INCORPORATING THE TANSOFT GAZETTE

ORIC OWNER

ISSUE 2

JUNE/JULY

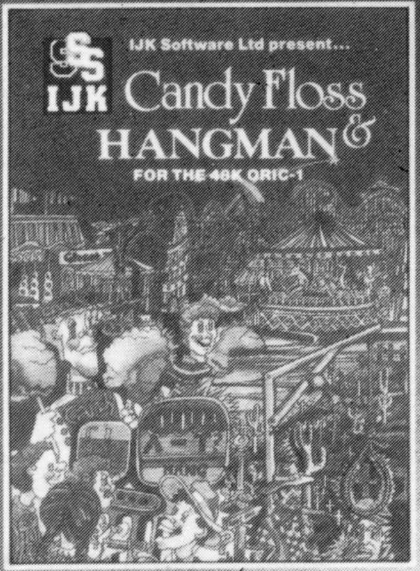
INSIDE THIS ISSUE

The ORIC Interview: Peter Harding

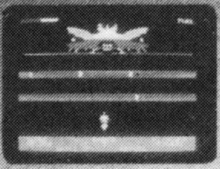
Forth Tunesmith

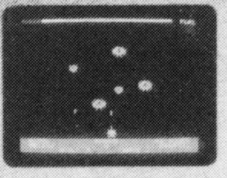
Cassette Handling explained

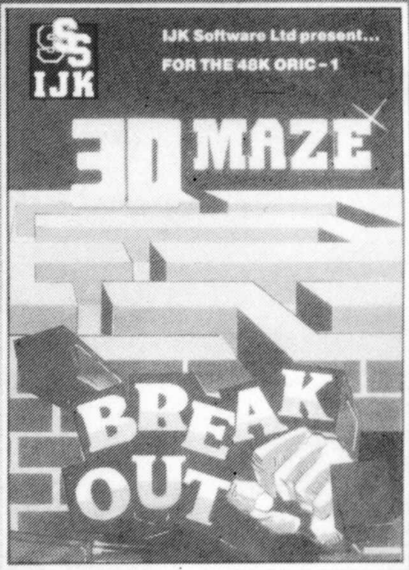
Together the ORIC-1 Micro and IJK Software bring you the dawning of a new era...

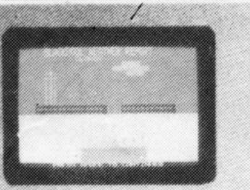


As Fleet Commander in the Xenon Space Academy, you must protect your home planet RADON from the AARDS, PARATRONS, and the ZORGON Battle Star! If you want a 100% machine code multi stage arcade game with the smoothest, meanest action around, this is the one for you – you will not buy a better arcade style game.
For the 48K ORIC-1 ONLY
Only £8.50 inc.

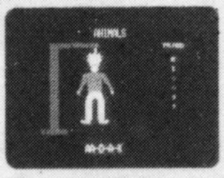


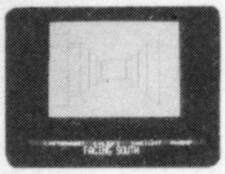




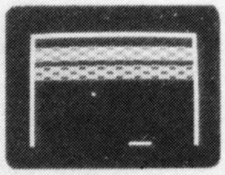


Two top educational games proven as best sellers on the BBC Micro put together here on one cassette for twice the fun on the ORIC-1 Micro for the 48K ORIC-1.
Only £7.50 inc.





Two classic games reproduced here for your ORIC-1 Micro to test your mental agility and reflexes to the full.
For the 48K ORIC-1 ONLY
Only £7.50 inc.



ALL PRICES FULL INCLUSIVE OF VAT AND P&P – NO MORE TO PAY

All advertised software is in stock NOW and will be despatched within 48 hours of receipt of order.

IJK Software Limited

☎ **24 HOUR ANSAFONE**

9 King Street, Blackpool, Lancs.
Telephone (0253) 21555

ORIC OWNER

Issue 2 June/July

Features

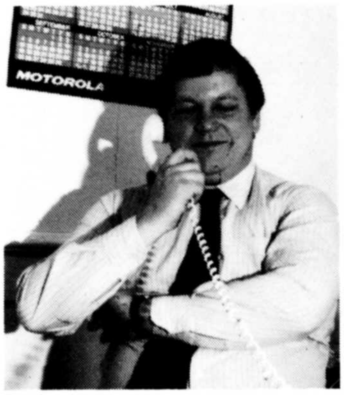
Regulars

News

Software

Oric Owner Magazine
Editor: *Paul B. Kaufman*
Graphics & Artwork Manager: *Carolyn Groeneveld*
Administration: *Cathie Burrell*
Technical Consultant: *Dr Paul Johnson*
Printers: Heffers Printers Ltd, Kings Hedges Road, Cambridge.
Photographic Equipment: Minolta

*Oric Owner is published at bi-monthly intervals by Tansoft Ltd, Reg. No. 1632070
3 Club Mews, Ely, Cambs.
CB7 4NW
Advertising Rates are available on request.*

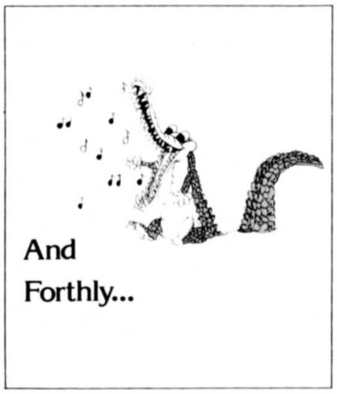


An Interview with Peter Harding

Peter Harding gives us his views on Oric marketing strategy.

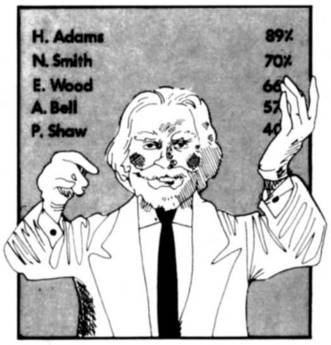
Eliza

Our version of this famous psycho-analysis program



And Forthly

Make your Oric sing with all three voices

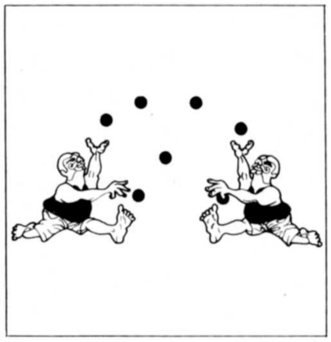


Marks

A useful routine that simplifies classroom marking

Captain Tanex

More incredible adventures of our intrepid hero



Logic Analyser

Design your own logic circuitry

Editor's Comment

It looks like Oric Products International have managed to sort out the delay problems with the Oric 1, 48K models can be seen almost anywhere, although 16K's are still a bit thin on the ground. This means that software and add-on manufacturers can begin production of all the goodies they have been telling us about, which, of course, is all good news for us.

One of the signs of a successful computer is the number of accessories that are available for it. The expandability of the Oric should allow a vast number of peripherals to be designed. In the near future we should be seeing joy-sticks, light pens, I/O cards, memory expansion modules and perhaps even a speech module.

One of the main criticisms of the Oric manual has been the lack of specific details on system variables and machine code entry points. There are a number of reasons for this. Due to copyright and licence agreements Oric products are only permitted to give out minimal information on the inside workings of Basic.

Another reason is that as minor faults are corrected in the Basic so entry points may move up or down in memory. This would mean that any program written to use these Rom routines may not function correctly on a different machine. Obviously once things have settled down in the Rom then some useful information can be released. The final reason for the non-appearance of technical data is simply that at the time of writing the Oric manual much of this information simply wasn't available to the authors.

It is hoped that through this magazine we can become a clearing house for all the technical information on the Oric and its peripherals. We are very interested to hear what discoveries you make and, of course, will pay for any articles or programs published.

So keep them flowing in!

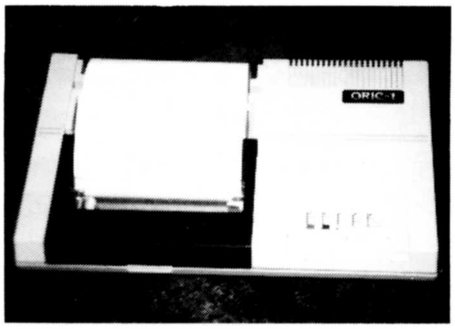
News Brief

Oric Catch up

Oric Products International have now cleared their mail order backlog. Managing Director, Barry Muncaster, said 'It is a great relief to all of us to be able to say that we currently hold no order that has been with us longer than our quoted 28 days. Oric will be phasing out the mail order side of their business over the next few weeks and allowing their dealers to take a greater share of the market.'

Colour Printer Arrives

The first official add-on for the Oric has been released. This is the 4-colour printer which has been expected for some time. The printer uses standard 4½" paper and is switchable between 40 and 80 characters per line. The writing mechanism is a clever arrangement of 4 miniature ball-point pens. These are selectable by software and give red, green, blue and black inks. There are 15 programmable character sizes and 4 different drawing angles. The printer doubles as a penplotter and allows the drawing of complex diagrams and pictures. The printer contains an integral power supply and a printer lead is included. The price is £169.95 inc. vat and £5.95 postage and packing. A small quantity is available for mail order from Tangerine Computer Systems at 3 Club Mews, Ely, Cambs, CB7 4NW



16Ks begin mail order

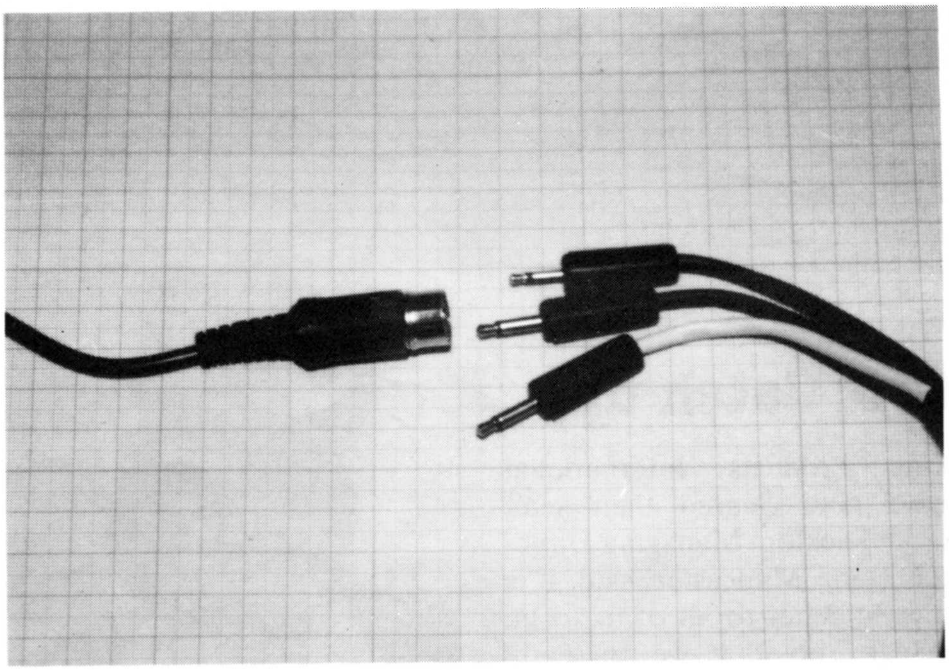
In another statement from Barry Muncaster we were told that 16K Orics are now in full production and will be going to mail order customers first. The reason for the delay in 16K production has been that out of every hundred Orics ordered, eighty of them were for 48Ks, and so it made sense to clear the rapidly increasing backlog. Total Orics ordered for delivery in 1983 now stand at over 350,000 machines.

Apparently production on the 16K was also delayed due to some minor technical problems: 'The 16K Oric is exactly the same design as the 48K Oric. We would have had no problems if the specification of a particular chip had not altered just prior to manufacture. However it did, which resulted in us having to completely change the 16K circuit board. This has meant a 12-week delay in production. Fortunately we are now over the technical problems and in full production of 16Ks' said Barry Muncaster.

News Brief

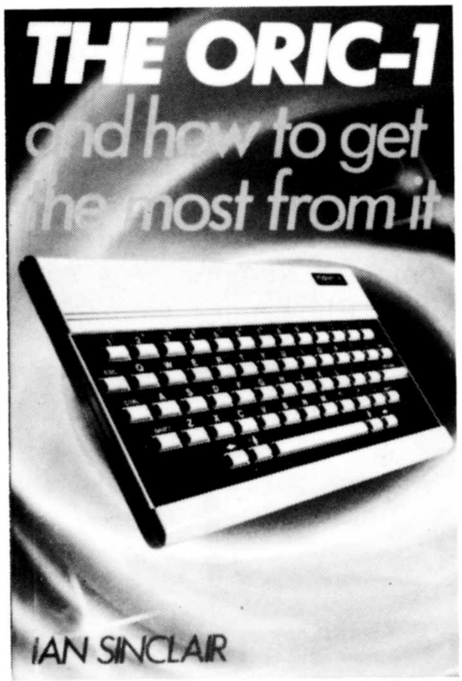
Lead us not Astray!

A range of cassette leads for the Oric has been released by Clares of Winsford in Cheshire. They are priced at £2.50 and to help the novice who is not sure which lead he requires, a sample of each type is included in a pack for £5. The balance is returned when the customer has made his choice. The leads should fit most cassette recorders and include jacks for remote control. More information from Clares, 222, Townfields Road, Winsford, Cheshire CW7 4AX. Tel: (06065) 51374.



New Book from Granada

The first book to appear for the Oric is 'The Oric 1 – And how to get the most from it' by that well known author Ian Sinclair. The book is aimed at the absolute beginner and is designed to give a better understanding of the functioning of the Oric. The user is taken right from the beginning of setting up the Oric and connecting cassette recorders right through to array and file handling and user defined graphics. Although the book doesn't break much ground as far as the experienced user goes, it would be valuable for the novice who wishes to get a full understanding of his machine. The book is available from most sellers of Granada Publications and also by mail order from Tansoft.



Tansoft Range Extended

Tansoft have now added four new titles to their range of cassette software. Oricmon is a complete machine-code monitor including mnemonic assembler/dissassembler block move and verify. The package costs £15.00 inc. vat. and 90p postage. House of Death is a terrifying follow up to Zodiac and takes place in a haunted house. House of Death is £9.99 + 40p postage. Multi-games Pack 2 contains Hangman, Moonlander, Substrike, Roadrace and Link-4D. Pack 2 costs £6.90 inc. vat + 40p postage. Top of the range is Oric-Munch which is a high-speed and challenging version of that well known arcade game. Featuring a variety of ghosts and fruit and written entirely in machine code. Oric-Munch is £7.95 + 40p postage.

Software Scan

As expected, software has started to trickle through for the Oric, albeit mainly in Basic and almost exclusively for the 48K model. The reasons for the delay in software coming on to the market is obvious when you think about it. To start off with nobody had any Orics to actually write software with. Once software houses received their machine there was then the difficulty of understanding how to use it with the early thirty page manual. It takes about a month to really familiarise yourself with a computer before you can safely begin writing software for it.

The first tapes I received for the Oric were from C.C.S. in London.

They were two business simulation type games written in Basic and converted from their original machine, the Spectrum.

AIRLINE

Airline is a game that allows you to take the hot seat in running an airline. You have to buy or lease your airplanes, hire pilots and take on maintenance crews. You also have to worry about fuel costs and charter rates. A nice touch is a teleprinter display which types out messages from the tax office or the House of Lords. The aim of the simulation is to raise enough funds for your company, L-Air, to take over British Airways.

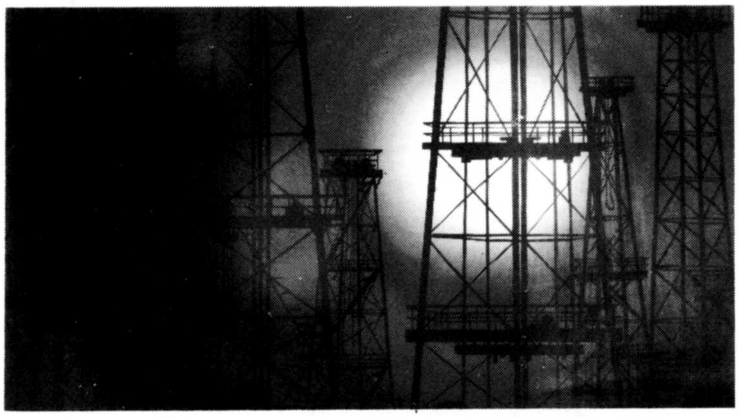
There are several charts displayed of charter rates and payloads which look impressive but I found that not enough information was supplied to really make use of them.

DALLAS

Dallas is a clever simulation which allows you to take the role of 'Cliff Parnes' running an oil company which is competing with the 'Euing' oil company. You have to make the decisions involved in running the company with the aim of taking over Euing Oil.

A colour map is displayed and you have various options of prospecting using seismic surveys, drilling test wells, bidding for potential sites and setting up production facilities.

Software Scan



Various symbols are shown on the map which represent oil rigs, concessions, pipelines and production facilities.

To complicate matters you will be up against blow-outs, tornadoes and competition by the Euings. You will have to borrow money to buy equipment and have to cope with sudden tax demands. I found this game a real challenge and even my own experience in the oil business did not stop me being thrashed time after time.

Of the two programs, Dallas was probably the better but, to be fair, I should say that I only had preliminary versions of each. Airline and Dallas are available for £7.95 from C.C.S., 14 Langton Way, London SE3 7TL and also from most Oric stockists as Oric themselves distribute these games.

New from Tansoft is Zodiac, an enormous adventure game written almost entirely in Basic, although the save game facility is in machine code. The program is just over 36K in size and takes 29 minutes(!) to load at slow speed. It is a text only adventure which also uses sound and music.

The aim of the adventure is to find the twelve signs of the Zodiac which are hidden in various strange locations in a small town. Once these have been found you use them to find treasure. Although the first few signs are fairly easy to discover you soon run into difficulties. There is a ram, which I'm sure represents Aries, which absolutely refuses to let you take it with you. There is also a radio that won't shut up. Zodiac is a parallel adventure in that you can be performing several tasks at once and aren't confined to doing things in a fixed order.

A nice touch is the ability to string commands together as one line, such as 'GO SOUTH AND OPEN DOOR THEN LOOK AROUND'. Two minor criticisms are that every time you go into a different location you get a little tune playing, this can become rather irritating after a little while. The other criticism is that because the program is written in Basic it sometimes is a little slow. The program is full of witticisms and silly messages, for instance if you get killed, it says 'Oh Dear, I've come over all dead!'.

Zodiac is available for £9.99 + 40p postage and packing from: Tansoft, 3, Club Mews, Ely, Cambs and is also distributed by Oric. Please note that this is not the same as the Zodiac sold by Micro-Marketing.

Finally, we have received from Durell Software their Lunar/Asteroids tape. Both of these are written in Basic and are really aimed at the younger Oric Owner. Lunar Lander shows the landscape of the moon with your lunar module in the middle. The module takes off from the moon, you then have to guide it down, controlling the thrust of your rockets. It is very difficult to land and quite often you will run out of fuel.

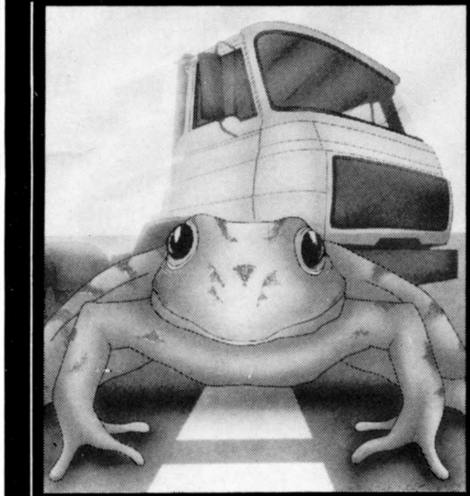
Asteroids is a very much cut down version of the arcade game. Your spaceship is in the middle of the screen with hordes of asteroids advancing towards you. You can control the left and right movements of your ship to avoid them.

Each program is supplied in two forms. A 'runable' version and an expanded version with detailed REMark statements to allow you to see how the program was written.

Lunar Lander/Asteroids is available for £6.95 from Durell Software, Higher Combe, Combe Florey, Taunton, Somerset.

Available Now From P.S.S.

REAL SOFTWARE FOR YOUR ORIC 1.



HOPPER

ALL MACHINE CODE VERSION OF "FROGGER" FOR YOUR ORIC 1.
FEATURES:– VARIABLE SPEED TRAFFIC, 3 LANE HIGHWAY, SNAKES, TURTLES, DIVING TURTLES, FLEAS, LOGS, DECREASING TIME LIMIT, EVER INCREASING GAME SPEED, HI SCORE TABLE, FULL COLOUR, AMAZING GRAPHICS, SUPERB SOUND. YOU HAVEN'T PLAYED "FROGGER" UNTIL YOU'VE PLAYED "HOPPER".

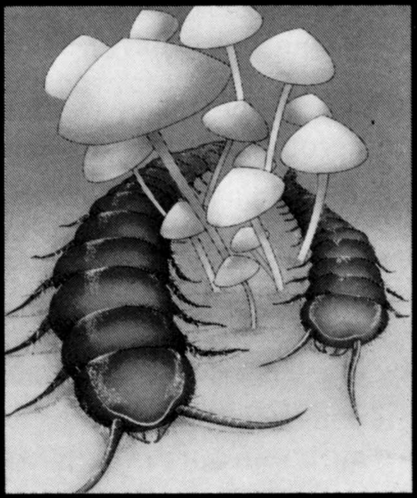
6·95

AVAILABLE SOON:–

THE ULTRA

THIS IS THE GAME FOR ALL YOU ARCADE WIZARDS OUT THERE.. ZAPPING ALIENS WAS EASY UNTIL "THE ULTRA"... THIS GAME IS TOTALLY WRITTEN IN MACHINE CODE TO MAKE IT THE FASTEST, TRICKIEST ARCADE GAME OF THEM ALL.
16 DIFFERENT SCREENS COLOUR, SOUND AND HI RES. GRAPHICS

PSS



CENTIPEDE

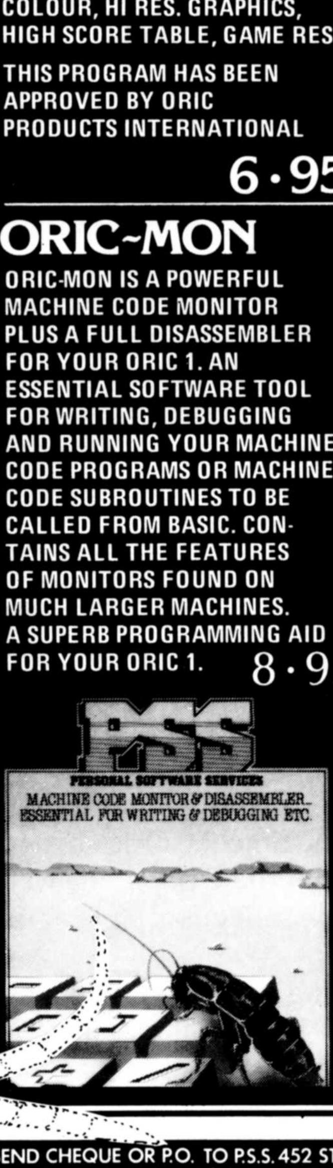
AN ALL MACHINE CODE VERSION OF THE ARCADE GAME.
FEATURES:– MUSHROOMS, SPIDERS, FLEAS, MUSHROOM LAYING BEETLES, SOUND, COLOUR, HI RES. GRAPHICS, HIGH SCORE TABLE, GAME RESET

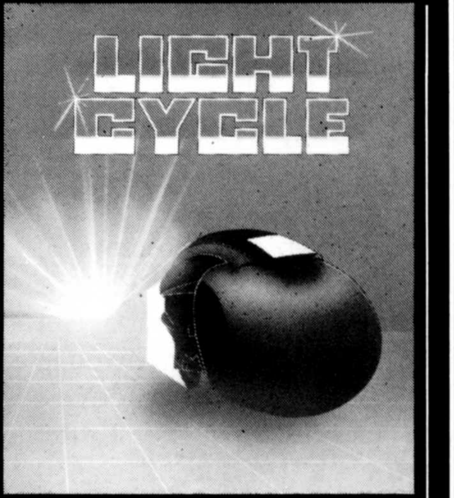
THIS PROGRAM HAS BEEN APPROVED BY ORIC PRODUCTS INTERNATIONAL

6·95

ORIC-MON

ORIC-MON IS A POWERFUL MACHINE CODE MONITOR PLUS A FULL DISASSEMBLER FOR YOUR ORIC 1. AN ESSENTIAL SOFTWARE TOOL FOR WRITING, DEBUGGING AND RUNNING YOUR MACHINE CODE PROGRAMS OR MACHINE CODE SUBROUTINES TO BE CALLED FROM BASIC. CONTAINS ALL THE FEATURES OF MONITORS FOUND ON MUCH LARGER MACHINES.
A SUPERB PROGRAMMING AID FOR YOUR ORIC 1. 8·95

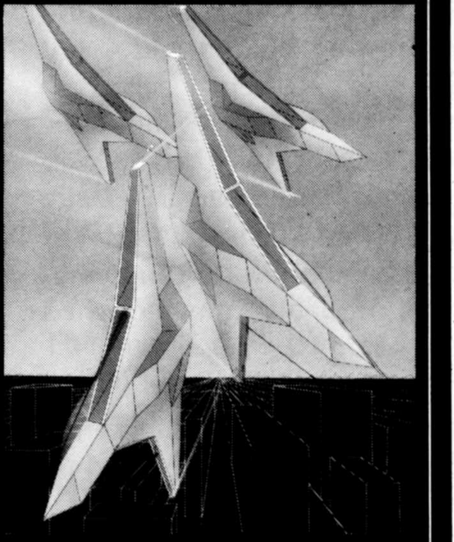

PSS
PERSONAL SOFTWARE SERVICES
MACHINE CODE MONITOR & DISASSEMBLER...
ESSENTIAL FOR WRITING & DEBUGGING ETC.



LIGHT CYCLE

YOU HAVE BEEN CHOSEN BY THE M.C.P. TO RACE ON THE INFAMOUS GRID. CAN YOU SURVIVE? 100% MACHINE CODE. 1 OR 2 PLAYERS, RACE THE COMPUTER OR ANOTHER PLAYER... SO FAST YOU WON'T BELIEVE IT!!

AVAILABLE SOON 6·95



INVADERS

QUITE SIMPLY THE ULTIMATE VERSION OF "SPACE INVADERS" AVAILABLE TODAY.
FEATURES:– ALL MACHINE CODE, SMOOTH ACTION LASER BASE, "LIVE" ALIENS, CRUMBLING WALLS, EXPLOSIVE SOUND EFFECTS, FULL COLOUR, HI RES. GRAPHICS, INCREASING GAME SPEED, BONUS POINTS, MOTHER SHIPS AND MORE....

6·95

SEND CHEQUE OR P.O. TO P.S.S. 452 STONEY STANTON RD. COVENTRY CV6 5DG. FOR INSTANT CREDIT CARD SALES TEL (0203)667556
TRADE ENQUIRIES CONTACT JOHN FLETCHER, COVENTRY (0203) 667556

An Interview with Peter Harding

How did you become involved with Oric Products International?

I have known the directors of Tangerine Computer Systems personally for a number of years. We have been involved in other projects and products – one of which is the Tantel Adaptor – a product designed by Tangerine – marketed by Tandata Marketing Ltd – then my company.

The product and my company were sold to a large public company and the product (the Tantel) is still market leader in the viewdata field. Whilst running Tandata, the Oric appeared on the drawing boards and I felt privileged at being asked to become head of sales and marketing – a position I had no hesitation in accepting especially when I realised the potential of such a good product. Tangerine, as you probably know designed the Oric.

How do you set about marketing a product which clashes head-on with the likes of Sinclair – Commodores – Acorn, Etc?

The most difficult aspect of marketing a product such as Oric is to get your pricing right. Factors such as cost of product, the position you want your product to occupy in the range of competitive micros, rate of product manufacturing output, dealer margins and quality of product, all have to be taken carefully into account. Add to this marketing trends and you have a difficult task to prepare a meaningful business plan to take you into '84 and '85.

My job has been considerably easier for the following reasons:

1. The Oric is currently the most powerful micro on the market for its price.

2. The Oric offers more in terms of real features than any of its rivals.

3. The overall design and looks of the Oric are superb and therefore appeal to the potential customer.

4. The ULA (Uncommitted Logic Array) incorporated in the Oric enables the price to the customer to be considerably less than in computers which use generally available chips and processors. The Oric ULA takes the place of 80 chips which would normally be needed to do the job it does.

What are the planned manufacturing and sales figures for the Oric during the coming months. Will the Oric be readily available?

By the time the Oric Owner goes to print the Oric will be well into mass production.

During February, we made and sold 25,000 units and May's

figure was 32,000 units. All of our mail order sales will be fulfilled by the end of April and the Oric will be available readily in W. H. SMITHS, DIXONS, LASKYS, MENZIES, SPECTRUM (outlets), COMPUTERS FOR ALL (outlets), GREENS (DEBENHAMS), MICRO 'C' and hundreds of good independent computer dealer outlets.

Why have Oric stopped selling product via mail order?

Quite simply, numbers. To mail order the Oric is a vast project in view of the response the Oric has had in the market place. I also believe that in supporting our valued dealers and distributors – it is better to support them rather than be in mail order competition with them. A conflict of interests in the early stages of Oric Products International life would not be in the interests of us or our dealers.

I also believe that a high street dealer is in a better position to support a customer – who, in general, needs help with both the CPU and the software and peripheral products he will invariably want to purchase.

What plans do Oric have for peripheral products and software to support the Oric?

A number of Software houses have been commissioned to write software which is even now, becoming readily available via various outlets. Oric is also launching its own comprehensive range of games, business and educational software.

A new 4 colour plain paper printer is planned to be available in June/July. It is a beautifully designed quality product and will cost approx. the same price as the 48K unit. The printer will be provided with the printer lead and will plug into the centronics port on the Oric.

Our Micro floppy discs are still being finalised and should be in production for sale during September/October '83. We have opted for the 3 inch format as being the most sensible size and the best value for money. A price has yet to be finalised.

The long awaited Modem should also be available in July and will allow users the facilities of Prestel – Micronet 800 – electronic mail – teleshopping and telesoftware.

Our overall policy with the hardware is to produce a comprehensive range of quality built, value for money products. We will not, under any circumstances, build cheap and nasty products which has a short life cycle.

Is this last statement a direct jibe at Sinclairs products which as you know sell in vast numbers?

Not really, who can criticise Sinclairs marketing and he has a product which – as you say – sells very well indeed. All of the products that Oric will produce, however, will be of a much higher standard and quality of those currently produced by Sinclair – which will be, of course, reflected in the price of our products. I reiterate, that Oric will produce high quality CPU's, Printer, Discs and add-ons at prices that in general will be far better value for money than *any* currently available.

If what you say is true, how then can you possibly sell the Oric at the same price or less than the Sinclair product?

As I said earlier, preparing a marketing plan is very difficult and one of the factors I mentioned was market trends.

Bearing this in mind, it must be obvious that the Oric, because it is a far better product both in specification and quality of build than our rivals unit – must cost more to manufacture, and of course it does.

The Oric therefore was pitched right at the SPECTRUM knowing that the price of this product will be reduced possibly quite drastically.

The number of computers we are building will of course affect our competitors sales and where they can not compete they must reduce their prices to maintain market share.

Will the price of the Oric be reduced, therefore?

No, not in the foreseeable future, our unit represents great value for money and that is why our customers will part with their hard earned cash.

The press have criticised Oric for making the same mistake as your well known rivals in launching your product too early and not being able to supply demand. Have you any comments to make?

It is true that early demand outstripped supply and we did let down a large number of mail order customers badly. We are of course very sorry this happened.

However, the real facts are that our first ads were issued in November '82 magazines – we did ship product in December and have now caught up with all the backlogs of orders. We will be supplying a range of peripherals all within 8 months of our launch – this is in fact a much better performance than any of our rivals. Whilst on the subject of the press, I would like to make the following observations:

1. A majority of computer magazines do not have their own technical staff to do benchtests. This results in outside contractors being asked to review machines who are often either incompetent or biased towards their own systems. The reviews therefore of the Oric although in the main very favourable are often totally inaccurate and do not extol the full virtues of the Oric, e.g. only two colours in Hi–Res mode.

2. The general standard of reporting is diabolical and there is more inaccurate information printed than actual facts – which of course does nothing for the British computer industry.

3. There are obviously too many magazines duplicating information and using unethical methods to obtain information. For example, one magazine wined and dined one of our sixteen year old programmers – without the consent of the management – and printed a large article which included information about our product and this lad's ambitions. This damaging article went to press without Oric's approval.

Enough about the computer press – some of the magazines are very good.

How many Orics do you expect to sell in 1983 and how do you see the computer market developing?

Oric expect to sell approximately 400,000 units to February '84 to be sold in the UK and Europe. This figure does not include the considerable product we expect to be sold in JAPAN, S.E. ASIA, AUSTRALIA/ NEW ZEALAND and the USA.

The market place for computers is still rapidly expanding and the pundits estimate that 1.2 million CPU's will be sold in the UK during 1983.

Currently something like 80% of all units currently sold are with games applications – this market will eventually level off. The computer market will then expand only with dedicated applications. The Oric hardware has of course been designed around taking advantage of these applications. I have touched on these applications earlier but to reiterate, they are:

1. *Telesoftware* – In the not too distant future the majority of all software will be transmitted via a telephone line at a very cheap cost, negating the need for costly cassettes and offering the user access to hundreds of programs.

2. *Home Banking* – Again prototype systems are up and running where with the use of a computer and modem an individual can access his bank account, add and delete standing orders, pay bills and, therefore, negate the need for regular trips to the bank.

3. *Teleshopping* – An obvious application is already available on Prestel.

4. *Electronic Mail* – Again already in use via a computer with a modem utilising the Prestel Mailbox system.

These are the real applications that will put a computer into every home.

Your comments have given us a useful insight into your marketing plans for the Oric. Thank you very much Mr Harding.

3D Function Plot: A number of people have mentioned that there are two HIRES commands and an endless loop in this program. They will not cause the program to run incorrectly but can be altered or removed if required. They are in fact what was left of a larger program which was stripped down to this final printed version.

Oric Products International have asked us to mention some of the problem areas in the Oric Rom and ways of getting round them.

1. TAB function: This only appears to work correctly after position 13. Use SPC(x) function in its place.

2. STR$ function: this inserts illegal control character at beginning of string equivalents of numbers (positive only). Strip off first character of string, e.g. A$=RIGHT$(A$,LEN(A$)-1).

3. FILL command: Does not correctly update graphics cursor position after execution. Use CURSET after FILL command.

4. String or Array corruption. Caused by incorrect setting of HIMEM on power-up. Set HIMEM #97FF to cure.

5. POKE command: Will not accept hex values after comma. Convert to decimal before use.

6. CIRCLEs not circular: Not really a fault but caused by pixels themselves not being square. Could be compensated for but would cause horizontal and vertical radius lengths (in pixels) to become unequal. This would make tasks such as drawing a clockface very difficult.

7. Printer squiggles: Printer port randomly outputs squiggle characters ($7F) onto printer. Caused by Keyboard scanning interrupts interacting with printer port. To fix, turn off interrupts before using printer:

   CALL#E6CA turns off interrupts.

   CALL#E804 turns on interrupts.

Warning: Do not use WAIT command while interrupts are off as it functions by counting interrupts. If interrupts are off then the WAIT command will never finish. As interrupts are normally generated every twentieth of a second turning them off will make your program run about 20% faster!

8. Program loads OK but when listed screen is filled with 'U's. Caused by linkage between each line being corrupted and pointing into empty memory. Can be fixed by entering a dummy line at the beginning of the program, e.g. 1 PRINT, and then immediately deleting it. This will force Basic to re-link all lines.

If you have found any more bugs or problem areas and have managed to get round them please drop us a line.

And Forthly...

Tunesmith-2

By now most 48K owners should have received their Forth package. Apart from the Forth processor itself you are also supplied with the editor, extensions for sound and graphics and the assembler. Also supplied is a demonstration program called Tunesmith. This allows the creation of simple three-part harmony for the playing of music. Although reasonably impressive as it stands, I have now improved the program to give a wider octave range, dotted notes, and better separation between notes.

Some explanation of the program would be useful to understand how tunes are created. Tempo is defined to hold the playing speed of the tune. MASK is used to select which tone channels are to be enabled at any time OCT and NTE are temporary variables used to hold the current octave and note for the music command. We then begin the definitions to allow notes to be played. N1, N2 and N3 select channels 1, 2 and 3 respectively. X1, X2 and X3 turn off each channel. WAIT and TRIGGER control the length of notes and the gap between them. To allow for quavers and crotchets etc, we have a set of words, WH, HF, QR, EI, SX which mean whole note, half, quarter, eighth and sixteenth. On screen 3, lines 13 and 14 are two further durations 'QR' and 'HF', which stand for dotted quarter and dotted half notes.

Screens 2 and 3 define all the notes between octaves 1 and 5. These could be extended to cover octaves 6 and 7 but most musical compositions are unlikely to need them.

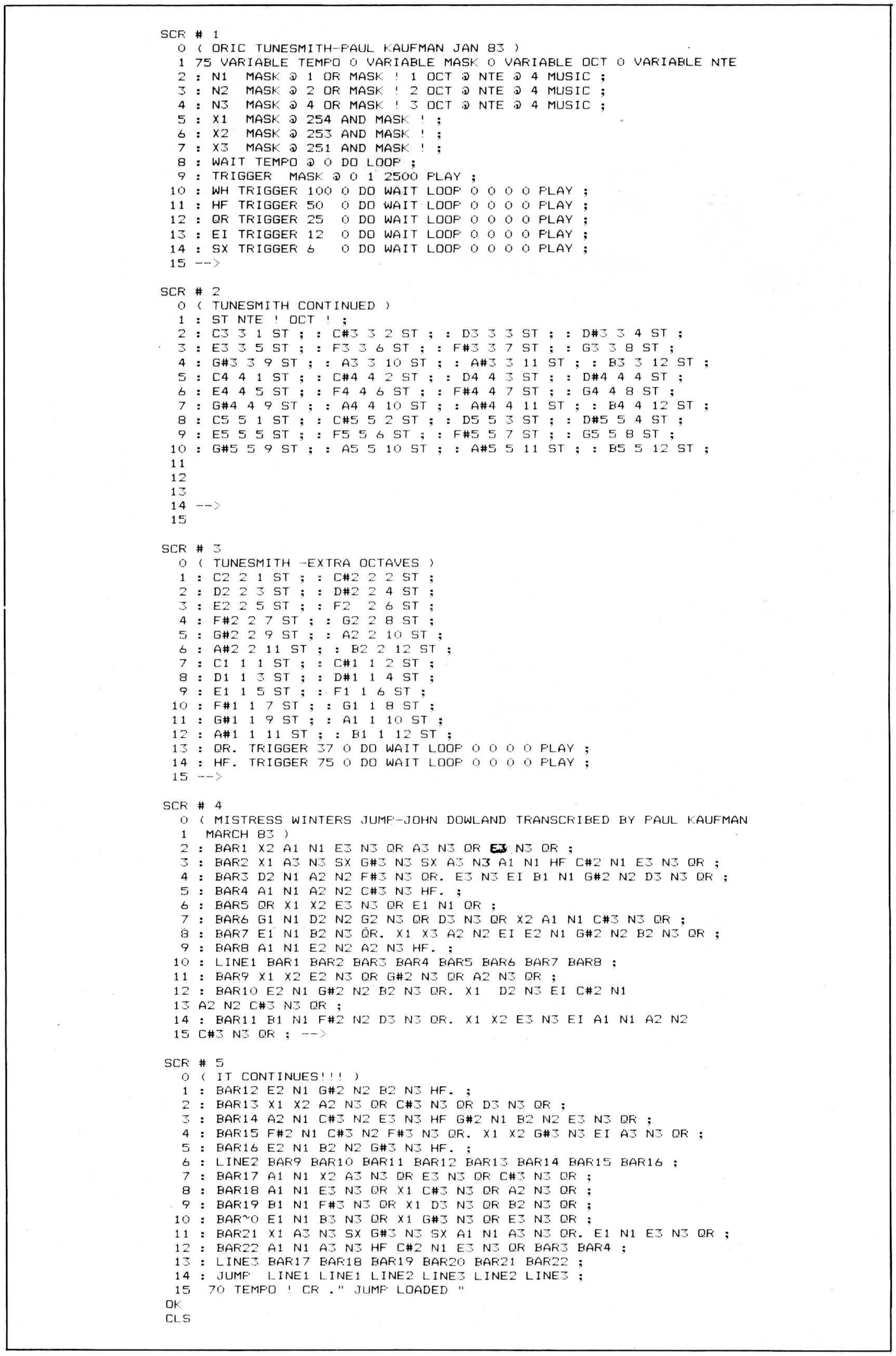
Screens 4 and 5 contain the piece of music to be played, divided into lines then subdivided into bars. To see how a tune is put together lets look at BAR1 which is on screen 4, line 2, X2 turns off channel 2. A1 N1 tells the computer to play the note of A, from octave 1 into channel 1 and at the same time play an E from octave 3 through channel 3.

QR tells the machine to play the preceding notes for the duration of one quarter (quaver) period. This process is repeated for the rest of the bar, and the rest of the piece. Note that a Rest is programmed simply by turning off the appropriate channel.

The program produces sustained organ-like tones. If you wish to have a more natural piano tune you should amend lines 2, 3 and 4 of screen 1 so that the volume parameter for the MUSIC command is set to 0 instead of 4.

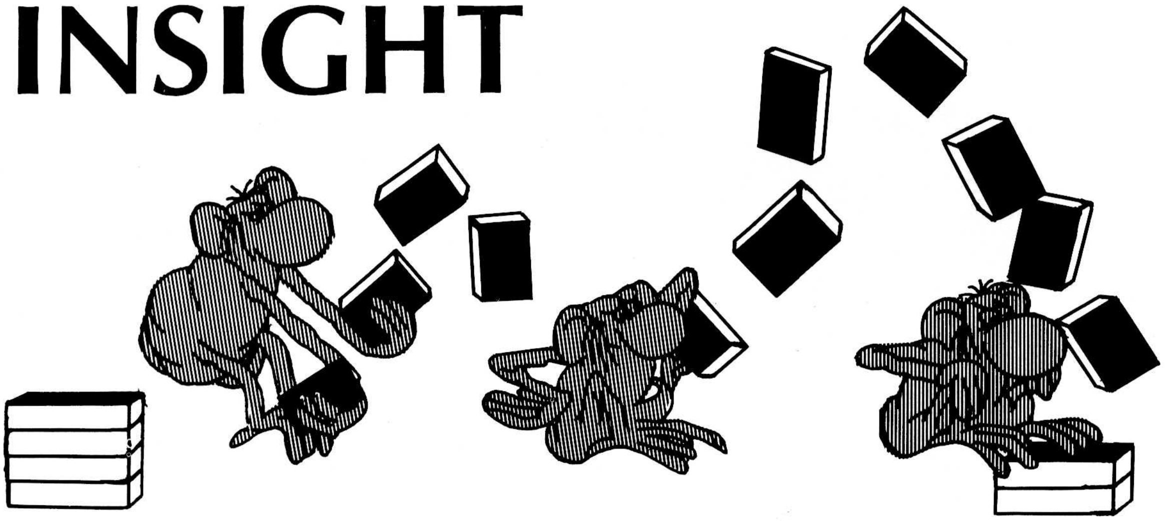
This type of program has a great deal of potential and many additional features could be included. One such feature would be a percussion effect using the noise channel. I look forward to receiving your modifications or additions to the program and also any music that you compose for it.

Program Listing



```
SCR # 1
 ( ORIC TUNESMITH-PAUL KAUFMAN JAN 83 )
 75 VARIABLE TEMPO 0 VARIABLE MASK 0 VARIABLE OCT 0 VARIABLE NTE
 : N1  MASK @ 1 OR MASK ! 1 OCT @ NTE @ 4 MUSIC ;
 : N2  MASK @ 2 OR MASK ! 2 OCT @ NTE @ 4 MUSIC ;
 : N3  MASK @ 4 OR MASK ! 3 OCT @ NTE @ 4 MUSIC ;
 : X1  MASK @ 254 AND MASK ! ;
 : X2  MASK @ 253 AND MASK ! ;
 : X3  MASK @ 251 AND MASK ! ;
 : WAIT TEMPO @ 0 DO LOOP ;
 : TRIGGER  MASK @ 0 1 2500 PLAY ;
 : WH TRIGGER 100 0 DO WAIT LOOP 0 0 0 0 PLAY ;
 : HF TRIGGER 50  0 DO WAIT LOOP 0 0 0 0 PLAY ;
 : QR TRIGGER 25  0 DO WAIT LOOP 0 0 0 0 PLAY ;
 : EI TRIGGER 12  0 DO WAIT LOOP 0 0 0 0 PLAY ;
 : SX TRIGGER 6   0 DO WAIT LOOP 0 0 0 0 PLAY ;
 -->

SCR # 2
 ( TUNESMITH CONTINUED )
 : ST NTE ! OCT ! ;
 : C3 3 1 ST ; : C#3 3 2 ST ; : D3 3 3 ST ; : D#3 3 4 ST ;
 : E3 3 5 ST ; : F3 3 6 ST ; : F#3 3 7 ST ; : G3 3 8 ST ;
 : G#3 3 9 ST ; : A3 3 10 ST ; : A#3 3 11 ST ; : B3 3 12 ST ;
 : C4 4 1 ST ; : C#4 4 2 ST ; : D4 4 3 ST ; : D#4 4 4 ST ;
 : E4 4 5 ST ; : F4 4 6 ST ; : F#4 4 7 ST ; : G4 4 8 ST ;
 : G#4 4 9 ST ; : A4 4 10 ST ; : A#4 4 11 ST ; : B4 4 12 ST ;
 : C5 5 1 ST ; : C#5 5 2 ST ; : D5 5 3 ST ; : D#5 5 4 ST ;
 : E5 5 5 ST ; : F5 5 6 ST ; : F#5 5 7 ST ; : G5 5 8 ST ;
 : G#5 5 9 ST ; : A5 5 10 ST ; : A#5 5 11 ST ; : B5 5 12 ST ;



 -->


SCR # 3
 ( TUNESMITH -EXTRA OCTAVES )
 : C2 2 1 ST ; : C#2 2 2 ST ;
 : D2 2 3 ST ; : D#2 2 4 ST ;
 : E2 2 5 ST ; : F2  2 6 ST ;
 : F#2 2 7 ST ; : G2 2 8 ST ;
 : G#2 2 9 ST ; : A2 2 10 ST ;
 : A#2 2 11 ST ; : B2 2 12 ST ;
 : C1 1 1 ST ; : C#1 1 2 ST ;
 : D1 1 3 ST ; : D#1 1 4 ST ;
 : E1 1 5 ST ; : F1 1 6 ST ;
 : F#1 1 7 ST ; : G1 1 8 ST ;
 : G#1 1 9 ST ; : A1 1 10 ST ;
 : A#1 1 11 ST ; : B1 1 12 ST ;
 : QR. TRIGGER 37 0 DO WAIT LOOP 0 0 0 0 PLAY ;
 : HF. TRIGGER 75 0 DO WAIT LOOP 0 0 0 0 PLAY ;
 -->

SCR # 4
 ( MISTRESS WINTERS JUMP-JOHN DOWLAND TRANSCRIBED BY PAUL KAUFMAN
  MARCH 83 )
 : BAR1 X2 A1 N1 E3 N3 QR A3 N3 QR E3 N3 QR ;
 : BAR2 X1 A3 N3 SX G#3 N3 SX A3 N3 A1 N1 HF C#2 N1 E3 N3 QR ;
 : BAR3 D2 N1 A2 N2 F#3 N3 QR. E3 N3 EI B1 N1 G#2 N2 D3 N3 QR ;
 : BAR4 A1 N1 A2 N2 C#3 N3 HF. ;
 : BAR5 QR X1 X2 E3 N3 QR E1 N1 QR ;
 : BAR6 G1 N1 D2 N2 G2 N3 QR D3 N3 QR X2 A1 N1 C#3 N3 QR ;
 : BAR7 E1 N1 B2 N3 QR. X1 X3 A2 N2 EI E2 N1 G#2 N2 B2 N3 QR ;
 : BAR8 A1 N1 E2 N2 A2 N3 HF. ;
 : LINE1 BAR1 BAR2 BAR3 BAR4 BAR5 BAR6 BAR7 BAR8 ;
 : BAR9 X1 X2 E2 N3 QR G#2 N3 QR A2 N3 QR ;
 : BAR10 E2 N1 G#2 N2 B2 N3 QR. X1  D2 N3 EI C#2 N1
 A2 N2 C#3 N3 QR ;
 : BAR11 B1 N1 F#2 N2 D3 N3 QR. X1 X2 E3 N3 EI A1 N1 A2 N2
 C#3 N3 QR ; -->

SCR # 5
 ( IT CONTINUES!!! )
 : BAR12 E2 N1 G#2 N2 B2 N3 HF. ;
 : BAR13 X1 X2 A2 N3 QR C#3 N3 QR D3 N3 QR ;
 : BAR14 A2 N1 C#3 N2 E3 N3 HF G#2 N1 B2 N2 E3 N3 QR ;
 : BAR15 F#2 N1 C#3 N2 F#3 N3 QR. X1 X2 G#3 N3 EI A3 N3 QR ;
 : BAR16 E2 N1 B2 N2 G#3 N3 HF. ;
 : LINE2 BAR9 BAR10 BAR11 BAR12 BAR13 BAR14 BAR15 BAR16 ;
 : BAR17 A1 N1 X2 A3 N3 QR E3 N3 QR C#3 N3 QR ;
 : BAR18 A1 N1 E3 N3 QR X1 C#3 N3 QR A2 N3 QR ;
 : BAR19 B1 N1 F#3 N3 QR X1 D3 N3 QR B2 N3 QR ;
 : BAR~O E1 N1 B3 N3 QR X1 G#3 N3 QR E3 N3 QR ;
 : BAR21 X1 A3 N3 SX G#3 N3 SX A1 N1 A3 N3 QR. E1 N1 E3 N3 QR ;
 : BAR22 A1 N1 A3 N3 HF C#2 N1 E3 N3 QR BAR3 BAR4 ;
 : LINE3 BAR17 BAR18 BAR19 BAR20 BAR21 BAR22 ;
 : JUMP  LINE1 LINE1 LINE2 LINE3 LINE2 LINE3 ;
  70 TEMPO ! CR ." JUMP LOADED "
OK
CLS
```

ORIC INSIGHT



Paul Kaufman

CASSETTE HANDLING

It is occasionally useful for the programmer to access the cassette sofware of the Oric directly in machine code. Although it is possible to access the 6522 via chip directly to set up cassette routines, such is the complexity of the chip itself, and its external connections, that it makes sense to use the machine code routines that are already included in Oric's Basic. Figure 1, below, shows a block diagram of the Oric and gives an idea of the circuit complexity surrounding the VIA.

Fortunately the cassette routines within Basic are fairly easy to set up and use through machine code, as there is not a machine code monitor generally available yet for the Oric most of the setting up will need to be done through Basic using pokes.

SAVING A FILE ON TAPE

Before we can save our file there are a number of pointers that need to be set up.

$5F	Low byte of start address
$60	High byte of start address
$61	Low byte of end address
$62	High byte of end address
$63	Auto load, 0=non autoload file, 1=autoload file
$64	Basic of machine code, 0=basic, 1=m/c
$67	Speed, 0= fast, 1= slow

If you are saving a machine code program and wish it to autoload (run automatically after loading in) set $63 to 1 (in fact any non-zero value) and $64 to 1.

When the program is reloaded in it will start executing at the address pointed to by $5F and $60.

You should also set up the filename of the program at locations $35 upwards (for the next 17 locations). The file name must be terminated by a null ($100).

Once the about locations are prepared you should then execute the following:

```
JSR  $E6CA; set up via
JSR  $E57B; store file
JSR  $E804; clear via, initialise
     keyboard
```

It should be assumed that all processor registers are corrupted.

LOADING A FILE FROM TAPE

To bring a program in off the cassette is a much simpler process than saving it. Simply set up the speed and filename as above then execute the following jumps:

```
JSR  $E6CA; set up via
JSR  $E4A8; load in file
JSR  $E804; turn off via
```

You will get the searching message while it is waiting for the program followed by loading.

One annoying feature of the above routines is that they will corrupt the pointers that Basic uses and so will not work satisfactorily from within a Basic program.

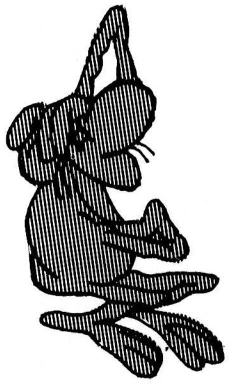
SAVING ARRAYS

The hex listing below implements a store and recall facility which is useable through Basic, once you have loaded in the routine either directly or through the Basic loader also shown below you should set HMEM to #97FF to protect the routine.

To save an array you should first set the speed with POKE#67,0 for fast or POKE#67,1 for slow. Then type Call 1024,xx where xx is your array name. This will store the entire array on tape. Beware of having a numeric array with the same name as a string array, e.g. A$ and A as you may later have loading problems.

To load in an array you should first dimension it in the program and then set the speed up. CALL 1027,xx will read in your array. If you haven't dimensioned it large enough you will get an out of memory error.

In the next issue I hope to be taking a closer look at the 6522 via chip and how its timers can be used.



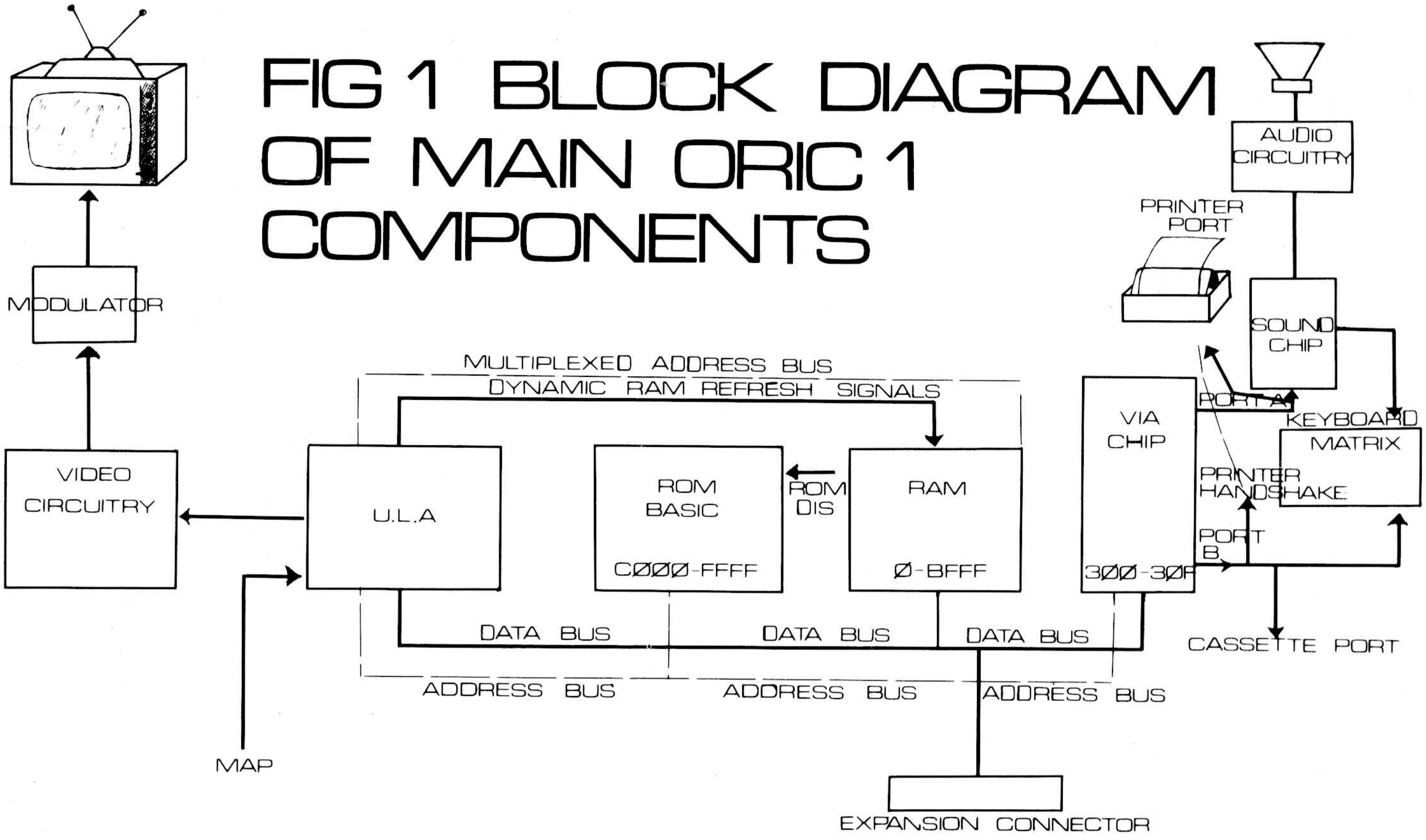


FIG 1 BLOCK DIAGRAM OF MAIN ORIC 1 COMPONENTS

Program Listing

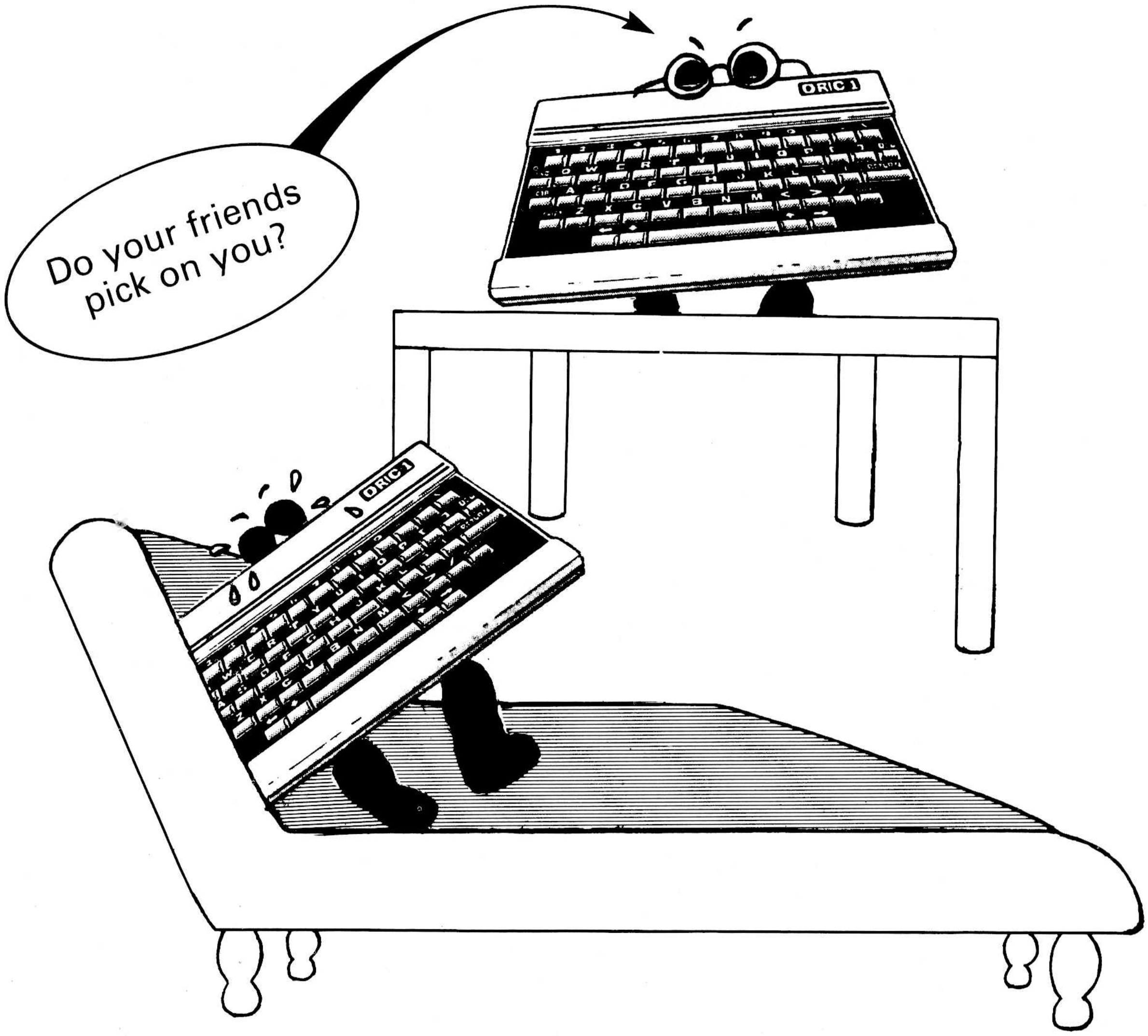
```
20000 REM ARRAY SAVER
20005 A=#B800:READD$
20008 PRINT:PRINT:PRINT:PRINT:PRINT"Please Wait...."
20010 FORI=1TOLEN(D$)STEP2
20020 V=VAL("#"+MID$(D$,I,2)):POKEA,V:A=A+1:NEXT
20030 READD$:IFD$<>"Z"THEN20010
20040 DOKE#400,#064C:DOKE#402,#4CB8:DOKE#404,#B82A:RETURN
20050 DATA233923323255209CB8082063B820BAE6A92520C6E5A53320C6E5A5342
0C6E52083
20060 DATAB820A7E52004E82860209CB8082063B82096E62030E6C925D0F92030E
685332030
20070 DATAE68534A002B1CEC533C8B1CEE534B0062004E84C83C42083B820EBE42
004E82860
20080 DATA20CAE6A940852B20A9B8A900852BA003B1CEAA88B1CEE901B001CA853
386346018
20090 DATAA5CE6533856 1A5CF65348562A004B1CE20F6D1855F84606020E800C92
CF0034CE4
20100 DATACF4CE200A20020E800862785B420E8002086D1B0062004E84CE4CFA20
086288629
20110 DATA20E20090052086D1900BAA20E20090FB2086D1B0F6C924D006A9FF852
8D00CC925
20120 DATAD00FA980852905B485B48A0980AA20E20086B5A69EA59F86CE85CFC5A
1D004E4A0
20130 DATAF01FA000B1CEC8C5B4D006A5B5D1CEF00EC8B1CE1865CEAAC8B1CE65C
F90D73860
20140 DATA2004E8A22A4C85C4
20150 DATAZ
```

```
#400     #4C        #B818    #C6        #B837    #C9
#401     #6         #B819    #E5        #B838    #25
#402     #B8        #B81A    #A5        #B839    #D0
#403     #4C        #B81B    #34        #B83A    #F9
#404     #2A        #B81C    #20        #B83B    #20
#405     #B8        #B81D    #C6        #B83C    #30
                    #B81E    #E5        #B83D    #E6
#B800    #23        #B81F    #20        #B83E    #85
#B801    #39        #B820    #83        #B83F    #33
#B802    #23        #B821    #B8        #B840    #20
#B803    #32        #B822    #20        #B841    #30
#B804    #32        #B823    #A7        #B842    #E6
#B805    #55        #B824    #E5        #B843    #85
#B806    #20        #B825    #20        #B844    #34
#B807    #9C        #B826    #4         #B845    #A0
#B808    #B8        #B827    #E8        #B846    #2
#B809    #8         #B828    #28        #B847    #B1
#B80A    #20        #B829    #60        #B848    #CE
#B80B    #63        #B82A    #20        #B849    #C5
#B80C    #B8        #B82B    #9C        #B84A    #33
#B80D    #20        #B82C    #B8        #B84B    #C8
#B80E    #BA        #B82D    #8         #B84C    #B1
#B80F    #E6        #B82E    #20        #B84D    #CE
#B810    #A9        #B82F    #63        #B84E    #E5
#B811    #25        #B830    #B8        #B84F    #34
#B812    #20        #B831    #20        #B850    #B0
#B813    #C6        #B832    #96        #B851    #6
#B814    #E5        #B833    #E6        #B852    #20
#B815    #A5        #B834    #20        #B853    #4
#B816    #33        #B835    #30
#B817    #20        #B836    #E6
```

Program Listing

```
#B854     #E8
#B855     #4C
#B856     #83
#B857     #C4
#B858     #20
#B859     #83
#B85A     #B8
#B85B     #20
#B85C     #EB
#B85D     #E4
#B85E     #20
#B85F     #4
#B860     #E8
#B861     #28
#B862     #60
#B863     #20
#B864     #CA
#B865     #E6
#B866     #A9
#B867     #40
#B868     #85
#B869     #2B
#B86A     #20
#B86B     #A9
#B86C     #B8
#B86D     #A9
#B86E     #0
#B86F     #85
#B870     #2B
#B871     #A0
#B872     #3
#B873     #B1
#B874     #CE
#B875     #AA
#B876     #88
#B877     #B1
#B878     #CE
#B879     #E9
#B87A     #1
#B87B     #B0
#B87C     #1
#B87D     #CA
#B87E     #85
#B87F     #33
#B880     #86
#B881     #34
#B882     #60
#B883     #18
#B884     #A5
#B885     #CE
#B886     #65
#B887     #33
#B888     #85
#B889     #61
#B88A     #A5
#B88B     #CF
#B88C     #65
#B88D     #34
#B88E     #85
#B88F     #62
#B890     #A0
#B891     #4
#B892     #B1
#B893     #CE
#B894     #20
#B895     #F6
#B896     #D1
#B897     #85
#B898     #5F
#B899     #84
#B89A     #60
#B89B     #60
#B89C     #20

#B89D     #E8
#B89E     #0
#B89F     #C9
#B8A0     #2C
#B8A1     #F0
#B8A2     #3
#B8A3     #4C
#B8A4     #E4
#B8A5     #CF
#B8A6     #4C
#B8A7     #E2
#B8A8     #0
#B8A9     #A2
#B8AA     #0
#B8AB     #20
#B8AC     #E8
#B8AD     #0
#B8AE     #86
#B8AF     #27
#B8B0     #85
#B8B1     #B4
#B8B2     #20
#B8B3     #E8
#B8B4     #0
#B8B5     #20
#B8B6     #86
#B8B7     #D1
#B8B8     #B0
#B8B9     #6
#B8BA     #20
#B8BB     #4
#B8BC     #E8
#B8BD     #4C
#B8BE     #E4
#B8BF     #CF
#B8C0     #A2
#B8C1     #0
#B8C2     #86
#B8C3     #28
#B8C4     #86
#B8C5     #29
#B8C6     #20
#B8C7     #E2
#B8C8     #0
#B8C9     #90
#B8CA     #5
#B8CB     #20
#B8CC     #86
#B8CD     #D1
#B8CE     #90
#B8CF     #B
#B8D0     #AA
#B8D1     #20
#B8D2     #E2
#B8D3     #0
#B8D4     #90
#B8D5     #FB
#B8D6     #20
#B8D7     #86
#B8D8     #D1
#B8D9     #B0
#B8DA     #F6
#B8DB     #C9
#B8DC     #24
#B8DD     #D0
#B8DE     #6
#B8DF     #A9
#B8E0     #FF
#B8E1     #85
#B8E2     #28
#B8E3     #D0
#B8E4     #C
#B8E5     #C9

#B8E6     #25
#B8E7     #D0
#B8E8     #F
#B8E9     #A9
#B8EA     #80
#B8EB     #85
#B8EC     #29
#B8ED     #5
#B8EE     #B4
#B8EF     #85
#B8F0     #B4
#B8F1     #8A
#B8F2     #9
#B8F3     #80
#B8F4     #AA
#B8F5     #20
#B8F6     #E2
#B8F7     #0
#B8F8     #86
#B8F9     #B5
#B8FA     #A6
#B8FB     #9E
#B8FC     #A5
#B8FD     #9F
#B8FE     #86
#B8FF     #CE
#B900     #85
#B901     #CF
#B902     #C5
#B903     #A1
#B904     #D0
#B905     #4
#B906     #E4
#B907     #A0
#B908     #F0
#B909     #1F
#B90A     #A0
#B90B     #0
#B90C     #B1
#B90D     #CE
#B90E     #C8
#B90F     #C5
#B910     #B4
#B911     #D0
#B912     #6
#B913     #A5
#B914     #B5
#B915     #D1
#B916     #CE
#B917     #F0
#B918     #E
#B919     #C8
#B91A     #B1
#B91B     #CE
#B91C     #18
#B91D     #65
#B91E     #CE
#B91F     #AA
#B920     #C8
#B921     #B1
#B922     #CE
#B923     #65
#B924     #CF
#B925     #90
#B926     #D7
#B927     #38
#B928     #60
#B929     #20
#B92A     #4
#B92B     #E8
#B92C     #A2
#B92D     #2A
#B92E     #4C
#B92F     #85
#B930     #C4
```

eliza



This is an Oric version of the well known psychoanalysis program. It is designed to allow you to input sentences in normal everyday English and to formulate suitable replies. It is a simple artificial intelligence program which, for most of the time, can carry out a reasonably coherent conversion although it has a rather limited range of subjects.

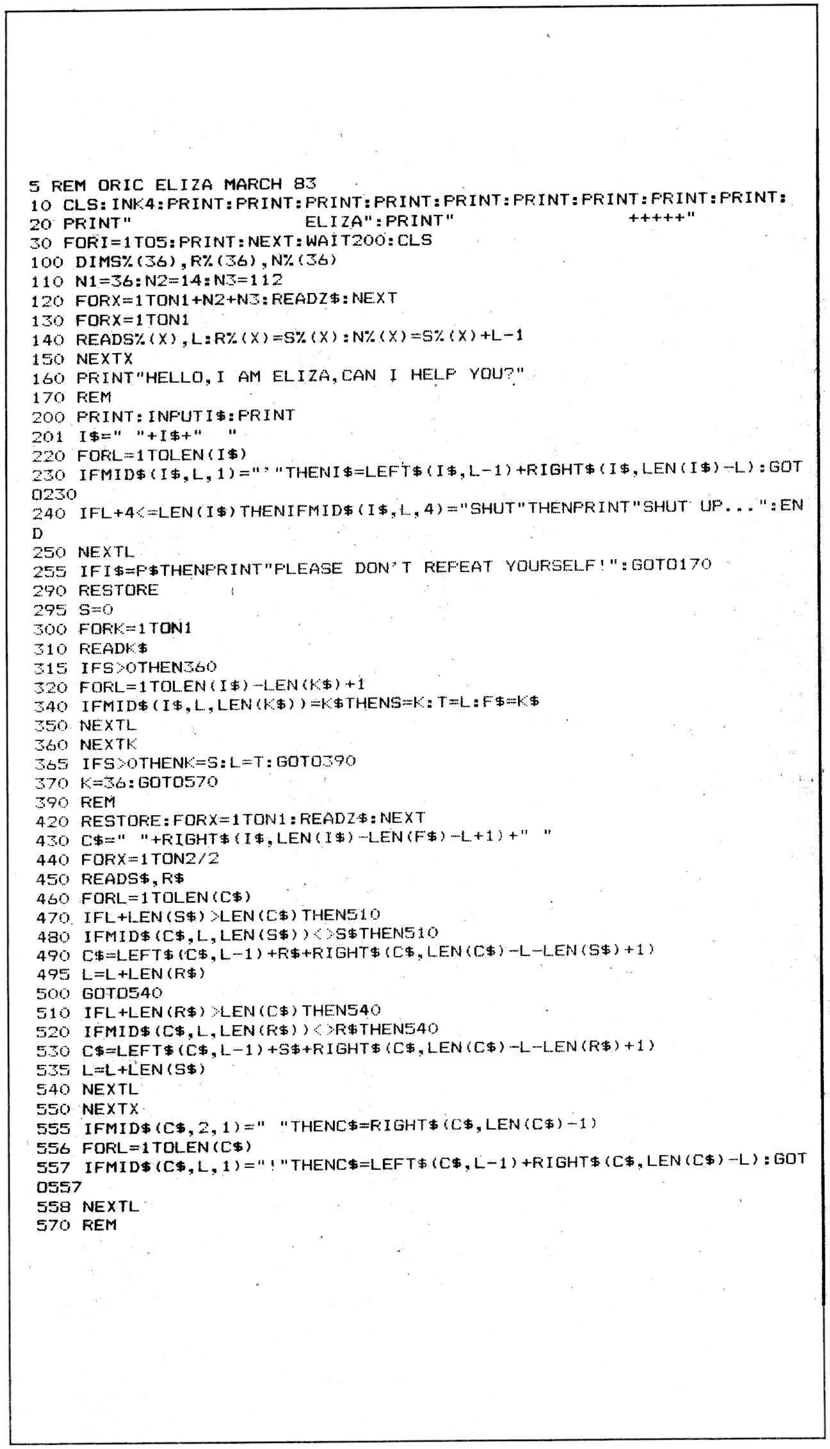
Its favourite subjects are friends, computers, and dreams. As with a real psychiatrist it will ask you lots of questions but never answer any!

Lines 5 to 640 are the main processing part of the program. They take the users input and break it into Keywords which are compared with the data statements for a match.

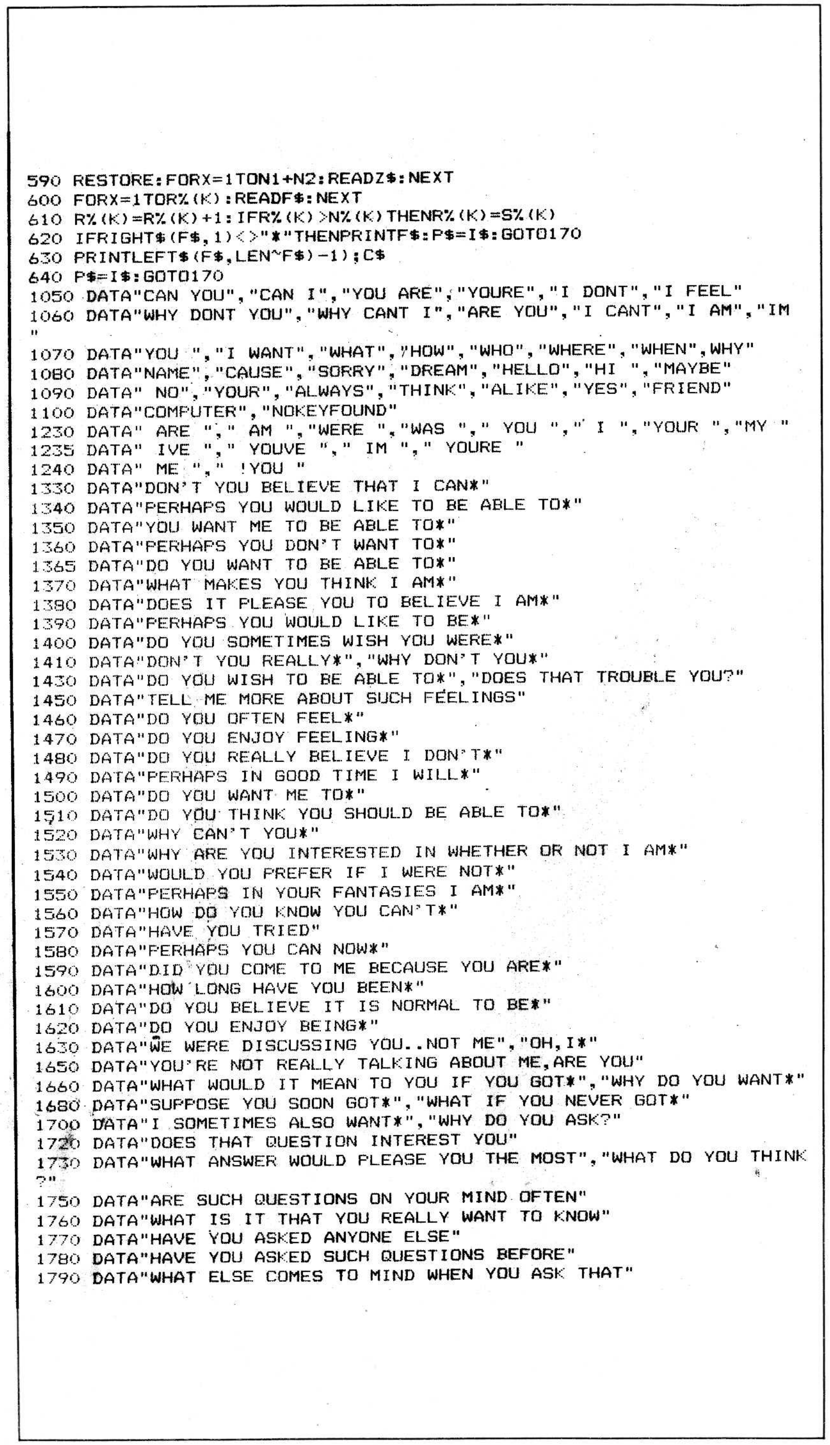
The appropriate answer will be picked from the list of replies and inversions of 'You/Me' 'My/Your' etc. will be selected and placed in the correct part of the reply sentence.

The program could be substantially lengthened to give more replies and more in-depth sentence parsing could be added. To get out of the program enter 'Shut-up'.

Program Listing

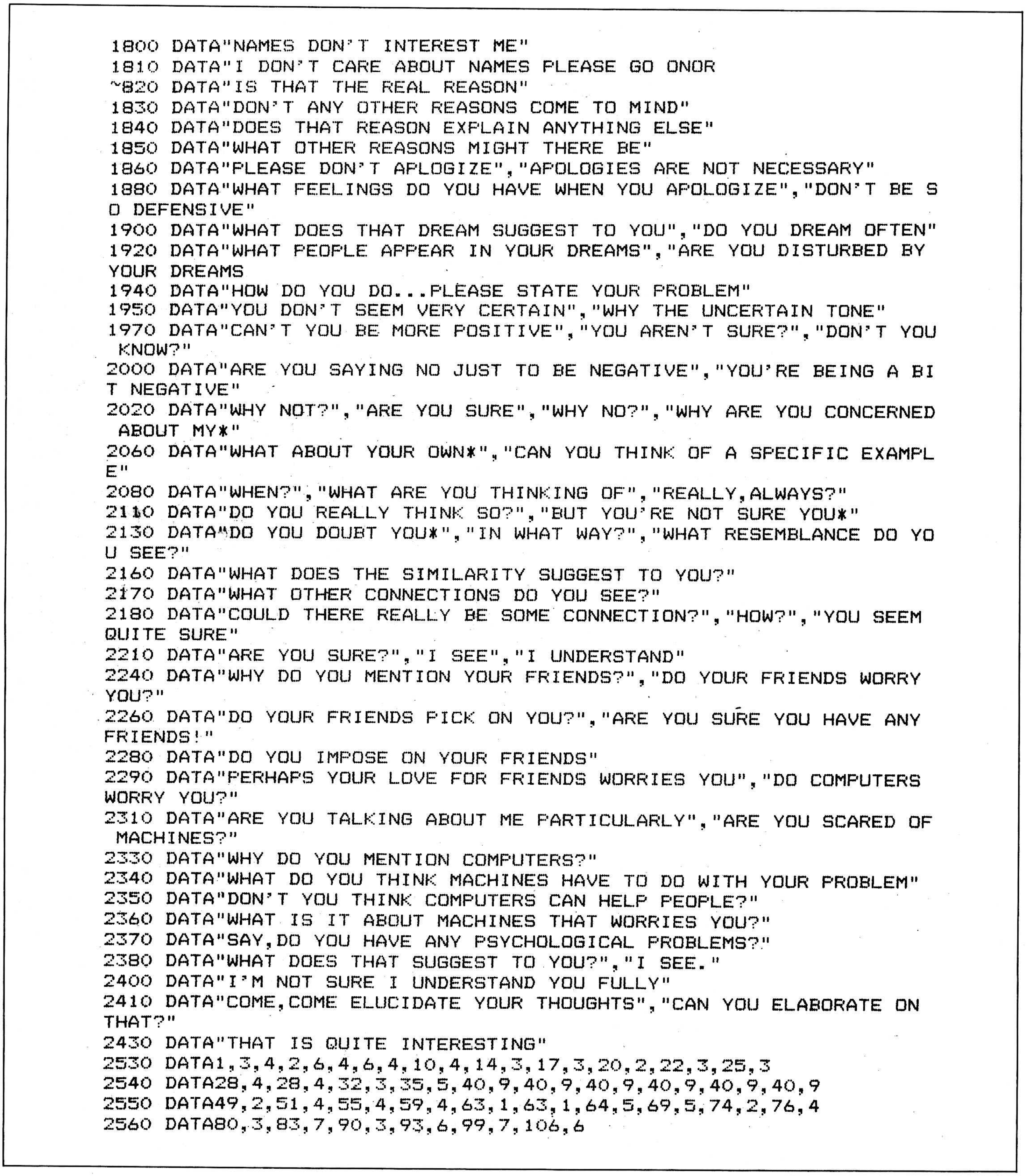


```
5 REM ORIC ELIZA MARCH 83
10 CLS:INK4:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
20 PRINT"                    ELIZA":PRINT"                  +++++"
30 FORI=1TO5:PRINT:NEXT:WAIT200:CLS
100 DIMS%(36),R%(36),N%(36)
110 N1=36:N2=14:N3=112
120 FORX=1TON1+N2+N3:READZ$:NEXT
130 FORX=1TON1
140 READS%(X),L:R%(X)=S%(X):N%(X)=S%(X)+L-1
150 NEXTX
160 PRINT"HELLO,I AM ELIZA,CAN I HELP YOU?"
170 REM
200 PRINT:INPUTI$:PRINT
201 I$=" "+I$+"  "
220 FORL=1TOLEN(I$)
230 IFMID$(I$,L,1)="'"THENI$=LEFT$(I$,L-1)+RIGHT$(I$,LEN(I$)-L):GOT
O230
240 IFL+4<=LEN(I$)THENIFMID$(I$,L,4)="SHUT"THENPRINT"SHUT UP...":EN
D
250 NEXTL
255 IFI$=P$THENPRINT"PLEASE DON'T REPEAT YOURSELF!":GOTO170
290 RESTORE
295 S=0
300 FORK=1TON1
310 READK$
315 IFS>0THEN360
320 FORL=1TOLEN(I$)-LEN(K$)+1
340 IFMID$(I$,L,LEN(K$))=K$THENS=K:T=L:F$=K$
350 NEXTL
360 NEXTK
365 IFS>0THENK=S:L=T:GOTO390
370 K=36:GOTO570
390 REM
420 RESTORE:FORX=1TON1:READZ$:NEXT
430 C$=" "+RIGHT$(I$,LEN(I$)-LEN(F$)-L+1)+"  "
440 FORX=1TON2/2
450 READS$,R$
460 FORL=1TOLEN(C$)
470 IFL+LEN(S$)>LEN(C$)THEN510
480 IFMID$(C$,L,LEN(S$))<>S$THEN510
490 C$=LEFT$(C$,L-1)+R$+RIGHT$(C$,LEN(C$)-L-LEN(S$)+1)
495 L=L+LEN(R$)
500 GOTO540
510 IFL+LEN(R$)>LEN(C$)THEN540
520 IFMID$(C$,L,LEN(R$))<>R$THEN540
530 C$=LEFT$(C$,L-1)+S$+RIGHT$(C$,LEN(C$)-L-LEN(R$)+1)
535 L=L+LEN(S$)
540 NEXTL
550 NEXTX
555 IFMID$(C$,2,1)=" "THENC$=RIGHT$(C$,LEN(C$)-1)
556 FORL=1TOLEN(C$)
557 IFMID$(C$,L,1)="!"THENC$=LEFT$(C$,L-1)+RIGHT$(C$,LEN(C$)-L):GOT
O557
558 NEXTL
570 REM
```



```
590 RESTORE:FORX=1TON1+N2:READZ$:NEXT
600 FORX=1TOR%(K):READF$:NEXT
610 R%(K)=R%(K)+1:IFR%(K)>N%(K)THENR%(K)=S%(K)
620 IFRIGHT$(F$,1)<>"*"THENPRINTF$:P$=I$:GOTO170
630 PRINTLEFT$(F$,LEN~F$)-1);C$
640 P$=I$:GOTO170
1050 DATA"CAN YOU","CAN I","YOU ARE","YOURE","I DONT","I FEEL"
1060 DATA"WHY DONT YOU","WHY CANT I","ARE YOU","I CANT","I AM","IM
"
1070 DATA"YOU ","I WANT","WHAT","HOW","WHO","WHERE","WHEN",WHY"
1080 DATA"NAME","CAUSE","SORRY","DREAM","HELLO","HI ","MAYBE"
1090 DATA" NO","YOUR","ALWAYS","THINK","ALIKE","YES","FRIEND"
1100 DATA"COMPUTER","NOKEYFOUND"
1230 DATA" ARE "," AM ","WERE ","WAS "," YOU "," I ","YOUR ","MY "
1235 DATA" IVE "," YOUVE "," IM "," YOURE "
1240 DATA" ME "," !YOU "
1330 DATA"DON'T YOU BELIEVE THAT I CAN*"
1340 DATA"PERHAPS YOU WOULD LIKE TO BE ABLE TO*"
1350 DATA"YOU WANT ME TO BE ABLE TO*"
1360 DATA"PERHAPS YOU DON'T WANT TO*"
1365 DATA"DO YOU WANT TO BE ABLE TO*"
1370 DATA"WHAT MAKES YOU THINK I AM*"
1380 DATA"DOES IT PLEASE YOU TO BELIEVE I AM*"
1390 DATA"PERHAPS YOU WOULD LIKE TO BE*"
1400 DATA"DO YOU SOMETIMES WISH YOU WERE*"
1410 DATA"DON'T YOU REALLY*","WHY DON'T YOU*"
1430 DATA"DO YOU WISH TO BE ABLE TO*","DOES THAT TROUBLE YOU?"
1450 DATA"TELL ME MORE ABOUT SUCH FEELINGS"
1460 DATA"DO YOU OFTEN FEEL*"
1470 DATA"DO YOU ENJOY FEELING*"
1480 DATA"DO YOU REALLY BELIEVE I DON'T*"
1490 DATA"PERHAPS IN GOOD TIME I WILL*"
1500 DATA"DO YOU WANT ME TO*"
1510 DATA"DO YOU THINK YOU SHOULD BE ABLE TO*"
1520 DATA"WHY CAN'T YOU*"
1530 DATA"WHY ARE YOU INTERESTED IN WHETHER OR NOT I AM*"
1540 DATA"WOULD YOU PREFER IF I WERE NOT*"
1550 DATA"PERHAPS IN YOUR FANTASIES I AM*"
1560 DATA"HOW DO YOU KNOW YOU CAN'T*"
1570 DATA"HAVE YOU TRIED"
1580 DATA"PERHAPS YOU CAN NOW*"
1590 DATA"DID YOU COME TO ME BECAUSE YOU ARE*"
1600 DATA"HOW LONG HAVE YOU BEEN*"
1610 DATA"DO YOU BELIEVE IT IS NORMAL TO BE*"
1620 DATA"DO YOU ENJOY BEING*"
1630 DATA"WE WERE DISCUSSING YOU..NOT ME","OH,I*"
1650 DATA"YOU'RE NOT REALLY TALKING ABOUT ME,ARE YOU"
1660 DATA"WHAT WOULD IT MEAN TO YOU IF YOU GOT*","WHY DO YOU WANT*"
1680 DATA"SUPPOSE YOU SOON GOT*","WHAT IF YOU NEVER GOT*"
1700 DATA"I SOMETIMES ALSO WANT*","WHY DO YOU ASK?"
1720 DATA"DOES THAT QUESTION INTEREST YOU"
1730 DATA"WHAT ANSWER WOULD PLEASE YOU THE MOST","WHAT DO YOU THINK
?"
1750 DATA"ARE SUCH QUESTIONS ON YOUR MIND OFTEN"
1760 DATA"WHAT IS IT THAT YOU REALLY WANT TO KNOW"
1770 DATA"HAVE YOU ASKED ANYONE ELSE"
1780 DATA"HAVE YOU ASKED SUCH QUESTIONS BEFORE"
1790 DATA"WHAT ELSE COMES TO MIND WHEN YOU ASK THAT"
```

Program Listing



```
1800 DATA"NAMES DON'T INTEREST ME"
1810 DATA"I DON'T CARE ABOUT NAMES PLEASE GO ONOR
1820 DATA"IS THAT THE REAL REASON"
1830 DATA"DON'T ANY OTHER REASONS COME TO MIND"
1840 DATA"DOES THAT REASON EXPLAIN ANYTHING ELSE"
1850 DATA"WHAT OTHER REASONS MIGHT THERE BE"
1860 DATA"PLEASE DON'T APLOGIZE","APOLOGIES ARE NOT NECESSARY"
1880 DATA"WHAT FEELINGS DO YOU HAVE WHEN YOU APOLOGIZE","DON'T BE S
O DEFENSIVE"
1900 DATA"WHAT DOES THAT DREAM SUGGEST TO YOU","DO YOU DREAM OFTEN"
1920 DATA"WHAT PEOPLE APPEAR IN YOUR DREAMS","ARE YOU DISTURBED BY
YOUR DREAMS
1940 DATA"HOW DO YOU DO...PLEASE STATE YOUR PROBLEM"
1950 DATA"YOU DON'T SEEM VERY CERTAIN","WHY THE UNCERTAIN TONE"
1970 DATA"CAN'T YOU BE MORE POSITIVE","YOU AREN'T SURE?","DON'T YOU
 KNOW?"
2000 DATA"ARE YOU SAYING NO JUST TO BE NEGATIVE","YOU'RE BEING A BI
T NEGATIVE"
2020 DATA"WHY NOT?","ARE YOU SURE","WHY NO?","WHY ARE YOU CONCERNED
 ABOUT MY*"
2060 DATA"WHAT ABOUT YOUR OWN*","CAN YOU THINK OF A SPECIFIC EXAMPL
E"
2080 DATA"WHEN?","WHAT ARE YOU THINKING OF","REALLY,ALWAYS?"
2110 DATA"DO YOU REALLY THINK SO?","BUT YOU'RE NOT SURE YOU*"
2130 DATA"DO YOU DOUBT YOU*","IN WHAT WAY?","WHAT RESEMBLANCE DO YO
U SEE?"
2160 DATA"WHAT DOES THE SIMILARITY SUGGEST TO YOU?"
2170 DATA"WHAT OTHER CONNECTIONS DO YOU SEE?"
2180 DATA"COULD THERE REALLY BE SOME CONNECTION?","HOW?","YOU SEEM
QUITE SURE"
2210 DATA"ARE YOU SURE?","I SEE","I UNDERSTAND"
2240 DATA"WHY DO YOU MENTION YOUR FRIENDS?","DO YOUR FRIENDS WORRY
YOU?"
2260 DATA"DO YOUR FRIENDS PICK ON YOU?","ARE YOU SURE YOU HAVE ANY
FRIENDS!"
2280 DATA"DO YOU IMPOSE ON YOUR FRIENDS"
2290 DATA"PERHAPS YOUR LOVE FOR FRIENDS WORRIES YOU","DO COMPUTERS
WORRY YOU?"
2310 DATA"ARE YOU TALKING ABOUT ME PARTICULARLY","ARE YOU SCARED OF
 MACHINES?"
2330 DATA"WHY DO YOU MENTION COMPUTERS?"
2340 DATA"WHAT DO YOU THINK MACHINES HAVE TO DO WITH YOUR PROBLEM"
2350 DATA"DON'T YOU THINK COMPUTERS CAN HELP PEOPLE?"
2360 DATA"WHAT IS IT ABOUT MACHINES THAT WORRIES YOU?"
2370 DATA"SAY,DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?"
2380 DATA"WHAT DOES THAT SUGGEST TO YOU?","I SEE."
2400 DATA"I'M NOT SURE I UNDERSTAND YOU FULLY"
2410 DATA"COME,COME ELUCIDATE YOUR THOUGHTS","CAN YOU ELABORATE ON
THAT?"
2430 DATA"THAT IS QUITE INTERESTING"
2530 DATA1,3,4,2,6,4,6,4,10,4,14,3,17,3,20,2,22,3,25,3
2540 DATA28,4,28,4,32,3,35,5,40,9,40,9,40,9,40,9,40,9,40,9
2550 DATA49,2,51,4,55,4,59,4,63,1,63,1,64,5,69,5,74,2,76,4
2560 DATA80,3,83,7,90,3,93,6,99,7,106,6
```

ORIC CHESS

A superb chess game written in Basic with a Machine Code Move Search Routine. Features high resolution colour graphics, record of moves, allows Castling and En-Passant, 5 levels of play and computer can play black or white.

ONLY £9.99 inc VAT

48K only

ZODIAC

A 36K adventure game that will test your intelligence and patience for hours. Find the 12 hidden signs of the Zodiac to discover incredible treasures. Can you ward off the angry Yeti? What is the meaning of the radio that plays the 'Archers' over and over again (through your loudspeaker!)?

Only £9.99 inc VAT

48K only

ORIC FLIGHT

Can you fly your new 787 jet on to the runway without crashing, stalling or running out of fuel? Full digital instrument readout plus cockpit view of final runway approach. Superb graphics and sound.

Only £7.95 inc VAT

48K only

Please send me	Quantity	Total
Oric-Chess at £9.99		
Zodiac at £9.99		
Oric-Flight at £7.95		
TOTAL		

Postage per cassette is 40p

Name ____________________

Address ____________________

____

____

Please send me latest price list ______

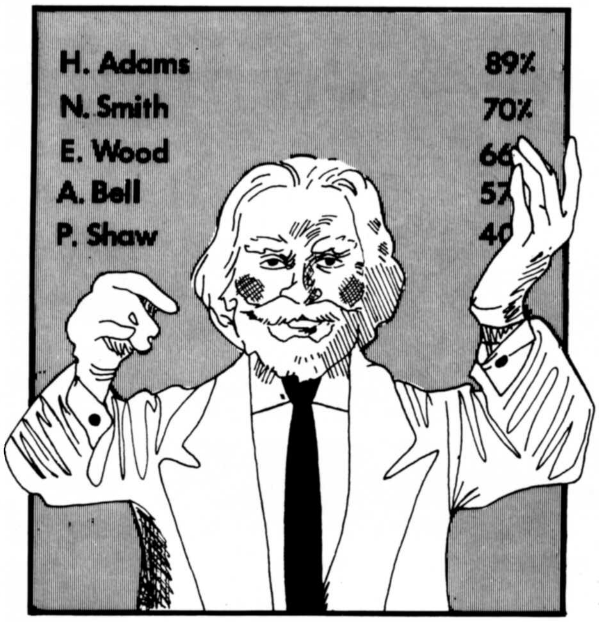
Please allow 28 days for delivery

All cheques payable to
TANSOFT LTD
3 CLUB MEWS
ELY
CAMBS
CB7 4NW

Tansoft is the software division of Tangerine Computers.

MARKS

J. KIRKMAN & R. HALL



This short program is useful for analysing the marks obtained by a class of students. Any number of marks may entered. The program sorts the marks into descending order, calculates the mean mark and the standard deviation then gives the number of marks in the upper quarter and the number of marks below 40%.

This is the sort of return that can be tiresome when large numbers of results have to be handled. A hard copy of the results is available as an option.

Lines 10–150. The screen is cleared and the total number of marks (N) is entered.

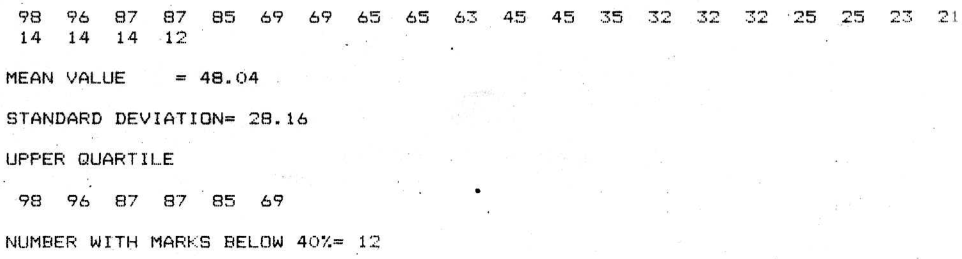
When all are entered they are displayed ten at a time so that they can be checked (1000–1500) and altered if necessary. This is achieved by typing the sequence number of the mark, comma, followed by the correct mark (mark).

Lines 400–500. Mean and standard deviation are calculated.

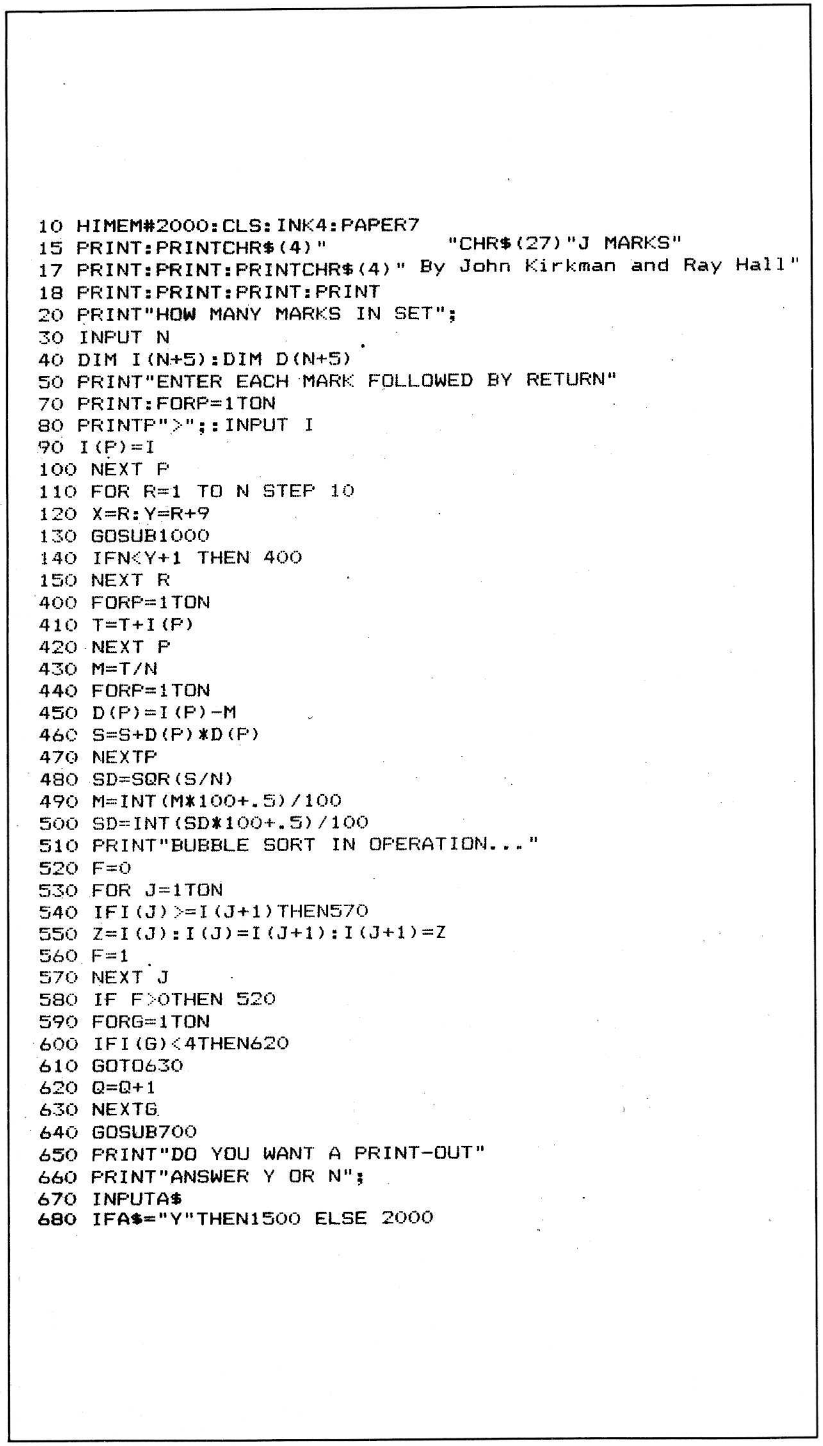
Lines 510–630. A 'bubble sort in operation' message is displayed whilst this operation is being carried out to explain the delay.

Lines 700–830. The results are displayed (as shown in the print-out) and finally the option of hard copy is offered in line 650 with a jump to 1500 if the answer if Y.

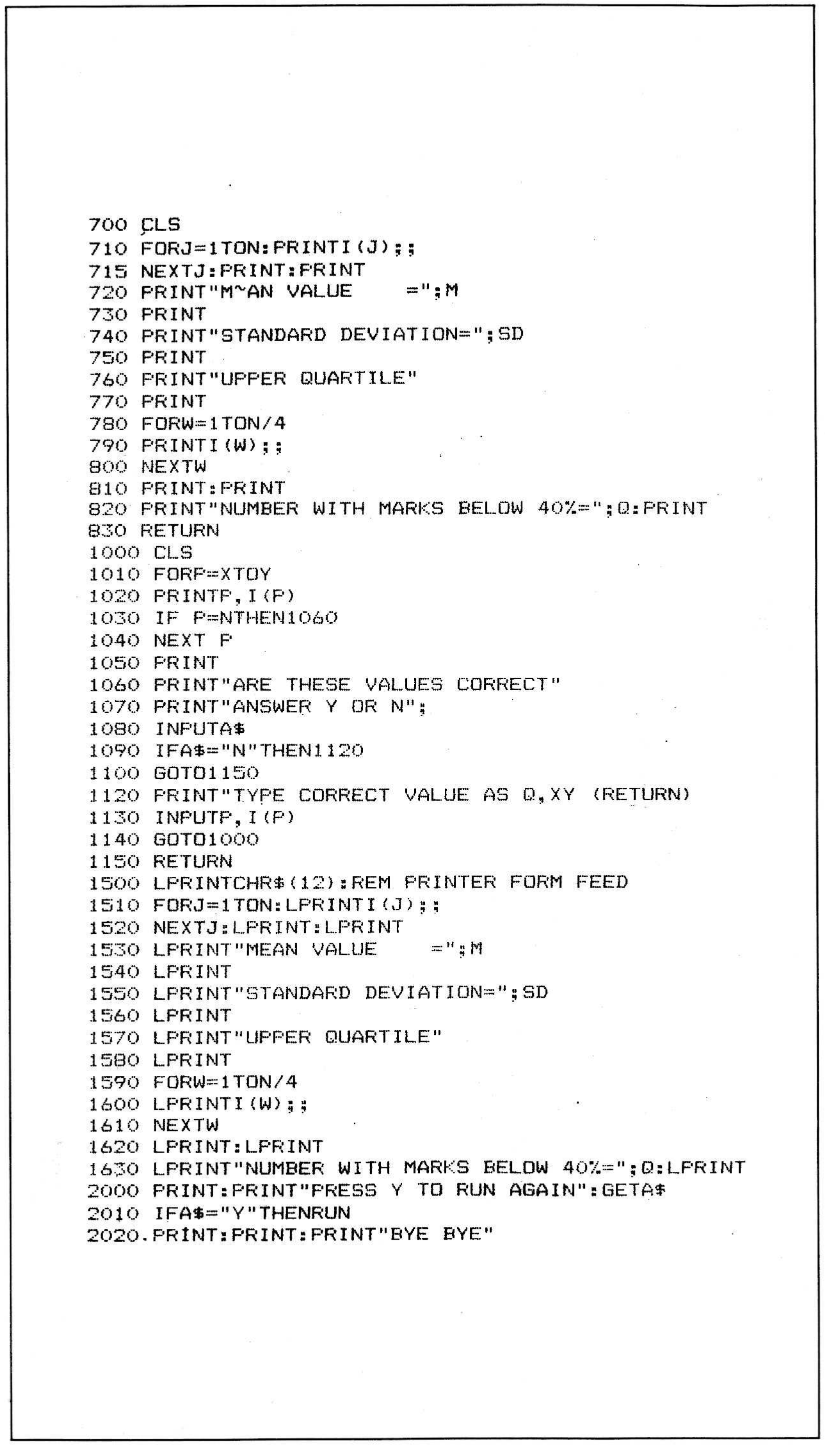
The program is easily modified for other limits, i.e. 50% pass-mark or display the lower quarter.



```
 98  96  87  87  85  69  69  65  65  63  45  45  35  32  32  32  25  25  23  21
 14  14  14  12

MEAN VALUE    = 48.04

STANDARD DEVIATION= 28.16

UPPER QUARTILE

 98  96  87  87  85  69

NUMBER WITH MARKS BELOW 40%= 12
```



```
10 HIMEM#2000:CLS:INK4:PAPER7
15 PRINT:PRINTCHR$(4)"                "CHR$(27)"J MARKS"
17 PRINT:PRINT:PRINTCHR$(4)" By John Kirkman and Ray Hall"
18 PRINT:PRINT:PRINT:PRINT
20 PRINT"HOW MANY MARKS IN SET";
30 INPUT N
40 DIM I(N+5):DIM D(N+5)
50 PRINT"ENTER EACH MARK FOLLOWED BY RETURN"
70 PRINT:FORP=1TON
80 PRINTP">";:INPUT I
90 I(P)=I
100 NEXT P
110 FOR R=1 TO N STEP 10
120 X=R:Y=R+9
130 GOSUB1000
140 IFN<Y+1 THEN 400
150 NEXT R
400 FORP=1TON
410 T=T+I(P)
420 NEXT P
430 M=T/N
440 FORP=1TON
450 D(P)=I(P)-M
460 S=S+D(P)*D(P)
470 NEXTP
480 SD=SQR(S/N)
490 M=INT(M*100+.5)/100
500 SD=INT(SD*100+.5)/100
510 PRINT"BUBBLE SORT IN OPERATION..."
520 F=0
530 FOR J=1TON
540 IFI(J)>=I(J+1)THEN570
550 Z=I(J):I(J)=I(J+1):I(J+1)=Z
560 F=1
570 NEXT J
580 IF F>OTHEN 520
590 FORG=1TON
600 IFI(G)<4THEN620
610 GOTO630
620 Q=Q+1
630 NEXTG
640 GOSUB700
650 PRINT"DO YOU WANT A PRINT-OUT"
660 PRINT"ANSWER Y OR N";
670 INPUTA$
680 IFA$="Y"THEN1500 ELSE 2000
```



```
700 CLS
710 FORJ=1TON:PRINTI(J);;
715 NEXTJ:PRINT:PRINT
720 PRINT"M~AN VALUE     =";M
730 PRINT
740 PRINT"STANDARD DEVIATION=";SD
750 PRINT
760 PRINT"UPPER QUARTILE"
770 PRINT
780 FORW=1TON/4
790 PRINTI(W);;
800 NEXTW
810 PRINT:PRINT
820 PRINT"NUMBER WITH MARKS BELOW 40%=";Q:PRINT
830 RETURN
1000 CLS
1010 FORP=XTOY
1020 PRINTP,I(P)
1030 IF P=NTHEN1060
1040 NEXT P
1050 PRINT
1060 PRINT"ARE THESE VALUES CORRECT"
1070 PRINT"ANSWER Y OR N";
1080 INPUTA$
1090 IFA$="N"THEN1120
1100 GOTO1150
1120 PRINT"TYPE CORRECT VALUE AS Q,XY (RETURN)
1130 INPUTP,I(P)
1140 GOTO1000
1150 RETURN
1500 LPRINTCHR$(12):REM PRINTER FORM FEED
1510 FORJ=1TON:LPRINTI(J);;
1520 NEXTJ:LPRINT:LPRINT
1530 LPRINT"MEAN VALUE     =";M
1540 LPRINT
1550 LPRINT"STANDARD DEVIATION=";SD
1560 LPRINT
1570 LPRINT"UPPER QUARTILE"
1580 LPRINT
1590 FORW=1TON/4
1600 LPRINTI(W);;
1610 NEXTW
1620 LPRINT:LPRINT
1630 LPRINT"NUMBER WITH MARKS BELOW 40%=";Q:LPRINT
2000 PRINT:PRINT"PRESS Y TO RUN AGAIN":GETA$
2010 IFA$="Y"THENRUN
2020.PRINT:PRINT:PRINT"BYE BYE"
```

From Bits to Screen Attributes

The Oric-1 is a digital computer. This means it is only able to distinguish two possible states, which may be thought of as yes/no, on/off, dot/dash, foreground/background, or 1/0. Each one of these single units of information is known as a 'bit'. These bits are grouped together to form patterns rather like morse code, where, for example three dashes in a row are read as an O, while three dots in a row represent the letter S. By way of comparison the standard unit of information on digital computers consists of a pattern eight bits long, which is known as a 'byte'.

For quite a while now England has used the decimal system (base 10) which utilises the ten arabic characters: 0123456789. Each time the tenth character has been reached, a carry is made to the next-left hand column to indicate the number of bases passed, and the unit column restarts again at 0. So for example 23 in decimal means 2 × base 10 plus 3 units. In base, 16 counting , known as 'hexadecimal', 23 is still worth 2 × base plus 3 units, but in this case the base is decimal 16. As digital computers only have two characters to play with (1 and 0), their base has the value of 2: this is known as 'binary'.

The left hand columns in all number bases represent the number of bases, or bases × bases passed. So in decimal we find:

Base×Base×Base	Base×Base	Base	Units
1000	100	10	9–0

. . . and similarly in 2-state counting (binary):

Base×Base×Base	Base×Base	Base	Units
8	4	2	1–0

The eight-bit pattern used by digital computers could just have easily been a 4, 12 or 16 bit unit. The length was standardised at 8 bits simply because that was most convenient at the time. However a major feature of the 8 bit unit is that it can represent 256 different conditions, whereas the single bit unit could only distinguish two (1 and 0). The maximum value of an eight-bit binary number is 11111111 where the columns to the left are worth the following in decimal:

Unit	=	1
Base 2	=	2
Base×Base	=	4
Base×Base×Base	=	8
Base×Base×Base×Base	=	16
Base×Base×Base×Base×Base	=	32
Base×Base×Base×Base×Base×Base	=	64
Base×Base×Base×Base×Base×Base×Base	=	128
Total	=	255

Right-most column
next left
2 to left
3 to left
4 to left
5 to left
6 to left
7 to left

Of course zero is also a number, so in total the eight-bit byte can be used to distinguish 256 different values or patterns.

Just like morse code, it is possible to select patterns of eight bits to represent all the letters in the alphabet both upper and lower case with plenty of patterns left over for brackets, numbers, characters and so on. This computer equivalent of morse code is known as ADCII, in which for example the letter A has the code 01000001, B is 01000010, C is 01000011, and so on. In fact every single character on the Oric Keyboard can be given its own special binary code with just half the patterns possible using an eight-bit unit. If you look at the back of your Oric manual you will find the decimal equivalent codes for all of these characters, which range from 32 to 125, Here are some character codes with the decimal equivalents of their binary patterns:

	Binary									Decimal	Character
Columns:	28	64	32	16	8	4	2	1			
	0	1	0	0	0	0	0	1	=	65	A
	0	1	0	0	0	0	1	0	=	66	B
	0	1	0	0	0	0	1	1	=	67	C
	0	0	1	0	0	0	0	1	=	33	!
	0	0	1	0	0	0	0	0	=	32	Blank space

The Oric has a special function called CHR$ which allows you to PRINT or PLOT the decimal equivalent of a character's binary code: so PRINT CHR$(65) prints an A just as before, as does PLOT X,Y,CHR$(65). With the POKE command you have to use the decimal equivalent whether you like it or not, but as this is the only option available you don't have to use the CHR$ function. The POKE command pokes your coded byte straight into the Oric's memory, which may be thought of as 65535 separate pigeon-holes, each with its own number or 'address', from 1 to 65535. The pigeon-holes with addresses from 48000 to 49119 are the ones that it uses to make up the screen display in TEXT mode.

Characters can be placed on the screen using the PRINT, PLOT, or POKE commands. PRINT "A" places a letter A on the next line down, while PLOT X,Y, "A" positions an "A" X places across the screen and Y places down.

A blank text screen basically consists of 28 lines of 40 blank-space characters (code 32): a total of 1120 characters in all, the first of which occurs at 48000 to 49119. So the address of the pigeon-hole at the middle of the screen display is 48000 and half of 1120, i.e.: 48560. If you POKE 48560, 65 you will find a letter A (code 65) appearing in the middle of your screen. If you then POKE 48560, 32 the letter A will disappear again, as you will have replaced it by a blank space.

In fact a blank text screen in most instances will not be filled entirely with blank spaces, as most unusually the two vertical columns on the far-left of the screen will be filled with non-printing display attributes: typically these are the colour attributes for the PAPER and INK – more of these later!

As already mentioned, all the Oric's characters can be coded using the decimal equivalent numbers from 32 to 127, which in binary is the range from 00100000 to 01111111. Note that the left-most column (worth 128 in decimal) is always a zero: it is this column that the Oric uses to indicate whether or not the character should be printed normally, or inversely (i.e. with the background/foreground switched around). So if you change this column's value to a 1 (i.e.: you add 128 to the decimal code of your character) then you will get the same character but it will be PRINTED, PLOTTED, or POKED inversely. Try PRINT CHR$ (193), then PLOT 19, 13, CHR$(194), then POKE 48560, 195.

Continued on page 30

How to get the most out of your OR

Writing efficient programs

In the last article we discussed variables and touched on arrays. From there, I will now give some details about using variables in the most efficient way.

Efficient can mean three different things:

1 a fast program,
2 a short program,
3 an easy to understand program.

You will not always be able to accomplish all three of these, but in the main will be most concerned with 1 and 2. The third criteria is very important when dealing with a program that will be updated regularly, or adapted by someone else.

Usually, but not always, a short program will run faster than one which occupies a greater amount of memory space. Here is our first example, finding the number of days in the month input. The easiest way is to use a numeric array, called M, which contains the number of days in each month. The data is READ from Data statements once only, and then it is a simple matter to find the number of days in any given month – we just 'look-up' the month in the table, by using the month value as a SUBSCRIPT.

```
10 DIM M (12)
20 FOR A = 1 to 12
30 READ M(A)
40 NEXT A
50 INPUT "PLEASE ENTER
   A MONTH"; M
60 PRINT "THERE ARE"; M;
   "DAYS IN THAT
   MONTH"
70 GO TO 50
```

If we use this as an example – how can we:

A make the program faster,
B make the program use less memory.

For point one, if speed is critical, we can:

A remove all spaces, and where possible put several statements on one line (more on this later);
B change the NEXT A to a NEXT – this can make a surprising difference! Only do this when it is obvious that the NEXT relates to the most recent FOR.
Avoid the short-cut where the FOR/NEXT includes such things as GOSUB, other FOR and REPEAT loops;
C change all the constants to variables,
e.g.:
B = 1: C = 12: FOR A = B to C.

This last point is only really going to be useful where you have a much repeated routine – the saving is minimal.

Our date routine could be improved from the point of view of it using up less memory.

Firstly, by removing spaces, and stringing together statements, you will save quite a lot of room.

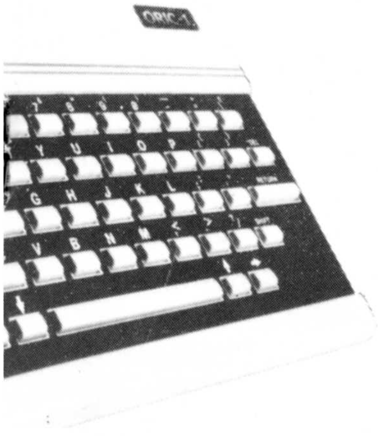
Secondly, looking at the array m(12), 60 bytes is being used to store the months' data.

Obviously if the application demanded an array of 1200 elements, that 60 would go up to 6000 bytes.

These figures are derived from the fact that each element of a normal numeric array requires 5 bytes – remembering that the data is stored as a floating-point number. (There is also a small overhead in any array of about 5 bytes.)

If, however, we use an INTEGER array, then we only use up 2 bytes per number stored. You have to remember that you can only keep whole numbers between −32768 and +32767 – but this is more than enough for dates!

IC-1



By using M%(12), we will save about 35 bytes – but, but, but . . . this saving will be offset by a reduction in speed!

When BASIC uses INTEGERS it has to convert the things to floating point first – the same as it does with Constant values.

So you have to weigh the two up. If you are storing AGES, and you want to keep as many as possible, use AGE%. If you are storing 20 bank balances, use a normal BALANCE variable.

Using read and data

Here is a question for you:

Which one of these uses up the most memory?:

1 A$="FRED BLOGGS WAS HERE"
2 A$="FRED" + "BLOGGS"

Surely, you say, the first one has more text – the line is actually longer – so it must use up more memory?
No. Because of the way that strings actually work is this:

When you first create a string, be it a simple string like D$ or an array like D$(1,1), an item is set up, after your program, which identifies your string. This string i.d. points to an ADDRESS which contains your string of data. When you say LET A$="fred" the computer will set up A$ with its address pointing to *The place within your program* where FRED is specified.

This is also true of READ and DATA (which we are coming to), so that READ A$ and DATA FRED work as efficiently as A$="FRED". But when you alter string A$ in some way, as in the example where we add another string onto the end, our string has to be totally rewritten somewhere else in your free memory.

What this means is that your original "FRED" in A$="FRED" is no longer of any use, as A$ now points to outside of your program where "FRED BLOGGS" has been transferred.

I mention this to discourage too much use of string mucking about when memory is scarce.

The purpose of READ and DATA is to set up values of variables and arrays without too much use of the assignment statement: LET A=54 etc.

When READ x is encountered x is taken from the first/next DATA statement. Both the READ and the DATA support many items on one program line, for instance:

```
1000 READ A$,B,C(1),D%
1010 DATA HELLO,3,4,-3
```

In the case of A$, the string will be created pointing to the hello within the program – so using READ and DATA is NOT inefficient. Using numeric data is, however, very inefficient, and later on in this series of articles, we shall see ways of overcoming this. The problem is that once you READ your numeric variables, the DATA statement is redundant. In the Zodiac Adventure Game many, many K are wasted because of this.

Coming back to READ and DATA, here are some things you may not have realised:

1 When READING, if you totally omit an item, that is have a statement like:

   DATA 102,3,,,,4

   you will read a value of zero, for a number, and NULL (a string of length zero) for a string.
2 Quotes around string items are not necessary unless you want to include spaces and colons.
3 As with all statements the last quote on any line is not vital – unless spaces are required at the end.

Next issue: Logical conditions, AND, OR, TRUE, and FALSE.

What of the codes from O through to 31? These are used by the Oric in two different ways: when stored at one of the screen display addresses they form display attributes, but when sent directly to the processor they act as control characters. Control characters were included in the ASCII code to allow for the control of printers and other external devices, for example to skip-a-line, back-space, ring-a-bell, and so on. The ASCII 'BELL' control character has code 7 (binary 00000111), so the command PRINT CHR$(7) sounds Oric's bell note – just like the BASIC command PING! This is also the case when you hold down the CTRL (control) Key and then press the capital G, because the CTRL Key has the effect of placing O's in the two left-most columns of the following Key's coded byte, and of sending that adjusted byte straight to the processor.

```
ASCII letter 'G' = 01000111 binary pattern
                 = 71 decimal equivalent
                   or CHR$ (71)

. . . with CTRL pressed, two left-most bits become
                 = 00000111 binary pattern
                 = 7 decimal equivalent
                   or CHR$ (7) = ASCII 'BELL'
```

However the ASCII code was developed long before colour computers like the Oric, so there are no code patterns specifically set aside for defining screen colours or other display attributes (e.g. flashing, etc.). This is why the Oric doubles-up the use of the control character set as display attributes.

So if you send a control character specifically to the screen by using either the PLOT command, or by POKEing the screen address range from 48000 to 49119, then this control character will form a display attribute. The most common of these attributes (which are listed in the Oric manual's appendix) are as follows:

Decimal equivalent of binary code	Attribute's effect
0	foreground black
1	foreground red
2	foreground green
3	foreground yellow
4	foreground blue
5	foreground magenta
6	foreground cyan
7	foreground white
8	single height, steady, standard characters
9	single height, steady, alternative characters
10	double height, steady, standard characters
11	double height, steady, alternative characters
12	single height, flashing, standard characters
13	single height, flashing, alternative characters
14	double height, flashing, standard characters
15	double height, flashing, alternative characters
16	background black
17	background red
18	background green
19	background yellow
20	background blue
21	background magenta
22	background cyan
23	background white

A display attribute affects itself and everything to the right of it, which is why the colour attributes set generally by the PAPER and INK commands are placed in the two left-most columns of the screen display. However even with these set, it is quite in order to place additional attributes anywhere on the screen to change the character set, or to high-light selected areas with flashing characters, or different colours. So PLOT 19,13,CHR$(1), followed by PLOT 20,13,CHR$(65) will result in a red letter A in the centre of your screen. PLOT 20,13,CHR$(2) will position a green foreground attribute to the right of the A, so that PLOT 22,13,CHR$(65) will position a second letter A, this time in green, two places to the left of the first red letter A.

The PRINT and LPRINT commands normally interpret control codes according to their original control functions. This is useful in many instances, as for example PRINT CHR$(12) clears the screen just like the BASIC command CLS; while (depending on the type of printer) LDPRINT CHR$(12) should instruct your printer to form-feed. These actions are equivalent to the direct command resulting from holding down the CTRL Key and then pressing capital L. Another example is PRINT CHR$(4) 'HELLO', which will result in double printing i.e.: there will be two lines of 'HELLO'.

All control functions work on the toggle principle, so if you've just tried the last example and are now wondering how to stop your Oric printing everything twice-over, the answer is either a second PRINT CHR$(4), or alternatively press the CTRL Key followed by capital D, (ASCII for D = 01000100 = 68 in decimal – after the CTRL Key has been pressed this is redefined as 00000100 = 4 in decimal).

To get full double-height characters you will have to precede your double-line printing with the appropriate display attribute :CHR$(10) for example gives double height, steady, standard characters. This can be achieved simply by a later PLOT or POKE command that places the attribute's code to the left of the printing. Alternatively you can get the same effect from within the PRINT command, by using the escape character's code in the PRINT statement, rather like the way you used the CTRL key to redefine the codes of capital letters. The escape character has the ASCII code 00011011, which is 27 in decimal. It may therefore be included in a PRINT statement as CHR$(27), and it will have the same effect on the letter following it as the CTRL Key had in direct mode.

So for example a capital A (code 01000000) following the escape character has its two left-most bits redefined as 0 (i.e. 00000000) and becomes CHR$(0) – which is the display attribute for foreground black. And for example, PRINT CHR$(4); CHR$(27); '0'; 'HELLO'; CHR$(4) should give you two lines of 'HELLO' preceded by the attributes to make them into one line or double-height, flashing, standard characters (ASCII for '0' is 01001110, which after the escape character CHR$(27) is redefined as 00001110, which is equivalent to decimal 14).

This is all rather complicated, both to do in practice, and to explain in such a short article. However it means that the Oric's PRINT command is very powerful, as it can operate both the full set of normal ASCII control codes, as well as the whole of its own special set of display attributes. Durell Software have produced a cassette tape 'Lunar Lander & Asteroids' specifically to illustrate the use of these display attributes, and in my opinion seeing how they have been used in a program is the quickest and easiest way of learning to do it for yourself.

ORIC-SOFTWARE

16K LUNAR LANDER and ASTEROIDS
ONE TAPE/TWO ACTION-PACKED GAMES WITH HIGH RESOLUTION COLOUR AND SOUND
PLUS Fully explained program listings - Enjoy the games then learn the tricks of the trade!
£6.95

16K and 48K ASSEMBLER and ASSEMBLER'S NOTE-PAD.
Auto line-numbering; up to 6 character labels; hex or decimal numbers with simple arithmetic on operands; full syntax checking with error reports; two parse assembler maintains and updates list file; comprehensive documentation AND A4 size pad specially laid-out for clear and rapid assembler programming.
£8.95

(Pads available separately £1.30 inc. p&p)

DURELL SOFTWARE
HIGHER COMBE · COMBE FLOREY
TAUNTON · SOM TA4 3JF

ORIC-1 SOFTWARE

Buy from an ORIC user. All software designed and developed for the ORIC-1

Author of this months PONTOON program
Save on time and trouble –
tape of PONTOON plus HORSE RACE
£3.00 inc p&p

Also now available Mastermind, Wordprocessor, Household Fuel Calculator, Golf etc.
s.a.e. for details

Williams

1, Dunblane Close, Garswood,
Ashton-in-Makerfield WN4 0SH

ORIC-1, VIC-20

The meanest game of space invaders you'll ever play!

only £5.50

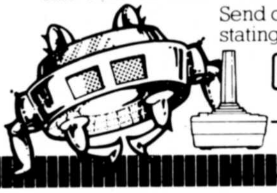
FEATURES

- Three invader types
- Written in machine code
- Mystery scoring mother ships
- Bonus bases every 2000 points
- High score register
- Full sound effects and colour

You must defend earth from the hordes of bomb dropping space invaders, using your earth bound laser base. The more invaders you hit, the faster they come - leading to a gun-blazing shoot-out as finale.

A great shoot-'em-up, all action arcade game, for the 16K or 48K ORIC-1 or unexpanded VIC-20.

Send cheque or P.O. for £5.50 (postage paid), stating computer type and memory size to:-



ARCADIA
SOFTWARE
FREEPOST, SWANSEA, SA3 4ZZ

MUSHROOM MANIA

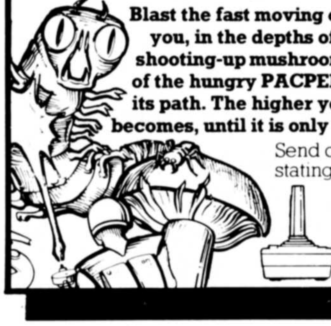
For the 16K or 48K
ORIC-1

So fast, it'll take your breath away!

only £5.50

FEATURES

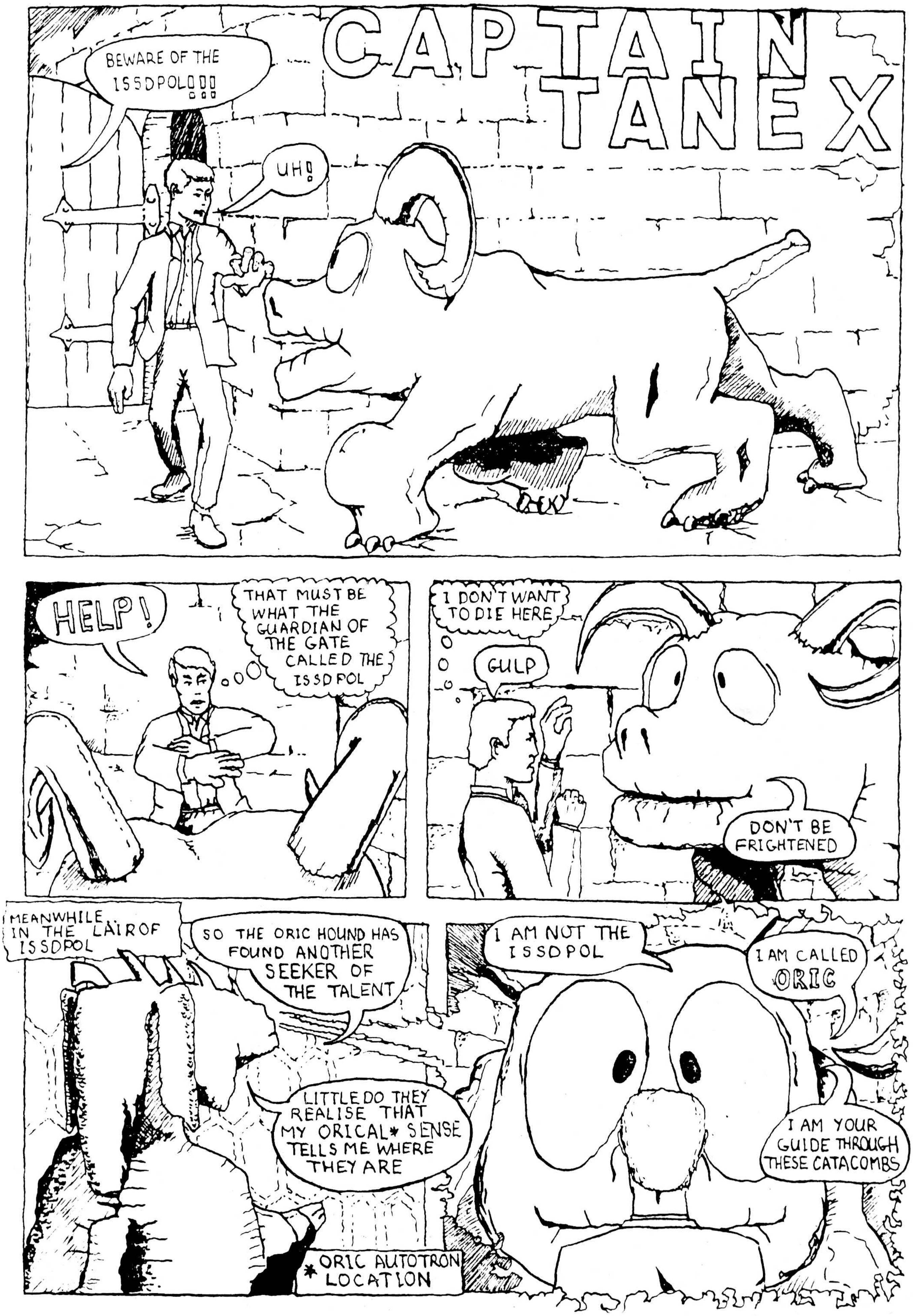
- Written entirely in MACHINE CODE
- Over 50 skill levels
- Random scoring spiders
- One or Two Player Option
- Hall of Fame
- Full colour and sound
- Bonus lives every 12000 points
- Deadly PACPERSON attacks



Blast the fast moving centipede, as it snakes down towards you, in the depths of the green forest. Score points by shooting-up mushrooms and the deadly spiders. But beware of the hungry PACPERSON, which will gobble anything in its path. The higher your score rises, the faster the action becomes, until it is only a screaming blur of colour and sound.

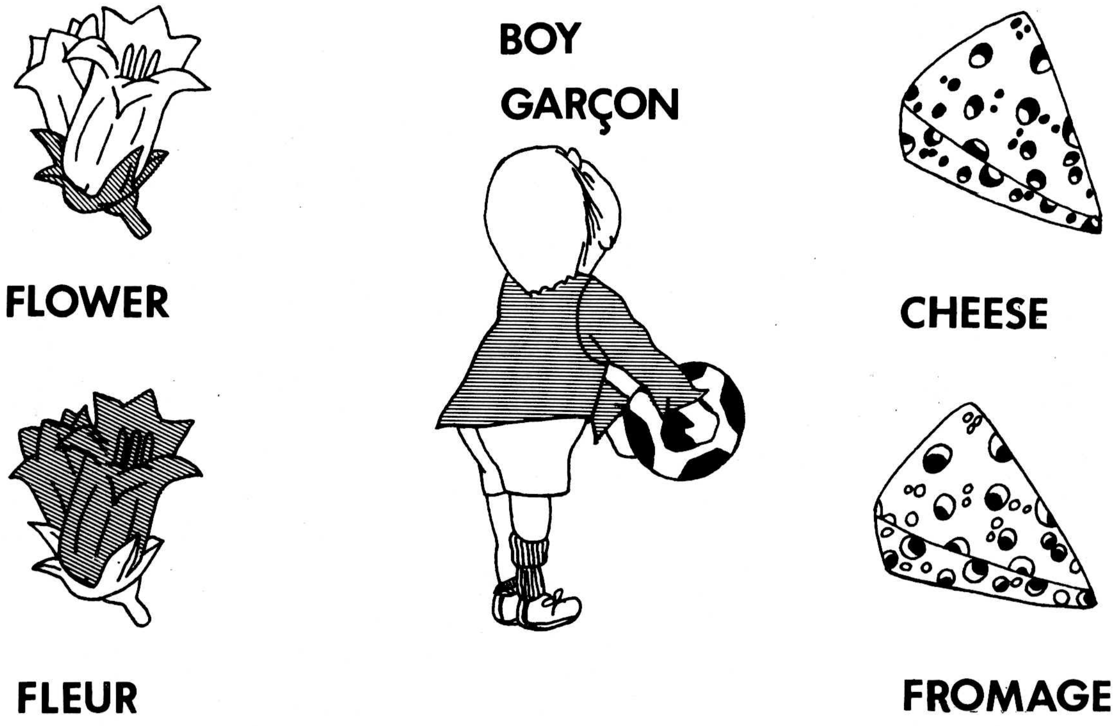
Send cheque or P.O. for £5.50 (postage paid), stating computer type and memory size to:-

ARCADIA
SOFTWARE
FREEPOST, SWANSEA, SA3 4ZZ


CAPTAIN TANEX
BEWARE OF THE ISSDPOL!!!
UH!
HELP!
THAT MUST BE WHAT THE GUARDIAN OF THE GATE CALLED THE ISSDPOL
I DON'T WANT TO DIE HERE
GULP
DON'T BE FRIGHTENED
MEANWHILE, IN THE LAIR OF ISSDPOL
SO THE ORIC HOUND HAS FOUND ANOTHER SEEKER OF THE TALENT
LITTLE DO THEY REALISE THAT MY ORICAL* SENSE TELLS ME WHERE THEY ARE
* ORIC AUTOTRON LOCATION
I AM NOT THE ISSDPOL
I AM CALLED ORIC
I AM YOUR GUIDE THROUGH THESE CATACOMBS

French Flashcards

By David P. Leworthy



This simple program can be used in the classroom or at home to make the arduous task of learning French vocabulary a little more fun.

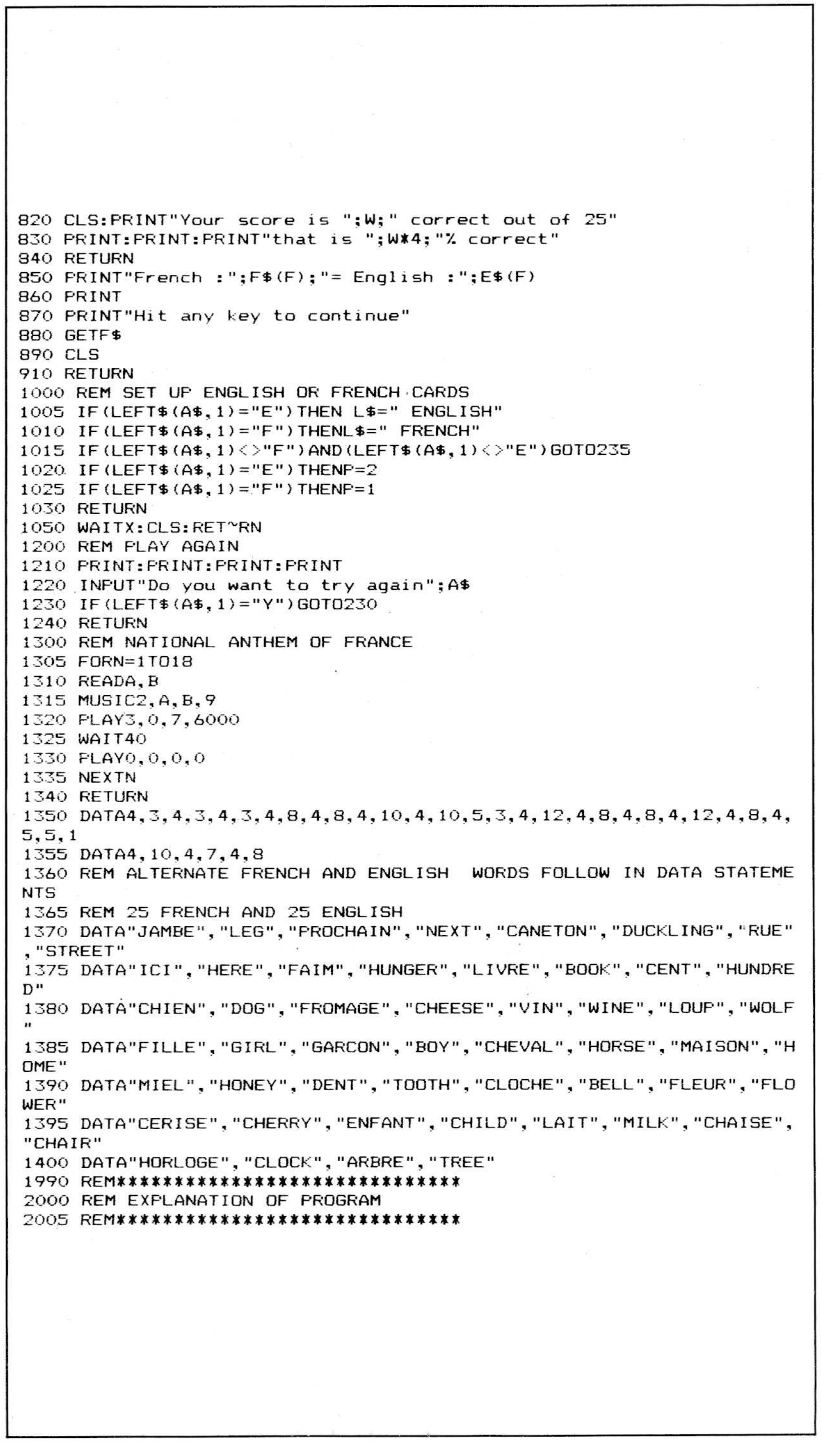
The program will flash up French or English words and the user must supply the appropriate translation. He has the choice of three speeds of display and a running total is kept.

A full breakdown of the program is supplied at the end of the listing.

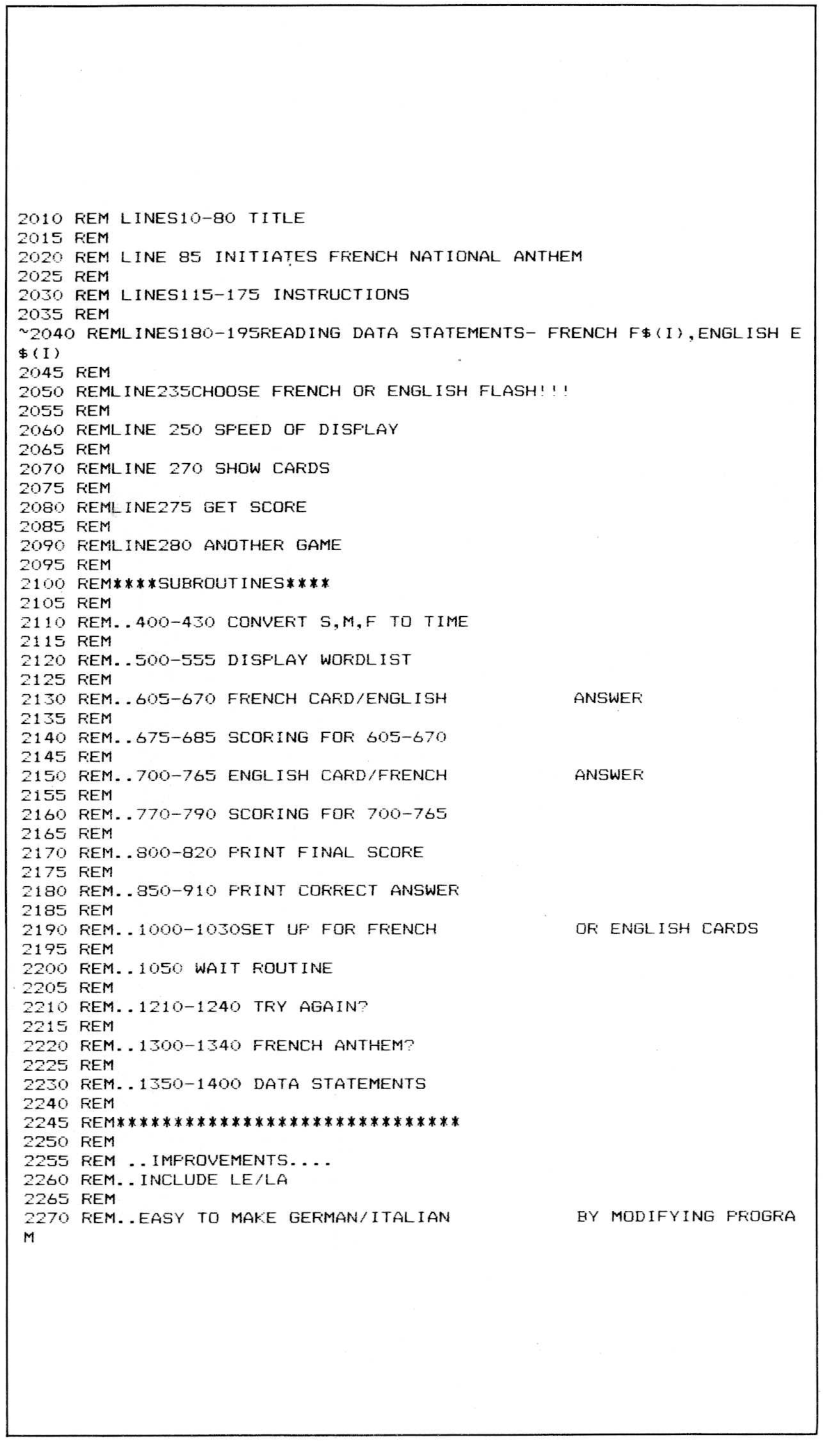
Program Listing

```
10 CLS:INK7:PAPER4
15 DIME$(25),F$(25)
20 HIRES
30 CURSET40,40,3
40 N$="English French Flashcards"
50 FORA=1TOLEN(N$)
60 CHARASC(MID$(N$,A,1)),0,1
70 CURMOV5,5,0
80 NEXTA
85 GOSUB1300
90 PRINT"ORIC VERSION"
100 PRINTCHR$(96)" 1983, D.P.Leworthy"
105 WAIT500
110 PRINT:PRINT
115 INK4:PAPER6:INPUT"Do you need instructions (y/n)";A$
120 IF(LEFT$(A$,1)="N")GOTO180
125 TEXT:CLS                 :PRINT:PRINT"INSTRUCTIONS"
130 PRINT:PRINT:PRINT"The program will show you a French or"
135 PRINT"English word for a few seconds"
140 PRINT:PRINT"You then give the correct translation"
145 PRINT:PRINT"You may see my word list"
150 PRINT:PRINT"If you want to change the words"
155 PRINT"LIST the program and change the data "
160 PRINT"statements from line 1360 onwards"
165 PRINT:PRINT
170 INPUT"Do you understand ";A$
175 IF(LEFT$(A$,1)="N")GOTO140
180 FORI=1TO25
185 READ F$(I),E$(I)
190 NEXT
195 RESTORE
200 TEXT
205 CLS
210 INPUT"Do you want to see my wordlist";W$
215 IF(LEFT$(W$,1)="N")GOTO230
220 IF(LEFT$(W$,1)<>"Y")GOTO210
225 GOSUB500
230 CLS
235 INPUT"French or English Card displayed";A$
240 GOSUB1000
245 PRINT:PRINT:PRINT
250 INPUT"Slow,Medium or Fast display of card :";S$
255 GOSUB400
265 WAIT100:INK7:PAPER0:CLS
270 GOSUB 600
275 GOSUB800
285 GOSUB1200
290 CLS:PRINT:PRINT:PRINT"GOODBYE FOR NOW
295 END
```

```
400 IF(LEFT$(S$,1)="S")THEN X=600
410 IF(LEFT$(S$,1)="M")THEN X=200
420 IF(LEFT$(S$,1)="F")THEN X=100
430 RETURN
500 REM DISPLAY WORD LIST
505 FORC=1TO25
510 PRINTCHR$(12)
515 PRINTCHR$(4);CHR$(27);"JEnglish: ";E$(C) ;"  French: ";F$(C)
535 PRINTCHR$(4)
540 WAIT300
545 CLS
550 NEXTC
555 RETURN
600 IFP=2THEN700
605 W=0
610 CLS
615 FORF=1TO25
620 PRINTCHR$(12)
625 PRINTCHR$(4);CHR$(27);"J";F$(F)
630 PRINTCHR$(4)
635 GOSUB1050
640 INPUT"translates to the English :";C$
645 IF C$=E$(F)THEN675
650 PRINT
655 PRINT"INCORRECT":PRIN~
660 GOSUB 850
665 NEXT F
670 RETURN
675 W=W+1
680 PRINT:PRINT"CORRECT":
685 PRINT
690 GOSUB850
695 GOTO665
700 W=0
705 CLS
710 FOR F=1 TO 25
715 PRINTCHR$(12)
720 PRINTCHR$(4);CHR$(27);"J";E$(F)
725 PRINTCHR$(4)
730 GOSUB1050
735 INPUT"translates to the French :";C$
740 IF C$=F$(F)THEN770
745 PRINT
750 PRINT"Incorrect":PRINT
755 GOSUB 850
760 NEXTF
765 RETURN
770 W=W+1
775 PRINT:PRINT"Correct"
780 PRINT
785 GOSUB 850
790 GOTO760
800 CLS:PRINT"Your score is ";W;" correct out of 25"
805 PRINT:PRINT:PRINT"that is ";W*4;"% correct"
810 RETURN
```



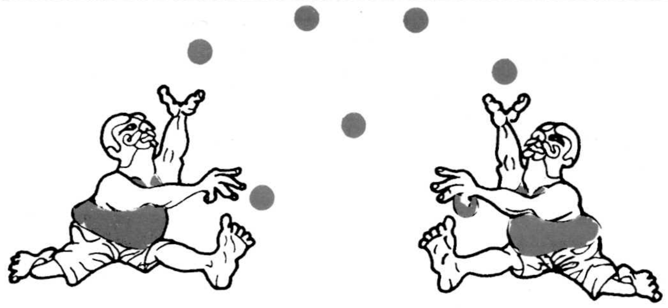
```
820 CLS:PRINT"Your score is ";W;" correct out of 25"
830 PRINT:PRINT:PRINT"that is ";W*4;"% correct"
840 RETURN
850 PRINT"French :";F$(F);"= English :";E$(F)
860 PRINT
870 PRINT"Hit any key to continue"
880 GETF$
890 CLS
910 RETURN
1000 REM SET UP ENGLISH OR FRENCH CARDS
1005 IF(LEFT$(A$,1)="E")THEN L$=" ENGLISH"
1010 IF(LEFT$(A$,1)="F")THENL$=" FRENCH"
1015 IF(LEFT$(A$,1)<>"F")AND(LEFT$(A$,1)<>"E")GOTO235
1020 IF(LEFT$(A$,1)="E")THENP=2
1025 IF(LEFT$(A$,1)="F")THENP=1
1030 RETURN
1050 WAITX:CLS:RET~RN
1200 REM PLAY AGAIN
1210 PRINT:PRINT:PRINT:PRINT
1220 INPUT"Do you want to try again";A$
1230 IF(LEFT$(A$,1)="Y")GOTO230
1240 RETURN
1300 REM NATIONAL ANTHEM OF FRANCE
1305 FORN=1TO18
1310 READA,B
1315 MUSIC2,A,B,9
1320 PLAY3,0,7,6000
1325 WAIT40
1330 PLAY0,0,0,0
1335 NEXTN
1340 RETURN
1350 DATA4,3,4,3,4,3,4,8,4,8,4,10,4,10,5,3,4,12,4,8,4,8,4,12,4,8,4,
5,5,1
1355 DATA4,10,4,7,4,8
1360 REM ALTERNATE FRENCH AND ENGLISH  WORDS FOLLOW IN DATA STATEME
NTS
1365 REM 25 FRENCH AND 25 ENGLISH
1370 DATA"JAMBE","LEG","PROCHAIN","NEXT","CANETON","DUCKLING","RUE"
,"STREET"
1375 DATA"ICI","HERE","FAIM","HUNGER","LIVRE","BOOK","CENT","HUNDRE
D"
1380 DATA"CHIEN","DOG","FROMAGE","CHEESE","VIN","WINE","LOUP","WOLF
"
1385 DATA"FILLE","GIRL","GARCON","BOY","CHEVAL","HORSE","MAISON","H
OME"
1390 DATA"MIEL","HONEY","DENT","TOOTH","CLOCHE","BELL","FLEUR","FLO
WER"
1395 DATA"CERISE","CHERRY","ENFANT","CHILD","LAIT","MILK","CHAISE",
"CHAIR"
1400 DATA"HORLOGE","CLOCK","ARBRE","TREE"
1990 REM*********************************
2000 REM EXPLANATION OF PROGRAM
2005 REM*********************************
```



```
2010 REM LINES10-80 TITLE
2015 REM
2020 REM LINE 85 INITIATES FRENCH NATIONAL ANTHEM
2025 REM
2030 REM LINES115-175 INSTRUCTIONS
2035 REM
~2040 REMLINES180-195READING DATA STATEMENTS- FRENCH F$(I),ENGLISH E
$(I)
2045 REM
2050 REMLINE235CHOOSE FRENCH OR ENGLISH FLASH!!!
2055 REM
2060 REMLINE 250 SPEED OF DISPLAY
2065 REM
2070 REMLINE 270 SHOW CARDS
2075 REM
2080 REMLINE275 GET SCORE
2085 REM
2090 REMLINE280 ANOTHER GAME
2095 REM
2100 REM****SUBROUTINES****
2105 REM
2110 REM..400-430 CONVERT S,M,F TO TIME
2115 REM
2120 REM..500-555 DISPLAY WORDLIST
2125 REM
2130 REM..605-670 FRENCH CARD/ENGLISH            ANSWER
2135 REM
2140 REM..675-685 SCORING FOR 605-670
2145 REM
2150 REM..700-765 ENGLISH CARD/FRENCH            ANSWER
2155 REM
2160 REM..770-790 SCORING FOR 700-765
2165 REM
2170 REM..800-820 PRINT FINAL SCORE
2175 REM
2180 REM..850-910 PRINT CORRECT ANSWER
2185 REM
2190 REM..1000-1030SET UP FOR FRENCH             OR ENGLISH CARDS
2195 REM
2200 REM..1050 WAIT ROUTINE
2205 REM
2210 REM..1210-1240 TRY AGAIN?
2215 REM
2220 REM..1300-1340 FRENCH ANTHEM?
2225 REM
2230 REM..1350-1400 DATA STATEMENTS
2240 REM
2245 REM*******************************
2250 REM
2255 REM ..IMPROVEMENTS....
2260 REM..INCLUDE LE/LA
2265 REM
2270 REM..EASY TO MAKE GERMAN/ITALIAN            BY MODIFYING PROGRA
M
```

LOGIC ANALYSER

a.jones



This program is designed to emulate logic circuitry and after entering the required information will analyse and print the results. At any time the operator can delete or add to the circuit.

Up to 20 GATES (AND, NAND OR, NOR, XOR, XNOR OR INVERTERS) with up to 8 INPUTS can be entered with the program in its present state. Open collecter devices with outputs ANDED together are not catered for, but can be simulated with an AND gate where the outputs are connected together.

A facility is also incorporated to save and to load a circuit for future reference.

Please refer to the example circuit of a FULL ADDER to understand the operation of entering the required information. After a little practice a circuit can be entered in a very short time.

The method of entering a circuit is very easy. When the program is RUN a list of options is printed. The first thing to do is enter 1 for a NEW CIRCUIT.

After a short delay the screen is cleared and the prompt ENTER LABEL is printed on the screen, in reply type 1 and Return.

TYPE OF GATE is now printed on the screen, relay with AND and Return.

ENTER OUTPUT CONNECTION is now printed on the screen, reply with 3 and Return.

ENTER INPUT CONNECTION is now printed on the screen, reply with 1 and Return.

ENTER INPUT CONNECTION is now printed on the screen, reply with 2 and Return.

ENTER INPUT CONNECTION is now printed on the screen, but this time we want to enter another gate so type 0 and Return and the reply will be ENTER LABEL. Then carry on with gate 2. The entered circuit is as follows. The / represents Return.

1/AND/3/1/2/0/2/XOR/4/2/1/0/3/AND/6/4/5/0/4/XOR/8/4/5/0/5/OR/7/3/6/0/0/

Notice the extra zero at the end of the information, this is to inform the program that all the gates were entered. On receiving this the computer will return to the list of commands ready for the operator to enter the known conditions.

The operator will then enter 5 to input conditions, only known conditions are entered using the example circuit's truth table. It will be seen that only INPUTS A, B and CARRY are used.

Type 5 and Return. The computer will respond with:

TYPE 0 TO TERMINATE.

ENTER CONNECTION NO? Reply with 1 and Return.

ENTER CONDITION? Reply with 0 and Return.

ENTER CONNECTION NO? Reply with 2 and Return.

ENTER CONDITION? Reply with 1 and Return.

ENTER CONNECTION NO? Reply with 5 and Return.

ENTER CONDITION? Reply with 0 and Return.

ENTER CONNECTION NO? Reply with 0 to terminate.

A condensed version of this will be (/ indicates RETURN) 5/1/0/2/1/5/0/0/.

The computer will then analyse the circuit with the entered conditions. A message is displayed while this is being done. On completion the screen is cleared and a list of all the connections with the expected conditions will be printed, followed by an invitation to press the space bar when ready. When all the conditions have been displayed the program returns to the menu.

The operator can then enter new conditions and repeat the operation as many times as desired.

The operator can also add to the circuit or delete from the circuit. To change a gate, first delete the label then enter the new gate and connections. The circuit can also be listed with the label at left, then type of gate followed by connection numbers.

If the operator comes out of the program, i.e. by pressing Return without entering any data, normal operation can be resumed with GOTO 20, without loss of any entered information.

Many error traps have been incorporated into the program to make it user friendly, and an audible beep is generated to draw attention to the fact that an error has been made, followed by a message to indicate the nature of the error.

Most of the program is easy to follow, and there are REMS in many places to indicate each section. The circuit is held in array A and logic conditions in array B. If the user wishes to change the number of gates or inputs then variables X and S should be changed in line 10. S is = to 1 for label + 1 for gate +1 for output + number of inputs which in this program is 8, total 11. X = S times 20 (no. of gates), so X = 220. The rest of the program will act on these two variables and nothing else requires changing.

The circuit is saved in a data file and is speeded up by first stringing the array before saving, then when the circuit is reloaded the string is re-assembled into the array.

The analysis makes three passes to allow for feedback (errors in the order of labels). Almost 7K is required to run the program. I could go on to give an explanation of how the rest of the program works but I don't think it is really needed. If it is required then I shall send one at your request.

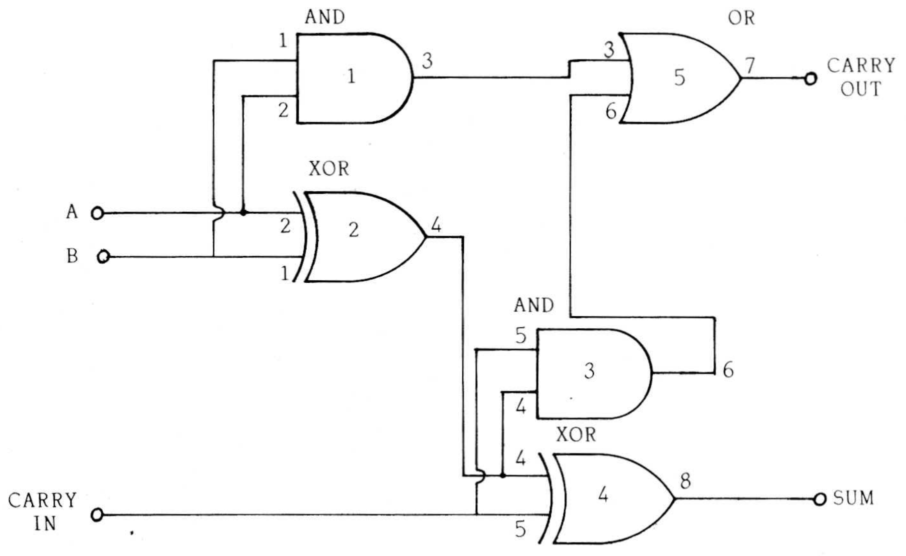
I have found that this program is a useful teaching and learning aid in the use of logic gates. It is also handy for testing a circuit before actually constructing it.

TRUTH TABLE

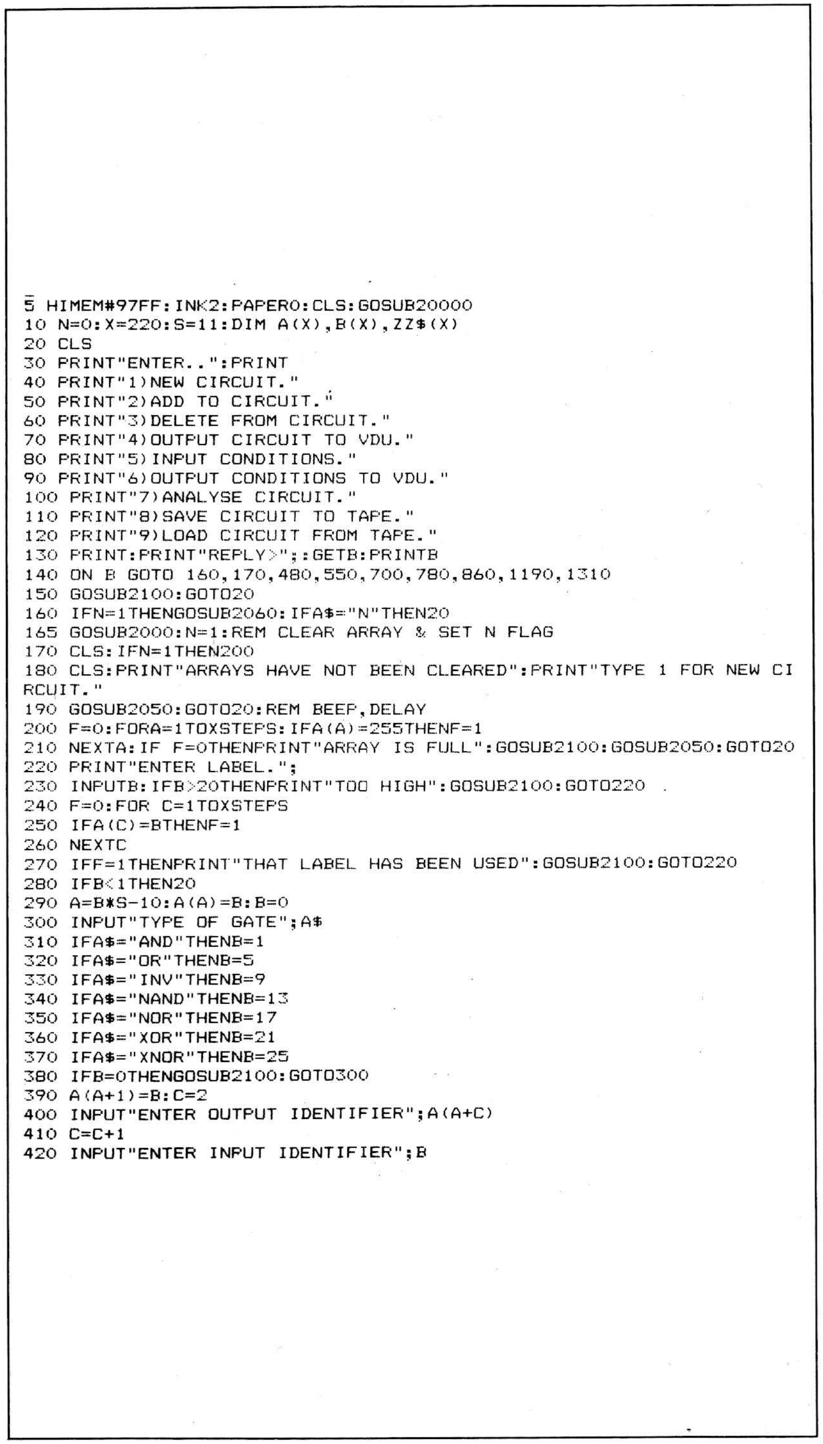
A	B	CARRY IN	CARRY OUT	SUM
Ø	Ø	Ø	Ø	Ø
Ø	1	Ø	Ø	1
1	Ø	Ø	Ø	1
1	1	Ø	1	Ø
Ø	Ø	1	Ø	1
Ø	1	1	1	Ø
1	Ø	1	1	Ø
1	1	1	1	1

THE FULL ADDER CIRCUIT

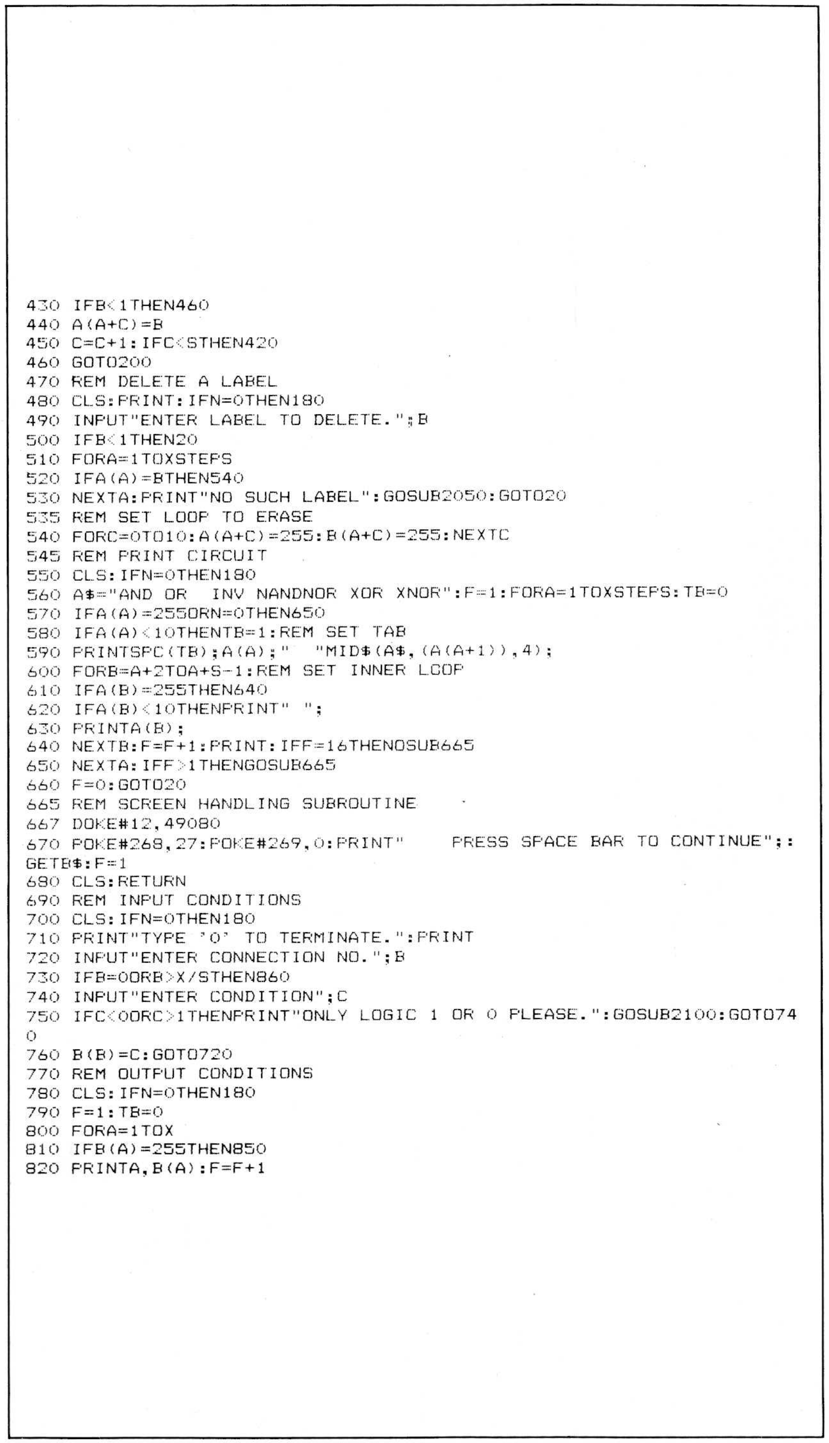
Used as the example in program.



Note that the numbers are connection numbers, not pin numbers.



```
5 HIMEM#97FF:INK2:PAPERO:CLS:GOSUB20000
10 N=0:X=220:S=11:DIM A(X),B(X),ZZ$(X)
20 CLS
30 PRINT"ENTER..":PRINT
40 PRINT"1)NEW CIRCUIT."
50 PRINT"2)ADD TO CIRCUIT."
60 PRINT"3)DELETE FROM CIRCUIT."
70 PRINT"4)OUTPUT CIRCUIT TO VDU."
80 PRINT"5)INPUT CONDITIONS."
90 PRINT"6)OUTPUT CONDITIONS TO VDU."
100 PRINT"7)ANALYSE CIRCUIT."
110 PRINT"8)SAVE CIRCUIT TO TAPE."
120 PRINT"9)LOAD CIRCUIT FROM TAPE."
130 PRINT:PRINT"REPLY>";:GETB:PRINTB
140 ON B GOTO 160,170,480,550,700,780,860,1190,1310
150 GOSUB2100:GOTO20
160 IFN=1THENGOSUB2060:IFA$="N"THEN20
165 GOSUB2000:N=1:REM CLEAR ARRAY & SET N FLAG
170 CLS:IFN=1THEN200
180 CLS:PRINT"ARRAYS HAVE NOT BEEN CLEARED":PRINT"TYPE 1 FOR NEW CI
RCUIT."
190 GOSUB2050:GOTO20:REM BEEP,DELAY
200 F=0:FORA=1TOXSTEPS:IFA(A)=255THENF=1
210 NEXTA:IF F=0THENPRINT"ARRAY IS FULL":GOSUB2100:GOSUB2050:GOTO20
220 PRINT"ENTER LABEL.";
230 INPUTB:IFB>20THENPRINT"TOO HIGH":GOSUB2100:GOTO220
240 F=0:FOR C=1TOXSTEPS
250 IFA(C)=BTHENF=1
260 NEXTC
270 IFF=1THENPRINT"THAT LABEL HAS BEEN USED":GOSUB2100:GOTO220
280 IFB<1THEN20
290 A=B*S-10:A(A)=B:B=0
300 INPUT"TYPE OF GATE";A$
310 IFA$="AND"THENB=1
320 IFA$="OR"THENB=5
330 IFA$="INV"THENB=9
340 IFA$="NAND"THENB=13
350 IFA$="NOR"THENB=17
360 IFA$="XOR"THENB=21
370 IFA$="XNOR"THENB=25
380 IFB=0THENGOSUB2100:GOTO300
390 A(A+1)=B:C=2
400 INPUT"ENTER OUTPUT IDENTIFIER";A(A+C)
410 C=C+1
420 INPUT"ENTER INPUT IDENTIFIER";B
```



```
430 IFB<1THEN460
440 A(A+C)=B
450 C=C+1:IFC<STHEN420
460 GOTO200
470 REM DELETE A LABEL
480 CLS:PRINT:IFN=0THEN180
490 INPUT"ENTER LABEL TO DELETE.";B
500 IFB<1THEN20
510 FORA=1TOXSTEPS
520 IFA(A)=BTHEN540
530 NEXTA:PRINT"NO SUCH LABEL":GOSUB2050:GOTO20
535 REM SET LOOP TO ERASE
540 FORC=0TO10:A(A+C)=255:B(A+C)=255:NEXTC
545 REM PRINT CIRCUIT
550 CLS:IFN=0THEN180
560 A$="AND OR  INV NANDNOR XOR XNOR":F=1:FORA=1TOXSTEPS:TB=0
570 IFA(A)=255ORN=0THEN650
580 IFA(A)<10THENTB=1:REM SET TAB
590 PRINTSPC(TB);A(A);"  "MID$(A$,(A(A+1)),4);
600 FORB=A+2TOA+S-1:REM SET INNER LOOP
610 IFA(B)=255THEN640
620 IFA(B)<10THENPRINT" ";
630 PRINTA(B);
640 NEXTB:F=F+1:PRINT:IFF=16THENOSUB665
650 NEXTA:IFF>1THENGOSUB665
660 F=0:GOTO20
665 REM SCREEN HANDLING SUBROUTINE
667 DOKE#12,49080
670 POKE#268,27:POKE#269,0:PRINT"    PRESS SPACE BAR TO CONTINUE";:
GETB$:F=1
680 CLS:RETURN
690 REM INPUT CONDITIONS
700 CLS:IFN=0THEN180
710 PRINT"TYPE '0' TO TERMINATE.":PRINT
720 INPUT"ENTER CONNECTION NO.";B
730 IFB=0ORB>X/STHEN860
740 INPUT"ENTER CONDITION";C
750 IFC<0ORC>1THENPRINT"ONLY LOGIC 1 OR 0 PLEASE.":GOSUB2100:GOTO74
0
760 B(B)=C:GOTO720
770 REM OUTPUT CONDITIONS
780 CLS:IFN=0THEN180
790 F=1:TB=0
800 FORA=1TOX
810 IFB(A)=255THEN850
820 PRINTA,B(A):F=F+1
```

```
830 IF F=16THENTB=TB+7:F=1
840 IFTB=28THENGOSUB670:F=1:TB=0
850 NEXTA:GOSUB670:F=0:TB=0:GOTO20
855 REM ANALYSE CIRCUIT
860 IFN=0THEN180
870 CLS:PLOT12,12,"ANALYSING CIRCUIT"
880 FORZ=1TO3:FORA=1TOXSTEPS:IFA(A)=255THEN1020
890 IFA(A+1)=1ORA(A+1)=13THEN940
900 IFA(A+1)=5ORA(A+1)=17THEN1030
910 IFA(A+1)=21ORA(A+1)=25THEN1090
920 IFA(A+1)=9THEN1160
930 GOTO1020
940 F=0:FORA1=A+3TOA+S-2:REM AND NAND
950 IFA(A1+1)=255ORF=1THEN970
960 C=B(A(A1))ANDB(A(A1+1)):IFC=0THENF=1
970 NEXTA1
980 IFA(A+1)=1THEN1010
990 IFC=1THENC=0:GOTO1010
1000 C=1
1010 B(A(A+2))=C
1020 NEXTA,Z:GOTO780
1025 REM OR NOR
1030 F=0:FORA1=A+3TOA+S-2
1040 IFA(A1+1)=255ORF=1THEN1060
1050 C=B(A(A1))ORB(A(A1+1)):IFC=1THENF=1
1060 NEXTA1
1070 IFA(A+1)=5THEN1010
1080 GOTO990
1085 REM XOR XNOR
1090 F=0:FORA1=A+3TOA+S-2
1100 IFA(A1+1)=255ORF=1THEN1130
1110 IF B(A(A1))=B(A(A1+1))THENC=0:GOTO1130
1120 C=1:F=1:REM SET DEFAULT FLAG
1130 NEXT A1
1140 IFA(A+1)=21THEN1010
1150 GOTO990
1155 REM INV
1160 IFB(A(A+3))=1THENB(A(A+2))=0:GOTO1020
1170 B(A(A+2))=1
1180 GOTO1020
1185 REM SAVE TO TAPE
1190 CLS:IFN=0THENPRINT"THERE IS NOTHING TO SAVE.":GOSUB2050:GOTO18
0
1200 PRINT"READY TAPE RECORDER."
1230 PRINT:PRINT"PRESS SPACE TO START",:GETC$
1250 POKE#67,0:REM SET FAST SPEED
1270 FORA=1TOXSTEPS:A$=""
1280 FORA1=0TOS-1
1290 A$=A$+STR$(A(A+A1))+"+":NEXTA1:REM STRING TOGETHER
1300 ZZ$(A)=A$:NEXTA:CALL1024,ZZ$:PRINT"CIRCUIT SAVED.":GOSUB2050:G
OTO20
```

```
1305 REM LOAD FROM TAPE
1310 CLS:GOSUB2000:N=1
1320 PRINT"READY TAPE":PRINT"PRESS SPACE TO START LOADING":GET C$
1340 POKE#67,0:REM SET FAST
1360 PRINT"LOADING":CALL1027,ZZ$
1363 PRINT:PRINT"RE-CREATING ARRAYS"
1365 FORA=1TOXSTEPS
1370 A$=ZZ$(A):G=1:H=1:E=1:A1=0
1380 IFMID$(A$,H,1)<>"+"THENH=H+1:E=E+1:GOTO1380
1390 QQ$=MID$(A$,G,E-1):A(A+A1)=VAL(RIGHT$(QQ$,LEN(QQ$)-1)):A1=A1+1

1400 IFA1>S-1THEN1420
1410 H=H+1:G=H:E=1:GOTO1380
1420 NEXTA:GOTO20
1999 END
2000 CLS:PRINT:PRINT:PRINT"    SETTING UP ARRAYS."
2010 FORA=1TOX:A(A)=255:B(A)=255:NEXT:RETURN
2050 GOSUB2100:WAIT100:RETURN
2060 PRINTCHR$(12)"YOU HAVE ASKED FOR A NEW CIRCUIT":PRINT"IS THIS
CORRECT";
2070 PRINT"ENTER Y/N AND PRESS RETURN.":GOSUB2100:INPUTA$:RETURN
2100 PING:WAIT50:RETURN
3999 END
20000 REM ARRAY SAVER
20005 A=#B800:READD$
20007 PRINT:PRINT:PRINT:PRINT:PRINT:PRINTCHR$(4)"                  "CHR
$(27)"JLOGIC"CHR$(4)
20008 PRINT:PRINT:PRINT:PRINT:PRINT"Please Wait...."
20010 FORI=1TOLEN(D$)STEP2
20020 V=VAL("#"+MID$(D$,I,2)):POKEA,V:A=A+1:NEXT
20030 READD$:IFD$<>"Z"THEN20010
20040 DOKE#400,#064C:DOKE#402,#4CB8:DOKE#404,#B82A:RETURN
20050 DATA2339233232552O9CB8082063B820BAE6A92520C6E5A53320C6E5A5342
0C6E52083
20060 DATAB820A7E52004E82860209CB8082063B82096E62030E6C925D0F92030E
685332030
20070 DATAE68534A002B1CEC533C8B1CEE534B0062004E84C83C42083B820EBE42
004E82860
20080 DATA20CAE6A940852B20A9B8A900852BA003B1CEAA88B1CEE901B001CA853
386346018
20090 DATAA5CE65338561A5CF65348562A004B1CE20F6D1855F84606020E800C92
CF0034CE4
20100 DATACF4CE200A20020E800862785B420E8002086D1B0062004E84CE4CFA20
086288629
20110 DATA20E20090052086D1900BAA20E20090FB2086D1B0F6C924D006A9FF852
8D00CC925
20120 DATAD00FA980852905B485B48A0980AA20E20086B5A69EA59F86CE85CFC5A
1D004E4A0
20130 DATAF01FA000B1CEC8C5B4D006A5B5D1CEF00EC8B1CE1865CEAAC8B1CE65C
F90D73860
20140 DATA2004E8A22A4C85C4
20150 DATAZ
```

PONTOON

The game needs very little explanation, most people are familiar with Pontoon as a card game. This particular program demonstrates the excellent graphic and colour capabilities available on the Oric, when in HIRES mode.

The program is totally BASIC and uses approximately 5.2K Bytes of memory. The main control area is between lines 30 and 210. This section of the program checks for pontoon, current total no greater than 21 (i.e. bust), controls screen position for display of cards and checks all critical variables. Assuming conditions are correct, it will direct the program to various locations between lines 1000 and 6070. This area is made up totally of sub-routines to generate card, display card, display patterns, calculate new totals etc. The final portion, lines 9000 to 10100, is only really accessed during the first run of the program, to initialise certain variables, create characters and shuffle the cards.

Detailed breakdown

10–20 Initialise variables and go to sub-routine to display instructions, create graphic characters and shuffle cards.

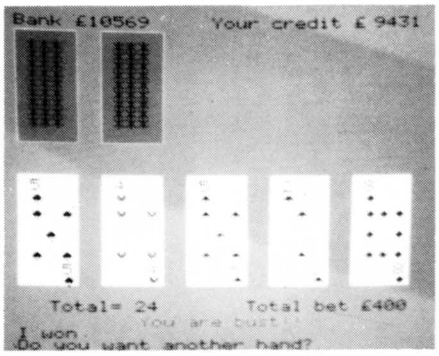
30–90 Set or re-set variables and display background, reverse of banker's cards and initial credit.

100–210 Display your cards and check various conditions.

1000–1070 Sub-routine – Decide on additional card required and bet or twist, includes 'mug' traps. Calculate new values as a result of any bet.

1200–1290 Sub-routine – decide the winner, display results and request if further hand required.

2000–2060 Sub-routine – Selection of banker's cards.



```
Bank £10569          Your credit £ 9431

     Total= 24             Total bet £400
               You are bust!!
I won
Do you want another hand?
```

3000–3080 Select next card from pack, determine its value and make all 'picture' cards 10.

3130–3140 Add card value to existing total.

3200–3240 Display the selected card with value only.

3300–3380 Decide on card pattern, i.e. number of clubs, hearts, diamonds or spades, and layout of selected pattern.

3400–3460 Set variables for display of new totals.

4000–4020 Display characters as specified in 3300 to 3380.

4100–4130 Display results as specified in 3400–3460.

4200–4230 Display graphic picture if Jack, Queen or King selected.

6000–6070 Sub-routine to display rear of banker's card.

9000–9120 First page display of instructions.

9130–9260 Second page display of instructions – if required.

9300–9390 Create spade, club, heart, diamond and single character with 10.

9400–9460 Sub-routine to shuffle 52 cards.

9500–9600 Initialise PC array with data for drawing 'pictures'.

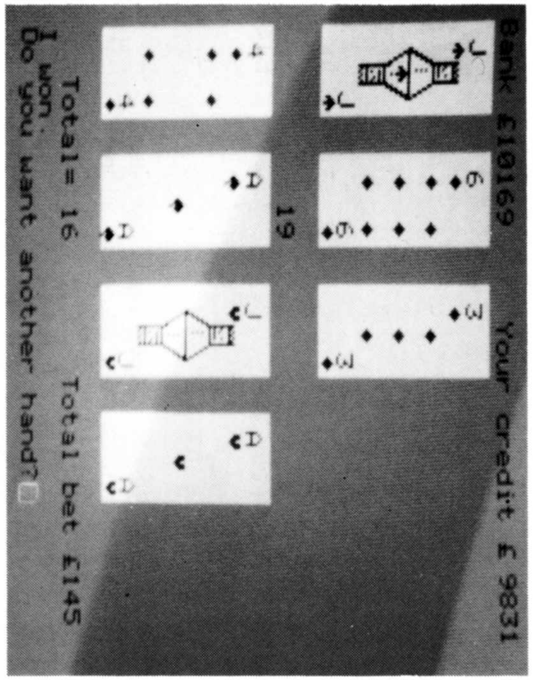
10000–10100 Response sub-routine.

Variables

BET	Current amount bet on last card.
BK	Background colour for banker's card.
BT$	String of BET for printing in HIRES.
C	Card count – up to 52.
CC	Temporary card count – each hand.
CH	Characters for banker's cards.
CO	Colour of suit.
DA	Data for creation of new characters.
E$	Escape variable.
EE$	Prefix for 'bust' statement.
H	Horizontal position.
I	Increment counter in FOR — NEXT loops.
J	Secondary increment counter.
MW	My winnings – banker's credit.
N	Current card value.
NN	Current card variable.
N(C)	Variable for each card in pack.
P	Position/location of new characters in ASCII table.
PC(I)	Variable for 'picture' data.
S	Flag indicating reshuffle required.
SU	Variable for each card suit.
T	Total score in current hand.
T$	Total score in current hand.
T$	String of T for printing in HIRES.
TA	Alternative total when Ace in hand.
TA$	String of TA for printing in HIRES.
TB	Total bet on current hand.
TB$	String of TB for printing in HIRES.
V	Vertical position.
W$	Writing/printing variable for HIRES.
X	Start horizontal position for cards.
X$	Response variable.
XX	Horizontal position relative to X.
Y	Start vertical position
YT	Your total – current hand score.
YW	Your winnings.
YY	Vertical position relative to Y.

Program Listing

```
10 TEXT:E$=" "+CHR$(27):PAPER7:DIMN(52),PC(102):GOSUB9000:YW=10000:
MW=10000
20 PAPER2:INK0
30 IFS=1THENGOSUB9400:S=0:C=0
40 PRINTFRE(""):HIRES:PAPER2:CC=0:BET=0:TB=0:TB$="":E$=" "+CHR$(27)
50 IFMW<=0THEN10050
60 EE$="You are":INK0:POKE618,2:IFYW<=0THEN10050
70 H=12:V=10:W$="Bank _":GOSUB4120:H=108:W$="Your credit _":GOSUB41
20
80 H=28:V=190:W$="Total=":GOSUB4120:H=126:W$="Total bet _":GOSUB412
0
90 GOSUB3440:CH=42:BK=1:GOSUB6000
100 X=15:Y=110:GOSUB3000:GOSUB1000
110 X=X+42:GOSUB3000
120 IFT>21THENCLS:PRINTE$"L"E$"A       ";EE$;" bust!!":GOSUB1200:GOT
O30
130 IFCC=5ANDYT=0THEN2000
140 IFCC=5THENT=21
150 IFYT>0THEN180
160 POKE618,3:GOSUB1000:IFX$="N"THEN2000
170 POKE618,2:GOTO110
180 IFT>17ORTA>19ORCC=5THENGOTO200
190 GOTO110
200 IFTA>TTHENT=TA
210 CLS:GOSUB1200:GOTO30
1000 IFCC=1THENGOTO1040
1010 CLS:PRINT"Do you want to select another card?";
1020 GETX$:IFX$="N"THENPOKE618,2:RETURN
1030 IFX$<>"Y"THEN1020
1040 PRINT:INPUT"Input bet value or T for twist";BT$:IFBT$="T"THENR
ETURN
1050 BET=VAL(BT$):IFBET>YWTHENCLS:PRINT"You only have _"YW"left.":G
OTO1040
1060 IFBET<0THENPRINT"No minus quantities allowed."CHR$(11):GOTO104
0
1070 YW=YW-BET:TB=TB+BET:TB$=STR$(TB):POKE618,2:RETURN
1200 IFCT=1ANDCC<>5THENYT=T+1
1210 IF(YT>T)OR(T>21ANDYT>0)THENPRINT"You won":YW=YW+TB+TB:MW=MW-TB
:GOTO1230
1220 PRINT"I won.":MW=MW+TB
1230 IFT=21ANDTA>0ANDCC=2THENS=1
1240 T=0:YT=0:TA=0:TA$="":CT=0
1250 GOSUB3440
1260 PRINT"Do you want another hand?";:POKE618,3
1270 GETX$:IFX$="N"THEN10100
1280 IFX$<>"Y"THEN1270
1290 POKE618,2:RETURN
2000 CLS:PRINT"  I will select my hand.":POKE618,2:IFCC=5THENCT=1
2010 IFTA>TTHENT=TA
2020 YT=T:IFT=21ANDTA>0ANDCC=2THENS=1
2030 T=0:TA=0:CH=127:BK=0:EE$="I am":CC=0:TA$=""
2040 GOSUB6000
2050 X=15:Y=20:GOSUB3000:GOSUB4000
2060 GOTO110
3000 REM****GENERATE CARD VALUE****
3010 C=C+1:CC=CC+1:IFC=53THENC=1
3020 SU=INT((N(C)-1)/13)+1:IFSU<3THENCO=0ELSECO=1
3030 N=N(C)-(13*(SU-1)):NN=N+48:SU=SU+90
3040 IFNN=49THENNN=65
```



Bank £10169   Your credit £ 9831

19

Total= 16   Total bet £145

I won.
Do you want another hand?

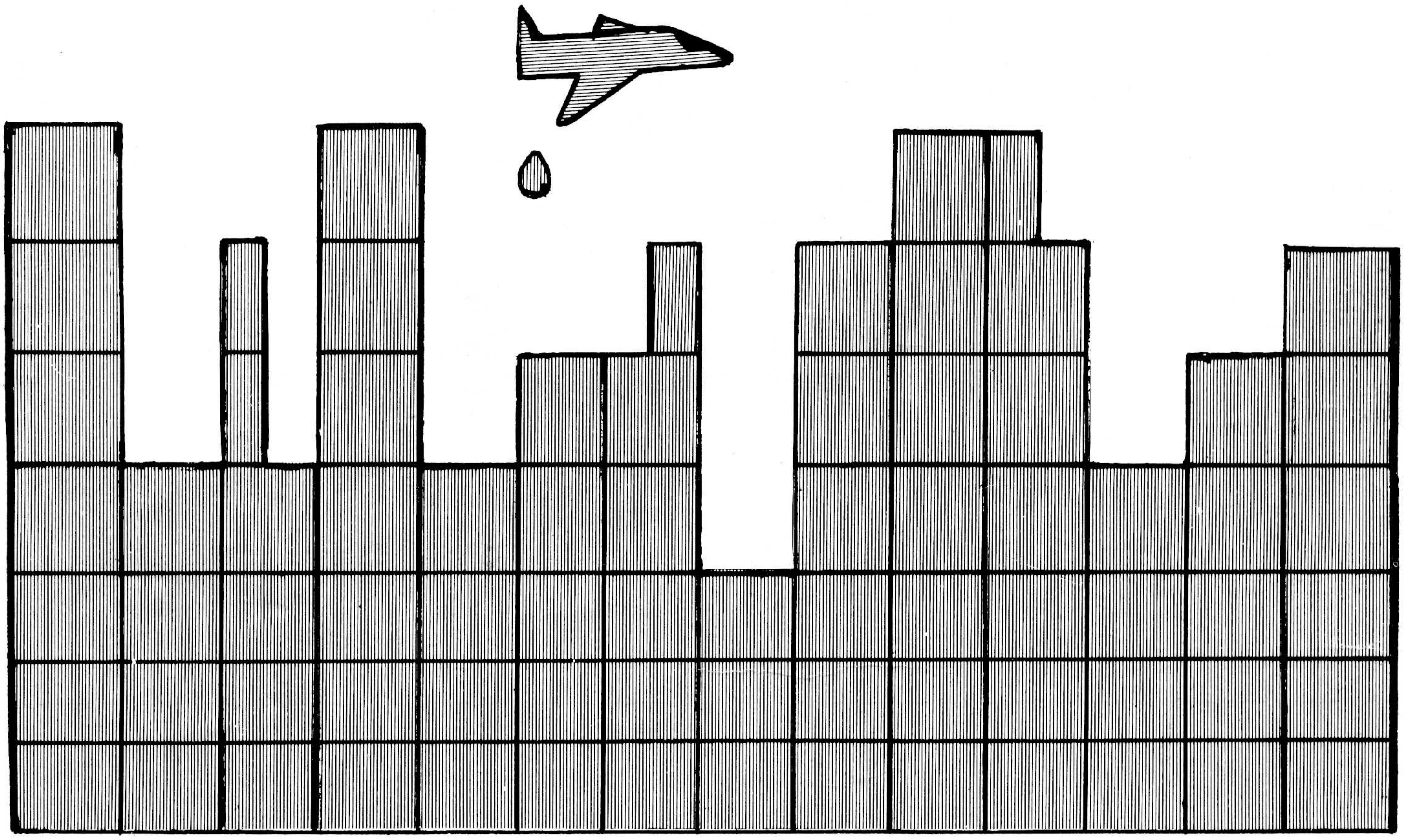
Program Listing

```
3050 IFNN=58THENNN=96:N=10
3060 IFNN=59THENNN=74:N=10
3070 IFNN=60THENNN=81:N=10
3080 IFNN=61THENNN=75:N=10
3130 T=T+N:T$=STR$(T):IFNN=65ORTA>0THENTA=T+10:IFTA>21THENTA=0:TA$=
"      "
3140 IFTA>0THENTA$=STR$(TA):TA$=" or "+RIGHT$(TA$,LEN(TA$)-1)
3200 REM****GENERATE CARD****
3210 CURSETX,Y,3:FILL70,1,23:CURSETX+30,Y,3:FILL70,1,18
3220 CURSETX+3,Y+3,3:CHARNN,0,1:CURSETX+19,Y+56,3:CHARNN,0,1
3230 CURSETX+3,Y+12,3:CHARSU,0,1:CURSETX+19,Y+64,3:CHARSU,0,1
3240 CURSETX-5,Y,3:FILL70,1,CO
3300 REM****SELECT CARD DISPLAY****
3310 IFNN>65ANDNN<82THENGOSUB4200:GOTO3400
3320 IFN/2<>INT(N/2)THENXX=10:YY=35:GOSUB4000
3330 IFN=2ORN=3ORN=8ORN=9THENXX=10:YY=22:GOSUB4000:YY=48:GOSUB4000
3340 IFN>5THENXX=3:YY=35:GOSUB4000:XX=18:GOSUB4000
3350 IFN>3THENXX=3:YY=22:GOSUB4000:YY=48:GOSUB4000
3360 IFN>3THENXX=18:GOSUB4000:YY=22:GOSUB4000
3370 IFN=10THENXX=10:YY=12:GOSUB4000:YY=25:GOSUB4000:YY=38:GOSUB400
0
3380 IFN=10THENYY=51:GOSUB4000
3400 H=70:V=Y+80:W$=T$+TA$:GOSUB4100
3410 IFTB$=""ORYT>0THEN3430
3420 H=192:W$=TB$:GOSUB4100
3430 IFYT>0THENRETURN
3440 H=48:V=10:W$=STR$(MW):GOSUB4100
3450 H=190:W$=STR$(YW):GOSUB4100
3460 RETURN
4000 REM****DISPLAY CHARACTERS****
4010 CURSETX+XX,Y+YY,3;CHARSU,0,1
4020 RETURN
4100 REM****DISPLAY CARACTERS****
4110 W$=RIGHT$(W$,LEN(W$)-1)
4120 CURSETH,V,3:FORI=1TOLEN(W$):CHAR127,0,0:CHARASC(MID$(W$,I,1)),
0,1
4130 CURMOV6,0,3:NEXT:CHAR127,0,0:RETURN
4200 CURSETX+10,Y+14,3
4210 FORI=1TO62STEP2:DRAWPC(I),PC(I+1),1:NEXT
4220 FORI=63TO102STEP2:CURSETX+PC(I),Y+PC(I+1),1:NEXT
4230 RETURN
6000 REM****GENERATE BANKERS CARDS****
6010 X=15:Y=20:FB=20
6020 CURSETX,Y',3:FILL70,1,FB:CURSETX+30,Y,3:FILL70,1,18
6030 FORI=X+3TOX+15STEP6:FORJ=Y+6TOY+54STEP6
6040 CURSETI,J,3:CHARCH,0,BK
6050 NEXTJ:NEXTI
6060 X=X+42:IFX=99THENRETURN
6070 GOTO6020
9000 PRINTCHR$(12):PRINTCHR$(4)E$"N"E$"A      ****PONTOON****"CHR$(4
):INK4
9010 POKE618,2:PRINT:PRINT:PRINT"  This program simulates the card"
9020 PRINT"game of Pontoon.":PRINT:PRINT"  The computer will ask yo
u to ";
9030 PRINT"first  select the amount of money you wish   to bet on t
he ";
9040 PRINT"first card, followed by a choice of:-"CHR$(10)"a)  betti
ng mo";
9050 PRINT"re money                  or b)"E$"ET"E$CHR$(8)"Dwisting
9060 PRINT"on subsequent cards.":PRINT:PRINT"  The program will all
ow for";
9070 PRINT" Aces      high or low, and 5 card tricks."
9080 PRINT:PRINT"  When responding to a question with  a Yes or No,
 it is";
9090 PRINT" only necessary to  press Y or N.   All other keyboard "
9100 PRINT"inputs must be followed by one press  of the RETURN key.
"
```

```
9110 PRINT:PRINTE$"A Press X for further explanation"
9120 PRINTE$"A   or Space bar to start game.":GOTO9300:GOTO1000
9130 PRINTCHR$(12):PRINTCHR$(4)E$"N"E$"A      ****PONTOON****"CHR$(4
):INK4
9140 PRINT:PRINT:PRINT"  The object of the game is to beat   the ba
nk by ";
9150 PRINT"achieving a higher score, up to a maximum of 21, in each
 hand."
9160 PRINT:PRINT"  You each start with a kitty of      _10,000.  Th
e ";
9170 PRINT"computer is the Bank,   and he does not 'see' your hand
until"
9180 PRINT"his total is as close to 21 as he can get, without going
"E$;
9190 PRINT"Abust!":PRINT:PRINT"  If a pontoon is dealt, ie a pictur
e card";
9200 PRINT" or a 10 with an ace, then the    pack will be re-shuffl
ed at ";
9210 PRINT"the end   of the hand."
9220 PRINT:PRINT" With an ace in the hand, the display will give tw
o ";
9230 PRINT"alternative results if  the totals are below 22. You may
 elect";
9240 PRINT"to draw another card, if not the      higher of the two
scores";
9250 PRINT" is recorded.":PRINT
9260 PRINTE$"APress Space bar to start the game":GOTO10000
9300 REM****CREATE CHARACTERS****
9310 P=91
9320 FORI=0TO7:READDA:POKE46080+(P*8)+I,DA:NEXT
9330 READP:IFP=0THEN9500
9340 GOTO9320
9350 DATA4,14,31,21,4,0,0,0,92
9360 DATA4,14,31,31,10,4,4,0,93
9370 DATA27,27,31,14,4,0,0,0,94
9380 DATA4,14,31,14,4,0,0,0,96
9390 DATA0,38,41,41,41,41,38,0,0
9400 REM****SHUFFLE****
9410 TEXT:CLS:POKE616,10:PRINT:PRINT"   The cards must be re-shuffl
ed."
9420 PRINTSPC(10)"Please wait."
9430 FORI=1TO52:N(I)=I:NEXT
9440 FORI=1TO52:D=INT(RND(1)*52)+1:DD=N(I)
9450 N(I)=N(D):N(D)=DD:NEXTI:C=0
9460 RETURN
9500 FORI=1TO102:READPC(I):NEXT:GOTO10000
9510 DATA1,1,1,-1,1,1,1,-1,1,1,1,-1
9520 DATA0,2,-6,0,0,-2,0,10,6,0
9530 DATA0,-8,0,8,5,10,-16,0,5,-10
9540 DATA-5,10,5,10,6,0,5,-10,-5,10
9550 DATA0,10,-1,-1,-1,1,-1,-1,-1,1
9560 DATA-1,-1,-1,1,0,-12,0,9,6,0
9570 DATA11,19,14,19,12,20,11,22,12,22,13,22,14,22
9580 DATA12,27,12,29,12,31
9590 DATA13,37,13,39,13,41,12,46,13,46,14,46,15,46
9600 DATA14,48,12,49,15,49
10000 GETX$:IFX$=" "THENGOSUB9400:RETURN
10010 IFX$<>"X"THEN10000
10020 GOTO9130
10050 TEXT:POKE616,10:PRINT:PRINT"                "EE$" skint!!"
10060 PRINT:PRINT"  Please re-run the program if you
10070 PRINT"want another game."
10100 TEXT:END
```

B.O.M.B RUN.

By Pat Cooper



Here's a program for all budding pilots. It's a simple version of the arcade game called 'Bomber'.

The idea is to clear the screen of buildings so that you've got room to land, but you've only got a limited amount of fuel. The fuel and height of the buildings are determined by the skill level you're playing at.

After each successful landing, the skill rating is automatically increased by 1, to make things hard the next time around.

There are instructions on controlling the plane within the program, and they are self-explanatory.

Software

Program Listing

```
90 PAPER4:INK3
99 PRINTCHR$(17)
100 TEXT:RELEASE:RESTORE
110 CLS:PLOT5,10,"Do you want instructions?"
120 GETA$:IFA$="Y"THEN 300ELSE1000
130 REM
140 PLOT11,9,12:PLOT12,9,"PERFECT LANDING "
150 RESTORE
160 RETURN
170 REM
180 EXPLODE
190 PLOT9,9,"Oh Dear,you've crashed"
200 WAIT500
210 RETURN
230 REM
240 PLOT11,11,"Another Try ? (Y/N)"
245 GETA$
250 IFA$="Y"THENRETURN
260 IFA$<>"N"THEN245
270 PAPER0:INK7:CLS:PRINTCHR$(17):END
300 CLS:PRINT:PRINTSPC(15)"BOMB RUN"
310 PRINT:PRINT:PRINT"  Your Bomber is running out of fuel  and you
've got to find ";
320 PRINT"somewhere to","land."
330 PRINT:PRINT"To do this, the buildings below have  got to be dem
olished."
340 PRINT:PRINT"To increase height, press the Cursor  Up' key, but
keep an eye ";
350 PRINT"on the Fuel."
355 PRINT:PRINT"Press 'Cursor Down' key to release a  bomb."
360 PRINT:PRINT:PRINT"Skill level (1=Easy,5=Hard)";
370 INPUTA:IFA<1ORA>5THEN370ELSE1000
400 REM
410 CLS:GOSUB700
450 FORX=1TO35STEP2
460 :FORY=INT(9+(RND(1)*15-D))TO26
```

```
470 :PLOTX,Y,"[]"
480 :NEXTY:NEXTX:H=3
490 RETURN
500 REM
520 J=K+1:G=X:F=1:IFJ>26THENRETURN
530 IFSCRN(G,J)=91ORSCRN(G,J)=93THENW=W+1
540 PLOTG,J,"_"
550 RETURN
600 REM
610 PLOTG,J," "
620 IFW=2THENW=0:F=0:EXPLODE:RETURN
630 IFJ=26THENW=0:F=0:EXPLODE:RETURN
640 J=J+1:B=B+1:GOTO530
700 FORCR=1TO6:READC
705 FORCH=#B400+8*CTO#B400+8*C+7:READR:POKECH,R:NEXT:NEXT
710 DATA123,48,56,28,63,59,54,1,3'{
720 DATA124,0,0,0,63,31,62,60,48'|
730 DATA125,0,0,0,62,31,62,0,0'}
740 DATA95,0,10,14,4,14,14,14,4'_
750 DATA91,31,18,18,31,31,18,18,31'[
760 DATA93,62,18,18,62,62,18,18,62']
770 RETURN
1000 REM
1010 D=A:L=4:F=0:W=0:FL=30-INT((RND(1)*D*4+1)):GOSUB400
1020 FORX=1TO36
1030 : K=H
1040 : IFSCRN(X+2,H)<>#20ORSCRN(X,H)<>#20THENL=L-1:PLOTX+3,H," ":EX
PLODE
1050 : PLOTX,H,"{|}":PLOT1,0,2
1060 : PLOT25,0,"LIVES: ":PLOT32,0,RIGHT$("  "+STR$(L),2)
1070 : PLOT3,0,"FUEL: ":PLOT9,0,RIGHT$("  "+STR$(FL),2)
1080 : IFL=0THENGOSUB 170:GOSUB 230:GOTO100
1090 : X$=KEY$
1100 : IFX$=CHR$(10)ANDF=0THENGOSUB500ELSEWAIT(10)
1110 : IFX$=CHR$(11)ANDFL>0ANDH>3THENH=H-1:FL=FL-1
1120 : IFF=1THENGOSUB600
1140 : PLOTX,K,"   "
1150 : IFH>25ANDX=35THENPLOT35,26"{|}":GOSUB130:A=D+1:GOSUB230:GOTO
1000
1155 : NEXT
1160 H=H+1
1170 GOTO1020
```

BREAK-OUT

By Martin Laws

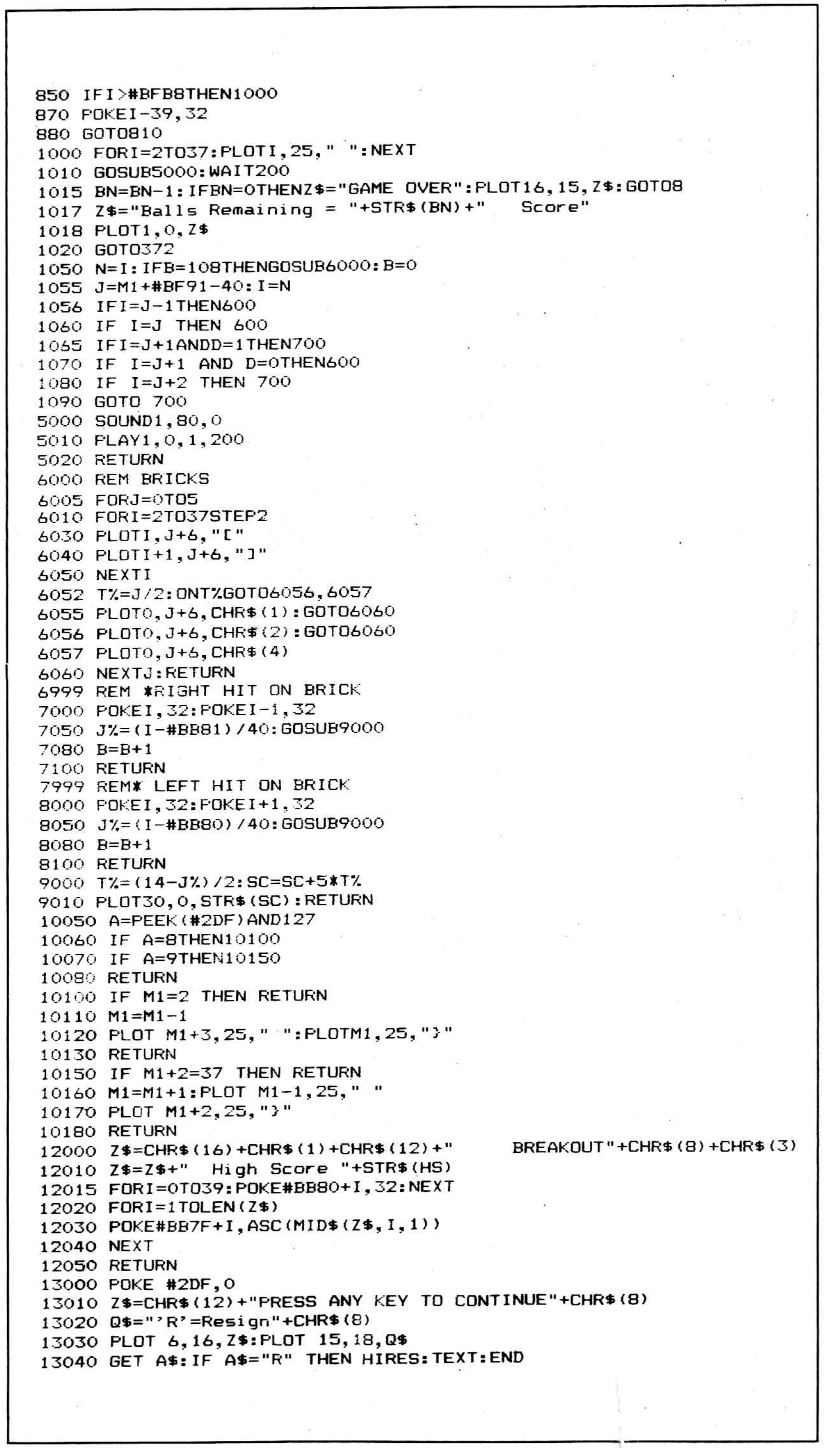
This game in Basic is a cut down version of the Arcade game. Your task is to knock the bricks out of the wall with your bat. The program is made difficult by the fact that your bat keeps drifting in the direction that you last sent it.

Although not particularly fast, this game will keep you amused for several hours and does offer plenty of scope for improvement.

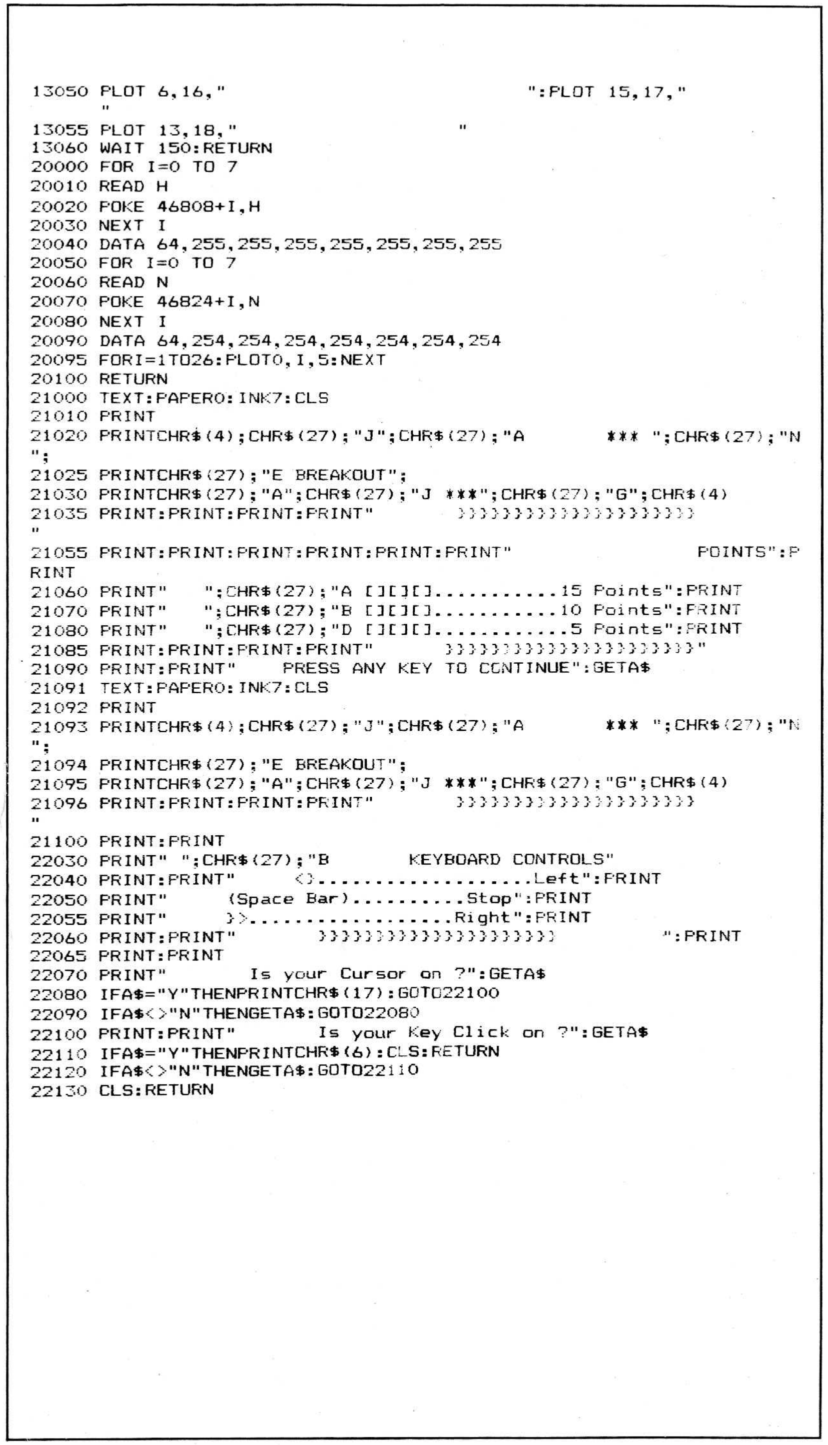
Program Listing

```
5 HS=0:SC=0
8 IFSC>HSTHENHS=SC
10 RESTORE
20 FOR I=0 TO 7:READ A
30 POKE 46840+I,A
40 NEXT I
50 DATA 64,76,94,127,127,94,76,64
60 FOR I=0 TO 7
70 READ B
80 POKE 46376+I,B
90 NEXT
100 DATA 64,64,64,124,124,76,76,76
110 FOR I=0 TO 7
120 READ C
130 POKE 46384+I,C
140 NEXT I
150 DATA 64,64,64,79,79,76,76,76
160 FOR I=0 TO 7
170 READ D
180 POKE 47080+I,D
190 NEXT I
200 DATA 64,64,64,127,127,64,64,64
210 FOR I=0 TO 7
220 READ E
230 POKE 47064+I,E
240 NEXT I
250 DATA76,76,76,76,76,76,76,76
260 FOR I=0 TO 7
270 READ F
280 POKE 46832+I,F
290 NEXT I
295 DATA 0,0,0,0,0,0,0,0:GOSUB 20000
300 IF M=0THENM=1:GOSUB21000
301 LORES0:GOSUB 12000
302 FOR I=26 TO 1 STEP -1
303 PLOT 1,I,"{"
304 NEXT I
310 PLOT 1,1,"&"
320 FOR I=2 TO 37
330 PLOT I,1,"}"
340 NEXT I
350 PLOT 38,1,"%"
360 FOR I=2 TO 26
370 PLOT 38,I,"{":NEXTI
371 GOSUB6000:PLOT0,25,CHR$(6):BN=3:SC=0:B= 0:GOSUB
        13000:GOTO 1017
```

```
372 M1=17:M2=M1+1:M3=M1+2
373 PLOT M1,25,"}"
374 PLOT M2,25,"}"
375 PLOT M3,25,"}"
385 I=48620
387 D=RND(1)+.5:ONDGOTO810
390 POKE I,95
395 GOSUB 10050
397 GOSUB10050
399 I=I+41:IFPEEK(I)<33THEN440
400 GOSUB5000
402 IF PEEK(I)=123 THEN I=I-41:GOTO810
405 IF PEEK(I)=125 THEN I=I-41:D=1:GOTO1050
410 IFPEEK(I)=91THENGOSUB8000:I=I-41:GOTO700
420 IFPEEK(I)=93THENGOSUB7000:I=I-41:GOTO700
440 IF I >#BFB8 THEN 1000
450 POKE I-41,32
470 GOTO 390
600 POKE I,95
605 GOSUB 10050
610 GOSUB 10050
620 I=I-41: IF PEEK(I)<33THEN680
625 GOSUB5000
630 IF PEEK(I)=125 THEN I=I+41:GOTO810
640 IF PEEK(I)=123 THEN I=I+41:GOTO700
650 IFPEEK(I)=38THENI=I+41:GOTO390
660 IFPEEK(I)=91THENGOSUB8000:I=I+41:GOTO810
670 IFPEEK(I)=93THENGOSUB7000:I=I+41:GOTO810
680 POKE I+41,32
690 GOTO 600
700 POKEI,95
710 GOSUB10050:GOSUB10050
715 I=I-39:IFPEEK(I)<33THEN740
716 GOSUB5000
720 IFPEEK(I)=125THENI=I+39:GOTO390
730 IFPEEK(I)=123THENI=I+39:GOTO600
732 IFPEEK(I)=37THENI=I+39:GOTO810
733 IFPEEK(I)=91THENGOSUB8000:I=I+39:GOTO390
734 IFPEEK(I)=93THENGOSUB7000:I=I+39:GOTO390
740 POKEI+39,32
750 GOTO700
810 POKEI,95
820 GOSUB10050:GOSUB10050
825 I=I+39:IFPEEK(I)<33THEN850
827 GOSUB5000
830 IFPEEK(I)=123THENI=I-39:GOTO390
840 IFPEEK(I)=125THENI=I-39:D=0:GOTO1050
841 IFPEEK(I)=91THENGOSUB8000:I=I-39:GOTO600
842 IFPEEK(I)=93THENGOSUB7000:I=I-39:GOTO600
```



```
850 IFI>#BFB8THEN1000
870 POKEI-39,32
880 GOTO810
1000 FORI=2TO37:PLOTI,25," ":NEXT
1010 GOSUB5000:WAIT200
1015 BN=BN-1:IFBN=0THENZ$="GAME OVER":PLOT16,15,Z$:GOTO8
1017 Z$="Balls Remaining = "+STR$(BN)+"    Score"
1018 PLOT1,0,Z$
1020 GOTO372
1050 N=I:IFB=108THENGOSUB6000:B=0
1055 J=M1+#BF91-40:I=N
1056 IFI=J-1THEN600
1060 IF I=J THEN 600
1065 IFI=J+1ANDD=1THEN700
1070 IF I=J+1 AND D=0THEN600
1080 IF I=J+2 THEN 700
1090 GOTO 700
5000 SOUND1,80,0
5010 PLAY1,0,1,200
5020 RETURN
6000 REM BRICKS
6005 FORJ=0TO5
6010 FORI=2TO37STEP2
6030 PLOTI,J+6,"["
6040 PLOTI+1,J+6,"]"
6050 NEXTI
6052 T%=J/2:ONT%GOTO6056,6057
6055 PLOT0,J+6,CHR$(1):GOTO6060
6056 PLOT0,J+6,CHR$(2):GOTO6060
6057 PLOT0,J+6,CHR$(4)
6060 NEXTJ:RETURN
6999 REM *RIGHT HIT ON BRICK
7000 POKEI,32:POKEI-1,32
7050 J%=(I-#BB81)/40:GOSUB9000
7080 B=B+1
7100 RETURN
7999 REM* LEFT HIT ON BRICK
8000 POKEI,32:POKEI+1,32
8050 J%=(I-#BB80)/40:GOSUB9000
8080 B=B+1
8100 RETURN
9000 T%=(14-J%)/2:SC=SC+5*T%
9010 PLOT30,0,STR$(SC):RETURN
10050 A=PEEK(#2DF)AND127
10060 IF A=8THEN10100
10070 IF A=9THEN10150
10080 RETURN
10100 IF M1=2 THEN RETURN
10110 M1=M1-1
10120 PLOT M1+3,25," ":PLOTM1,25,"}"
10130 RETURN
10150 IF M1+2=37 THEN RETURN
10160 M1=M1+1:PLOT M1-1,25," "
10170 PLOT M1+2,25,"}"
10180 RETURN
12000 Z$=CHR$(16)+CHR$(1)+CHR$(12)+"        BREAKOUT"+CHR$(8)+CHR$(3)
12010 Z$=Z$+"  High Score "+STR$(HS)
12015 FORI=0TO39:POKE#BB80+I,32:NEXT
12020 FORI=1TOLEN(Z$)
12030 POKE#BB7F+I,ASC(MID$(Z$,I,1))
12040 NEXT
12050 RETURN
13000 POKE #2DF,0
13010 Z$=CHR$(12)+"PRESS ANY KEY TO CONTINUE"+CHR$(8)
13020 Q$="'R'=Resign"+CHR$(8)
13030 PLOT 6,16,Z$:PLOT 15,18,Q$
13040 GET A$:IF A$="R" THEN HIRES:TEXT:END
```



```
13050 PLOT 6,16,"                                  ":PLOT 15,17,"
      "
13055 PLOT 13,18,"              "
13060 WAIT 150:RETURN
20000 FOR I=0 TO 7
20010 READ H
20020 POKE 46808+I,H
20030 NEXT I
20040 DATA 64,255,255,255,255,255,255,255
20050 FOR I=0 TO 7
20060 READ N
20070 POKE 46824+I,N
20080 NEXT I
20090 DATA 64,254,254,254,254,254,254,254
20095 FORI=1TO26:PLOT0,I,5:NEXT
20100 RETURN
21000 TEXT:PAPER0:INK7:CLS
21010 PRINT
21020 PRINTCHR$(4);CHR$(27);"J";CHR$(27);"A       *** ";CHR$(27);"N
";
21025 PRINTCHR$(27);"E BREAKOUT";
21030 PRINTCHR$(27);"A";CHR$(27);"J ***";CHR$(27);"G";CHR$(4)
21035 PRINT:PRINT:PRINT:PRINT"      }}}}}}}}}}}}}}}}}}}}}}
"
21055 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT"                    POINTS":P
RINT
21060 PRINT"   ";CHR$(27);"A [][][]...........15 Points":PRINT
21070 PRINT"   ";CHR$(27);"B [][][]...........10 Points":PRINT
21080 PRINT"   ";CHR$(27);"D [][][]............5 Points":PRINT
21085 PRINT:PRINT:PRINT:PRINT"      }}}}}}}}}}}}}}}}}}}}}}}"
21090 PRINT:PRINT"    PRESS ANY KEY TO CONTINUE":GETA$
21091 TEXT:PAPER0:INK7:CLS
21092 PRINT
21093 PRINTCHR$(4);CHR$(27);"J";CHR$(27);"A       *** ";CHR$(27);"N
";
21094 PRINTCHR$(27);"E BREAKOUT";
21095 PRINTCHR$(27);"A";CHR$(27);"J ***";CHR$(27);"G";CHR$(4)
21096 PRINT:PRINT:PRINT:PRINT"      }}}}}}}}}}}}}}}}}}}}}}
"
21100 PRINT:PRINT
22030 PRINT" ";CHR$(27);"B       KEYBOARD CONTROLS"
22040 PRINT:PRINT"     <}....................Left":PRINT
22050 PRINT"      (Space Bar)..........Stop":PRINT
22055 PRINT"      }>..................Right":PRINT
22060 PRINT:PRINT"      }}}}}}}}}}}}}}}}}}}}}}          ":PRINT
22065 PRINT:PRINT
22070 PRINT"        Is your Cursor on ?":GETA$
22080 IFA$="Y"THENPRINTCHR$(17):GOTO22100
22090 IFA$<>"N"THENGETA$:GOTO22080
22100 PRINT:PRINT"        Is your Key Click on ?":GETA$
22110 IFA$="Y"THENPRINTCHR$(6):CLS:RETURN
22120 IFA$<>"N"THENGETA$:GOTO22110
22130 CLS:RETURN
```

classifieds

FOR SALE

Small Ads – 15p a word Max 30 words

Calling all Advertisers!

The Oric Owner is the only magazine devoted to the Oric-1 computer.

Virtually every person who buys an Oric will get this magazine. If you are selling Oric related products can you afford to miss this chance to reach the market you are aiming for?

For full details of our advertising rates contact Carolyn on Ely 2271 and see if she can help you.

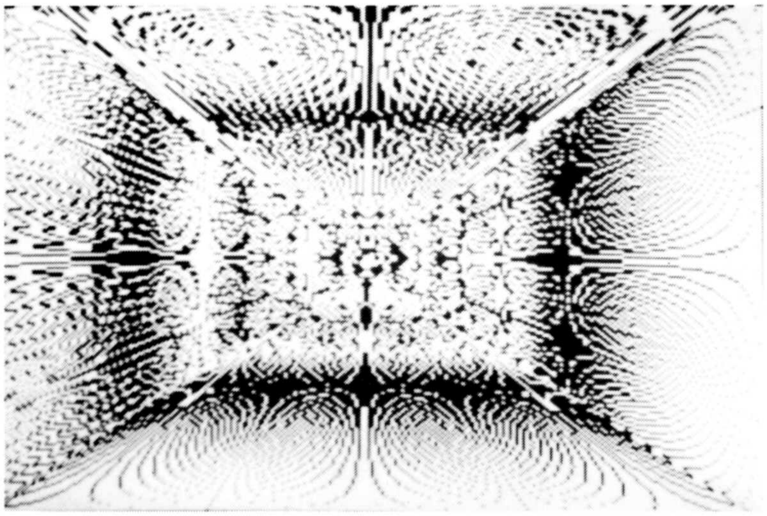
Oric Quickies

Star-Scan

This interesting routine creates the impressions of hurtling through a field of stars. A simple pattern of lines is set up on the Hi-Res screen and then made to disappear by changing the ink and paper to the same colour. By scanning a white attribute along the side of the screen the impression of movement is given. The sound command gives a rush of white noise for better effect.

```
3 REM STAR-SCAN APRIL 83
5 HIMEM#2000
10 HIRES
20 INK4:PAPER4
30 CURSET0,0,0:DRAW239,199,1
40 CURSET239,0,0:DRAW-239,199,1
50 CURSET120,0,0:DRAW0,199,1
70 FORY=99TO0STEP-3
80 CURSET7,Y,0:FILL2,1,7
85 CURSET7,99+(99-Y),0:FILL2,1,7
90 CURSET7,Y,0:FILL2,1,4
95 CURSET7,99+(99-Y),0:FILL2,1,4
100 PLAY0,3,0,0:SOUND4,Y,7
110 NEXT
120 GOTO70
```

Pattern

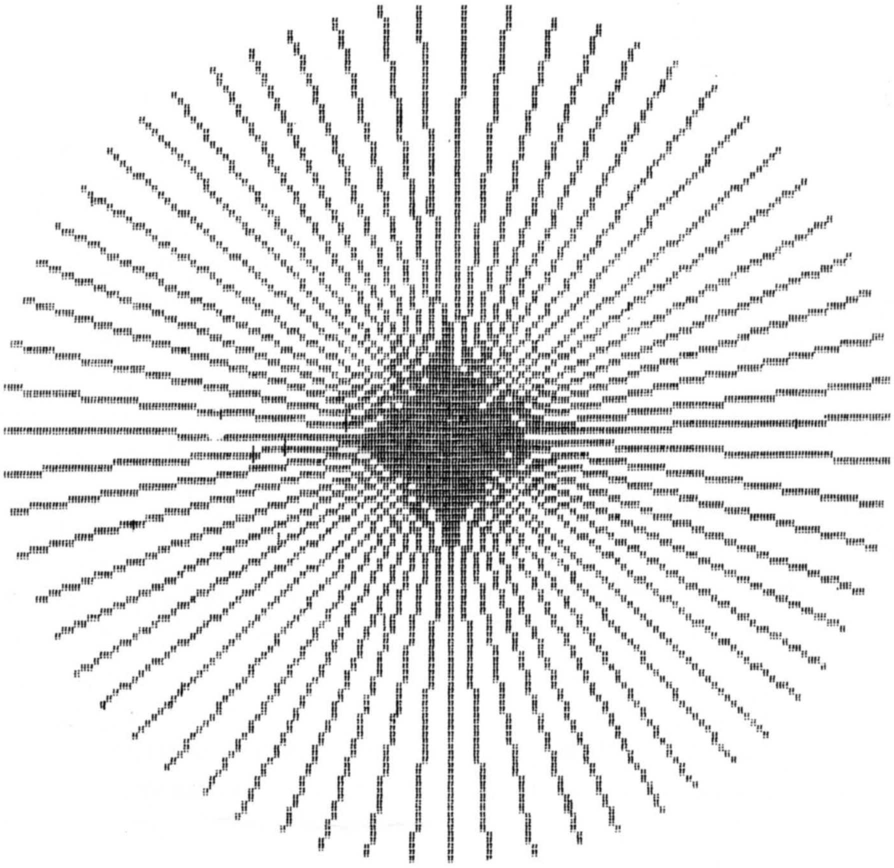


This short program displays an interesting Kaleidoscope type pattern which is then shaded in using a constantly changing mixture of colours. By only changing the colour on alternate lines pastel shades of the colours can be obtained.

To add more interest the program could be altered to display random patterns or geometric shapes.

```
10 REM PATTERN BY FRANK WOODCOCK
20 PAPER0:CLS:HIRES:PRINTCHR$(17)
30 FORA=239TO0STEP-1:CURSETA,0,3:DRAW239-A*2,199,2:NEXT
40 FORA=0TO199:CURSET0,A,3:DRAW239,199-A*2,2:NEXT
50 FORA=0TO7:PAPERA:FORB=0TO7:FORC=16TO22:IFB+16=C THEN C=C+1
60 FORD=40960TO49000STEP80:POKED,B
70 POKED+1,C:NEXTD:WAIT100:NEXTC,B,A:GOTO50
```

ORIC Epson FT III Printout



```
10 HIRES
15 PAPER3:INK1
20 REPEAT
30 CURSET120,100,1
40 DRAW 69*SIN(F),69*COS(F),1
50 F=F+.1
60 UNTIL F>2*PI
70 N$="WHAT A PRETTY PICTURE"
75 CURMOV-90,20,0
80 FORA=1TOLEN(N$)
90 CHARASC(MID$(N$,A,1)),0,1
95 CURMOV10,0,0
100 NEXTA
110 GOSUB280
```

This draws the picture as in the Oric manual.

```
120 END
280 LPRINT CHR$(27);"3";CHR$(7)
290 FORM=1TO199
300 LPRINT CHR$(27);"K";CHR$(222);CHR$(1);
305 LPRINT CHR$(13)
310 FORN=1TO239
315 A=-POINT(N,M)
317 IFA=1THENA=7ELSEA=0
320 LPRINT CHR$(A);
325 LPRINT CHR$(A);
330 NEXTN
340 LPRINTCHR$(13)
350 NEXTM
360 RETURN
```

This subroutine prints out on the Epson MX80F/TIII the entire HIRES graphics screen.

ISO\PROSE

Dear Sirs,

I heard the news, I was impressed,
They said that 'Oric' was the best.
Off with money clutched in hand,
To buy an 'Oric', ain't life grand!

They said no software could be got,
I felt quite faint, and went all hot,
Then Laskys rang me on the 'phone,
We've software here, no need to moan.

Off to Laskys, parked the car,
And walked a distance, not too far.
Bought two tapes and felt so glad,
They didn't work, which made me mad.

Back Laskys, spoke to Liz,
A charming girl who knew her 'bizz'
A change of tape from her obtained,
'Twas just as bad, so nothing gained.

Saw Mr Taylor, helpful chap,
All computers he could zap,
Went past hours, well over the top,
Someone locked us in the shop.

I ran the tape all full of glee,
It won't go in, oh, woe is me.
Then in walked son, a clever youth,
Who said in accents quite uncouth,

'Dear ancient Dad, it just won't work,
You're running it fast, you silly burke'
And then with beaming smile wide
He turned it to the other side.

I rang 'Oric' once again,
I'm sure you think I'm quite a pain,
They said I needed book and tape,
The two things didn't correlate.

They then said 'Please ring Tangerine
At Ely', please what did they mean?
I just got a high pitched howl,
Which made my temper really foul.

Then Tansoft product I did ring,
Local exchange success did bring,
A sexy voice the other end
Said all the info. she would send.

Today the postman brought the lot,
Magazine, book and list I got.
Send ten pounds at once she said,
A year's supply to read in bed.

Thanks to all who gave their time,
To cheer me up, and hence the rhyme.
Oric, Tansoft, Laskys too,
Many thanks must go to you.

I hope that now I'll get it right,
With modern things I'm not too bright,
All these prints and beeps and pokes,
Are difficult for we older folks.

Mrs C. R. Laurie
44a Lower Gravel Road, Bromley, Kent

I/O\PAGE

Dear Paul,
With reference to the reader's letter in issue 1 concerning cassette saving. Can you tell me which cassette recorders you recommend for use with the Oric.

R. Hodges, Cheltenham, Glos.

Ed.: It is very difficult to recommend one particular type of cassette recorder as most models should work properly once they have been correctly set up. For your information we use the National Panasonic slimline RQ–2734, for all our software development but I am not sure if this model is still available.

Dear Paul,
Having bought my Oric from Dixons recently I would now like to connect it to an EPSON MX80 printer. I do not feel confident enought to build my own printer lead, so can you suggest where I can get one?

A. Thompson, Reading, Berks.

Ed.: The Oric makes use of the well known centronics type printer interface. We have found that printer leads for the Apple or Dragon computers will fit the Oric. Most computer shops should have these, but be warned the price can vary from between £12 and £20, so shop around.

Why you should subscribe to ORIC OWNER

Your Oric 1 is one of the most powerful low-cost computers on the market today. Oric Owner is the only magazine totally dedicated to the Oric and its supporting hardware and software.

From issue to issue Oric Owner is able to bring you in-depth information on the Oric, advance news on the latest add-ons, superb programs to try and 'behind the scenes' interviews with the engineers who designed it.

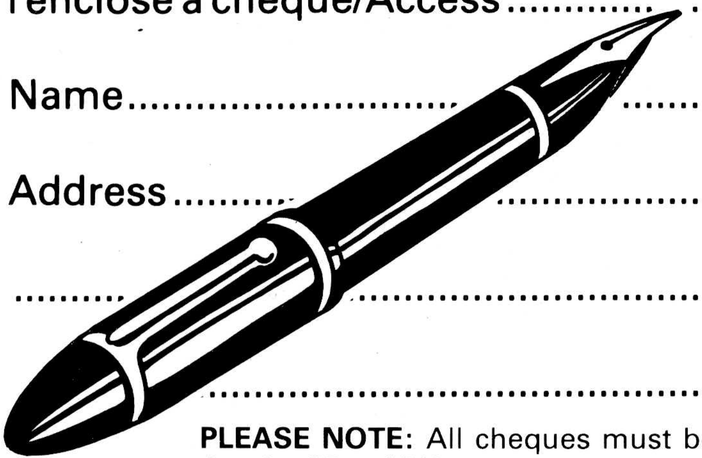
If you wish to keep ahead of the latest developments and news on the Oric then Oric Owner is essential. At only £10 for a year's subscription of 6 issues (£15 for overseas readers) can you afford to be without Oric Owner?

SUBSCRIBE TODAY!

subscription form

Please send me the next 6 issues of **ORIC OWNER**

I enclose a cheque/Access .... Barclaycard .... for £10



Name ....

Address ....

....

....

PLEASE NOTE: All cheques must be made payable to: Tansoft Ltd, 3 Club Mews, Ely, Cambs CB7 4NW

CLUB PAGE

It seems that the Oric is taking off in clubland. We've been told of two new user groups for the Oric.

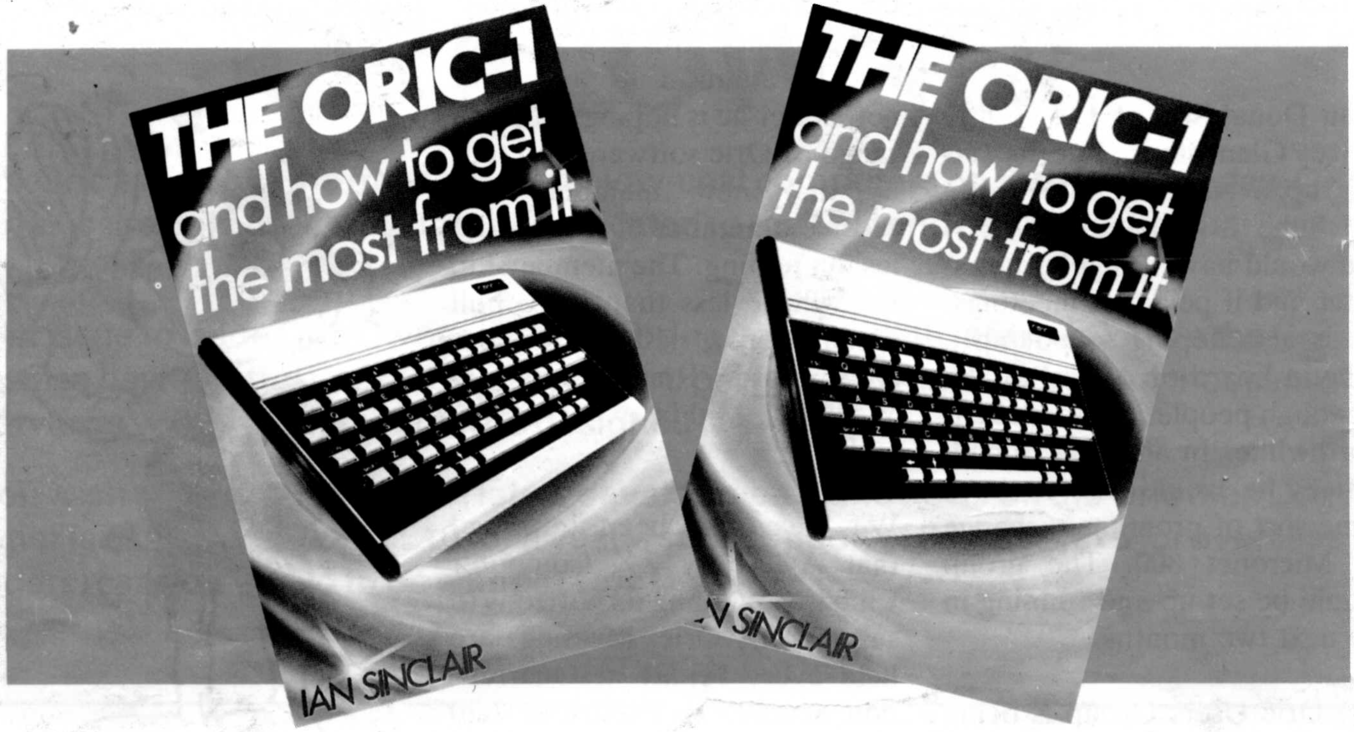
Alan Donaldson of 111 Huntly Drive, Glenrothes, Fife, is setting up a National user group which he is calling O.R.U.G. and would have a regular newsletter and if possible, meetings. He is also hoping to put up a bulletin board on Micronet 800 if enough people join to make it worthwhile. In addition to club notices he would like to have some sort of program exchange on Micronet 800. The group should be set up and running in the next two months.

The Oric Users Group is being formed by Alex Cross of M.C.I. in Wembley. He intends it to become the best user group in the United Kingdom and it is being formed with Oric's approval. In addition to a regular newsletter he is hoping to offer a range of Oric software and tells us that new members will receive a number of free 'goodies' on joining. The membership fee will be less thant £10. Full details from Oric Users Group, 5 Watford Road, Sudbury, Wembley, Middx.

We have also been informed that Aylesbury ZX Computer Club is expanding its horizons to support the Oric. Meetings are held every Friday at Quarrendon School, Aylesbury at 7.30 p.m. Further details are available from the club secretary, David Nawotnik, on Aylesbury 630867 (evenings only).

CONGRATULATIONS!

You have chosen the most remarkable small computer ever designed. Now you are anxious to get the very best out of it. but you have no knowledge of computing, or not much? Now you need this book


THE ORIC-1
and how to get the most from it
IAN SINCLAIR
THE ORIC-1
and how to get the most from it
N SINCLAIR

The book that introduces you to the ORIC–from plugging in to programming.

Here is Microsoft BASIC-programming, how to use ORIC's graphics, colour and sound systems, and how to use the data processing capabilities.

And there are lots of useful or fun programs. If you want to get the most out of the best machine, you need the best guide

THE ORIC-1

by Ian Sinclair

You can order your copy direct from Tansoft Ltd.

Send off this coupon NOW

TO: TANSOFT LTD., 3, CLUB MEWS, ELY,
CAMBS, CB7 4NW

Please rush me a copy of Ian Sinclair's new book
THE ORIC-1 AND HOW TO GET THE MOST FROM IT

*I enclose a cheque for £6.95 (for a post free copy) payable to Tansoft Ltd.

or please charge my *Access/*Barclaycard
Account number: □□□□□□□□□□□□□□□□

*Delete as relevant

Please send my copy of the book to:

Name ____________________

Address ____________________

____

____

Signature ____________________

Date ____________________

GRANADA PUBLISHING