Y1=4
7990 IF Y1=4 THEN PEN 1 ELSE PEN 0
8000 IF Y1=4 THEN PEN 1 ELSE PEN 0
8010 IF Y1=4 THEN PAPER 0 ELSE PAPER 1
8020 LOCATE X4+1, Y4+3:PRINT SPC(1);"ONE"
;SPC(2)
8030 PEN 0:PAPER 1
8040 IF Y1=6 THEN PEN 1 ELSE PEN 0
8050 IF Y1=6 THEN PAPER 0 ELSE PAPER 1
8060 LOCATE X4+1, Y4+5:PRINT"ELEVEN"
8070 IF TOGGLE=1 THEN RETURN
8080 IF (INKEY(0)=0 OR JOY(0)=1) AND Y1>3
THEN FOR DELAY=1 TO 50:NEXT:Y1=Y1-2
8090 IF (INKEY(2)=0 OR JOY(0)=2) AND Y1<7
THEN FOR DELAY=1 TO 50:NEXT:Y1=Y1+2
8100 IF (INKEY(9)=0 OR JOY(0)=16) THEN FOR
DELAY=1 TO 300:NEXT:GOTO 8120
8110 GOTO 7970
8120 Y2=9:TOGGLE=1:GOSUB 6760:TOGGLE=0:G
OSUB 8200
8130 IF CANCEL=1 THEN CANCEL=0:Y1=4:GOTO
7970
8140 IF Y1=4 THEN ACE=1
8150 IF Y1=6 THEN ACE=11
8160 RETURN
8170 ;
8180 ; --OK & RETURN MENU--
8190 ;
8200 IF Y2>11 THEN Y2=9
8210 IF Y2<9 THEN Y2=11
8220 IF Y2=9 THEN PEN 1 ELSE PEN 0
8230 IF Y2=9 THEN PAPER 0 ELSE PAPER 1
8240 LOCATE X4+1, Y4+9:PRINT SPC(1);"OK";
SPC(3)
8250 IF Y2=11 THEN PEN 1 ELSE PEN 0
8260 IF Y2=11 THEN PAPER 0 ELSE PAPER 1
8270 LOCATE X4+1, Y4+11:PRINT"RETURN"
8280 IF TOGGLE=1 THEN RETURN
8290 IF (INKEY(0)=0 OR JOY(0)=1) AND Y2<
```



```

8 THEN FOR DELAY=1 TO 100:NEXT:Y2=Y2-2
8300 IF (INKEY(2)=0 OR JOY(0)=2) AND Y2<
12 THEN FOR DELAY=1 TO 100:NEXT:Y2=Y2+2
8310 IF INKEY(9)=0 OR JOY(0)=16 THEN FOR
DELAY=1 TO 300:NEXT:GOTO 8330
8320 GOTO 8200
8330 PEN 0:PAPER 1
8340 IF Y2=9 THEN PAPER #6,2:CLS#6:RETUR
N
8350 IF Y2=11 THEN PAPER #6,2:CLS#6:CANC
EL=1
8360 RETURN
8370 '
8380 ' --YES & NO SELECTION--
8390 '
8400 X5=X6
8410 X7=X6+4
8420 LOCATE X6+3,Y6:PRINT"/"
8430 IF X5=X6 THEN PEN 1 ELSE PEN 0
8440 IF X5=X6 THEN PAPER 0 ELSE PAPER CO
8450 LOCATE X6,Y6:PRINT"YES"
8460 IF X5=X7 THEN PEN 1 ELSE PEN 0
8470 IF X5=X7 THEN PAPER 0 ELSE PAPER CO
8480 LOCATE X7,Y6:PRINT"NO"
8490 IF (INKEY(8)=0 OR JOY(0)=4) AND X5=
X7 THEN X5=X6
8500 IF (INKEY(1)=0 OR JOY(0)=8) AND X5=
X6 THEN X5=X7
8510 IF INKEY(9)=0 OR JOY(0)=16 THEN 853
0
8520 GOTO 8430
8530 IF X5=X6 THEN YES=1
8540 IF X5=X7 THEN NO=1
8550 PEN 0:PAPER 1
8560 RETURN
8570 ' --BANKERS CARDS--
8580 '
8590 '
8600 ' --DEAL BANKERS CARD NO. (1)--
8610 '
8620 Y=Y+13
8630 GOSUB 2000
8640 FOR DELAY=1 TO 1000:NEXT
8650 GOSUB 1680
8660 GOSUB 1600
8670 GOSUB 3470
8680 F1=CARD
8690 IF F>10 THEN F=10
8700 IF F=1 THEN F=11
8710 BANKTOT=BANKTOT+F
8720 RETURN
8730 '
8740 ' --DEAL BANKERS CARD NO. (2)--
8750 '
8760 Y=Y+13
8770 GOSUB 2000
8780 RETURN
8790 Y=Y+13:X=6
8800 PAPER 1
8810 PAPER #3,2
8820 CLS #3
8830 GOSUB 1680
8840 GOSUB 1600
8850 GOSUB 3470
8860 G=CARD

```

```

8870 IF F>10 THEN F=10
8880 IF F<>1 THEN 8910
8890 IF BANKTOT=10 THEN BANKTOT=BANKTOT+
11:GOTO 8920
8900 IF BANTOT>=7 THEN BANKTOT=BANKTOT+1
1:GOTO 8920
8910 BANKTOT=BANKTOT+F
8920 IF BANKTOT=21 THEN BPONTOON=1 ELSE
BPONTOON=0
8930 IF BPONTOON=1 THEN 9320
8940 IF BANKTOT>=17 THEN 9320
8950 '
8960 ' --DEAL BANKERS CARD NO. (3)--
8970 '
8980 X=X+5
8990 GOSUB 2000
9000 FOR DELAY=1 TO 1000:NEXT
9010 GOSUB 1680
9020 GOSUB 1600
9030 GOSUB 3470
9040 H=CARD
9050 IF F>10 THEN F=10
9060 BANKTOT=BANKTOT+F
9070 IF BANKTOT>=17 THEN 9320
9080 '
9090 ' --DEAL BANKERS CARD NO. (4)--
9100 '
9110 X=X+5
9120 GOSUB 2000
9130 FOR DELAY=1 TO 1000:NEXT
9140 GOSUB 1680
9150 GOSUB 1600
9160 GOSUB 3470
9170 I1=CARD
9180 BANKTOT=BANKTOT+F
9190 IF BANKTOT>=17 THEN 9320
9200 '
9210 ' --DEAL BANKERS CARD NO. (5)--
9220 '
9230 X=X+5
9240 GOSUB 2000
9250 FOR DELAY=1 TO 1000:NEXT
9260 GOSUB 1680
9270 GOSUB 1600
9280 GOSUB 3470
9290 J=CARD
9300 BANKTOT=BANKTOT+F
9310 IF BANKTOT<=21 THEN BFIVEUNDER=1 EL
SE BFIVEUNDER=0
9320 PEN 0:PAPER 1
9330 R=2:GOSUB 6970
9340 '
9350 ' --DECIDE WHO WINS THE HAND--
9360 ' --AND DISPLAY OUTCOME --
9370 '
9380 IF PONTOON=1 AND BPONTOON=1 THEN 95
40
9390 IF FIVEUNDER=1 AND BFIVEUNDER=1 THE
N 9640
9400 IF FIVEUNDER=1 AND BPONTOON=1 THEN
9660
9410 IF BFIVEUNDER=1 AND PONTOON=1 THEN
9750
9420 IF PONTOON=1 THEN 10270
9430 IF FIVEUNDER=1 THEN 10330

```

9440 IF PONTON=1 THEN 10450
 9450 IF BEFIVEUNDER=1 THEN 10540
 9460 IF PLAYTOT>21 THEN 9950
 9470 IF BANKTOT>21 THEN 10020
 9480 IF BANKTOT>=PLAYTOT THEN 10090
 9490 IF PLAYTOT>BANKTOT THEN 10180
 9500
 9510 '--BANKER AND PLAYER BOTH HAVE--
 9520 '--PONTON. BANK WINS --
 9530
 9540 LOCATE 26,16:PEN 0:PRINT"THE PLAYER
 HAS"
 9550 LOCATE 26,17:PEN 3:PRINT"PONTOON."
 9560 LOCATE 26,18:PEN 0:PRINT"THE BANKER
 HAS"
 9570 LOCATE 26,19:PEN 3:PRINT"PONTOON."
 9580 LOCATE 26,20:PEN 3:PRINT"THE BANK W
 INS"
 9590 GOTO 10600
 9600
 9610 '--BANKER AND PLAYER BOTH HAVE--
 9620 '--FIVE & UNDER. BANK WINS --
 9630
 9640 LOCATE 26,16:PEN 0:PRINT"THE PLAYER
 HAS"
 9650 LOCATE 26,17:PEN 3:PRINT"FIVE & UND
 ER."
 9660 LOCATE 26,18:PEN 0:PRINT"THE BANKER
 HAS"
 9670 LOCATE 26,19:PEN 3:PRINT"FIVE & UND
 ER."
 9680 LOCATE 26,20:PEN 3:PRINT"THE BANK W
 INS"
 9690 GOTO 10600
 9700
 9710 '--PLAYER HAS PONTOON AND THE--
 9720 '--BANK HAS FIVE & UNDER --
 9730 '--THE PLAYER WINS --
 9740
 9750 LOCATE 26,16:PEN 0:PRINT"THE PLAYER
 HAS"
 9760 LOCATE 26,17:PEN 3:PRINT"PONTOON."
 9770 LOCATE 26,18:PEN 0:PRINT"THE BANKER
 HAS"
 9780 LOCATE 26,19:PEN 3:PRINT"FIVE & UND
 ER"
 9790 LOCATE 26,20:PEN 3:PRINT"PLAYER WIN
 S"
 9800 GOTO 10600
 9810
 9820 '--BANKER HAS PONTOON AND THE--
 9830 '--PLAYER HAS FIVE & UNDER --
 9840 '--THE BANK WINS --
 9850
 9860 LOCATE 26,16:PEN 0:PRINT"THE BANKER
 HAS"
 9870 LOCATE 26,17:PEN 3:PRINT"PONTOON."
 9880 LOCATE 26,18:PEN 0:PRINT"THE PLAYER
 HAS"
 9890 LOCATE 26,19:PEN 3:PRINT"FIVE & UND
 ER"
 9900 LOCATE 26,20:PEN 3:PRINT"THE BANK W
 INS"
 9910 GOTO 10600
 9920

9930 '--PLAYER BUSTS BANK WINS--
 9940
 9950 LOCATE 26,16:PRINT"THE PLAYER HAS"
 9960 LOCATE 26,17:PEN 3:PRINT"BUSTED." :P
 EN 0
 9970 LOCATE 26,18:PRINT"THE BANK WINS":P
 EN 0
 9980 GOTO 10600
 9990
 10000 '--BANKER BUSTS PLAYER WINS--
 10010
 10020 LOCATE 26,16:PRINT"THE BANKER HAS"
 10030 LOCATE 26,17:PEN 3:PRINT"BUSTED." :
 PEN 0
 10040 LOCATE 26,18:PEN 0:PRINT"PLAYER WI
 NS":PEN 0
 10050 MONEY=MONEY+BETTOT+BETTOT:GOTO 106
 00
 10060
 10070 '--BANKER WINS HAND--
 10080
 10090 LOCATE 26,16:PRINT"THE BANKER HAS"
 10100 LOCATE 26,17:PEN 0:PRINT"A TOTAL 0
 F":PEN 3:PRINT BANKTOT
 10110 LOCATE 26,18:PEN 0:PRINT"THE PLAYE
 R HAS"
 10120 LOCATE 26,19:PEN 0:PRINT"A TOTAL 0
 F":PEN 3:PRINT PLAYTOT
 10130 LOCATE 26,20:PEN 3:PRINT"THE BANK
 WINS"
 10140 GOTO 10600
 10150
 10160 '--PLAYER WINS HAND--
 10170
 10180 LOCATE 26,16:PRINT"THE PLAYER HAS"
 10190 LOCATE 26,17:PEN 0:PRINT"A TOTAL 0
 F":PEN 3:PRINT PLAYTOT
 10200 LOCATE 26,18:PEN 0:PRINT"THE BANKB
 R HAS"
 10210 LOCATE 26,19:PEN 0:PRINT"A TOTAL 0
 F":PEN 3:PRINT BANKTOT
 10220 LOCATE 26,20:PEN 3:PRINT"PLAYER WI
 NS"
 10230 MONEY=MONEY+BETTOT+BETTOT:GOTO 106
 00
 10240
 10250 '--FLAYER HAS PONTOON--
 10260
 10270 LOCATE 26,16:PEN 0:PRINT"THE PLAYE
 R HAS"
 10280 LOCATE 26,17:PEN 3:PRINT"PONTOON."
 10290 LOCATE 26,18:PEN 0:PRINT"THE BANK
 PAYS"
 10300 LOCATE 26,19:PRINT"DOUBLE YOUR"
 10310 LOCATE 26,20:PEN 3:PRINT"BET."
 10320 MONEY=MONEY+BETTOT+BETTOT:GOTO:G
 OTO 10600
 10330
 10340 '--PLAYER HAS FIVE & UNDER--
 10350
 10360 LOCATE 26,16:PEN 0:PRINT"THE PLAYE
 R HAS"
 10370 LOCATE 17:PEN 3:PRINT"FIVE & UN
 DER."
 10380 LOCATE 26,18:PEN 0:PRINT"THE BANK

```

PAYS"
10390 LOCATE 26,19::PRINT"DOUBLE YOUR"
10400 LOCATE 26,20:PEN 3:PRINT"BET."
10410 MONEY=MONEY+BETTOT+BETTOT+BETTOT:G
OTO 10600
10420 ,
10430 '--BANKER HAS PONTOON--
10440 ,
10450 LOCATE 26,16:PEN 0:PRINT"THE BANKE
R HAS"
10460 LOCATE 26,17:PEN 3:PRINT"PONTOON."
10470 LOCATE 26,18:PEN 0:PRINT"YOU PAY B
ANK"
10480 LOCATE 26,19:PEN 0:PRINT"DOUBLE YO
UR"
10490 LOCATE 26,20:PEN 3:PRINT"BET."
10500 MONEY=MONEY-BETTOT:GOTO 10600
10510 ,
10520 '--BANKER HAS FIVE & UNDER--
10530 ,
10540 LOCATE 26,16:PEN 0:PRINT"THE BANKE
R HAS"
10550 LOCATE 26,17:PEN 3:PRINT"FIVE & UN
DER."
10560 LOCATE 26,18:PEN 0:PRINT"YOU PAY B
ANK"
10570 LOCATE 26,19:PEN 0:PRINT"DOUBLE YO
UR"
10580 LOCATE 26,20:PEN 3:PRINT"BET."
10590 MONEY=MONEY-BETTOT:GOTO 10600
10600 ,
10610 '--END OF HAND RESULTS--
10620 ,
10630 FOR DELAY=1 TO 4500:NEXT
10640 PEN 0
10650 FOR I=16 TO 20
10660 LOCATE 26,1:PRINT SPC(14)
10670 NEXT
10680 IF MONEY <=0 THEN 12800
10690 LOCATE 26,16:PRINT"YOU NOW HAVE"
10700 LOCATE 26,17:PRINT MONEY;"CHIPS"
10710 LOCATE 26,18:PEN 0:PRINT"WOULD YOU
LIKE"
10720 LOCATE 26,19:PRINT"ANOTHER HAND"
10730 X6=26:Y6=20:GOSUB 8400
10740 IF YES=1 THEN YES=0:PLAYTOT=0:BAK
TOT=0:BET=0:BETTOT=0:PONTOON=0:EPONTOON=
0:FIVEUNDER=0:BFIVEUNDER=0:GOTO 470
10750 IF NO=1 THEN NO=0:W=1:GOTO 12800
10760 GOTO 10660
10770 ,
10780 '-- INTRODUCTION & HEADING PAGE--
10790 ,
10800 PAPER 2:PEN 0:INK 0,22:INK 3,22:NO
DE 1:CLS
10810 GOSUB 11190
10820 WINDOW #1,2,39,2,24
10830 PEN 0:LOCATE 15,3:PRINT"ab"
10840 LOCATE 15,4:PRINT"cdefghijklm"
10850 LOCATE 15,5:PRINT"nopqrstuvwxyz"
10860 PEN 3:LOCATE 9,4:PRINT CHR$(137);C
HR$(138)
10870 LOCATE 9,5:PRINT CHR$(139);CHR$(14
0)
10880 PEN 0:LOCATE 12,4:PRINT CHR$(145);
CHR$(146)
10890 LOCATE 12,5:PRINT CHR$(147);CHR$(1
48)
10900 PEN 3:LOCATE 26,4:PRINT CHR$(133);
CHR$(134)
10910 LOCATE 26,5:PRINT CHR$(135);CHR$(1
36)
10920 PEN 0:LOCATE 29,4:PRINT CHR$(141);
CHR$(142)
10930 LOCATE 29,5:PRINT CHR$(143);CHR$(1
44)
10940 INK 0,0:INK 3,6
10950 X=13:Y=7:R=0:GOSUB 2000:F=1:SUIT=2
:FOR DELAY=1 TO 1000:NEXT
10960 GOSUB 1600
10970 GOSUB 3470
10980 X=X+5:R=0:GOSUB 2000:F=13:SUIT=1:F
OR DELAY=1 TO 1000:NEXT
10990 GOSUB 1600
11000 GOSUB 3470
11010 PAPER 2
11020 A3$="WRITTEN BY - TIM BALDOCK "
11030 Y3=20
11040 I=38:D=1
11050 LOCATE 1,Y3:PRINT LEFT$(A3$,D)
11060 I=I-1:D=D+1:IF I<8 THEN 11090
11070 FOR DELAY=1 TO 100:NEXT
11080 GOTO 11050
11090 IF LEFT$(A3$,1)="G" THEN 11130
11100 A3$="GRAPHICS BY - TIBOR GYORE "
11110 Y3=Y3+1
11120 GOTO 11040
11130 LOCATE 10,23:PEN 3:PRINT"INSTRUCTI
ONS":X6=25:Y6=23:CO=2:GOSUB,8400:CO=1:PA
PER 2
11140 IF YES=1 THEN YES=0:GOTO 11310
11150 IF NO=1 THEN NO=0:RETURN
11160 ,
11170 '--BORDER AROUND SCREEN--
11180 ,
11190 FOR I=1 TO 24 STEP 4
11200 LOCATE 1,1+I:PEN 0:PRINT CHR$(226)
:PEN 3:LOCATE 40,1+I:PRINT CHR$(227)
11210 LOCATE 1,1+2:PEN 3:PRINT CHR$(227)
:PEN 0:LOCATE 40,1+2:PRINT CHR$(226)
11220 LOCATE 1,1+3:PEN 0:PRINT CHR$(229)
:PEN 3:LOCATE 40,1+3:PRINT CHR$(228)
11230 LOCATE 1,1+4:PEN 3:PRINT CHR$(228)
:PEN 0:LOCATE 40,1+4:PRINT CHR$(229)
11240 NEXT
11250 FOR I=1 TO 40 STEP 4
11260 LOCATE 1,1:PEN 0:PRINT CHR$(226)::
PEN 3:PRINT CHR$(227)::PEN 0:PRINT CHR$(
229)::PEN 3:PRINT CHR$(226)
11270 LOCATE 0+1,25:PEN 0:PRINT CHR$(226)
::PEN 3:PRINT CHR$(227)::PEN 0:PRINT CH
R$(229)::PEN 3:PRINT CHR$(228);
11280 NEXT
11290 RETURN
11300 ,
11310 '--INSTRUCTIONS--
11320 ,
11330 PAPER #1,2:CLS #1
11340 PEN 3:LOCATE 3,3:PRINT"INSTRUCTION
3":PEN 0

```

11350 LOCATE 3,5:PEN 0:PRINT"PONTOON IS
 A GAME OF CHANCE AND IS"
 11360 LOCATE 3,6:PRINT"SIMILAR TO BLACKJ
 ACK AND TWENTY-ONE."
 11370 PEN 3:LOCATE 3,9:PRINT"OBJECT OF T
 HE GAME.":PEN 0
 11380 LOCATE 3,11:PRINT"IN ORDER TO WIN
 THE HAND YOU MUST"
 11390 LOCATE 3,12:PRINT"GET A SCORE
 GREATER THAN THE"
 11400 LOCATE 3,13:PRINT"BANKERS. THIS SC
 ORE IS DERIVED FROM"
 11410 LOCATE 3,14:PRINT"THE SUM OF THE V
 ALUES OF THE CARDS"
 11420 LOCATE 3,15:PRINT"IN YOUR HAND. IF
 YOUR SCORE EXCEEDS"
 11430 LOCATE 3,16:PRINT"TWENTY-ONE, YOU
 LOSE. (BUST)"
 11440 GOSUB 12720
 11450 CLS #1
 11460 PEN 3:LOCATE 3,3:PRINT"PICTURE CAR
 DS.":PEN 0
 11470 LOCATE 3,4:PRINT"A PICTURR CARD I
 S ONE THAT HAS A"
 11480 LOCATE 3,5:PRINT"PICTURE ON ITS'
 FACR. THESE CARDS"
 11490 LOCATE 3,6:PRINT"ARR KING, QUE
 EN AND JACK. THE"
 11500 LOCATE 3,7:PRINT"VALUE OF THESE
 CARDS IS EQUAL TO"
 11510 LOCATE 3,8:PRINT"TEN. "
 11520 X=3:Y=10:F=13:GOSUB 2000:FOR I=1 T
 O 1000:NEXT
 11530 GOSUB 1600:GOSUB 3470
 11540 X=10:Y=10:F=12:GOSUB 2000:FOR I=1
 TO 1000:NEXT
 11550 GOSUB 1600:GOSUB 3470
 11560 X=29:Y=10:F=11:GOSUB 2000:FOR I=1
 TO 1000:NEXT
 11570 GOSUB 1600:GOSUB 3470
 11580 PAPER 2
 11590 GOSUB 12720
 11600 CLS #1
 11610 PEN 3:LOCATE 3,3:PRINT"PONTOON":PE
 N 0
 11620 LOCATE 3,5:PRINT"TO OBTAIN PONTOO
 N YOUR HAND MUST"
 11630 LOCATE 3,6:PRINT"CONSIST OF CARDS
 WITH THE VALUE TEN"
 11640 LOCATE 3,7:PRINT"AND AN ACE."
 11650 LOCATE 3,8:PRINT"IF YOU HAVE PONT
 OON AND THE BANKER"
 11660 LOCATE 3,9:PRINT"DOES NOT, YOU WI
 LL BE PAID TWICE"
 11670 LOCATE 3,10:PRINT"THE AMOUNT THAT
 YOU BET."
 11680 LOCATE 3,11:PRINT"IF THE BANKER H
 AS PONTOON AND YOU"
 11690 LOCATE 3,12:PRINT"DO NOT THEN YOU
 MUST PAY TWICE THE"
 11700 LOCATE 3,13:PRINT"AMOUNT OF YOUR B
 ET."
 11710 LOCATE 3,14:PRINT"IN THE EVENT TH
 AT THE BANKER ALSO"

11720 LOCATE 3,15:PRINT"HAS PONTOON YOU
 MUST PAY THE BANKER"
 11730 LOCATE 3,16:PRINT"THE VALUE OF YOU
 R BET."
 11740 LOCATE 3,17:PRINT"IF YOU HAVE PON
 TOON AND THE BANKER"
 11750 LOCATE 3,18:PRINT"HAS FIVE & UNDER
 YOU WILL ONLY BE"
 11760 LOCATE 3,19:PRINT"PAID THE VALUE O
 F YOUR BET."
 11770 GOSUB 12720
 11780 CLS #1
 11790 LOCATE 3,3:PRINT"IF IT WERE THE O
 THER WAY AROUND,"
 11800 LOCATE 3,4:PRINT"THAT IS THE BANKE
 R HAS PONTOON AND"
 11810 LOCATE 3,5:PRINT"YOU HAVE FIVE & U
 NDER, YOU MUST PAY"
 11820 LOCATE 3,6:PRINT"THE BANKER THE AM
 OUNT OF YOUR BET."
 11830 PEN 3:LOCATE 3,8:PRINT"FIVE & UNDE
 R":PEN 0
 11840 LOCATE 3,10:PRINT"FIVE & UNDER IS
 A SITUATION WHERE"
 11850 LOCATE 3,11:PRINT"THE PLAYER HAS A
 HAND OF FIVE CARDS"
 11860 LOCATE 3,12:PRINT"THAT DOES NOT E
 XCEED THE VALUE OF"
 11870 LOCATE 3,13:PRINT"TWENTY-ONE. IN
 THE EVENT THAT YOU"
 11880 LOCATE 3,14:PRINT"HAVE FIVE & UNDE
 R THE BANK PAYS YOU"
 11890 LOCATE 3,15:PRINT"DOUBLE THE AMOUN
 T THAT YOU HAVE BET."
 11900 LOCATE 3,16:PRINT"IF THE BANKER HA
 S PONTOON OR FIVE &
 11910 LOCATE 3,17:PRINT"UNDER, YOU MUST
 PAY HIM THE AMOUNT"
 11920 LOCATE 3,18:PRINT"THAT YOU HAVE BE
 T."
 11930 GOSUB 12720
 11940 CLS#1
 11950 LOCATE 3,3:PEN 3:PRINT"THE ACE":PE
 N 0
 11960 LOCATE 3,5:PEN 0:PRINT"THE ACE MAY
 BE TAKEN AS THE VALUE"
 11970 LOCATE 3,6:PRINT"OF ONE OR ELEVEN
 AT THE CHOICE OF"
 11980 LOCATE 3,7:PRINT"THE PLAYER."
 11990 LOCATE 3,9:PEN 3:PRINT"BETTING"
 12000 LOCATE 3,11:PEN 0:PRINT"BEFORE EA
 CH HAND THE PLAYER MUST"
 12010 LOCATE 3,12:PRINT"PLACE A BET. TH
 IS IS KNOWN AS"
 12020 LOCATE 3,13:PRINT"BUYING YOUR F
 IRST CARD. THE"
 12030 LOCATE 3,14:PRINT"PLAYER MAY THE
 N PURCHASE HIS"
 12040 LOCATE 3,15:PRINT"CONSECUTIVE CARD
 S FOR ONE, TWO OR"
 12050 LOCATE 3,16:PRINT"ATIVE CHIPS."
 12060 GOSUB 12720
 12070 CLS#1
 12080 LOCATE 3,3:PEN 3:PRINT"SLIPPING"
 12090 LOCATE 3,5:PEN 0:PRINT"WHEN THE PL

```

AYER FLIPS HE PURCHASES"
12100 LOCATE 3,6:PRINT"A CARD WITHOUT B
UYING IT. A PLAYER"
12110 LOCATE 3,7:PRINT"MAY ONLY FLIP W
HEN HE HAS A TOTAL"
12120 LOCATE 3,8:PRINT"OVER OR EQUAL TO
TWELVE."
12130 LOCATE 3,10:PEN 3:PRINT"SITTING"
12140 LOCATE 3,12:PEN 0:PRINT"THE PLAYER
SITS WHEN HE HAS A HAND"
12150 LOCATE 3,13:PRINT"THAT HE IS SATIS
FIED WITH THE PLAYER"
12160 LOCATE 3,14:PRINT"MAY ONLY SIT ON
SIXTEEN OR OVER."
12170 LOCATE 3,15:PRINT"ONCE THE PLAYER
SITS, THE BANK THEN"
12180 LOCATE 3,16:PRINT"PLAYS HIS HAND."
12190 LOCATE 3,18:PEN 3:PRINT"NOTE - ";:
:PEN 0:PRINT"THE BANK DOES NOT KNOW WHA
T"
12200 LOCATE 3,19:PRINT"YOU HAVE IN YO
UR HAND SO HE MAY"
12210 LOCATE 3,20:PRINT"DECIDE TO SIT BE
LOW YOUR TOTAL."
12220 GOSUB 12720
12230 CLS#1
12240 LOCATE 3,3:PEN 3:PRINT"RESHUFFLING
"
12250 LOCATE 3,5:PEN 0:PRINT"THE PACK CO
NSISTS OF FIFTYTWO CARDS"
12260 LOCATE 3,6:PRINT"EACH OF THESE CA
RDS MUST BE USED"
12270 LOCATE 3,7:PRINT"BEFORE THE PACK I
S RESHUFFLED."
12280 LOCATE 3,9:PEN 3:PRINT"THE CONTROL
S"
12290 LOCATE 3,11:PEN 0:PRINT"THESE ARE
THREE CONTROLS IN THE"
12300 LOCATE 3,12:PRINT"GAME THESE ARE U
P,DOWN AND CONFIRM"
12310 LOCATE 3,13:PRINT"YOU MAY USE THE
CURSOR CLUSTER ON"
12320 LOCATE 3,14:PRINT"THE KEYBOARD OR
A JOYSTICK."
12330 GOSUB 12720
12340 CLS#1
12350 WINDOW #2,3,7,9,16:PAPER #2,2
12360 LOCATE 3,3:PEN 3:PRINT"MENUS"
12370 LOCATE 3,5:PEN 0:PRINT"THESE ARE F
OUR DIFFERENT MENUS."
12380 LOCATE 3,6:PRINT"THESE ARE MAIN ME
NU , BETTING MENU,"
12390 LOCATE 3,7:PRINT"ACE MENU AND THE
CONFIRM MENU."
12400 X4=30:Y4=9:Y1=10:TOGGLE=1:GOSUB 65
90:GOSUB 7180:PAPER 2
12410 LOCATE 3,9:PEN 3:PRINT"MAIN MENU":
PEN 0
12420 LOCATE 3,11:PRINT"TO SELECT ANY
OF THESE"
12430 LOCATE 3,12:PRINT"OPTIONS SIMPLY
MOVE THE"
12440 LOCATE 3,13:PRINT"HIGHLIGHTED BAR
OVER THE"
12450 LOCATE 3,14:PRINT"DESIRED FUNCTION
"
"
12460 GOSUB 12720:CLS#2
12470 GOSUB 6590:GOSUB 7610:PAPER 2
12480 LOCATE 3,9:PEN 3:PRINT"BETTING MEN
U":PEN 0
12490 LOCATE 3,11:PRINT"AFTER SELECTING
THE BET"
12500 LOCATE 3,12:PRINT"OPTION YOU ARE
PRESENTED"
12510 LOCATE 3,13:PRINT"THE MENU ON THE
RIGHT, YOU"
12520 LOCATE 3,14:PRINT"MUST SELECT";:P
EN 3:PRINT" ONE , TWO";:PEN 0:PRINT" OR"
12530 LOCATE 3,15:PEN 3:PRINT"FIVE";:PEN
0:PRINT" CHIPS TO BET."
12540 GOSUB 12720
12550 CLS#2
12560 GOSUB 6590:GOSUB 7950:PAPER 2
12570 LOCATE 3,9:PEN 3:PRINT"ACE MENU":P
EN 0
12580 LOCATE 3,11:PRINT"WHEN YOU ARE DEA
LT AN ACE"
12590 LOCATE 3,12:PRINT"YOU MUST SELECT
IT'S"
12600 LOCATE 3,13:PRINT"VALUE. THIS CAN
BE EITHER"
12610 LOCATE 3,14:PEN 3:PRINT"ONE";:PEN
0:PRINT" OR";:PEN 3:PRINT" ELEVEN.":PEN
0
12620 GOSUB 12720
12630 CLS#2
12640 Y4=1:Y2=9:GOSUB 6780:GOSUB 8200:PA
PER 2
12650 LOCATE 3,9:PEN 3:PRINT"CONFIRM MEN
U"
12660 LOCATE 3,11:PEN 0:PRINT"THIS MENU
WILL APPEAR"
12670 LOCATE 3,12:PRINT"AFTER ANY SELECT
ION FROM"
12680 LOCATE 3,13:PRINT"THE OTHER MENUS.
"
12690 LOCATE 3,14:PEN 3:PRINT"OK";:PEN 0
:PRINT" CONFIRMS SELECTION"
12700 LOCATE 3,15:PEN 3:PRINT"RETURN";:P
EN 0:PRINT" CANCELS SELECTION"
12710 TOGGLE=0:PAPER 2:PEN 0
12720 LOCATE 8,23:PRINT" PRESS ";:PEN 3:
PRINT"FIRE ";:PEN 0:PRINT"TO CONTINUE "
12730 :PAPER 2:PEN 0
12740 IF INKEY(9)=0 OR JOY(0)=16 THEN 12
760
12750 GOTO 12740
12760 RETURN
12770 ,
12780 ,--CASHIERS WINDOW--
12790 ,
12800 WINDOW #2,3,21,5,15
12810 PAPER #2,1
12820 CLS #2
12830 PEN 0
12840 LOCATE 3,5:PRINT CHR$(149);
12850 FOR I=1 TO 18
12900 LOCATE 3+I,5:PRINT CHR$(150);
12870 NEXT
12880 LOCATE 21,5:PRINT CHR$(151)

```

```

12990 FOR I=1 TO 9
12900 LOCATE 3,5+I:PRINT CHR$(152):LOCAT
E 2,5+I:PRINT CHR$(155)
12910 NEXT
12920 LOCATE 3,15:PRINT CHR$(154);
12930 FOR I=1 TO 17
12940 LOCATE 3+I,15:PRINT CHR$(156);
12950 NEXT
12960 LOCATE 21,15:PRINT CHR$(155)
12970 IF W=1 THEN GOTC 13160
12980
12990 '---RUN OUT OF CHIPS--
13000
13010 PEN 3:LOCATE 4,6:PRINT"AT THE CHAS
HIER"
13020 PEN 0:LOCATE 4,7:PRINT"YOU HAVE LO
ST ALL"
13030 LOCATE 4,8:PRINT"YOUR CHIPS.":PEN
0
13040 LOCATE 4,9 PRINT"WOULD YOU LIKE TO
"
13050 LOCATE 4,10:PRINT"BUY SOME MORE."
13060 X6=4:Y6=11:GOSUB 8400:PAPER 1:PEN
0
13070 IF NO=1 THEN NO=0:W=1:GOTO 12800
13080 IF YES=1 THEN YES=0:LOAN=LOAN+20:M
ONEY=MONEY+20:GOTO 13100
13090 GOTO 13020
13100 LOCATE 12,41:PRINT"YOU NOW"
13110 LOCATE 4,12:PRINT"HAVE 20 CHIPS."
13120 LOCATE 4,13:PRINT"THAT WILL COST"
13130 LOCATE 4,14:PRINT"YOU $20. THANKYO
U"
13140 FOR DELAY=1 TO 4000:NEXT:GOTO 430
13150
13160 '---END OF GAME STATISTICS--
13170
13180 PEN 3:LOCATE 4,6:PRINT"AT THE CASH
IER"
13190 IF MONEY<0 THEN OWE=ABS(MONEY):MON
BY=0:LOAN=LOAN+OWE
13200 PEN 0:LOCATE 4,7:PRINT"YOU HAVEN";M
ONEY
13210 LOCATE 4,8:PRINT"CHIPS. AT ONR"
13220 LOCATE 4,9:PRINT"DOLLAR PER CHIP"
13230 LOCATE 4,10:PRINT"THAT'S $";MONEY
13240 LOCATE 4,11:PRINT"YOU FINISHED UP"
13250 IF MONEY<20+LOAN THEN 13200
13260 LOCATE 4,12:PEN 3:PRINT"WINNING $"
:(MONEY-20)-LOAN
13270 GOTO 13290
13280 LOCATE 4,12:PEN 3:PRINT"LOSING $";
(LOAN+20)-MONEY
13290 LOCATE 4,13:PEN 0:PRINT"ANOTHER GA
ME"
13300 X6=4:Y6=14:GOSUB 8400
13310 IF YES=1 THEN RUN
13320 IF NO=1 THEN CALL 0
13330 GOTO 13270

```



PCW 8256 and 8512 owners

If you are reading this magazine for the first time you have already missed reviews on The Cracker and the Poly Series software published last month.

Already in the pipeline and due for publication next month are hard-hitting reviews of the latest significant releases including:

Tasword 8000
dBase II
Cardbox

And that not all!

Copies of DR Draw, Cambase, Micro Word and Brainstorm are in the queue for reviewing.

It makes sense to spend just \$3.50 on The Amstrad User and get an independent opinion of software that may cost many hundreds of dollars to purchase.

As the number of PCW users grows, so too our commitment to providing as much relevant information as possible.

Contributions are most welcome and rewarded.

Tornado Low Level

When did you last play a game where nothing shot at you? Can't remember? Then perhaps you should take a look at this neat little aeroplane escapade, in which your mission is to attack five targets scattered around the local countryside.

Of course the pilot doesn't have it all his own way. Whilst searching for the targets you must dodge trees, tall buildings and electric power lines. In a 3D fashion one can carefully fly behind things.

The cassette cover boasts of 'the fastest, smoothest scrolling yet to be seen on the Amstrad'. This may well be true - pulling back the joystick sweeps back the wings and increases your height and speed.

Unfortunately most of the work is done close to the ground at the slowest possible speed as the weaponry will only fire at close range. This is when the scrolling jerks a little, losing the ultrasmooth quality.

Be prepared for the first flights to be rather brief until the delicate knack of hugging the

terrain is achieved. Once learnt, it suddenly becomes easier to line up the enemy - a small, rather unimaginative dark circle.

Other features of the game include altimeter, fuel gauge, time remaining, map and ammo stock. This has to be called up in a status report. Whilst flying there is no information displayed on the screen so you end up relying on judgement alone for continuity's sake.

The graphics in TLL are only fair to average. There are stretches of water composed of millions of multi-coloured dots which I though looked quite clever but everything on the land is very blocky like the early Atari games.

Once you've completed the first mission each subsequent mission has the 5 targets located in more difficult places. However, I would say this game has little long term appeal due to the limited scope of the surrounding countryside.

Originality	7
Speed	7
Graphics	3
Ease of Use	3
Entertainment Value	5
Documentation	4
Ability to Hold Interest	4
TAUINDEX	59%

Way of the Exploding Fist

After reading many reviews and reports all praising The Way of the Exploding Fist, I was more than eager to get started. After loading the first two blocks, the title page appeared and anybody thinking that the game consisted of poor graphics would surely be proven wrong at this early stage. So what did the title page comprise of you might ask? Well I am sure that everybody has at least seen a poster or some type of advertisement publicising the game, showing a determined Martial Artist executing a head punch in order to break a wooden board with chinese lettering inscribed upon it, with an oriental type building in the background. Well that is exactly what the title page comprises of, illustrated in excellent colour and graphics.

In a nutshell, the object of the game is for you to kick, punch, sweep and somersault your way from novice to the ultimate glory of Tenth Dan.

Due to some incredible programming, eighteen traditional Karate movements are available at your disposal. These include a variety

of punches and kicks, leg sweeps, and even somersaults. But before you are able to use these movements to its fullest extent, practice is the name of the game.

There are one or two player options, therefore allowing you to choose which way you want to be slaughtered, by a friend or the computer. You also have the option to either control your man by joystick or keyboard.

To advance from Dan to Dan you are required to win a number of bouts against your opponent. Before you defeat your opponent you have to obtain two full points by succeeding in striking him in one form or another. The points are not scored by which action you choose to take, but how well you execute the movement.

So for all those pugnacious Bruce Lee fanatics who think that they can win by performing a series of untimed wild, high flying head kicks, think again!

In the event of an untimed aggressive movement which has penetrated your opponents defence, you will

(continued overleaf)

Way of the Exploding Fist

only score half a point. A full point is represented by a Yin and Yan symbol with a half point represented accordingly with a half symbol. Despite all these fantastic facilities available, I do have one complaint. Unlike the Commodore version the Amstrad version of the game does not include a bonus round nor does the setting change as you progress from Dan to Dan.

Nevertheless, I highly recommend the Way of the Exploding Fist to anybody that is tired of zapping aliens, or has sore feet from jumping from platform to platform or just anyone that is looking for the best Martial Arts stimulation game that is available for the Amstrad today.

Ease of Use	6
Entertainment Value	7
Originality	5
Ability to hold Interest	7
Speed	7
Documentation	7
Use of Graphics	8
TAU INDEX	84%

Highway Encounter

The Scene: A highway. The objective: Push the Lasertron weapon through 30 zones to defeat the alien stronghold.

The earth has been invaded! Again! But this isn't your average aim and shoot invaders, and it is NOT a road racing game. In fact HE is a strikingly original game in which you battle strange creatures and road hazards in your progression from Zone 30 to Zone Zero.

You have 4 automatic Vortons to assist in your task, the idea being to clear the way whilst they push for you. If they are all blown up you must do everything yourself.

Your 'direct control' Vorton looks similar to a

Dalek and he's smooth moving and just as dangerous when spitting fireballs at the road-running aliens. Deadly though he is, the Vorton is a real horror to steer. Most of your early games will be spent out of control until the basics of direction are mastered.

With several creatures and Vortons carcering around the screen at once the speed and movement are very good. The standard of graphics is high; sound is also OK, but it must be said the actual playing area is rather small.

Other features include joystick/keyboard option, demonstration mode, information screen, good 3D animation, high score

table and a first class loader screen. There is a grid pattern drawn on the road to assist perspective and to help line up you attacking shots (or glorious retreats).

Highway Encounter is addictive and interesting and comes up well on a green monitor but I still suggest you see it before buying, however. At \$30 I felt the game was overpriced, representing only fair value for money.

Originality	8
Use of graphics	5
Ease of use	2
Speed	7
Documentation	4
Entertainment value	4
Ability to hold interest	5
TAU INDEX	62%

CHILLER

What do you call a game with cluttered screens, jerky movements, awful speed and flickery graphics? Chiller, that's what! And this time call it successful as well..... This has sold well according to the UK software charts (nine months in top 30 peaking at No.4) but for the life of me I can't see why.

The storyline is quite cool, being based on the Michael Jackson Thriller video. The idea is to go through the forest, cinema, graveyard and ghetto, facing perils and collecting crucifixes designed to ward off enough evil, to arrive at the next screen. Having done all this you reach the Haunted House where your girlfriend is being held captive - rescue her then go back via the same screens to your car.

Chiller is best described as 'platforms and ladders' because you'll need these skills to jump around each scene quickly.

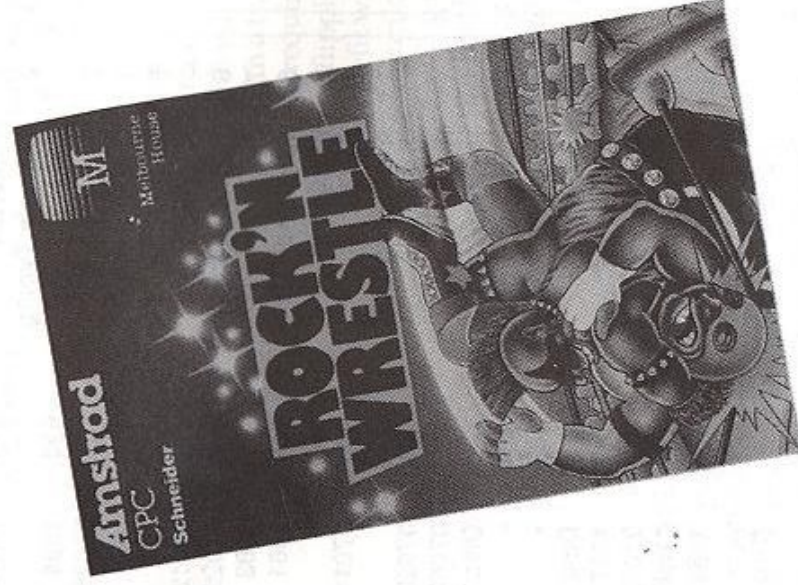
Your main problem is avoiding contact with anything that moves. The spiders and little men may look primitive but they will drain your energy as quick as lightning.

The music is quite boppy and is probably the best thing about the game although you can't turn it off. The other main point to remember is that Chiller is one of the budget series and only cost about \$7.

Summary - Chiller does initially have additive qualities but the whole thing looks amateurish and it wouldn't be too hard to find a magazine listing of equal graphic quality. It has sold, but I suspect when you pull out a game to impress your friends it won't be this one.

Speed	1
Graphics	2
Documentation	3
Ease of Use	4
Originality	5
Ability to Hold Interest	4
Entertainment Value	3
TAU INDEX	39%

ROCK'N WRESTLE



Wham, Bam, Thank you Mam! Rock n' wrestle from Melbourne House will have you grunting, groaning and grappling with the toughest 'rasslers' in the world today. There are eight other opponents to compete against before Gorgeous Greg (that's you) earns a crack at Lord Toff - the Champ.

Each contest has a three minute time limit and you must wear down the opponent and pin him within the time. There are lots of genuine wrestling moves to master - over 20 in fact. Many of these look quite spectacular.

Rock n' Wrestle has a very good loader screen and a comprehensive instruction leaflet, along with 1 or 2 player mode and joystick/keyboard option. Because of the large number of tricks to learn, the best thing to do is select '2 player' mode. That way you can practice on a guy who doesn't fight back.

When on the offensive, good timing and dexterity with moves will score for you, but when in a fix such as an aeroplane spin, wriggling the joystick in mad Decathlon style is the way to go. Each player has an energy level which decreases whilst you are on the receiving end of things, and as this level plunges towards zero, it becomes much harder to wriggle out of trouble.

One must say the sound effects are terrible - they consist only of a horrid scraping sound whenever contact is made. Speaking of noise, the game lacks atmosphere - after all what is wrestling without lots of grunts, yells, cheering and general hubbub? None of that here, just a placid bunch of open-mouthed onlookers. I reckon defeating Redneck McCoy deserves at least a victory tune.

I did like the way the ropes bend when you run into them, and putting Molotov Mick onto a suplex was just great, but I'm afraid the main characters in the fight are too blocky and ill-defined. There just wasn't enough put into their design and movement to compete with the better looking kung-fu games. One final point - on cassette the game is saved at two speeds - the fast load takes 5.5 minutes but the slow load takes an abysmal 10 minutes.

Summary - The programmers have obviously spent much effort on the manoeuvres and gameplay, and it WAS fun beating up my little brother, but the jerkiness and overall presentation let down what could really have been a "stunning" game.

Graphics	2
Speed	3
Ease of Use	4
Ability to hold interest	5
Entertainment value	6
Originality	6
Documentation	7
TAU INDEX	60%

**To see if
you've made it into
our Hall of Fame turn
to the next page**

The Amstrad User Hall Of Fame

AMSTRAD TOP 25

The official Amstrad Software Guide as
supplied by Mitsubishi Electric AWA Pty Ltd

GAME	SCORE/TIME	ACHIEVER
Airwolf	1500/25 mins	Heath Corcoran
Battle for Midway	8 carriers:spood 1:level 3	Steve Alatakis
Beach Head	132500/16 mins	Anthony Eden
Chuckie Egg	395960/45 mins	Tony Barberi
Codename Mat	870/45 mins	Gill Cherry
Combat Lynx	81450/no time specified	Steve Alatakis
Decathlon	615830/208 mins	Anthony Eden
Defend or Die	44500/15 mins	Darrell Harris
Er-Bert	68350/21 mins	Anthony Eden
Galactic Plague	118690/80 mins	B. Welch/G.Cherry
Gilligan's Gold	107403/9.75 mins	Alex Smyth
Grand Prix Rally II	47984/10.5 mins	Justin McShane
HARRIER Attack	337600/14.5 mins	Michael Hopkirk
Haunted Hedges	466460/35 mins	Lorraine Martin
Hunchback	213300/3 mins	Allison Pilbeam
Hunter Killer	17/67 mins	Chris Catalfamo
Jet Set Willy	33 items/11 mins	Inga Callahan
Knight Lore	98%/44 mins	Umut Akcelik
Minder	\$17749/no time specified	Steve Alatakis
Moontbuggy	152400/26.75 mins	Alex Smyth
Raid	416950/26 mins	Allan Etherington
Roland in the Caves	9748815/6 mins	Anthony Eden
Roland goes Digging	\$659.80/40 mins	Trant Eddy
Roland on the Ropes	736900/92 mins	Allison Pilbeam
Roland in Time	72/18 mins	Paul Azzopardi
Sorcery +	91500/14 mins	Mike Nicolai
Sorcery +	128200/50 mins	Karl Webber
Space Hawks	72300/7 mins	Murray Taylor
Spannerman	53500/no time specified	Allison Pilbeam
Splat	7280/45 mins	Neil Campbell
Star Commando	193810/133 mins	Alex Smyth
Survivor	223160/19.5 mins	Alex Smyth
Way of Exp. Fist	558900/66 mins	Gavern Cherry
Wild Bunch	10539/no time specified	Steve Alatakis
Yie Ar Kung Fu	7065940/3 hours	Gavern Cherry
Zorro	29100/22.5 mins	Jason Scott
3-D Monster Chase	1320.7 keys/7 mins	Adam Broadway

Pos. No.	Masterfile	Soft No. Tape	Disc
1	Masterfile	914	1914
2	Pitman's Typing	924	1924
3	Tutor	.	07028
4	SuperCalc II	.	Disc
5	Tasword 6128		
6	Grand Prix		
7	Rally II	06012	07012
8	3D Cyrus Chess	06026	07026
9	Sorcery+	.	1983
10	3D Grand Prix	961	1961
11	Wordhang	101	.
12	Airwolf	06011	07011
13	Assault on Port		
14	Stanley	06022	07022
15	Raid	.	07002
16	Microfile(Flexifile)	.	Disc
17	Fantastic Voyage	984	.
18	Flight Simulator	168	.
19	Taspell	.	Disc
20	Roland in Caves	117	1117
21	Spitfire 40	.	07052
22	Advanced Amsword	164	1164
23	3D Stuntrider	183	1183
24	Tasword 464D		
25	(464 Disc Drive)		Disc
26	Classic Adventure/		
27	Morden's Quest		
28	(for PCW)		
29	Roland Ahoy!	192	1192
30	MasterCalc	905	1905
31	Satellite Warrior	949	1949



To get your name in our
"HALL OF FAME" register on the form
below, or a copy, and if possible, send
a photo of the screen.

THIS NEXT PART MUST BE COMPLETED

Witness' Name _____

Address _____

Telephone Number _____

Occupation _____

I confirm that the above claimed score is accurate and
genuine
Signed _____

Name _____

Address _____

Game _____ Telephone Number _____

Achieved (date) _____ Score _____

Signed _____ Game lasted (mins) _____

Post this form along with your tips for playing the
game to:

Amstrad Achievers, The Amstrad User, Suite 1,
245 Springvale Road, Glen Waverley, Vic 3150

Hedge-Hog

A game from Ian Barnes

Hedge-Hog is a BASIC program which was originally designed around one of the few playable games for the Dick Smith Wizard home computer. (No, I didn't have one, but one of my friends did).

The idea of the game is to keep the Hedge-Hog on the path as he collects a number of bunches of cherries. Unfortunately you have no control over the Hedge-Hog himself, and so must move sections of the path to form a trail for him to follow. He will be killed if he comes off the path. By the way, I'm not English, it's just that Hedge-Hog sounds better than Echidna.

The program is all in BASIC, and some parts may seem to be very similar to others. This is due to my trying to increase the speed of the program, at the expense of program length. However, the whole program is only about 13k long, which is a fairly reasonable amount of typing.

A break-down of the program follows:

10 - 120	Call routines to set up the game
130 - 760	Draw playing field
770 - 860	Main program loop
870 - 1380	Move tiles
1390 - 1980	Move Hedge-Hog
1990 - 2860	Set up tiles
2870 - 3230	Draw starting screen
3240 - 3530	Data
3540 - 3640	Dead!

The game has ten speed levels, and anyone who feels that the game is too slow should try playing at level 9 (I usually play at about level 5).

When typing the program in you should type it as it is printed, as any missing lines will cause the program to do a number of interesting things, none of which help you to play. After the program is working you can remove the REMs.

```
10 KEY 135,"SYMBOL AFTER 200:paper 0:pen
1: border 1:mode 2:ink 0,1:ink 1,26:11st
"+CHR$(13)
20 ' HEDGE-HOG
30 ' BY IAN BARNES
40 ' JULY 1986
50 '
60 'Use a joystick or
70 'the cursor keys
80 '
90 '
100 GOSUB 2900 'Starting screen
110 GOSUB 2030 'set up variables
120 vscore=0:vrack=0
130 '
140 'start of rack
150 '
160 WHILE NOT dead
170 PRINT CHR$(23)+CHR$(1);
180 xa=280:ya=102:xb=0:yb=1.6:d=3
190 PAPER#1,0
200 IF vrack=1 AND diff<2 THEN diff=diff
+0.1
210 IF vrack=3 THEN diff=diff+0.1:IF vra
ck=4 THEN RESTORE 3460
220 t=0
230 '
240 ' print a new screen
250 '
260 ' first, the border tiles
270 '
280 FOR loop=5 TO 35 STEP 4
290 WINDOW#1,loop,loop+3,1,4
300 READ v:IF v=12 THEN PAPER#1,2 ELSE P
APER#1,0
310 PRINT#1,s(v);
320 NEXT
330 FOR loop=5 TO 21 STEP 4
340 WINDOW#1,5,8,loop,loop+3
350 READ v:IF v=12 THEN PAPER#1,2 ELSE P
APER#1,0
360 PRINT#1,s(v);
370 WINDOW#1,33,36,loop,loop+3
380 READ v:IF v=12 THEN PAPER#1,2 ELSE P
APER#1,0
390 PRINT#1,s(v);
400 NEXT
410 PLOT -4,-4,2:PRINT CHR$(23)+CHR$(1);
420 p=0
430 '
440 ' print cherries
450 '
```

```

460 TAG: READ vcount
470 FOR LOOP=1 TO vcount
480 READ X,Y: MOVE X,Y
490 PRINT CHR$(255);
500 c(loop,0)=x:c(loop,1)=y
510 NEXT
520 count=0
530 TAGOFF
540
550
560
570 PAPER#1,2
580 FOR LOOP=1 TO 6
590 FOR LOOP1=1 TO 5
600 WINDOW#1,5+LOOP*4,5+LOOP*4,LOOP1*4+1,
    LOOP1*4+4
610 R=INT(RND*7)+1
620 IF LOOP=8 THEN IF (loop1=2) THEN IF
    r=8 OR r=3 OR r=4 THEN r=INT(RND*2+1)
630 IF loop=3 THEN IF loop1=3 OR loop1=4
    THEN r=INT(RND*3)+2+5
640 PRINT#1,S(R);:B(LOOP,LOOP1)=R
650 TEXT:NEXT
660 PLOC -4,-4,J
670
680
690
700 od=-1:c=3*(3-diff)
710 x=RND*4+1:y=RND*3+1
720 IF x=3 AND y<>1 THEN Y10
730 b(x,y)=0:PAPER 0
740 WINDOW#0,5+x*4,5+x*4,1+y*4,y*4+4:CLS
750 WINDOW#0,1,40,1,25
760 LOCATE 15,25:PRINT"SCORE: ",USING"###
#";vscore;
770
780
790
800 WHILE (count<>vcount) AND NOT dead
810 GOSUB 800
820 GOSUB 1390
830 VEND
840 TAGOFF:CLS
850 VEND
860 GOTO 3570
870
880
890
900 xc=INT(xa-4)/16+1:yc=INT((404-ya)/16
)+1
910 IF t THEN ON dir GOSUB 1060,1150,124
0,1330:RETURN
920 xi=1:yi=1
930
940
950
960
970 IF (JOY(0) AND 1) OR (NOT INKEY(0))
    THEN IF b(x,y+1) THEN dir=4:t=-1
980 IF (JOY(0) AND 2) OR (NOT INKEY(2))
    THEN IF b(x,y-1) THEN dir=3:t=-1
990 IF (JOY(0) AND 4) OR (NOT INKEY(8))
    THEN IF b(x+1,y) THEN dir=2:t=-1
1000 IF (JOY(0) AND 6) OR (NOT INKEY(1))
    THEN IF b(x-1,y) THEN dir=1:t=-1

```

```

1010 FOR loop=0 TO 60:WBXT
1020 RETURN
1030
1040
1050
1060 WINDOW#1,x*4+xi+1,4+x*4+xi,y*4+1,y
    +4
1070 PRINT#1,S(B(x-1,y));
1080 LOCATE x*4+xi,y*4+1:PRINT wipe$;
1090 IF (xc)=x*4+xi AND xc<=4+x*4+xi AND
    yc>=y*4+1 AND yc<=y*4+4 THEN od=-1:xa
    xa+16
1100 xi=x1+1:IF xi=5 THEN b(x,y)=b(x-1,
    y):b(x-1,y)=0:x=x-1:t=0
1110 RETURN
1120
1130
1140
1150 WINDOW#1,9+x*4-xi,12+x*4-xi,y*4+1,y
    +4
1160 PRINT#1,S(B(x+1,y));
1170 PRINT CHR$(31)+CHR$(13+x*4-xi)+CHR$(
    y*4+1)+wipe$;
1180 IF (xc)=9+x*4-xi AND xc<=12+x*4-xi
    AND yc>=y*4+1 AND yc<=y*4+4 THEN od=-1:
    xa=xa-16
1190 xi=xi+1:IF xi=5 THEN b(x,y)=b(x+1,y
    ):b(x+1,y)=0:x=x+1:t=0
1200 RETURN
1210
1220
1230
1240 WINDOW#1,5+x*4,8+x*4,y*4+yi-3,y*4+y
    +1
1250 PRINT#1,S(B(x,y-1));
1260 LOCATE 5+x*4,y*4+yi-4:PRINT"
";
1270 IF (xc)=5+x*4 AND xc<=8+x*4 AND yc>
    =y*4+yi-3 AND yc<=y*4+yi THEN od=-1:ya=
    ya-16
1280 yi=yi+1:IF yi=5 THEN b(x,y)=b(x,y-1
    ):b(x,y-1)=0:y=y-1:t=0
1290 RETURN
1300
1310
1320
1330 WINDOW#1,5+x*4,8+x*4,5+y*4-y1,8+y*4
    -y1
1340 PRINT#1,S(B(x,y+1));
1350 LOCATE 5+x*4,9+y*4-y1:PRINT"
";
1360 IF (xc)=5+x*4 AND xc<=8+x*4 AND yc>
    =5+y*4-y1 AND yc<=9+y*4-y1 THEN od=-1:y
    a=ya+16
1370 yi=yi+1:IF yi=5 THEN b(x,y)=b(x,y+1
    ):b(x,y+1)=0:y=y+1:t=0
1380 RETURN
1390
1400
1410
1420 IF d>3 THEN GOSUB 1950 ELSE ON d+1
    GOSUB 1690,1710,1730,1750
1430
1440
1450
1460

```

1440 Call &BD19
1450 used to remove some
1460 of the flickering

```

1470 '
1480 ' remove old hedge-hog
1490 ' and print new one
1500 '
1510 IF ya<340 AND ya>100 THEN CALL &BD1
1520 '
1530 xa=xa+xb*diff:ya=ya+yb*diff:TAG
1540 IF od<>-1 THEN MOVE ox,oy:PRINT CHR
$(240+od);
1550 IF TEST(xa+8,ya-8)=3 OR TEST(4,4)=
3 THEN GOTO 1770
1560 MOVE xa,ya:PRINT CHR$(240+d);
1570 TAGOFF:ox=xa:oy=ya:od=d
1580 IF TEST(xa+8,ya-8)<>0 THEN dead=-1
1590 IF xa<64 AND d=2 THEN xa=550 ELSE I
F xa>550 AND d=0 THEN xa=66
1600 IF ya>340 THEN CALL &BD19
1610 RETURN
1620 '
1630 ' tests to see if hedgehog
1640 ' is about to come off its
1650 ' path.
1660 ' and change direction
1670 ' if it is.
1680 '
1690 IF (TEST(xa+20,ya-8)<>1 AND TEST(0
,0)<>3) THEN IF TEST(xa+24,ya-24)=1 THEN
d=4:xb=2:yb=-2:ndi=1 ELSE IF TEST(xa+24
,ya+8)=1 THEN d=7:xb=2:yb=2:ndi=3
1700 RETURN
1710 IF (TEST(xa+8,ya-20)<>1 AND TEST(0
,0)<>3) THEN IF TEST(xa-8,ya-24)=1 THEN
d=5:xb=-2:yb=-2:ndi=2 ELSE IF TEST(xa+24
,ya-24)=1 THEN d=4:xb=2:yb=-2:ndi=0
1720 RETURN
1730 IF (TEST(xa-6,ya-8)<>1 AND TEST(0,
0)<>3) THEN IF TEST(xa-8,ya-24)=1 THEN d
=5:xb=-2:yb=-2:ndi=1 ELSE IF TEST(xa-6,y
a+8)=1 THEN d=6:xb=-2:yb=2:ndi=3
1740 RETURN
1750 IF TEST(xa+8,ya+6)<>1 AND TEST(0,0
)<>3 THEN IF TEST(xa-8,ya+8)=1 THEN d=6:
xb=-2:yb=2:ndi=2 ELSE IF TEST(xa+24,ya+8
)=1 THEN d=7:xb=2:yb=2:ndi=0
1760 RETURN
1770 '
1780 ' remove cherries when they
1790 ' are run over.
1800 '
1810 FOR loop=1 TO vcount
1820 IF xa>c(loop,0)-16 THEN IF xa<c(loop
p,0)+16 THEN IF ya>c(loop,1)-16 THEN IF
ya<c(loop,1)+16 THEN v=loop:loop=vcount
1830 NEXT
1840 TAGOFF:PRINT CHR$(23)+CHR$(0);:TAG:
MOVE c(v,0),c(v,1):PRINT CHR$(143);
1850 TAGOFF:PRINT CHR$(23)+CHR$(1);:TAG
1860 count=count+1
1870 vscore=vscore+INT(25*diff)
1880 LOCATE#3,22,25:PRINT#3,USING"###";
vscore;
1890 GOTO 1550
1900 '
1910 ' moving at an angle

```

```

1920 '
1930 xv=xa-(xa\64)*64:yv=ya-(ya\64)*64
1940 h=h+1:IF h<3 THEN RETURN
1950 IF (ndi=1 OR ndi=3) THEN IF (xv>21
AND xv<27) THEN 1970 ELSE RETURN
1960 IF (ndi=0 OR ndi=2) THEN IF NOT (yv
>52 AND yv<58) THEN RETURN
1970 d=ndi:IF d=0 THEN xb=2:yb=0 ELSE IF
d=1 THEN xb=0:yb=-2 ELSE IF d=2 THEN xb
=-2:yb=0 ELSE xb=0:yb=2
1980 RETURN
1990 '
2000 ' set up all characters
2010 ' used on the tiles
2020 '
2030 SYMBOL AFTER 180
2040 SYMBOL 184,0,0,&F7,&63,&63,&63,&7F,
&7F
2050 SYMBOL 185,0,0,&80,0,0,0,&31,&4A
2060 SYMBOL 186,0,0,0,&20,&20,&20,&A3,&6
5
2070 SYMBOL 187,0,0,0,0,0,0,&30,&48
2080 SYMBOL 188,&63,&63,&63,&F3,3,3,3,3
2090 SYMBOL 189,&4A,&72,&42,&3D,0,0,0,0
2100 SYMBOL 190,&29,&69,&B9,&A7,&3,&D,&1
1,&12
2110 SYMBOL 191,&48,&70,&44,&B8,0,0,0,0
2120 SYMBOL 192,3,3,3,3,3,3,3
2130 SYMBOL 193,0,&3C,&18,&18,&18,&F8,&F
9,&1A
2140 SYMBOL 194,&C,0,0,0,0,&F3,&2D,&29
2150 SYMBOL 195,0,0,0,0,0,0,&20
2160 SYMBOL 196,3,3,7,0,0,0,0
2170 SYMBOL 197,&1A,&1A,&BD,0,0,0,0,0
2180 SYMBOL 198,&49,&49,&87,3,&D,&11,&12
,&C
2190 SYMBOL 199,&20,&40,&80,0,0,0,0,0
2200 SYMBOL 200,&FF,&3F,&3F,&3F,&7F,&7F,
&FF,&FF
2210 SYMBOL 201,&FF,&FC,&FC,&FC,&F8,&F8,
&F8,&F0
2220 SYMBOL 202,&83,&8F,&FF,&FF,&FF,&FF,
&FF,&FF
2230 SYMBOL 203,&FF,&FF,&FF,&FF,&FF,&FF,
&FF,&FF
2240 SYMBOL 204,&F0,&F0,&E0,&E0,&C0,&C0,
&80,0
2250 SYMBOL 205,&FF,&FF,&FF,&FF,&FE,&FE,
&F0,&80
2260 SYMBOL 206,&FE,&FC,&F0,&C0,0,0,0,0
2270 SYMBOL 207,&FF,&3F,&3F,&3F,&1F,&1F,
&1F,&F
2280 SYMBOL 208,&FF,&FC,&FC,&FC,&FE,&FE,
&FF,&FF
2290 SYMBOL 209,&F,&F,7,3,3,1,0
2300 SYMBOL 210,&FF,&FF,&FF,&FF,&FF,&FF,
&FF,&FF
2310 SYMBOL 211,&C1,&F1,&FF,&FF,&FF,&FF,
&FF,&FF
2320 SYMBOL 212,&7F,&3F,&F,3,0,0,0,0
2330 SYMBOL 213,&FF,&FF,&FF,&FF,&7F,&F,1
,1
2340 SYMBOL 214,0,0,0,0,&C0,&F0,&FC,&FE
2350 SYMBOL 215,&80,&80,&F0,&FE,&FF,&FF,
&FF,&FF

```

2360 SYMBOL 216, 0, &80, &C0, &E0, &E0, &E0, &F
0, &F0
2370 SYMBOL 217, &FF, &FF, &FF, &FF, &FF, &FF, &FF,
&FF, &FF
2380 SYMBOL 218, &FF, &FF, &FF, &FF, &FF, &FF, &FF,
&FF, &83
2390 SYMBOL 219, &F0, &F8, &F8, &F8, &FC, &FC,
&FC, &FF
2400 SYMBOL 220, &FF, &FF, &FF, &FF, &FF, &FF, &FF,
&FF, &FF
2410 SYMBOL 221, 1, 1, &F, &7F, &FF, &FF, &FF, &FF, &
FF
2420 SYMBOL 222, 0, 0, 0, 0, 3, &F, &3F, &7F
2430 SYMBOL 223, &FF, &FF, &FF, &FF, &FF, &FF, &FF,
&F1, &C1
2440 SYMBOL 224, &FF, &FF, &FF, &FF, &FF, &FF, &FF,
&FF, &FF
2450 SYMBOL 225, 0, 1, 3, 3, 7, 7, &F, &F
2460 SYMBOL 226, &FF, &FF, &FF, &FF, &FE, &FC, &FC,
&FC, &FF
2470 SYMBOL 227, &F, &F, &1F, &1F, &3F, &3F, &3F, &
3F, &FF
2480 SYMBOL 228, &FF, &3F, &3F, &3F, &3F, &3F, &3F,
&3F, &FF
2490 SYMBOL 229, &FF, &FC, &FC, &FC, &FC, &FC, &FC,
&FC, &FF
2500 SYMBOL 230, &81, &81, &FF, &FF, &FF, &FF, &FF,
&FF, &FF
2510 SYMBOL 231, &FF, &FF, &FF, &FF, &FF, &FF, &FF,
&81, &81
2520 SYMBOL 232, &FF, &80, &80, &80, &80, &80, &80,
&80, &80
2530 SYMBOL 233, &FF, 1, 1, 1, 1, 1, 1, 1, 1
2540 SYMBOL 234, 1, 1, 1, 1, 1, 1, 1, 1, &FF
2550 SYMBOL 235, &80, &80, &80, &80, &80, &80, &80,
&80, &80
2560 SYMBOL 236, &FF, 0, 0, 0, 0, 0, 0, 0, 0
2570 SYMBOL 237, 1, 1, 1, 1, 1, 1, 1, 1
2580 SYMBOL 238, 0, 0, 0, 0, 0, 0, 0, &FF
2590 SYMBOL 239, &8C, &80, &80, &80, &80, &80, &80,
&80, &80
2600 SYMBOL 240, 0, &70, &FC, &FB, &FB, &FC, &7
0, 0
2610 SYMBOL 241, &3C, &7E, &7E, &7E, &3C, &24,
&18, &18
2620 SYMBOL 242, 0, &E, &3F, &DF, &DF, &3F, &E,
0
2630 SYMBOL 243, &18, &18, &24, &3C, &7E, &7E,
&7E, &3C
2640 SYMBOL 244, &70, &F8, &FC, &FC, &7A, &36,
&E, 0
2650 SYMBOL 245, &E, &1F, &3F, &3F, &5E, &6C, &
70, 0
2660 SYMBOL 246, 0, &70, &6C, &5E, &3F, &3F, &1
F, &E
2670 SYMBOL 247, 0, &E, &36, &7A, &FC, &FC, &F3
, &70
2680 SYMBOL 250, 3, 5, &1A, &62, &F5, &FF, &FF,
0
2690
2700
2710
2720
2730
2740
2750
2760
2770
2780
2790
2800
2810
2820
2830
2840
2850
2860
2870
2880
2890
2900
2910
2920
2930
2940
2950

CHR\$(237)+CHR\$(205)+CHR\$(206)+" "+CHR\$(2
37)+CHR\$(235)+CHR\$(238) CHR\$(238)+CHR\$(2
34)
34)
2740 S(2)=CHR\$(232)+CHR\$(207)+CHR\$(208)+
CHR\$(233)+CHR\$(239)+CHR\$(209)+CHR\$(210)+
CHR\$(211)+CHR\$(239)+" "+CHR\$(212)+CHR\$(2
13)+CHR\$(235)+CHR\$(238)+CHR\$(238)+CHR\$(2
34)
2750 S(3)=CHR\$(232)+CHR\$(236)+CHR\$(236)+
CHR\$(233)+CHR\$(215)+CHR\$(214)+" "+CHR\$(2
37)+CHR\$(218)+CHR\$(217)+CHR\$(216)+CHR\$(2
37)+CHR\$(235)+CHR\$(220)+CHR\$(219)+CHR\$(2
34)
2760 S(4)=CHR\$(232)+CHR\$(236)+CHR\$(236)+
CHR\$(233)+CHR\$(239)+" "+CHR\$(222)+CHR\$(2
21)+CHR\$(239)+CHR\$(225)+CHR\$(224)+CHR\$(2
23)+CHR\$(235)+CHR\$(227)+CHR\$(226)+CHR\$(2
34)
2770 S(5)=CHR\$(232)+CHR\$(200)+CHR\$(208)+
CHR\$(233)+CHR\$(202)+CHR\$(203)+CHR\$(203)+
CHR\$(211)+CHR\$(218)+CHR\$(217)+CHR\$(217)+
CHR\$(223)+CHR\$(235)+CHR\$(220)+CHR\$(226)+
CHR\$(234)
2780 S(6)=CHR\$(232)+CHR\$(236)+CHR\$(236)+
CHR\$(233)+CHR\$(230)+CHR\$(230)+CHR\$(230)+
CHR\$(230)+CHR\$(231)+CHR\$(231)+CHR\$(231)+
CHR\$(231)+CHR\$(235)+CHR\$(238)+CHR\$(238)+
CHR\$(234)
2790 S(7)=CHR\$(232)+CHR\$(228)+CHR\$(229)+
CHR\$(233)+CHR\$(239)+CHR\$(228)+CHR\$(229)+
CHR\$(237)+CHR\$(239)+CHR\$(226)+CHR\$(229)+
CHR\$(237)+CHR\$(235)+CHR\$(226)+CHR\$(229)+
CHR\$(234)
2800 S(8)=" "+CHR\$(200)+CHR\$(201)+" "+CH
R\$(202)+CHR\$(203)+CHR\$(204)+" "+CHR\$(205
) +CHR\$(206)+" "
2810 S(9)=" "+CHR\$(207)+CHR\$(208)+" "+CH
R\$(209)+CHR\$(210)+CHR\$(211)+" "+CHR\$(2
12)+CHR\$(213)+" "
2820 S(10)=" "+CHR\$(215)+CHR\$(214)+" "
" +CHR\$(218)+CHR\$(217)+CHR\$(216)+" "+CH
R\$(220)+CHR\$(219)+" "
2830 S(11)=" "+CHR\$(222)+CHR\$(221)+
" "+CHR\$(225)+CHR\$(224)+CHR\$(223)+" "+CH
R\$(227)+CHR\$(226)+" "
2840 S(12)=CHR\$(184)+CHR\$(185)+CHR\$(186)
+CHR\$(187)+CHR\$(188)+CHR\$(189)+CHR\$(190)
+CHR\$(191)+CHR\$(192)+CHR\$(193)+CHR\$(194)
+CHR\$(195)+CHR\$(196)+CHR\$(197)+CHR\$(198)
+CHR\$(199)
2850 wipe\$=" "+CHR\$(8)+CHR\$(10)+" "+CHR\$(
6)+CHR\$(10)+" "+CHR\$(0)+CHR\$(10)+" "
2860 RETURN
2870
2880
2890
2900 SYMBOL AFTER 97
2910 SYMBOL 97, &3C, &3C, &3C, &3C, &3C, &3C, &
3C, &3C
2920 SYMBOL 98, 0, 0, &FF, &FF, &FF, &FF, 0, 0
2930 SYMBOL 99, &3C, &7E, &FF, &FF, &FF, &FF, &
7E, &3C
2940 SYMBOL 100, 0, 0, &7, &1F, &3F, &3E, &
3C
2950 SYMBOL 101, 0, 0, &E0, &F8, &FC, &7C,
&7C

```

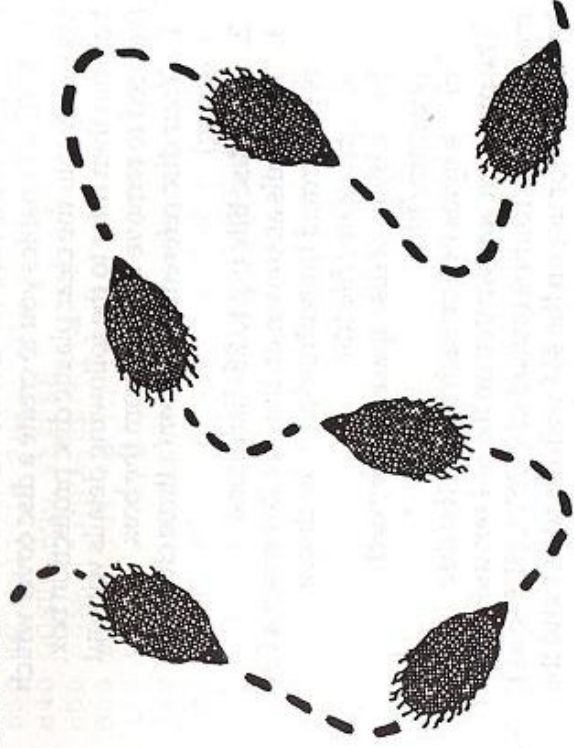
&3C
2960 SYMBOL 102,&3C,&7C,&FC,&F8,&F8,&E0,
0,0
2970 SYMBOL 103,&3C,&3E,&3F,&1F,&7,0
,0
2980 ,
2990 , Print starting screen
3000 ,
3010 MODE 1
3020 DEFINIT A-Z:DEFSTR S
3030 INK 0,9:INK 1,25:INK 2,25:INK 3,6,2
4:SPEED INK 1,1
3040 BORDER 9:PAPER 0:PEN 3
3050 PEN 3:LOCATE 15,20:PRINT"RANDOM NOI
SE"
3060 PEN 1:LOCATE 17,22:PRINT"PRESENTS"
3070 WINDOW#0,6,36,2,18
3080 FOR loop=1 TO 16
3090 READ as
3100 FOR loop1=1 TO 15
3110 LOCATE LOOP1,17:IF MID$(as,loop1,1)
=" " THEN PEN 1:PRINT CHR$(97+INT(RND*7)
):ELSE PEN 2:PRINT MID$(as,loop1,1);
3120 LOCATE 31-LOOP1,17:IF MID$(as,31-lo
op1,1)=" " THEN PEN 1:PRINT CHR$(97+INT(
RND*7)):ELSE PEN 2:PRINT MID$(as,31-loo
p1,1);
3130 NEXT:PRINT:PRINT:NEXT
3140 WINDOW#0,1,40,25,1
3150 PEN 1
3160 INK 2,26
3170 LOCATE 7,22:PRINT"ENTER DIFFICULTY
LEVEL (0-9)"
3180 LOCATE 11,24:PRINT"(9 IS HARDEST LE
VEL)"
3190 Is=INKEY$:IF Is="" OR Is<"0" OR Is>
"9" THEN 3190
3200 diff=1.1+VAL(Is)/10
3210 CLS:INK 0,3:INK 1,25:INK 2,9:INK 3,
6
3220 BORDER 3:DIM S(12),c(15,1)
3230 RETURN
3240 ,
3250 , data for starting screen
3260 ,
3270 DATA"
"
3280 DATA"
"
3290 DATA" a a dbbb adbe dbbe dbbb
"
3300 DATA" a a a c f a a a
"
3310 DATA" cbbc cbb a a a cbb
"
3320 DATA" a a a ce a a bc a
"
3330 DATA" a a gbbb agbf gbbf gbbb
"
3340 DATA"
"
3350 DATA"
"
3360 DATA" a a dbbe dbbe

```

```

3370 DATA"
"
3380 DATA"
"
3390 DATA"
"
3400 DATA"
"
3410 DATA"
"
3420 DATA"
"
3430 ,
3440 , data for racks
3450 ,
3460 DATA 12,4,3,4,3,12,4,3,2,1,6,6,
4,3,2,1
3470 DATA 7,180,374,308,374,436,374,88,8
4,88,284,536,84,536,284
3480 DATA 4,3,4,3,4,3,4,3,2,1,6,6,4,3,2,
1,12,12
3490 DATA 6,116,374,244,374,372,374,502,
374,88,148,536,148
3500 DATA 4,3,12,4,3,12,4,3,5,5,2,1,
4,3,2,1
3510 DATA 7,116,374,308,374,502,374,88,3
10,536,310,88,84,536,84
3520 DATA 3,12,4,3,4,3,12,4,2,1,4,3,2,1,
4,3,1,2
3530 DATA 6,88,84,88,220,244,374,372,372,372
,536,342,536,220
3540 ,
3550 , dead!!
3560 ,
3570 PLOT -800,-800,1:As="YOU'RE DEAD!";
3580 FOR LOOP=1 TO 13:LOCATE 2,LOOP+2:PR
INT MID$(As,LOOP,1);
3590 LOCATE 38,LOOP+2:PRINT MID$(As,LOOP
,1);
3600 NEXT
3610 LOCATE 12,12:PRINT"YOUR SCORE WAS:"
:USING"###";vscore;
3620 FOR LOOP=0 TO 40:As=INKEY$:NEXT
3630 WHILE INKEY$="":WEND
3640 RUN

```



COPYCAT

A disc utility from Dennis Shanaha

Here is yet another DISC COVER but, judge for yourself, I think it is the most useful yet.

How often have you been sitting there with a handful of discs and needing to know what side of which disc is the program you think you called COPYCAT, or was it COPICAT, or KOPICAT? Also, is your original version of the program still on disc.... you think you called it DISCOVER? Which disc, which title, how much space is left on the disc, how long is the file, do I have to do a CAT on each side of each disc 'til I find it?

Not being one to keep tidy records of my jaunts through my 664, and being frustrated trying to find different versions of similar programs, I decided this computer should help me straighten things out. COPYCAT is the result and it works well.

COPYCAT enables you to create a disc cover which fits neatly into the clear plastic disc protection box. You can then refer to the following details without the need to remove the disc from the box:

1. Your disc reference number (a three character string)
2. Your disc title (up to 26 characters)
3. All details as shown on the monitor when a CAT is performed for both sides of the disc, ie.

- a) program/file title
- b) k bytes of disc space used by each program/file
- c) k bytes free on each side of the disc

The program was written on the 664 but uses a short machine code routine instead of COPYCHR\$ so as to be suitable for use on the 464 with disc drive and the 6128. The printer I use has a 17cpi (compressed

printing), double width printing and line feed can be adjusted in 1/144" increments. The program uses L of 9/72" and 1/6". With minimal adjustments to the program it should be able to run with any dot matrix printer.

Shown below is an example of the output from COPYCAT. Don't worry if you have a mass of short files, as the program will extend the length of the cover to fit all the files; when it does this it lets you know with a "CONTINUED OVERLEAF" note at the bottom of the front cover.

```
DISC No. 009 AMSTRAD USER Vol.1
-----
SIDE=1 72K free
ANSFILE1.BAS-2 CASSLAB .BAS-7 HAND14 .LOG-1 ML1ST67 .BAS-51
ANSFILE2.BAS-2 CODEWORD .BAS-2 LEADTEST .BAS-1 POKERSX .BAS-21
ANSFILE3.BAS-3 D3KCHT .BAS-7 LOCODER .BAS-1 RANDMSX .BAS-41
ANSFILE4.BAS-3 FILECOPY.COM-3 LPV11 .BAS-3 WINDMILL .BAS-11
ANSTEST . -1 GETREX .BAS-2 ML1ST1 .BAS-6 YAHTZEE .BAS-171
ANSTEST .BAK-1 GUNFLIGHT .BAS-9 ML1ST4 . -4
BANKCAT .BAS-3 HAND11 .LOG-1 ML1ST45 .BAS-6
-----
SIDE=2 3K free
BUDGET .BAS-6 GRAFLOT .BAS-7 MCR1WS .BAS-2 NOTEPLAY .BAS-11
BUSHUNT .BAS-4 HEADREAD .BAK-5 MENU10 .BAS-3 PERCENTS .BAS-31
CALENDAR .BAS-3 HEADREAD .BAS-5 MENU13 .BAS-4 POKER .BAS-161
CLOCKFACE .BIN-17 SUBARCH .BAS-3 ML1ST1 .BAS-6 QUIZ .BAS-61
CLOCK1 .BAS-15 JUPIC1 .BAS-1 ML1ST2 . -2 REC1PE .BAS-131
CLOCKFT1 .BAS-15 JUPIC2 .BAS-1 NOWER .BAS-6 RXAGEN .BAS-31
FILECOPY.COM-3 JUFICS .BAS-1 MUSL1ST1 .BAS-5 SCROMP .BAS-21
GRAFLOAD .BAS-1 KABOOM .BAS-2 MUSL1ST2 .BAS-2 SORTS .BAS-31
```



```

10 'COPYCAT by Dennis Shanahan
20 MEMORY 39999
30 FOR i=1 TO 7
40 READ N%
50 POKE (40000+i),N%
60 NEXT
70 DATA &cd,&60,&bb,&32,&10,&a4,&c9
80 MODE 1
90 PRINT"C O P Y C A T by Dennis S
hanahan"
100 LOCATE 8,4:PRINT "1. Input NEW Lable
Details"
110 LOCATE 8,6:PRINT "2. Make NEW COVER(
S)"
120 LOCATE 10,22:PRINT"Select <1> or <2>
"
130 k$=INKEY$:IF k$="" THEN 130
140 ON INSTR("12",k$) GOTO 420,570
150 GOTO 130
160 k$=INKEY$:IF K$="" THEN 160
170 MODE 2:PRINT "The Computer is Readin
g the Catalog"
180 CAT
190 FOR tm=1 TO 250:NEXT tm
200 FOR y=5 TO 22
210 ct=ct+1:ct$(ct)=": "
220 FOR x=1 TO 61 STEP 20
230 FOR xinc=0 TO 15
240 IF xinc=12 THEN ct$(ct)=ct$(ct)+"-":
GOTO 270
250 IF xinc=13 THEN GOTO 270
260 LOCATE x+xinc,y:CALL 40001:ct$(ct)=c
t$(ct)+CHR$(PEEK(42000))
270 NEXT xinc
280 ct$(ct)=ct$(ct)+" "
290 NEXT x
300 MID$(ct$(ct),69)=":!"
310 IF LEFT$(ct$(ct),5)<>": " THEN 40
0
320 res=""
330 FOR xinc=1 TO 10
340 LOCATE xinc,y+1:CALL 40001:res=res+C
HR$(PEEK(42000))
350 NEXT xinc
360 IF side=2 THEN 380
370 ct$(0)=ct$(0)+res:GOTO 390
380 ct$(sd)=ct$(sd)+res:ct$(ct)="":full=
ct
390 y=22
400 NEXT y
410 RETURN
420 '***** new lable *****
430 CLS:CLEAR:DIM ct$(37):ct=0
440 LOCATE 14,1:PRINT "LABEL TITLE"
450 LOCATE 1,4:INPUT "Disc No. (max. 3 c
haracters) ";dno$
460 LOCATE 1,6:INPUT "Disc Name(max.26 c
haracters) ";names$
470 LOCATE 1,10:PRINT "Place disc, for w
hich lable is to"
480 PRINT:PRINT "be made, in drive with
side No.1 up.."
490 PRINT:PRINT "Then, press any key to
continue."
500 ct$(0)=SPACES(8)+"SIDE=1 ":side=1
510 GOSUB 160
520 MODE 1
530 PRINT "Remove disc then insert with
side No. 2"
540 PRINT:PRINT "UP... Then press any
key."
550 ct$(ct)=SPACES(8)+"SIDE=2 ":side=
2:sd=ct
560 GOSUB 160
570 '***** new cover(s) *****
580 MODE 1
590 LOCATE 14,1:PRINT "NEW COVERS"
600 LOCATE 1,10:INPUT "How many Copies";
cp
610 FOR c=1 TO cp
620 GOSUB 660
630 PRINT #8,CHR$(27);CHR$(64):'printer
reset
640 NEXT c
650 GOTO 10
660 'set printer to 17CPI and LF=9/72"
670 PRINT #8,CHR$(27);CHR$(15);CHR$(27);
CHR$(51);CHR$(18);
680 GOSUB 690: GOTO 700
690 FOR n=1 TO 69:PRINT #8,"-";:NEXT n:P
RINT #8: RETURN
700 PRINT #8,": DISC No. ";dno$;:GOSUB 7
10:GOTO 720
710 PRINT #8,CHR$(27);CHR$(14);:RETURN:'
printer double width
720 PRINT #8," ";names$:GOSUB 730:GOTO 74
0
730 PRINT #8,CHR$(27);CHR$(15);:RETURN:'
printer condensed
740 GOSUB 690
750 PRINT #8,CHR$(27);CHR$(51);CHR$(24);
:'printer LF=1/6"
760 FOR p=0 TO 37
770 IF ct$(p)=" THEN p=37: GOTO 820
780 IF p=17 AND full>18 THEN GOSUB 890:G
OSUB 690
790 IF p=0 OR p=sd THEN GOSUB 710
800 PRINT #8,ct$(p)
810 IF p=0 OR p=sd THEN GOSUB 730
820 NEXT p
830 IF full<18 THEN GOTO 850
840 GOSUB 690:RETURN
850 sp$=":"+SPACES(67)+"!:"
860 FOR n=full+1 TO 18
870 PRINT #8,sp$
880 NEXT n:GOTO 840
890 PRINT #8,": C O N T I N U E D O V
E R L E A F";SPACES(31);"!";:RETURN

```

A full screen Editor for ams-Forth

by Petr Luk

FORTH is a fascinating language. It demands more discipline from the programmer, but provides more freedom than other high-level languages. It is not very widely used on micros, because it is not normally bundled with the machine (the English Jupiter ACE was released in late 1982 with FORTH as the only language, but it was not a commercial success). Expressions must be entered in Reverse Polish Notation ("Polish" after the originator of the prefix notation Jan Lukasiewicz, whose name was considered to be too difficult to pronounce; "Reverse" because it is postfix, the reverse of prefix). The interpreters and compilers of other languages convert the usual algebraic expressions into RPN before evaluating them, so this notation is more suitable to the machine while it requires some extra programming effort.

The biggest obstacle to learning FORTH is lack of published programs which can be used as

examples, compounded by the fact that there are three standards (fig.79, and 83) and every implementation has various extensions.

There are two fig-FORTH implementations for the CPCs. The later one, 4TH, by Abersoft, was featured in the tutorial series in the U.K. Amstrad Computer User, March to August 1985. ams-FORTH is considerably cheaper and comes with a very elementary line-based editor, which means that it is necessary to re-type a whole line to correct a mistake. (I have not seen the 4TH editor, and ams-FORTH does not have ASSEMBLER).

The following program can ease the pain of entering programs, as well as indicate how things can be done in FORTH. It is not necessarily the best FORTH, being a very close translation from PASCAL (and my first FORTH attempt), but it is a start. A word-processor which produces uncompressed ASCII text and saves

it with a header (i.e. as a binary file) could be used for entering the program for subsequent loading SCR-LOAD, but test it first.

It compiles to about 2.3k; a similar BASIC program needs around three times the memory. I have recovered most of the 2.3k by lowering FENCE, FORGETTING all words after VLIST, re-installing LIST and FREE (use SPELL to see how they are constructed; the new LIST must not call the now forgotten EDITOR), and then making this editor a part of the system. Obviously, this requires considerable care and patience. The speed could be improved by coding DISP in machine-language and a CASE structure for testing the keys, but that can be done later. The value in HI could be increased to 7D00h, without running into problems when crossing the 7FFFh boundary. The extra 1k could provide an extra screen, or the value in LO could be increased to 5500h for more program space.

```
OK
2 1 0 L157 L157 L157
BLOCK 0
0 ( screed 0) ( LKS B60521)
1 A full-screen RAM-disc editor for ams-FORTH 1.11. Use
2 { n L157 } for hard copy, n LOAD to compile. Screen 0 (lines
3 0 to 15) for comments only; 0 LOAD crashes the system !
4 The editor is in overstrike mode at all times, and allows
5 insertion/deletion of characters; this affects all text from the
6 cursor position to the end of the last screen. The text must be
7 formatted into lines and screens before LOAD is invoked.
8 COPY allows copying of any one line within the screens area to
9 the line at the cursor. It will overwrite the existing text but
10 does not delete the original line. COPY and GUY require the
11 absolute line number (unsigned); invalid entries are ignored.
12 SEARCH tries to match the input string through the whole screens
```

13 area. Corrections may be made if a match is found, but each
14 search must be completed to avoid stack build-up.
15 ESC is available, but exit by CLR to reset the DEL key.

```
Block 1
0 ( screed 1)
1 FURTH SP: DECIMAL LK FREE
2
3 H1 @ LU @ - CH / 1 64 CONSTANT CH ( chars/line)
4
5 23 CONSTANT DL ( no of display lines)
6
7 0 VARIABLE TL ( top line on screen)
8
9 0 VARIABLE CC ( cursor col)
10
11 0 VARIABLE CL ( cursor line)
12
13 0 VARIABLE MA ( memory address)
14
15 0 VARIABLE T1 ( temporary store)
16
17 0 VARIABLE T2 ( temporary store)
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
Block 2
0 ( screed 2)
1 CS C, NEXT, SMUDGE
2 CREATE CUS ( set curs pos from CC, CL)
3 ED C, SB C, CC, 53 C, ( 1d DE, LCCJ 1d H, E)
4 ED C, SB C, UL, 68 C, ( 1d DE, LULL 1d L, E)
5 CD C, BB75, CS C, NEXT, SMUDGE ( call TXT SET CUR exit)
6 CREATE KEYDEF CS C, ( push BC)
7 SE C, 0 C, 6 C, B C, CD C, BB27, ( NU/A=wup)
8 SE C, 0 C, 6 C, 5 C, CD C, BB2D, ( SU/A=S)
9 SE C, 1 C, 6 C, 7 C, CD C, BB27, ( NK/A=wrt)
10 SE C, 2 C, 6 C, A C, CD C, BB27, ( ND/A=wdn)
11 SE C, 2 C, 6 C, 6 C, CD C, BB2D, ( SD/A=6)
12 SE C, 8 C, 6 C, 8 C, CD C, BB27, ( NL/A=wlt)
13 SE C, 7 C, 6 C, F C, CD C, BB27, ( NCOPY=15)
14 SE C, 44 C, 6 C, 1 C, CD C, BB27, ( NLAB=1)
15 -->
```

OK

```
5 4 3 LIST LIST LIST
Block 3
0 ( screed 3)
1 SE C, 44 C, 6 C, 2 C, CD C, BB2D, ( STAB=2)
2 SE C, 4F C, 6 C, 3 C, CD C, BB27, ( NDEL=3)
3 SE C, 4F C, 6 C, 4 C, CD C, BB2D, ( SDEL=4)
4 C1 C, CS C, NEXT, SMUDGE ( pop BC exit)
5 CREATE DELKEY ( restore DEL) CS C, ( push BC)
6 SE C, 4F C, 6 C, 7F C, CD C, BB27, ( NDEL=127)
7 C1 C, CS C, NEXT, SMUDGE
8 CREATE INS ( insert T1+1 spaces at cursor) CS C, ( push BC)
9 2A C, H1, 2B C, ( 1d HL, lmemory topJ dec HL)
10 ED C, 4B C, 11, 5D54, ( 1d BU, lshittJ 1d D, H 1d E, L)
11 3 C, 1A C, 1B C, ( inc BC loop1: 1d A, LDEL dec DE)
12 FE C, 21 C, 3B C, FA C, ( cp "!" jr c, loop1)
13 13 C, ED C, 52 C, ( inc DE sbc HL, DE)
14 ED C, 42 C, 3B C, 16 C, ( sbc HL, BC jr c, exit)
15 -->
```

Block 4

```
0 ( scred 4)
1 626B , 9 C, 5B C, ( 1d L,E 1d H,D add HL,BC ex DE,HL)
2 ED C, 4B C, MA , ( 1d BC,[memory address])
3 B7 C, ED C, 42 C, ( loop2: or A sbc HL,BC)
4 38 C, 9 C, ( jr c, exit)
5 9 C, 7E C, ( add HL,BC 1d A,HLJ)
6 36 C, 20 C, 12 C, ( 1d [HLJ,space 1d [DEJ,A)
7 281B , 1A C, F2 C, ( dec DE dec HL jr loop2)
8 C1 C, C3 C, NEXT , SMUDGE ( exit)
9 CREATE DEL ( delete TI chars at cursor) CS C, ( push BC)
10 2A C, MA , ED C, 4B C, F1 , ( 1d HL,[ma] 1d BC,[shift])
11 5054 , 9 C, ( 1d D,H 1d E,L add HL,BC)
12 ED C, 4B C, HI , ( 1d BC,[memory top])
13 B7 C, ED C, 42 C, ( loop: or A sbc HL,BC)
14 30 C, 9 C, ( jr nc, exit add HL,BC)
15 -->
```

Block 5

```
0 ( scred 5)
1 7E C, 36 C, 20 C, 12 C, ( 1d A,HLJ 1d [HLJ,space 1d [DEJ,A)
2 251B , 1B C, F2 C, ( inc DE inc HL jr loop)
3 B1 C, C3 C, NEXT , SMUDGE ( exit)
4 CREATE AMIT ( c --- ) ( prints all codes via [X] WK CHR)
5 B1 C, L5 C, 7D C, ED C, B5D , C1 C, C3 C, NEXT , SMUDGE
6 DECIMAL
7 : DISP ( from MA to end of current display, reset cursor)
8 [L @ DL + CH * LU @ + 1 - MA @ DU [ @ AMIF DUUP CUS ;
9 : ND? ( accept a valid line number) 7 EMIT
10 1 QUERY IS WOKU HERE DUP DU @ 1+ + SWAP 1+ DU 1.C@ BASE @
11 DIGIT IF DRUP ELSE 0= LEAVE THEN DUUP DUP HERE SWAP
12 IF NUMBER DRUP SWAP ELSE 80 ERASE THEN DUP IF SWAP DUP DUP
13 0< SWAP ML > OR IF DRUP 0= ELSE SWAP THEN THEN CUS ;
14 : $= ( string match) SWAP -DUP IF OVER + SWAP DU DUP C@
15 -->
```

OK

Block 6 LIST LIST LIST

Block 6

```
0 ( scred 6)
1 1 DE - IF 0= LEAVE ELSE 1+ THEN DUUP ELSE DRUP 0= THEN ;
2 : ROUND ( SMUDGE 1 ( called recursively by SEARCH) BEGIN
3 BEGIN BEGIN CUG [L @ CL @ 2 - + DUP DUUP 2DUUP 11 ;
4 2 EMIT 30 EMIT 4 .K 3 SPACES 16 /MUD
5 2 .R DB EMIT 2 .R CUS 3 EMIT
6 0< SWAP ML > OR WHILE 11 [L @ 0< IF 1 - THEN EMIT
7 EMIT REPEAT .S
8 CL @ DUUP 3 < SWAP DL > OR WHILE DL 2 / CL @ DUP 3 <
9 IF SWAP MINUS 1+ + ELSE SWAP - THEN
10 [L @ + DL MINUS MAX ML MIN DUP IL ;
11 [L * LU @ + MA ! 30 EMIT 15 SPACES CR DL 2 / CL : DISP
12 REPEAT
13 [L @ [L * LU @ + CL @ 1 - + MA ! KEY 127 AND
14 DUUP 1 = IF DRUP 0 0 [L ! 1NS THEN
15 -->
```

Block 7

```
0 ( scrd 7)
1 DUP 2 = IF DRUP 0 CH CC @ - T1 ! INB THEN
2 DUP 3 = IF DRUP 0 1 T1 ! DEL THEN
3 DUP 4 = IF DRUP 0 24 EMIT CH CC @ - 1+ DUP T1 ! MA @ SWAP
4 TYPE ." Confirm" CUS 24 EMIT KEY 4 = IF DEL THEN THEN
5 DUP 12 = IF DRUP 0 24 EMIT 18 EMIT ." Goto line : " NO?
6 IF IL ! DRUP 5 THEN 24 EMIT THEN
7 DUP 13 = IF CR THEN
8 DUP 14 = IF DRUP 11 @ 12 ! 26 26 1 INK ." Search : " QUERY
9 13 WORD HI @ LU @ DU 2 $ EMIT EMIT ! HERE CR HERE 1+ $=
10 IF 1 LU @ - CH / 1L ! 30 7 EMIT EMIT ROUND 13 EMIT HERE 1+
11 HERE CR TYPE THEN LOOP 24 24 1 INK 12 @ IL ! 7 EMIT 5 THEN
12 DUP 15 = IF DRUP 0 24 EMIT 18 EMIT ." Copy line : " NO?
13 IF CH * LU @ + DUP T1 ! CH TYPE ." Confirm" CUS KEY
14 15 = IF T1 @ MA @ CH CMOVE THEN THEN 24 EMIT THEN
15 -->
```

Block 8

```
0 ( scrd 8)
1 DUP 0= IF DISP BEGIN ?TERMINAL 0= UNTIL THEN
2 DUP 5 = IF 30 EMIT THEN
3 DUP 6 = IF 24 LL ! LUS THEN
4 DUP DUP BL < SWAP 16 > AND IF DRUP 7 THEN
5 DUP 2DUP EMIT BL < 0= IF MA @ C! ELSE DRUP THEN
6 16 = UNTIL ! SMUDGE
7 : SCHED DECIMAL 2 MODE 24 24 1 INK 0 PEN 1 PAPER CR CR CR
8 ." Screen editor for ams-FORTH 1.1 860521" CR
9 ." P. LUKES, 26 Noll St., 100W00MBA, 4350" CR
10 ML 1+ ." lines, shown as: abs line scr:line in scr" CR
11 ." Normal cursor keys, shifted " 241 240 EMIT SPACE EMIT
12 ." : page toggle" CR ." Special function keys : " CR
13 ." CLR exit" CR
14 ." normal/shift : TAB insert space/s, DEL delete char/s" CR
15 -->
OK
```

7 LIST

Block 9

```
0 ( scrd 9)
1 ." (insertion only if there is free space at end)" CR
2 ." (confirm multiple delete by shift+DEL)" CR
3 ." ctrl+N search : " CR
4 ." enter string; cont by CLR until paper changes!" CR
5 ." ctrl+L goto : enter abs line no" CR
6 ." copy copy line : enter abs line no; confirm by COPY" CR
7 ." erase all screens (N/y) ." 7 EMIT KEYDEF EMPTY-BUFFERS
8 KEY 121 = CR IF LU @ DUP HI @ SWAP - BLANKS ." Erased"
9 ELSE ." Converting non-printing codes to blanks"
10 HI @ LU @ DU 1 CR BL < IF 1 1 BLANKS THEN LOOP THEN
11 CR ." Press a key " 7 EMIT KEY SP! 0 0 CH 1 - 24 WINDOW
12 30 EMIT 0 1L ! ROUND DELKEY 2 MODE WARM :
13 7 EMIT FREE
14
15
OK
:
```

Book Reviews

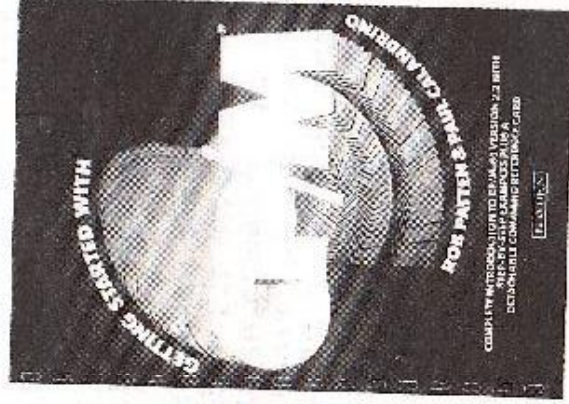
In addition to "High Energy Programs for the Amstrad" which has now been made generally available to non-subscribers at a cost of \$9.95 including postage (see elsewhere in this issue for details), there are four more books which are now stocked by Strategy Publications, and I am pleased to say that three of them cover CP/M.

Why books on this subject are hard to come by I shall never know, and why do they always seem to come from America? I hasten to add that the CP/M books are not Amstrad specific but then again it doesn't really matter when you are talking about the Control Program for Micro-computers.

Enough of my questions - let me give you a quick run-down on the titles.

Getting Started with CP/M

by Rob Patten and Paul Calandrino



It's a spiral bound book which lays flat - always useful when you are trying to type and read from a book at the same time. It is a complete guide to CP/M 2.2, the most widely used operating system for 8-bit computers. It explains how CP/M operates and how to use its command structures and functions. There are five chapters:

1. About CP/M - Data Storage and file structures, File names and extensions, Discette care, loading CP/M, CP/M prompt and drive designation, and general command structure.
2. CP/M commands - CP/M's

by Simon Antho

internal structure, line editing commands, built-in commands (DIR, ERA etc.), Transient commands (STAT, PIP etc.) and general error messages and conditions.

3. Work copy preparation - booting CP/M, disc formatting & duplication.

4. CP/M and application programs.

5. A sample session - file copy transfer and creating a file.

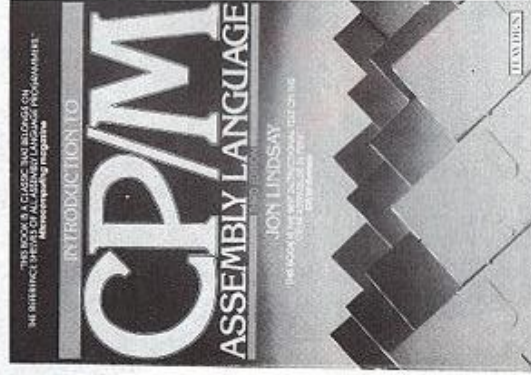
The Appendices contain a full Command/Function table and the obligatory glossary. Right at the end of the book is a most useful tear-out CP/M command referen card.

All in all, I believe the subjects which most beginners will find useful are covered. It contains 99 pages and costs \$29.95.

Introduction to CP/M Assembly Language

by Jon Lindsay

This book provides a clear, step-by-step presentation of CP/M functions, the use of program development tools, the structure of CP/M and the heart of assembly language programming, the BDOS calls. More than 50 program listings/routines illustrate applications of fundamental assembly language instructions and show the proper use of CP/M function calls. Guidelines for



avoiding common errors often encountered by the novice programmers are also included. As the author says, it is not a book which provides an in-depth study of CP/M and the interfacing with various hardware. It is an instructional text on the CP/M assembler with specific examples. Lindsay also deals with the most common algorithms used in assembly language programs, such as how to set up program loops, how to use counters, how to change ASCII numbers into binary etc. It's well indexed too.

A useful book if you want to start getting a little more serious. It consists of 151 pages and costs \$35.95.

Inside CP/M Plus

by David Cortesi
David Cortesi has written a number of books on CP/M, in particular a very successful title 'Inside CP/M - A Guide for Users and Programmers'. 'Inside CP/M Plus' follows the clarity and



authority that made the earlier book a valuable reference source on CP/M. As it is meant as a guide to users, it avoids technical terminology and the sorts of detail programmers love. It includes tutorial sections and reference sections which serve to cover only commands rather than describing every aspect of the system. Overall, the book gives users the background and understanding they need to use CP/M in a sophisticated way and thereby achieve maximum performance from their software rather than being at its mercy (or rather, at the mercy of the original designer of the application).

It weighs in at 261 pages with twelve chapters of pretty comprehensive reference material and is surprisingly just \$29.95.

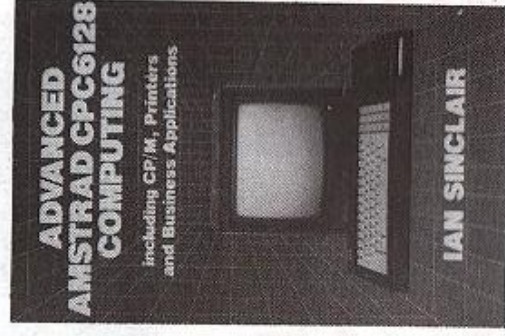
Advanced Amstrad CPC6128 Computing

By Ian Sinclair
Ian Sinclair produced a very

popular book "Amstrad Computing" (it still sells well) which introduced the new user/programmer to the CPC464. His new offering is built upon the success of that style and contrasts effectively the differences between the 6128 and earlier models. With the extra 64k available in the 6128, he provides explanations of its use with illustrative programs.

The level of the book is pitched at people who already know how to program in BASIC, and have a reasonable knowledge of computing terms. It introduces the reader to more advanced methods of BASIC and AMSDOS such as disc file maintenance, creating, searching and sorting, graphics and sound, and of course, the use of the extra memory. It also covers the principals of CP/M Plus and how to connect a second disc drive and make effective use of it. You will also find a useful skeleton database program called 'Filing Cabinet'.

This book is undoubtedly a valuable addition to the 6128 owner's library. It has 170 pages and costs \$27.95.



Poke a Monster

from P.T. Crowe (SCAUG)

This is a small program to utilise the DATA produced by the BUILD A SPRITE program in last month's magazine. All we have to do is poke it into memory to see the monsters fly around the screen. Firstly decisions have to be made:

1. Do you want to change the data in the Roland Game.
2. Poke in a new monster after Roland has been loaded?

The answer to the first is merely to replace the monster data in the Roland game to the new data. To achieve the second you will need to insert the program below into the first listing of Roland Takes a Running Jump. All this program does is to read DATA and then poke it in to the locations which hold the monster data.

Here's how it works:

RUN the Roland program using this amended listing. At any time after the program has run, press BREAK twice. You are then presented with a MENU. From the MENU select what you want to do. Within the listing below, are DATA lines which we have filled with two totally different monsters. The DATA within these lines can be replaced with your own creations. You can also add data lines, but one thing to remember is, that the last line of DATA must be:

DATA "****"

```
10 ON BREAK GOSUB 250
20 x=35043:y=35107:DIM pps(64),ps(64):MEM
MORY &3FFF:LOAD"rol-game",&8000:LOAD"scr
sens",&4000
30 MODE 1:INPUT"Which start screen":lev
1
40 POKE &8392,level
50 ENT-1,1,-2,1:ENV 1,15,-1,4:SPEED INK
2,2
60 CALL &8354
70 GOTO 250
80 ' *****
monster 1 ***** peek memory m
90 CLS#2:FOR x=35044 TO 35107:PRINT#2,x:
"USING"\":HEX$(PEEK(x)):NEXT
100 RETURN
110 ' ***** load monster 1 &
2 data *****
120 FOR n=1 TO 64:READ ps
```

```
130 IF ps="*" THEN RESTORE:GOTO 110
140 ps(n)=ps:NEXT
150 FOR nn=1 TO 64:READ pps
160 IF pps="*" THEN RESTORE:GOTO 150
170 pps(nn)=pps:NEXT
180 RETURN
190 '
200 ' ***** poke monster 1
*****
210 CLS#2:FOR n=1 TO 64:x=x + 1
220 p=VAL("&" + pps(n)):PRINT#2,x"USING"
\":ps(n):POKE x,p:NEXT
230 x=35043:RETURN
240 ' *** GO AND LOAD DATA *****
250 GOSUB 120
260 ' ***** SET UP MENU *
*****
270 CLS:MODE 2:WINDOW#1,40,80,18,25:WIN
DOW#2,1,80,1,17
280 LOCATE#1,1,1:PRINT#1,"To Peek at mon
ster 1 ----- Press ~ A ~"
290 LOCATE#1,1,2:PRINT#1,"To Peek at mon
ster 2 ----- Press ~ B ~"
300 LOCATE#1,1,3:PRINT#1,"To Poke monste
r 1 ----- Press ~ 1 ~"
310 LOCATE#1,1,4:PRINT#1,"To Poke monste
r 2 ----- Press ~ 2 ~"
320 LOCATE#1,1,5:PRINT#1,"To List -----
----- Press ~ L ~"
330 LOCATE#1,1,6:PRINT#1,"To Run -----
----- Press ~ R ~"
340 ps$=INKEY$
350 IF ps$="1" THEN GOSUB 210
360 IF ps$="2" THEN GOSUB 460
370 IF ps$="a" OR ps$="A" THEN GOSUB 90
380 IF ps$="b" OR ps$="B" THEN GOSUB 430
390 IF ps$="l" OR ps$="L" THEN GOTO 420
400 IF ps$="r" OR ps$="R" THEN n=0:nn=0:
GOTO 30
410 GOTO 340
420 CLS:LIST
430 ' ***** peek memory
monster 2 *****
440 CLS#2:FOR x=35108 TO 35171:PRINT#2,x
:"USING"\":HEX$(PEEK(x)):NEXT
450 RETURN
460 ' ***** poke monster 2 d
ata *****
470 CLS#2:FOR nn=1 TO 64:y=y + 1
480 pp=VAL("&" + pps(nn)):PRINT#2,y"USIN
G"\":pps(nn):POKE y,pp:NEXT
490 y=35107:RETURN
500 ' ***** WARRIOR1 *****
510 DATA 00,00,05,00,01,0f,08
520 DATA 00,03,60,0c,04,03,f0,0c
530 DATA 04,01,90,08,0e,00,60,0c
540 DATA 40,67,ef,ee,40,ff,5f,ff
550 DATA 66,dd,bf,bb,62,dd,ff,bb
560 DATA 73,cc,ff,22,51,88,0f,22
570 DATA 40,00,ff,22,40,11,99,88
580 DATA 40,11,00,88,40,03,00,0c
590 ' ***** robot *****
600 DATA 08,70,e0,00,0c,d0,b0,00
610 DATA c0,70,e0,00,c0,20,40,00
620 DATA c0,50,c0,00,f0,f0,f0,e0
630 DATA 70,84,12,f0,10,95,9a,b0
640 DATA 10,95,9a,b0,10,84,9a,b0
650 DATA 10,95,12,83,10,f0,f0,81
660 DATA 10,d0,b0,80,10,80,10,80
670 DATA 05,08,01,0a,07,0c,03,0e
680 DATA "****"
```


Gossip from the UK

❖ No sooner than the agreements transferring control of Sir Clive Sinclair's home computer division to AMSTRAD were exchanged, \$85 was chopped from the price of the Spectrum 128K. Obviously, the plan is to dispose of as much stock as possible before the revamped "SPAMSTRAD" is launched. (Sorry but I couldn't think of anything else to call it). One of the main problems with the old Sinclair range was reliability, something that most Amstrad users have not encountered. In the meantime we wait for news on Amstrad's plans - not that it will interest Amstrad owners too much as the new range will not be compatible with their machines.

❖ British Telecom have successfully prosecuted two computer hackers. The names Schifreen and Gold will go down in legal history as being the first people to be convicted of offences relating to unauthorised access to remote computers. The pair were charged with nine counts of forgery which related to unauthorised access of a particular Prestel computer, including a development test computer and Vampire, a Prestel status computer. The prosecution alleged that several Prestel accounts had been used without proper authorisation. What was interesting was the way that BT gathered its evidence. Apparently they used telephone call loggers and data monitors attached to the telephone lines of both defendants before they were caught. There seems to be a difference between this and telephone tapping, the latter requiring a special

warrant. I wonder what Telecom Australia are doing?

❖ *Tempest*, a well known and enjoyed arcade game, is soon to be available for the Amstrad. The official Activision version will be sold on the Eleetric Dreams label. With keyboard or joystick option, there are 99 screens each featuring a different shaped chasm.

❖ *Questprobe 3* has at last been released by US Gold. It's a graphic adventure featuring 'The Human Torch' and the 'Thing', the game is based on the Marvel comic characters in which you take the part of both heroes. (You can help your partner by switching roles). The 'baddies' are the Blob and a Ringmaster with his Circus of Crime - all out to stop you from saving Ms. Alicia Masters from the evil Dr. Doom.

❖ *Equinox* is an arcade adventure from Mikro-Gen, the producers of the "Wally" series of games. In this game Wally seems to have been replaced by a spherical cyclops. It is an exceedingly fast game with the potential for hours of manic action. The object is to clear rooms of radio-active waste whilst dodging the 'nasties'. I reckon it's difficult and can't believe that anyone will make it to the eighth level.

❖ *World Cup Carnival* from US Gold naturally hit the games charts pretty quickly but has caused quite a

stir. You see, it is an adaptation of Artic's World Cup Soccer yet, unknowingly, it is being bought with the impression that it is a new game has been reported that 25% of the copies have been returned. I wonder that was *after* England got knocked out of the World Cup?

❖ The complete range of MicroPro applications software, except the Wordstar family, have been taken over by Software Classics. They are CalcStar (spreadsheet), ReportStar (reports from DataStar information), SuperSort (sorting utility), DataStar (data entry) and InfoStar (a combination of DataStar and ReportStar).

Note: Release dates of software in the UK quite often change and availability in Australia obviously depends upon these and how quickly the importers can bring them into this country.



COMPUTER
ACCESSORIES

FROM PRIDE UTILITIE

SYSTEM X — 32 new extended Basic commands for the 464/664.
Cassette \$18.50 Disc \$26.50.

ODDJOB — A complete disc utility.
Disc only — \$29.95.

TRANSMAT — Easy tape to disc transfer Auto or non-auto modes. Cassette \$18.50 Disc \$26.50.

TOMCAT — Cassette back-up copier.
Cassette — \$17.50.

SPIRIT — Transfer genuine headerless programs/files onto disc. Cassette \$15.50.

SCRIPTOR — Enhance your DMP-1 printer. Cassette — \$17.50.

Some programs are sold on the understanding that they must not be used to infringe copyright.

8-BIT PRINTER PORT From K.D.S.

Allows you to send those codes above ASCII 127. Only \$54.50.

SIDEWAYS ROM From K.D.S. Holds 5 ROMS (8 or 16K) Accepts 27 series ROMS. Only \$95.00.

ALL PRICES INCLUDE P&P WITHIN AUSTRALIA. PHONE OR WRITE FOR FURTHER INFORMATION.



P.O. BOX 288,
MORRISSET 2264, NSW.
PHONE (049) 732754

Progedit

A 464/664/6128 program editing utility

Adapted by C.J. Collins

The BASIC on the Amstrad range of machines allows programs to be saved in ASCII format as well as the more tokenised format. ASCII format is, basically, exactly the same as the way in which a listing appears on the screen. This format can be easily modified by other programs. To save a program in ASCII format, append 'a' to the Save command.

Progedit loads the ASCII format of a program into memory and allows it to be modified in the following ways:

1. Expanded program listing
2. String searching
3. Global search and replace
4. String search with confirmed replace
5. REM statement removal

The expanded program listing lists the program to the printer, with each statement on a multi-statement line appearing on a separate line. Lines containing REMs are printed in bold, italic and 10 pitch. GOTOs, GOSUBs and RETURNs are flagged G, S and R in the lefthand margin.

The searches and replaces are done as might be expected with phrases found being highlighted. Sometimes the program seems to produce extra copies of lines after a long phase of searching and replacing. These can be cleaned up by saving the program and loading it in as a normal program without Progedit. BASIC will the remove all occurrences of extra lines.

The REM remover will not remove remarks that have been typed in with a " ".

It will only remove lines that start with "REM".

With search and replace, be careful which string you choose to use. If, for example, you use "the" as the search string, Progedit will also show you the "the" in "then". This is easy to fix - just put in a space at the end of your search string, ie. "the ".

It is sensible to load and resave all programs after they have been put through Progedit. All printer codes are to run an Epson or Epson compatible printer.

This is an adaptation of a program by Alan Dunlop-Walters that was originally written to run on the PCW range.

```
10 'progedit
20 MODE 2:INK 0,1:INK 1,24:BORDER 1:PEN
1:PAPER 0
30 POKE 46642,255:'change to POKE 46312,
255 for CFC464
40 DIM a$(2000)
50 rev$="< ":revofff$=" >"
60 PRINT:PRINT TAB(21)" PROGEDIT - Pro
gram Editing Utility"
70 PRINT:PRINT:PRINT" Note : Program to
be edited should previously have been sa
ved in ASCII format.":PRINT
80 PRINT:LINE INPUT"
ENTER to continue or 'A' to ABORT > ",r
es$
90 IF res$="A" THEN CALL 0
100 ON ERROR GOTO 960
110 PRINT:PRINT:LINE INPUT " Ent
er Name of File to Process(Include Exten
sion > ",prog$:PRINT
120 PRINT:PRINT TAB(33)"Loading File"
130 OPENIN prog$
140 FOR x=1 TO 2000
150 LINE INPUT #9,a$(x)
160 IF EOF THEN 180
170 NEXT x
180 CLOSEIN: last=x:amend=0
```

```

190 confirm=0:PRINT:PRINT"
Search and Display
bal Search/Replace
Search and Confirm Replace
ark Removal
Expanded Listing
"

```

```

200 PRINT:PRINT "      Operation >"
210 op$=INKEY$:IF op$="" THEN 210
220 IF op$="S" THEN 230 ELSE IF op$="G"
THEN 340 ELSE IF op$="C" THEN 450 ELSE I
F op$="R" THEN 510 ELSE IF op$="E" THEN
600 ELSE IF op$="X" THEN 690 ELSE PRINT
CHR$(7):GOTO 210

```

```

230 PRINT:PRINT TAB(11)"Search and Displ
ay Facility":PRINT
240 LINE INPUT "      Search String
>";ss$:PRINT

```

```

250 n=0
260 FOR x=1 TO last
270 a$=a$(x):IF a$="X" THEN 320
280 p=INSTR(p+1,a$,ss$):IF p=0 THEN 320
290 b$=LEFT$(a$,p-1)+rev$+MID$(a$,p,LEN(
ss$))+revoffs+MID$(a$,p+LEN(ss$)):PRINT
b$:PRINT:n=n+1

```

```

300 IF n=10 THEN PRINT:LINE INPUT "ENTER
to continue",etc$:PRINT:PRINT "...cont
inued...":PRINT:n=0
310 GOTO 280
320 NEXT x
330 GOTO 190

```

```

340 PRINT:PRINT TAB(11) "Global Search a
nd Replace Facility":PRINT
350 LINE INPUT "      Search String
>";ss$

```

```

360 LINE INPUT "      Replace by
>";r$:PRINT
370 FOR x=1 TO last
380 a$=a$(x):IF a$="X" THEN 440
390 p=INSTR(p+1,a$,ss$):IF p=0 THEN 440
400 b$=LEFT$(a$,p-1)+rev$+MID$(a$,p,LEN(
ss$))+revoffs+MID$(a$,p+LEN(ss$)):PRINT
b$:PRINT:n=n+1

```

```

410 a$=LEFT$(a$,p-1)+r$+MID$(a$,p+LEN(ss
$))
420 IF confirm THEN 480
430 a$(x)=a$:PRINT a$:PRINT:amend=1:GOTO
390

```

```

440 NEXT x:GOTO 190
450 confirm=1
460 PRINT:PRINT TAB(11) "Global Search a
nd Confirmed Replace Facility":PRINT
470 GOTO 350

```

```

480 PRINT a$:PRINT:PRINT"      R - t
o replace as shown      I - to l
gnore ":PRINT
490 well$=INKEY$:IF well$="" THEN 490
500 IF well$="I" THEN a$=a$(x):GOTO 390
ELSE IF well$="R" THEN 430 ELSE PRINT CH
R$(7):GOTO 490
510 PRINT:PRINT TAB(33)"REMark Removal":
PRINT

```

```

520 FOR x=1 TO last
530 a$=a$(x):IF INSTR(a$,"REM") THEN ame
nd=1:GOSUB 550
540 NEXT x:PRINT:PRINT TAB(36)"Finished"

```

```

:GOTO 190
550 p=INSTR(a$,"REM ")
560 a$=LEFT$(a$,p-2):a=VAL(a$(
570 IF LEN(a$)<6 THEN a$(x)="X" ELSE
x)=a$
580 IF a$(x)="X" THEN PRINT TAB(32) "
e";a:"Deleted"
590 RETURN

```

```

600 PRINT:PRINT TAB(38)"End"
610 IF amend THEN 630 ELSE PRINT:PRINT
AB(27)"No Amendments So No Receive"
620 FOR X=1 TO 1000:NEXT X:GOTO 80
630 PRINT:PRINT TAB(31)"Resaving Progr
"

```

```

640 OPENOUT prog$
650 FOR x=1 TO last
660 IF a$(x)="X" THEN 600
670 PRINT #9,a$(x)
680 NEXT :CLOSEOUT:GOTO 80
690 PRINT:PRINT TAB(11) "Expanded List:
"

```

```

700 PRINT:LINE INPUT"      Load Pri
ter,Then Press ENTER >";paps
710 PRINT:PRINT TAB(37) "RUNNING"
720 PRINT#8,CHR$(27);"P";CHR$(27);"W1";
heck$
730 PRINT#8,CHR$(27);"W0";CHR$(27);"M";
RINT#8

```

```

740 FOR x=1 TO last
750 IF a$(x)="X" THEN 850
760 a$=a$(x):q=0:GOSUB 880:PRINT#8,n$;"
";

```

```

770 IF r THEN PRINT#8,CHR$(27);"4";CHR$(
27);"G";CHR$(27);"P";
780 FOR y=1 TO LEN(a$):b$=MID$(a$,y,1)
790 IF b$=CHR$(34) THEN GOSUB 860
800 IF b$="." AND q=1 THEN 820
810 IF b$=":" AND q=0 THEN PRINT#8:PRINT
#8,TAB(10)";";GOTO 830
820 PRINT#8,b$:
830 NEXT :PRINT#8:PRINT#8

```

```

640 IF r THEN PRINT#8,CHR$(27);"5";CHR$(
27);"H";CHR$(27);"M";:r=0
850 NEXT x:GOTO 190
860 IF q THEN q=0:RETURN
870 q=1:RETURN
880 n$="      :IF INSTR(a$,"REM ") THEN r=
1 ELSE r=0

```

```

890 IF INSTR(a$,"GOTO") THEN MID$(n$,1,1
)="G"
900 IF INSTR(a$,"GOSUB") THEN MID$(n$,2,
1)="S"

```

```

910 IF INSTR(a$,"RETURN") THEN MID$(n$,3
,1)="R"
920 IF INSTR(a$,"THEN") THEN 940
930 RETURN
940 p=INSTR(a$,"THEN ") :js=MID$(a$,p+5):
IF VAL(js) THEN MID$(n$,1,1)="G"
950 RETURN

```

```

960 amdsoserr=(DERR AND &7F)
970 IF NOT EOF THEN PRINT:PRINT TAB(23)"
Program Has Too Many Lines...ABORTED":PR
INT CHR$(7):CLOSEIN:RESUME 80
980 IF amdsoserr=16 OR amdsoserr=18 THEN
PRINT CHR$(7):PRINT TAB(33)"NO SUCH FIL
E!":RESUME 100

```