2/7

Personal Computer World

A GUIDE TO BUSINESS SOFTWARE

US $2.00/FF8:80/LIRE 1700/DM 3.80/FL 4.00/BF 55
DKr 10.75/SKr 7.65/NKr 10.50/A Sch 28/S Fr 3.50/Pts 135

NOVEMBER 1979 50p

SEE YOU AT THE SHOW!

The 2nd Personal Computer World show~preview issue

Ask for our free colour brochure

DIAL NOW

031-225 2022

(24 HOURS ANSWERING)

MicroCentre are the UK Cromemco experts

Sample catalogue prices:

*System Two computer	1995
*System Three computer	3293
*Z-2H computer	4998
Extra 64K memory	893
3101 visual display unit	1147
3355 daisywheel printer	2297
HDD 11-mbytes hard disc	4022
ANSI Cobol compiler	55
ANSI Fortran IV compiler	55
16K extended Basic	55
Word processing system	55
Database management	55
Macro relocating assembler	55

Prices exclude VAT and delivery

MicroCentre also supply peripherals, applications sofware, and multi-user timesharing systems; a PROM programmer; analogue-digital interface; and much more. On site maintenance can be arranged throughout the UK.
*Computer systems include fast 4MHz Z80A micro, S-100 bus (21 slots), 64K memory, dual floppy discs, peripheral interfaces, etc. CP/M compatible operating system CDOS free with software.

With our in-depth experience and total commitment to the reliable Cromemco range we are Cromemco's leading UK distributor. Rely on us, as many others do, for expert support with your routine or special micro-computer applications.

Micro Centre

Complete Micro Systems Ltd.
132 St. Stephen Street,
Edinburgh EH3 5AA.
Tel: 031-225 2022.

Photo features Cromemco System 3 computer, 3101 VDU, and 3355 daisywheel printer.

Look out for us at Compec '79, stand no. 756

CONTENTS

VOLUME 2 No. 7 NOVEMBER 1979.

Founder
Angelo Zgorelec

Editor
Bruce Sawford

Technical Editor
David Tebbutt

Regular Contributors
Guy Kewney, Sheridan Williams, John Coll, Sue Eisenbach, Malcolm Peltu, Mike Knight, Dick Pountain

Consultants
John Coll, Mike Dennis, Charles Sweeten, Patrick Sutton, Michael James, R.W.Davy, David Hebditch, Sheridan Williams, Dr. Stephen Castell, Dr. D.J.Hand

Advertising Manager
Stephen England

Group Advertising Manager
Richard Howell

Production Manager
Dick Pountain

Art Director
Paul Carpenter

Art Assistants
Jimmy Egerton, Julia Davies

Cover Illustration
Hunt Emerson

Typesetter
Jane Hamnell

Published by Sportscene Publishers (PCW) Ltd., 14 Rathbone Place, London W1P 1DE, England. Tel: 01-637 7991/2/3. Telex: 8954139 A/B 'Bunch' G. London

Printed by Seymour Press Ltd., 334 Brixton Road, London SW9 7AG.

PCW welcomes all unsolicited material (written, photographic and illustrative) and although no guarantee can be given as to its safe return, reasonable care and attention will be exercised.

Guidelines for contributors
PCW welcomes articles of interest. Don't be put off if your style of writing is 'under developed'. . . true worth lies in the content, and shaping features comes naturally to us! Manuscripts should not exceed 3,000 words and authors are asked to use triple-spaced lines with a wide left-hand margin; diagrams, listings and/or photographs should be included wherever possible. Please enclose a stamped, self-addressed envelope if you would like your article returned.

Because of the foregoing, it is necessary to add that the views expressed in articles we publish are not necessarily those of *Personal Computer World*. Overall, however, the magazine will try to represent a balanced, though independent viewpoint.

Finally, before submitting an article, please check it through thoroughly for legibility and accuracy.

Subscription rates: Britain £8.00 for 12 issues, USA $20 for 12 issues (surface mail), Continent and elsewhere £9.80 for 12 issues. All prices include postage and packing. Supplies to specialist shops can be arranged by negotiation direct with the publishers.

Dr. Chris' Evans, psychologist and computer scientist, died of cancer early in October following a period of indifferent health. Although having been at the forefront of computer science for many years, it's doubly tragic that this should have happened at a time when Chris' was due to attract far wider recognition with the television serialisation of his best-selling book, "The Mighty Micro". With his interests firmly centered around the man/machine interface, his flair and energy are sure to be greatly missed. The staff of Personal Computer World extend their sympathy and condolences to his family.

Rostronics present the Z-Plus Dual Floppy Microsystem

Price **£3750** plus VAT

Features:

Over One Megabyte Disk Storage

4 MHz Z-80 Processor 64k Memory

Two Serial and Two Parallel I/O Ports

An alphanumeric keyboard with 95 keys and numeric pad,character display (24 lines of 80 characters per line) on a 15 inch screen

Including System Desk

MICRO COMPUTERS

ROSTRONICS LIMITED

115-117 WANDSWORTH HIGH STREET, LONDON SW18 4JP

Telephone: 01-870 4805 Telex: 8813089 INTPRM G

Personal Computer World

SUBSCRIPTIONS

Less than 2 years ago PCW became the first magazine in Europe to deal exclusively with the home and business use of Personal Computers. It has been an unqualified success. The current subscription list stands at well over 3000, with a staggering 70% renewal rate! PCW reader loyalty is already a byword in the publishing business. We aim to keep it that way. So if you are having difficulty in obtaining PCW at your newsagent, why not take out a subscription and have the magazine mailed to you direct? YOU KNOW IT MAKES SENSE!

ANNUAL SUBSCRIPTION RATES

UK: £8.00
USA: $20.00
Elsewhere: £9.80

Please send me the next 12 issues of PCW

NAME ______________________________

ADDRESS ______________________________

____

(Block capitals please)

I enclose cheque/P.O. for £ made payable to Sportscene Publishers Ltd.
Send coupon to PCW (Subscriptions), 14 Rathbone Place, London W1P 1DE. Allow at least 14 days for processing.

BINDERS KEEPERS LOSERS WEEPERS

Half the people you meet today are not preoccupied with pollution, perversion or persecution. It's worse than that — they've lost a copy or two of PCW and don't know where to find replacements.

So keep *your* copies of PCW in a beautiful bright blue binder. £2.95 worth of smart security.

Just check the coupon at the foot of the page.

BACK NUMBERS

THINK OF THE FUTURE-LOOK BACKWARDS!

PCW has long been regarded as by far the most authoratitive journal in its field.
Every issue published has contained a wealth of detailed technical information on all aspects of the rapidly growing Personal Computer Market.
Demand for our limited stock of back numbers has also become something of a rapid growth industry!
Consequently, early back numbers of PCW are going to be increasingly difficult to get hold of, so make absolutely certain of getting yours by checking the coupon at the foot of the page.

Volume 1 No. 1 May 1978
The launch of Nascom 1/77-68: The Mighty Micromite/The Gingerbread Mans Computer.

Volume 1 No. 2 June 1978
PCW review Research Machines 380Z/Computer in the classroom/The debut of E78 — The Europa Bus.

Volume 1 No. 3 July 1978
Buzzwords — The first A to Zzzz of computer terms/Pattern recognition/Johann Sebastiàn Byte.

Volume 1 No. 6 October 1978
Minotaur — a new computer game/Computing in the Primary School/Time Tabling for schools.

Volume 1 No. 8 December 1978
Computers and Art/3-D Noughts and Crosses/Meet Mickie — the micro with the bedside manner.

Volume 1 No. 9 January 1979
Super Scamps VDU/Hexadecimal Conversion/TRS-80 Level II Basic/8-Bit Analogue input — putting your pet to work.

Volume 1 No. 11 March 1979
Turning the Tables — a program for the T157/Motorola on sixteen cylinders/The SYM 1 - Computer on a board/Unravelling the Pet Bus.

Volume 1 No. 12 April 1979
Breaking the Software Barrier with Nascom 1/A program for the Z80/Vector MZ review/The Learn Machine.

Volume 2 No. 1 May 1979
Small computers for small organisations/The Sorcerers Wand/Chess Programming: Hints from an International Chess Master/Parkinsons Revas.

Volume 2 No. 2 June 1979
PCW review the MSI 6800/Witbit — disassemble your programs/The Multilingual Machine/Polytechnical Processing.

Volume 2 No. 3 July 1979
Vision link: Interfacing and Software for the Superscamp VDU/Pet Preening/The Soft Cursor — extended cursor graphics for the TRS-80.

Volume 2 No. 4 August 1979
The North Star Horizon — A PCW review/High Speed Cassette Interface for the SWTP 6800/Garage Accounting Program/Apple Medical Application.

Volume 2 No. 5 September 1979
New 10 part Pascal series/ESP — The ultimate interface?/Young Computer World with John Coll/Benchtest on the Compucolor

Volume 2 No. 6 October 1979
In Store — the first complete micro — the SHARP MZ-80K/Bet on a Pet: Business feature/. Cromenco System Three takes the PCW Benchtest.

Any one issue 70p; Any two issues £1.20; Any three issues £1.60; Any four issues £2.00. All additional issues @ 40p each. Binders @ £2.95. All prices include post and packing. Cheque or P.O. payable to Sportscene Publishers Ltd., 14 Rathbone Place, London W1P 1DE. Please allow up to 3 weeks for delivery and don't forget to state clearly your name and full address with your order.

Please send me the following copies of PCW. I enclose a cheque P.O. for £

Volume 1: ☐ 1 ☐ 2 ☐ 3 ☐ 6 ☐ 8 ☐ 9 ☐ 11 ☐ 12 / Volume 2: ☐ 1 ☐ 2 ☐ 3 ☐ 4 *

Please send me — PCW Binder(s) ☐

Name ______________________________

Address ______________________________

____

*Tick appropriate boxes

NEWSPRINT

Newsprint (now incorporating What's New) is where Guy Kewney *reports the happenings of the micro world. Product news, rumours, gossip, prediction and speculation. . .read Newsprint and stay ahead of events.*

A quack one?

It sounds like one of those old-fashioned quack cure-alls that relieves constipation, removes tar from teeth, dissolves kidney stones *and* cleans your spectacles. It's a £200 kit that uses either the Zilog Z80, or the Texas 9980 micro, and you can pull out one chip and replace it with the other, whenever you like.

My attitude is if it's true, I want one. There are few enough opportunities to get a cheap computer based on the TI chip family; the least expensive I know of is a board made by a Birmingham firm Brandauer, based on the 9900, a chip which has a full 16-bit data and address bus. It's really meant for the system builder, not the amateur (by this I mean you need an expensive terminal to get anything into or out of it). At the same time, there are so few Texas users that I'd be nervous of buying a computer that used only that chip. . . where would I go for help when things got stuck? And so, the option of the much more common Z80 attracts me.

So much for attitude, but what is it we're talking about? According to the preliminary specification it's a big board, a "double double Eurocard", with the processor section on one side, and the TV scanning circuit and keyboard interface which together provide input and output, on the other. You can cut this side off, and put the processor side in a standard Eurocard slot, says the designer.

If you don't cut it off, this side provides a display on a standard TV, and reads from a standard typewriter qwerty keyboard. On the TV side, it gives 16 lines of 64 characters, with a modulator described as 'on-board channel 36 wide-band UHF'.

Data is stored on/or retrieved from an audio tape recorder. The designer has modified the Kansas City computer users' tape standard (CUTS) to transmit in 64 byte blocks, with error checking. This is very important, although it does kill the possibility of compatibility on software or data from other systems.

So much for the more interesting points of the boring detail. The designer also provides preliminary information on a similar level about software and memory mapping.

The really interesting bit is, how does he do it? From the fact that he recommends buying two forty-pin central processor sockets, both 'multiple insertion' types at £7.00 extra, you can safely deduce that he does not expect to plug the chips into the same socket. Just as well; it wouldn't work!

Yet, even allowing for the fact that the 9980 has a restricted data bus only 8 bits wide, rather than the full 16-bit bus of its big brother the 9900, there are fundamental differences between the Zilog and the Texas central processors.

For example, Texas provides an on-chip communications register unit which gives direct serial communications to outside teletypewriter devices. Can a system which is built round the Z80's need for universal asynchronous receiver/transmitters (UARTs) also accommodate a chip with a CRU output?

But hold it, you say: why are these questions appearing in print?. . .don't they know the answers?

No, not yet. The designer is one B.B. Leather of 1 Willow Way, Loudwater, High Wycombe, Bucks HP11 1JR. He has no traceable telephone, and our letter pleading for a chat had not reached him at press time.

By the time you read this, the mystery will have been resolved. Watch this space!

As you were

We described the Philips MDCR as a diskette in the September issue. It is, of course, a cassette — a mini-cassette, in fact, as Philips rep Vic Drayton has been quick to tell me. He also points out that the bare device includes only read-write and motion control circuitry; software is needed for search ability, and a bit of logic for phase encoding; It is now available from two distributors: Swift Sasco of Gatwick Road, Crawley, tel: Crawley 28700; and Tekdata Electronics of Federation Road, Burslem, Stoke-on-Trent, Tel: Stoke 813631. It will be visible at Compec, the Wembley show, in a Pelco displayed Aim 65 system. Thank you, Vic.

Sybex training system

From America, the publishing company Sybex has 'published' a computer. It costs $300, and from that you can safely deduce that in the UK it will cost quite a bit more than £150.

Making this computer, (which uses the 6502 microprocessor) different from any other 6502 micro, is the fact that it is sold as a self study training system. Packaged with the machine — it looks astonishingly like a Sym 1 — are two books and two cassette tapes. One book is Programming the 6502, published by Sybex, and my friend Robin Bradbeer of the North London Poly tells me it is a good book. The other is a 6502 Applications book, also published by Sybex. On one tape, there is software, and on the other, a voice (probably human) giving instructions on how to use the board.

I'm afraid, on the basis of this information, I can't tell you why you should buy this package, rather than getting hold of the books separately, and buying a £75 Acorn which can be built up into a Eurocard system. If I hear of reasons, I'll print them. Sybex is at 2020 Milvia Street, Berkeley, California 94704.

Memory aid

Intel's 2118S

A memory chip from Intel — just another 16K RAM, it would seem — is in fact being offered as a boon for the memory designer. The part is called the 2118, and it contains 16K bits, each addressable separately and singly. That means you need eight chips to make a useable memory for a machine with an 8-bit data bus, and you get a minimum of 16K bytes.

What makes it special, says Intel, is "It is the first 16K by 1 (ie singly selectable bits) RAM to operate with a single 5V power supply and to offer very low levels of power with 150 mW drawn in operation and 11 mW on standby". It is voltage and pin compatible with future 64K bit RAMs, so boards designed with this will carry four times the memory — that is, at least 64K bytes — when the 64s are out. But not this year. Intel also says that this chip is designed to work with its 8202 dynamic RAM controller, which makes it as easy to use as static RAM.

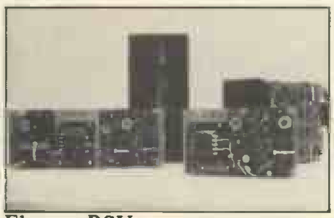
Sybex training system

Tandy trick

Cramming 117,740 bytes onto a Tandy diskette with capacity for 89,600 is a trick done by software from AJ Harding (Molimerx) by the simple expedient of deciding that all the information on a Tandy diskette is BASIC (and therefore is not truly eight-bit hexadecimal code, but upper case ASCII – which can all be stored in six bit codes, not eight). Together with this new product, a new word: "as with the regular system this buffer can be configurated as you wish". One can only offer sympathisement. Details from Bexhill (0424) 220391.

Floppy Power

Power supplies for floppy drives are not included in the average computer kit. You can make up one, or you can buy one, now, from HAL computers of Weybridge tel: Byfleet (09323) 45421.



Floppy PSUs

Owl's lisp

A new language? LISP is that, and costing £40 from Owl Computers it's probably a worthwhile experiment. Devotees claim that it's not so much a high level language as an assembly language for a high level machine, plus software to make your computer imitate that high level machine. Owl's version runs on the Apple II; it is supplied on disc or cassette, occupying 6K bytes of code, with a 44 page manual for a 16K byte or larger computer. Two demonstration programs are included.

It is aimed at "hobbyists who want hands on experience of the fundamental language of artificial intelligence research", amongst others. Owl is in Bishops Stortford, on 0279 52682.

Mass erase

You may never need to erase 104 programmable memory chips under ultra-violet – in fact statistics seem to show that most users of this form of read-only storage do just that – read only. Nonetheless, it may be worth your while knowing somebody who can cope with 104 at a time, because you may want to erase a u-v-e PROM that is soldered to a large board. That board will fit inside the big 100T PROM eraser now marketed by Microsystem Services. It's a fair bet that anybody who shells out the cost of a 100T will welcome the chance to recoup a bit by running an erasing service occasionally so if we hear of a sale – to a careless manufacturer of big EPROM systems who has to call back several thousand faulty boards, – we'll let you know. Meanwhile, back to the sun-ray lamp and guesswork timing. . .

Friends of Pascal

People take languages very seriously, and nobody likes to hear his mother tongue insulted. Not surprisingly, then, the language Pascal found itself amply provided with friends when a slightly negative comment was made by Abacus, about the package as supplied by the University of California at San Diego. (UCSD).

Oddly, Derek Rowe of Abacus was not attacking Pascal; he was announcing that it was available on a system he sells – the TEI system. Rowe's original comment was apparently designed to please Pascal freaks: he said that in the UK the demand for Pascal is very tentative, and that he found this reluctance rather disappointing.

He then blotted his copy book by warning the unwary that UCSD Pascal is not really suited for the amateur until it has been processed from its raw state into a purpose built package for a particular machine. Some exception has been taken! Those who 'speak Pascal' already, long to see others doing so too, and get very annoyed at anyone who seems to doubt their missionary zeal.

Yet the warning is worth repeating. What Rowe was trying to say was simple: if the inexperienced, BASIC-only programmer gets hold of the UCSD package, he won't have a clue how to select and tailor those portions that are dependant on the logical shape of your own computer.

"Most people who are looking to Pascal to give them a step up from BASIC are not systems programmers." Rowe said, "and if they were, they wouldn't be looking to Pascal, but to assembler. I think all serious programming should be done in assembler."

UCSD has now handed over the marketing of its Pascal to Softech, a US software corporation which is not required to be a non-profit outfit (UCSD was having tax problems over the success of Pascal, it seems). It remains to be seen what shape the product will take in their hands, given a stronger marketing drive. Meanwhile, Abacus is at 62 New Cavendish Street London W1M 7LD.

Off peak cheek?

The whole basis of the micro revolution has been the fact that you can have your own microcomputer for less money than the cost of a share in a large computer. So it takes a special kind of nerve for a London bureau to announce an 'off peak' time sharing service – for hobbyists. The bureau, Computer Time Sharing Services (CTSS), is prepared to let you use its machine for about a pound per hour. Quote from George Hertz, manager of CTSS:

"At these prices, many computer hobbyists will find time sharing more economical: all a user needs to get on to our system is a terminal with an RS-232C or V24 interface, a modem or acoustic coupler, plus his telephone. All this can be rented, or it can be purchased for less cost than most hobbyist computers. Yet it gives access to a system that is very much more sophisticated. No longer is data storage restricted by the limitations of cassettes or small floppy disks; the CTSS user can have many megabytes of online data storage for instant access."

All of which is very largely true. Exactly what it proves about the price of hobbyist systems in this country is probably unprintable. Until things change, CTSS is on 01-590 1155.

Switch to bits

A sub-miniature rotary encoder switch which will convert its ten positions into a four-bit binary code from 0 to 10 – or rather, from 0000 to 1010 – has been produced by Impectron. You could use it as a monitor select switch, or as the simplest form of direct input to a system. Alternatively you can set it so that when it points to 5, it gives out 1100 instead, and so introduces a whole new series of bugs! Details on 01-992 5388.

Solderless

Experimenters who do not rate their abilities as soldering operatives very highly will be pleased to see three 'solderless Breadboard units' from Lektrokit: two terminal strips, and one distribution strip. They have an adhesive on the back, or can be screwed down if you prefer. Details: Reading (0734) 669116.

Connections

Having brought Lektrokit's solderless breadboard, you can also buy a kit of wires to connect components together. Each kit has 350 wires, comes in a neat box, and has fourteen different lengths, insulated, bent over, and ready to push into the holes.

Photo-save

Your computer has just output a screen full of data onto a television. You know that if you write it down, you will acquire at least one error, and you can't afford any kind of printer. What do you do? Well it may seem obvious. . .

you take a camera and photograph the screen. A special Polaroid camera costs £128, and the supplier, John Davidson of GDS Ltd, will sell you a special hood to cut out reflections, for around £150 – or he will give you free designs and let you build your own hood. Phone Cambridge (0223) 51645 for details.

Star bores?

American software for Apple II computers is sold with a certain lack of style that makes it irresistible. Virginia company, Soft-One, has announced a two-volume package at $15 for each, with over a dozen programs on each volume.

But do we need things like 'Clock – turn your $1200 computer into a $5 clock with this program' and 'Story Teller tells simple stories; you supply the characters and the subject matter. Each one different' and 'Starwars – put the computer away Luke, and let the force be with you' and stuff? Yes? Somebody import it, then!

Coloured acorns

A colour computer for under £200 can now be put together from Acorn parts. Some may think it almost impudent of Acorn's Chris Curry to announce a colour video display board for his £65 kit (£75 built) and apologise for for the fact that it costs £88. Veterans of the hobby business may recall, somewhat wryly, that it was the PAL colour output board which Apple told us, here in Europe was responsible for its high price compared with the US price. (PAL is the system of colour television we have, and it is much better than the US system, which is NTSC – that's all you need to know to enjoy the fight).

Acorn is 'cheating' a little by using the Mullard teletext chip for colour characters and graphics, and there is a hidden extra: £12 for a UHF modulator board to provide a signal that the aerial socket will be able to tune in to. It's not a lot of money, though. Hidden extras are on the positive side too: the Acorn board can also give you a light pen facility.

Quest micropad

For computer users who can't type, a hand-writing reader in the form of an intelligent writing pad has been announced by Quest. Originally the Datapad was a large minicomputer hidden under the table, watching the position and direction of movement of a pen on a pad. It was good, but the minicomputer cost several thousand pounds, and it didn't do anything with the information; it just turned it into the sort of output you would normally get from a keyboard.

Not surprisingly, the original Datapad did not take the world by storm. Its little brother may do. It has a microprocessor built into the pressure sensitive pad – the micro is the Texas 9900. It's a lot cheaper, and, says the Quest subsidiary which makes it, every bit as good.

Micropad recognises the full English alphabet, with alpha, numeric and special characters 'allowing for a wide tolerance in style and shape'. You need not interpret this as implying that it will cope with a scrawl, because it won't; there is a little display to show you what it thinks you have written, and that display isn't there just for show. It makes the occasional mistake even then. The Micropad also recognises where on the form you are filling in, you have entered data: so if the computer is properly programmed, you can enter (say) '33' under Age, '38 Bloggs Drive' under address, and the machine will interpret this correctly.

Kits and bits

Kits and bits will be on show at the 'kits and bits' show, Breadboard '79 this Christmas. Last year, the first Breadboard attracted several microcomputer companies despite fears that it would prove to be the normal concoction of metal detectors and bad

TRS 80 MICROCOMPUTER

The 'silicon chip' age is here, now, at your nearest TANDY store. TRS-80, the world's biggest selling microcomputer, is bringing high technology down to earth in shops, factories, offices, schools, laboratories and homes everywhere.

Teaching maths, storing essential information, making calculations for stock control, management accounts, financial analysis, personal finance and performing a-thousand-and-one other functions.

No prior knowledge of computing is required to operate a TRS-80 system using the comprehensive step-by-step 232 pg. users manual. Continuous computer demonstrations are featured at the stores listed below.

* TRS-80 is fully wired and tested – NOT a Kit.
* Designed and manufactured by TANDY CORPORATION.
* 'Level' refers to version of BASIC programming language offered by a particular system.
* 'K' factor relates to size of Random Access Memory and degree of program and data storage a particular system offers.
* TRS-80 is a modular system capable of expansion to suit your needs exactly. Get details of 'expansion interfaces', 'upgrades' and system capabilities from your local TANDY store.

16K LEVEL II

INCLUDES NUMERIC KEY PAD

REGULAR PRICE
£596.85 Inc. VAT

£519 PLUS VAT

4K LEVEL I

£375 PLUS VAT

REGULAR PRICE
£431.25 Inc. VAT

32K DISC SYSTEM

INCLUDING FREE GENERAL LEDGER PROGRAMME

SYSTEM INCLUDES

26 1162	DISK DRIVE (0)
26 1161	DISK DRIVE (1)
26 9503	DISK CONNECTOR
26 1006	16k LEVEL II
26 1141	16k RAM EXPANSION INTERFACE
26 1401	LINE PRINTER CONNECTOR
26 1301	SYSTEM DESK
26 1302	LINE PRINTER STAND
26 1403	INTERFOLD PAPER
26 1152	TRACTOR FEED PAPER
26 1103	NUMERIC KEY PAD
26 310	DOS DISKETTE
26 2104	DISK BASIC MANUAL
26 305	BLANK DISKETTES (10)

FREE WORTH £69.95 Inc. VAT
26 1552 GENERAL LEDGER

REGULAR PRICE
£3,333.85 Inc. VAT

£2,899 PLUS VAT

Continuous computer demonstration stores

WEDNESBURY Bilston Road, (off Hollyhead Road); 021-556 6429
BRISTOL 5 Badminton Road, Downend; 0272 560234
SOUTHAMPTON Shopping Centre, 3 East Street;
LEEDS 72 Merrion Centre; 0532 42520
LEICESTER Able Jack, Abbotsford Road, Humberstone Park; 0533 58011
HENDON 21 Sentinel Square, Brent Street, N.W. 1; 01-202 7331
WIMBLEDON 124-126 The Broadway, S.W. 19; 01-542 6389
SALE 13-15 The Mall Shopping Centre; 061-973 0371

TANDY DEALER

Most items also available at Tandy Dealers. Look for this sign in your area.

Prices may vary at individual stores. Offers subject to availability.

Instant credit available in most cases.
Access, Barclaycard and Trustcard welcome.
Check your telephone directory for your nearest store.

KNOWN AS RADIO SHACK IN U.S.A.
MAKERS OF THE WORLD'S BIGGEST SELLING MICROCOMPUTER TRS-80.

TANDY

BRINGING HIGH TECHNOLOGY DOWN TO EARTH

amplifiers, loved by electronics experimenters. This year the organisers, Trident Exhibitions, say over 90 exhibition stands will feature "microcomputer systems, analysers, logic test accessories, hi fi amplifier kits, modulators, etc, as well as a variety of exciting construction. . . ." (oh well, what do you expect. . .) "kits and TV games. . .visitors can construct their own lie detectors."

Trident, you should be warned, is a company with an uncanny knack of turning an exhibition into an astonishing success. The last one I know of was Compec (now owned by IPC).

At this stage, the alphabetical list shows Acorn, Commodore, Compshop, Crofton, Henry's Radio, Lektrokit, Lotus Sound, A Marshall, Microdigital, Newbear, Transam and Vero as the more obviously computing exhibitors. The dates are December 4 to 8, Tuesday to Saturday; and the location is Royal Horticultural Halls, Elverton Street, Westminster.

Better BASIC structure

An 'extremely advanced' version of the BASIC language, called Structured Basic, is available from the big S100 specialist distributor, Comart. The company introduced this new software tool as a way of allowing programmers to write structured (good) programs, rather than unstructured (bad) programs. Commands such as REPEAT, WHILE, IF...THEN...ELSE..., and PROCEDURE are believed to make clear program design easier: they do not make it inevitable, however, and you can write as badly structured a program as anybody – even using Structured Basic!

Pascal for Elf

An unusual microprocessor, the RCA Cosmac 1802, has acquired a Pascal system through the Bicester firm, Golden River, which specialises in this device.

The micro is extraordinary in many ways: chiefly there's its use of complementary metal oxide silicon technology. This takes very little current, and makes it possible to run quite large systems off a dry cell battery. It's also noteworthy for having an assembler language which makes Motorola and Intel assemblers look like voice recognition. The advantage of its fiddly assembler has always been that frighteningly efficient programs could be written, using only a little memory.

In the days when nobody could afford Cmos memory to go with the Cmos processor, that was an important advantage, and a Pascal system that needed 20K bytes of memory would have been meaningless for most amateurs. Now, however, Cmos is much cheaper, and even more important, standard dynamic memory chips are being sold that use as little power. All this is good news for users of the cheap Elf system, and eventually, the availability of Pascal will be reassuring to them. Golden River is on Bicester 44551.

Elf prices down

Most of the items in the Elf II range have now been reduced in price. The Giant Board is £37.80, the 4K Static RAM board is £75.60, the ASCII Keyboard, £54.63 and the Kluge Prototype Board, £13.82. A video graphics board will be available soon, as will text editor, assembler and disassembler on cassette. Enquiries to HL Audio Ltd., 138 Kingsland Road, London E2 8BY (01-739 1582).


ELF II

DIY fibre optics

Do-it-yourself fibre optics for experimenters has been announced by Burr-Brown, the analogue to digital company in Watford. Two packages are available, each with sufficient parts to form a complete link with the addition of a power supply and TTL level signals. The difference between them is speed. Details on 0923 338337.

Look alike

Could Japan have struck at the hardest nut of all, the TRS-80? Being the most common computer in the world, a Japanese imitation makes sense, and a Matlock based company, Lowe Electronics, may have found one. They call it Genie and expect to sell it for £500 without video monitor. Supposedly, it runs all TRS-80 software.

Danish soft

Denmark's personal computer industry has software for us. From a company called Lisco Micro Data in Kolding, comes a package of languages including Focal, Tiny BASIC, and some applications, aimed at users of the 6502 microprocessor.

Gunnar List claims that his Tiny BASIC will run in 4K byte systems on Kim-1 and Sym-1 from Commodore and Synertek, with a version for the Rockwell Aim 65 due out soon. Cost is £12.30 including tape and manuals.

Focal costs more – £13.20 for the standard language interpreter, £17.60 for the extended interpreter, £4.70 for a mini manual and £9.40 for a 'user manual'. My typewriter won't cope with the subtleties of the Danish spelling, but as near as I can manage is: Lisco, Aprilvaenget 6, 6000 Kolding. The phone number is (05) 56 86 82.

From the Centre

A British system builder has joined the long list of American names offering systems based on the standard S100 layout. This is a £3,000 machine, so isn't for the user at home unless the user happens to have a generous employer.

The company, Computer Centre, is well known for the low prices it charges (especially on components such as memory boards) so it isn't surprising that boss Peter Norman has offered a 'basic kit form version for the scientific builder' at under £1,000.

The big machine is the OEM 2, with dual diskettes storing 2 million characters of data, a full 64K bytes of internal memory, and built-in software including the well respected CP/M operating system. This will allow the user to expand his external storage to 128 million characters without confusing the computer.

The basic kit version has only one diskette drive. Computer Centre is now in Swansea, at 9 De La Beche Street, Tel: 0792 460023.

Tape basic

Very probably, most people who move from tape cassettes to diskettes could manage quite well on tape, if only the data loaded into the computer or stored out onto tape, were less liable to be wrong.

Nascom software expert Tony Rundle, now with his own company, Starbase, has added an error checking system to the way that computer handles mag tape. It comes with the new version of BASIC for the Nascom 1. . . . only the Nascom 2, when it is available, will have BASIC as standard on the board. Rundle says that he was virtually forced to design a checksum cassette handling system because there was no other way of loading an 8K byte program.

Tape Basic costs £30 from Nascom itself; Tony Rundle is prepared to help and advise from his address at Waxhouse Gate, 15 High Street, St. Albans, Herts AL3 4EH.

From Japan

Diskettes from Japan are to be marketed in this country by a new company: DRG Business Machines of Weston-super-Mare. Both five inch minifloppies and full 8-inch floppy drives are offered, data compatible on 8-inch with IBM drives. Details of these and a controller for the 8-inch, on 0934 415398.

Please help

Ian Litterick was astonished to find that when he was first infected with the desire to become a computer owner, there was nobody to ask about pros and cons of different systems. Ian, a consumerist by nature and by training, instantly wrote off to MPs and civil servants suggesting that here was a hole that needed to be plugged.

"If you want information on big computers," he noticed, "you go to the National Computing Centre and pay for it. But if you want information on a micro, the sort of money the NCC wants for giving help can be almost as much as the micro."

His idea is a national Microcomputing Centre; and by dint of being a lot more energetic than the rest of us, he has bullied the NCC into studying the idea, and providing money for the study. The Department of Industry provides half the cost.

Litterick is compiling the informed opinions of people with informed opinions. . .he has even asked me, for example. That can't be good enough. So, readers, please help with advice. What has been most lacking when you were making your purchase decision? Would it help if the NMCC existed and put out a 'preferred specification' against which you could match your requirements, and compare prices? Send your opinions to PCW, or, if what you have to say is too harsh for our ears, to the NCC, which is at Oxford Road, Manchester M1 7ED.

Pet sophistication

Pet owners usually go for the Pet in the first place because it has BASIC: after a while, they start wanting to do more sophisticated programming. The Pet Machine Language Guide, from Abacus Software in Michigan (not to be confused with Abacus Computers in the UK) is aimed at these ambitious people. Cost of the guide to us Britons is $8.95.

Included are sections on using the Pet's input and output routines, clocks and timers, floating point, ASCII number conversion routines, and other complex arithmetic functions. Payments by Visa card is accepted; Abacus is at PO Box 7211, Grand Rapids, Mich 49510.

Cash in hand

It's competition time, and both Peterborough and the National Research Development Corporation have thousands of pounds ready to give away to those with bright micro ideas. Peterborough has a total of about £40,000 (that includes a free factory for a year in Peterborough). There are lots of details, musts and must-nots, all available from the organisers at Peterborough Council, and at the NCC respectively.

Superbrain 2xZ80

Causing some interest amongst enthusiasts with £3,000 to spare, is a computer that uses two Z80 micros. The Superbrain, as it is called, uses the second micro to control data coming in from the outside storage (two floppy disks) thereby speeding up the whole process.

The basic system is pretty big with a full quota of memory (64K bytes) and a wide variety of output methods. It will talk to standard computer terminals using the V24 protocol, and will drive an ordinary S100 bus from its S100 output port. Software includes the standard CP/M operating system, so the whole CP/M user library should be accessible to the user. The supplier, Computrade, is on Leatherhead (03723) 77374.

Superbrain – 2XZ80

Comma to you

Comma Computers is now officially the name of Computer Marketing, the company which got itself known as a terminal supplier, and moved into micros by selling American ICS micro courses. News of micros with the Comma label has been given in the past, and the company felt that to have the same name for computer and company would 'give a clearer, crisper image to the combined operation'. It all arises out of Computer Marketing's takeover of Micro Software systems in July; the Companies Registrar is blamed for the delay in changing names.

Comma managing director George Macfarlane has astounded his competitors in the terminal market with his willingness to stick his neck out; they speak of him in hushed tones because of his willingness to trust Decwriter's delivery promises; no doubt they will also shake their heads over his decision to sell the new Alpha Micro 16-bit system.

He says he plans to shift more than £7 million worth of micros and terminals in just over a year, "and we are celebrating the new name by announcing a novel microcomputer system called the Comma Copywriter". We look forward to seeing it, George, when can you bring it round?

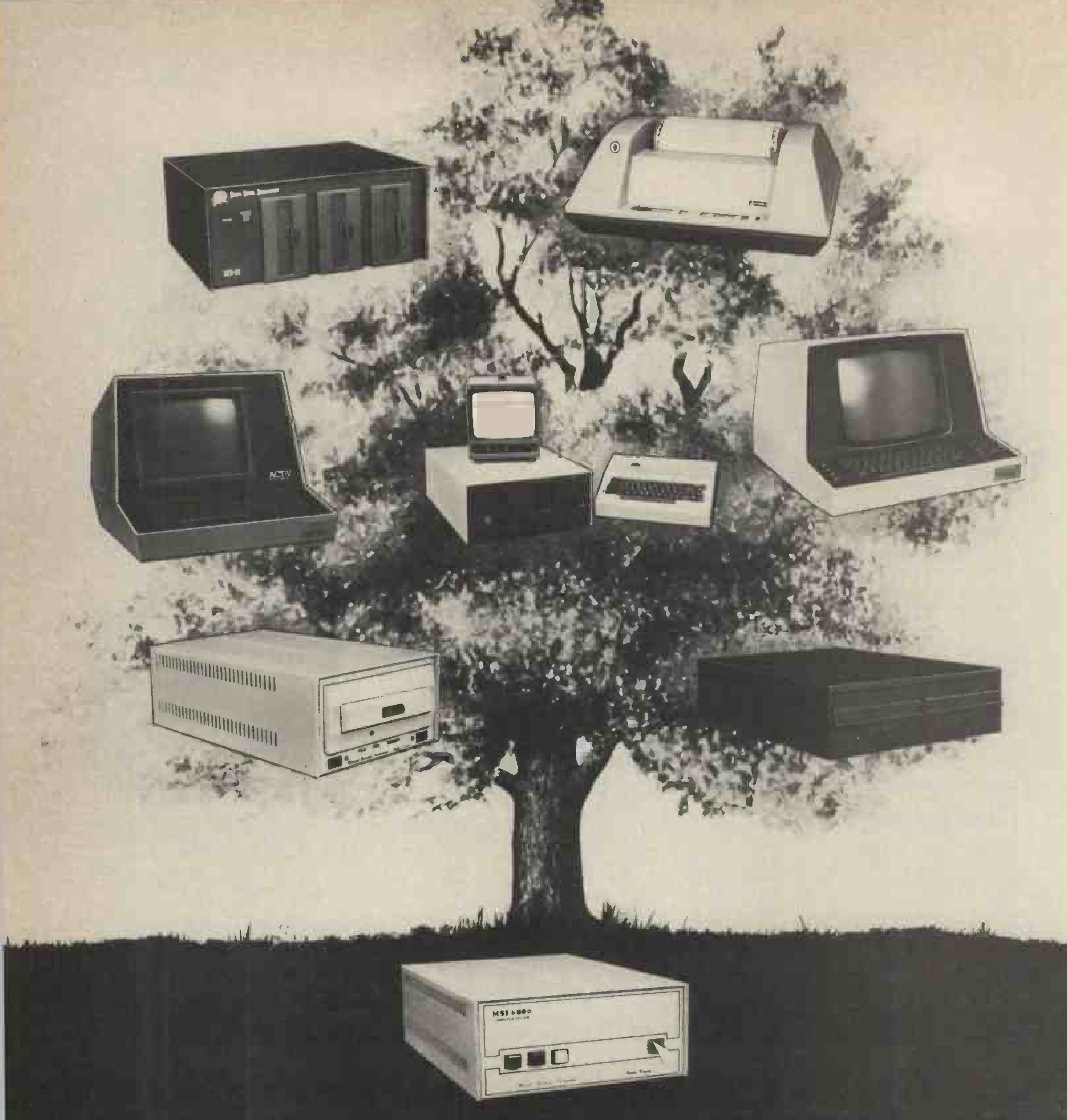
Supermicros, but when?

The supermicros, computers that would give us 'all the power of a middle sized minicomputer like Digital's PDP-11/34, on one chip', are knocking on the door.

The most encouraging news is of Motorola's big chip, the 68000, now 'available' on an evaluation board costing £1400, The word 'available' is a wild exaggeration, of course, with around a dozen of these boards so far sold, and with the UK micro people proudly displaying chip number 1065. Obviously the machine is not yet available to just anybody, so when will it be?

The best bet, from the point of the private buyer, is that it will remain a rare beast for at least another year, maybe two. Again, from the private buyer's viewpoint, this doesn't matter a lot. The biggest restraint on any private system is not the power of the processor. It is a question of how much memory it can control – both internal, and external. The Motorola 68000 will be able to control some 24M bytes of semiconductor memory. . .that's more internal storage than most private systems have disc storage. Couple a machine of that power, with its full quota of memory – or even a quarter of it – to a big, 50M byte disc, and it will eat anything on the market for £30,000 today.

But the same could be said of the humble Intel 8080, if one were to spend a little time and ingenuity giving it memory management; with a 50M byte disc, and a virtual disc storage system, to give the appearance of having more than the maximum 64K bytes internal storage (a neat trick, and an old, well proven one), it too could eat most small business systems. The problem is not the speed at which the processor can process data, but the slowness



MSI 6800: At the root of every good system.

seed

Strumech Engineering Electronic Developments Limited
Portland House, Coppice Side, Brownhills, Walsall, West Midlands. Telephone (279) 4321

with which data gets into it. A big, fast disc makes many times the difference of a big, fast processor.

That said, the signs are that inside two years, Motorola, Zilog and Intel will be able to offer chips (and support chips) with 16-bit architectures that will be attractive to private users. Motorola's is agreed to be the biggest and best; many, however, have disagreed as to whether it was not overambitious, with its 68,000 transistors on an area of silicon measuring 246 by 281 mills. Is it just too small for today's semiconductor expertise?

The most convincing argument that says 'yes, it can be built' has just been produced by Rockwell. Rockwell had a design for a supermicro of its own; it was going to be a descendant of the 6502, and it was going to be called the Super 65. Rockwell has abandoned Super 65 in favour of taking on the 68000. In exchange for the design, it will give Motorola its bubble memory designs.

Rockwell's decision tells us two things. First, it is convinced that it can make the 68000, and so the prospect of Motorola making it itself becomes that much more tenable. (Ones and twos don't count; hundreds per day would barely be convincing.) Second, it tells us that Rockwell is equally sure that it has time to learn the recipe for the 68000 before Motorola has got it down to a fine art and can make them for $10 each.

Courses

A new micro consultancy which wishes itself to be known by personal computing enthusiasts is Microsystems Consultants. . .for the reason that they run courses based on the Rockwell Aim 65. The courses are approved by Rockwell agent, Pelco, in the UK.

Managing director Markus Moser says he would like to help companies "with little or no knowledge of microprocessor applications", to get them to take the plunge and develop ideas and projects. Moser says he is an engineer with a degree in communications, and has worked for large companies like SCM, IBM and Data General on mini and micro projects.

For details of the courses, ring Camberley 27417, and to take the plunge, contact Fleet 29627.

IBM on the move

"If there were any serious point to personal computing, then IBM would make a personal computer." Next time some computer industry know-all tries to put down personal computing as something for excessively open minded people – along with astrology and roulette systems – the comeback is a number: the 5105.

This machine will be announced by Christmas, and available at under £2,500 by Spring – in America at least. So says the California market research group Creative Strategies International (CSI) in a report costing some £500.

According to CSI, this will be the specification of the Entry Level System, or 5105 (assuming IBM doesn't change the name to prove them wrong): that's 500 nanosecond cycle time, BASIC and monitor in read only memory, a minimum of 16K bytes memory, built-in video with 960 or 1920 character screen, mag tape cartridge for bulk storage and slow printer built in.

Options will include a language called ACL in firmware, memory extensions to 64K bytes or possibly 96K bytes; diskette storage up to 2M bytes; add on matrix printers and software, including word processing. A later option may be a 5M byte hard disc – probably a mini Winchester.

For those who got lost somewhere in the middle of that, it would be a pick up and carry home system rather like the Pet, with a printer and a better quality tape drive, possibly a little more powerful, and certainly priced at the top end of the market for what it is. The add-on list would take the price to around £10,000, for a system that would apparently compete with others based on the hobbyist, S100 bus, although at something of a price disadvantage. Software packages could give it an edge, however, for those needing something more than the IBM badge.

Final goodie: it may have an S100 bus adaptor.

Sprint 5

Get it right!

Last issue I said that a British Company, A J Harding (Molimerx), was responsible for Tandy software addition, Infinite Basic. How wrong I was! Freddy Nichols of Optronics (who also handles the product) tells me that in fact it was written in the States by Ron Johnson.

At our show

Showing for the first time – and where else but at the 79 PCW Show? – are systems based on the Microstar 1.2 and 2.4 megabyte modules. Access Data are the appointed distributors and the two micro computer packages will have full software backup for both word and data processing. For the stand demonstration, one will be showing the word processing capabilities of the 55cps Qume daisywheel printer, the other will be programmed for data processing using a Texas 820 dot matrix printer.

Access Data Communications are at 228 High Street, Uxbridge, Middlesex.

BENCH TEST

CHALLENGER C3 ~S1

At about the time that the 6502, 6800 and Z80 were emerging as the "big three" 8-bit microprocessors Ohio Scientific, Inc. began to advertise its solution to the problem of program portability. This was the Challenger CIII series — a range of systems centered on a novel MPU board which contained all those microprocessors and which could therefore utilize programs written to run on any one of the three. In designing this sytem, OSI have proved farsighted by predicting the decline in price of the actual processor chip relative to the accompanying hardware. To become really successful however, this scheme depends on the premise that people have a large number of assembly language programs which they need to transport from system to system. In the event, the arrival of BASIC (especially Microsoft's) and fairly widely implemented operating systems may have detracted from the original idea.

The Challenger III series offers a variety of memory sizes, peripherals and software configured around the basic board. Perhaps the most spectacular peripheral is the CD-74 74MB hard disc which comes with the top of the line system. Also catalogued are a voice I/O board, an A/D and D/A board, a multiplexing parallel board together with more standard serial and memory expansion boards. Also on offer are a variety of operating systems, starting with a simple DOS and graduating to a (not yet released) multi-user multiprogramming OS. CP/M is available, as are a Word Processing Package, Data Base Management System and a small business package.

The review machine, the S1 model, was a 56K RAM, twin floppy system with a Hazeltine 1410 terminal. Operating systems 65D, 65U, CP/M and application packages DMS (Data Base Management System) were provided on floppies together with most of the software documentation.

BY SUE EISENBACH

Hardware

The Challenger III model C3-S1 is housed in two cases, a light one containing the computer itself, and the other, heavier one, the disc drives. To open either box the cover has to be unscrewed. Both boxes are well ventilated. The computer has no fan and the operating instructions state that it should be run in an air-conditioned room with 6" clearance for ventilation. The fan in the floppy disc drive is small and noisy; attached to the box, for some reason it clatters when it is running.

The outstanding feature of any Ohio Scientific C3 computer is its CPU board. Called the model 510, it contains three microprocessors, the 6502A. the 6800 and the Z80. A software switching program is on the board so that choice of microprocessor is under program control. The PROM contains the 6502 and 6800 monitors as well as a floppy disc bootstrap. An RS-232 port, eight parallel lines and a clock (which supplies 4MHZ, 2MHZ and 1MHZ signals) are provided.

The memory comprises two to four OSI 520 16K static RAM cards. The fourth is only half populated, giving a maximum of 56K (as in the review machine).

The disc controller is an OSI 470 which can support 1 or 2 single or dual headed 8" floppy drives with soft sectored, single density recording format. Capacity varies from 230K Bytes to 290K Bytes depending on the operating system used. The disc drives are Siemans FDD 120-8.

There were problems arranging the test. The machine came from Computerland in Birmingham and travelled by train and van to reach me. It hadn't fared well during the journey. Inspection showed that not all of the PC boards were attached to the backplane of the computer. They could have been securely screwed down to the base of the box but this hadn't been done. After placing the loose boards back onto the bus and soldering up a few wires that had broken off in transit I turned my attention to the VDU. It failed to operate and investigation showed that a board was missing. Eventually, once equipped with a new VDU, the computer powered up successfully.

I experienced two hardware faults during testing. Firstly, the computer didn't always clear the memory when the reset button was pressed and secondly, when booting one of the CP/M discs, a few messages appeared on the screen and then the system crashed. This disc was however accessible (via the other CP/M disc) from the other drive.

My overall impression of the hardware was of a cleverly designed MPU board enclosed in a rather fragile mainframe.

System software

According to the sales literature, there are four operating systems for the Challenger III. The review machine was supplied with three:

OS-65D, OS-65U and OS-CP/M. OS-CP/M appeared to be a standard CP/M running on the Z80.

The other two operating systems were written by OSI and ran on the 6502. OSI do not provide an operating system to run on the 6800 as the fourth (a business/word processing system) also runs on the 6502.

65D is OSI's simplest operating system. It runs on any Ohio 6502 disc configuration (including those of the Challenger I and II) and is monitor type software. I was given two versions, one with BASIC and assembler, and one without. The disc without BASIC was designed for facilitating the execution of 6502, 6800 and Z80 machine code programs. It contains the operating system, a utilities package, I/O drivers and file handlers. The utility package provides software to use all three microprocessors. These include switches to the 6800 and 6502 monitors (in PROM on the MPU board), a Z80 monitor and Z80 and 6800 memory movers. For the 6800 there is also a MIKBUG simulator and 6800 LOAD and DUMP routines. MIKBUG itself cannot be executed on the C3 as it's not designed for such a large system. OSI explain how to alter MIKBUG programs for use under the OSI 6800 monitor and only provide the simulator for the execution of programs where there is no one available to do the alteration.

The utilities provided do not shield the user from the intricacies of data or processor transfers. To load 6800 or Z80 programs from disc the 65DOS must be entered, the utilities loaded, the program loaded and then the switch to the appropriate processor made. To save programs they must first be moved out of the way of the DOS, control switched back to the 6502, 65DOS booted in and finally the program saved. From

Disc drives unveiled. Notice the large opening in the back and small fan on the cover.

the documentation supplied, I could see no way of accessing disc files using the 6800. (The Z80 can access disc files under CP/M).

The second 65DOS disc supplied booted in BASIC along with the operating system. The BASIC utilities supplied are not provided with a 'LOAD and GO' facility and have to be explicitly executed — e.g. to see the disc directory one types RUN "DIR", to create a file RUN 'CREATE'. There are two ways of saving a BASIC program. The first is to exit from the BASIC system and then PUT the program onto a specifically named track (overwriting anything that might be there) and return to BASIC. The second method is to create a file before typing in any program. When creating the file its size must be declared and, unlike the previous method, the new file will be placed in free space. The user then types in a program and saves it in the usual manner. If the program is larger than the space allocated, nothing will be saved. In addition to the BASIC this 65DOS disc had an editor/assembler. Unfortunately no documentation was provided for these so I could not evaluate them.

The other OSI operating system provided, called OS-65U, is a BASIC only system. In most respects it felt like OS-65D with the BASIC booted in. The data file facilities under 65D are not as comprehensive as those under 65U. Both however have random and sequential files; in addition 65U has indexed sequential files and a FIND command.

The two operating systems are sufficiently similar that it is surprising that Ohio Scientific decided not to write one operating system with the features of both.

Basic

Each operating system came supplied with a BASIC — 65D and 65U BASIC occupy 9K. This includes 8K Microsoft BASIC and 1K of OSI add-ons (primarily file handling). The CP/M BASIC occupies 19K and is a slightly pared down version of 20K (Altair) Microsoft BASIC. Microsoft's BASICs are the industry standard and are upwardly compatible. Unfortunately OSI's file handling facilities are not the same as those written by Microsoft. The BASICs running under 65D and CP/M have comparable features (using different instructions) while 65U's are more sophisticated. The Data Base Management is written in 65U BASIC and utilizes its indexing instructions. For those readers with Pet experience 65D and 65U BASIC should seem familiar. In fact PET BASIC is easier to use with its screen editing.

65U BASIC contains a FLAG command which enables or disables a variety of system features, primarily error traps. Although there is no PRINT USING statement there is money mode output, which rounds to two decimal places with either left or right justification. File handling comprises: OPEN, CLOSE, PRINT%, INPUT%, INDEX and FIND. The INDEX is a pointer to a record in an open file which can be examined and altered. FIND searches from the current position of INDEX through the rest of the file for a given string (which can include 'don't care' characters). If found, INDEX points to the string; if not found it is set to 1,000,000,000.

TECHNICAL DATA

CPU(S):	6502A 2MHZ, 68B00 2MHZ, (Sic) Z80 4MHZ.
MEMORY:	32K - 56K STATIC RAM
KEYBOARD: SCREEN:	HAZELTINE 1410
CASSETTE:	N/A
DISC DRIVES:	2 DRIVES, 1 OR 2 HEADS PER DRIVE, 8" DISCS, SINGLE DENSITY.
PRINTER:	N/A
BUS:	OSI 48-LINE BUS
PORTS:	1 SERIAL, 1 PARALLEL, EXPANDABLE.
SYSTEM SOFTWARE:	OS-65D, OS-65U, OS-CP/M, WP-1B
LANGUAGES:	6502, 6800, Z80 ASSEMBLERS, BASIC, EXTENDED BASIC, FORTRAN, COBOL

The CP/M BASIC is a language you would expect to find on a machine in this price range. It has in line editing, PRINT USING statement, IF. . THEN. . ELSE, AUTO line number and RENUM.

I would have preferred more expansive error codes on all three BASICs ("OM IN LN 100"); fortunately the messages are the same. The tables with the BASIC reserved words should illustrate the differences between the languages. OSI claim that the 6502 is a superior microprocessor . . . after running the benchmark programs I don't see much between them.

Other software

Because CP/M runs on it, there is a large range of software available for the Challenger III. In particular, I was provided with two Microsoft compilers. . . one for 8080 Fortran IV and the second for Cobol-80. As these are completely standard (and good) software packages I will not describe them.

More interestingly, Ohio Scientific have written a comprehensive Data Base Management System designed to run under 65U O/S and aimed at the small business-man with no computer experience. OS-DMS boots in the DMS menu which is the first of several, the whole system being menu driven. The utilities, which can be altered by the programmer, are listed in the table below and show how comprehensive this system is. For security, passwords can be placed on any of the programs in the system. Unfortunately, the system might cause difficulties for a person without computer experience as most input is not checked for legality before being accepted. It is not difficult (contrary to statements in the documentation) to type in an answer that seems reasonable — only to get "SN IN LN 75" with no obvious way of getting back to the DMS system. Even when inputs are checked the user is just requested to type in another response — no range of acceptable data is offered. Before the non computer user would feel comfortable using this system, routines are necessary that buffer the operator from the programs and a rewrite of the documentation is needed.

CP/M BASIC with 65U & D marked U or D, B=Both

Commands:				
AUTO	CLEAR	CONT(B)	DELETE	EDIT
FILES	LIST(B)	LLIST	LOAD(B)	MERGE
NEW(B)	NULL(B)	RENUM	RESET	RUN(B)
SAVE	SYSTEM	TRON	TROFF	WIDTH
Program Statements:				
DEF(B)	DEFDBL	DEFINT	DEFSNG	DEFSTR
DIM(B)	END(B)	ERASE	ERROR	FOR(B)
GOSUB(B)	GOTO(B)	IF. .THEN(ELSE)	LET	NEXT(B)
ON. . .ERROR	ON. . .GOSUB	ON. . .GOTO(B)	OUT	POKE(B)
REM(B)	RESUME	RETURN(B)	STOP(B)	SWAP
WAIT				PEEK(B)
Input/Output Statements:				
CLOSE(U)	DATA(B)	FIELD	GET	INPUT
KILL	LINEINPUT	LSET	NAME	OPEN(U)
PRINT(B)	PUT	READ(B)	RESTORE(B)	RSET
Arithmetic Functions:				
ABS(B)	ATN(B)	CDBL	CINT	COS(B)
CSNG	ERL	ERR	EXP(B)	FRE(B)
INP	INT(B)	LOG(B)	LPOS	
POS(B)	RND(B)	SGN(B)	SIN(B)	SPC(B)
SQR(B)	TAB(B)	USR(B)	VARPTR	
String Functions:				
ASC(B)	CHR$(B)	FRE(B)	HEX$	INSTR
LEFT$(B)	LEN	MID$(B)	OCT$	RIGHT$(B)
SPACE$	STRING$	STR$(B)	VAL(B)	
Input/Output Functions:				
			EOF	LOC
LOF	MKD$	MKI$	MKS$	

Extensions		
Both	**65U**	**65D**
IF. . .THEN	INDEX	EXIT
IF. . .GOTO	PRINT%	DISK !<STRING>
WAIT	INPUT%	DISK OPEN, <DEVICE>, <STRING>
	FIND	DISK CLOSE, <DEVICE>
	FLAG NN	DISK GET, <RECORD NUMBER>
	PRINT $R,X	DISK PUT
	PRINT $L,X	

DATA BASE MANAGEMENT SYSTEM PROGRAMS

- Create New Master File
- Create New Key File
- Edit Master File
- Load Key File From Master
- Edit Key File
- Dump Key File
- Generate Mailing Labels From Master File
- Master File Merge or Load
- Diskette Copier
- Multi-File Multi-Format Report Writer
- Multi-Conditional Report Writer with Statistical Functions
- Multi-Conditional Statistical Package
- Sort a File
- Master File Record Inserter
- Master File Record Delete and Repack
- Inventory
- Order Entry
- General Ledger
- Personnel
- Payroll
- Accounts Receivable
- Accounts Payable
- Query

Benchmark

As well as running the Kilobaud benchmarks (see summary), I set up some disc tests.

These were run under OS-65U as Ohio Scientific state that this operating system provides the best file accessing facilities. All the files in these tests are 100 record files with 256 character records. Each record is composed of 8 fields (called A$ — H$). Tests 2 and 4 are designed to test the "randomness" of writing to and reading from files. If tests 2 and 4 take substantially longer to run than 1 and 3 then the operating system is probably using a sequential method for its random access.

Test 1. Fill A$ — H$ with 32 "A"s each. Open a datafile; using a FOR-NEXT loop write to records 0 to 99; close the file.

Test 2. As test 1 but writing the records to the file starting with the last record; that is the FOR-NEXT loop's step is -1.

Test 3. Open "Datafile" using a FOR-NEXT loop, read each record out of the file, close the file.

Test 4. As test 3 but reading from the file starting with the last record.

Disc test 1	19.9
Disc test 2	21.9
Disc test 3	83.1
Disc test 4	83.1

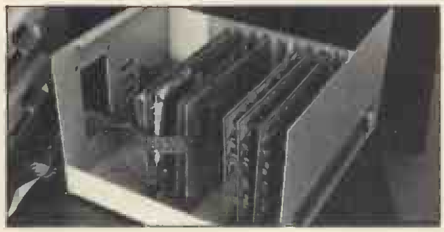
Business potential

The Challenger III is designed for use both as an end user system for running application packages and as a development system. For either use probably its greatest selling point is its hard discs. No other personal computer system on the market offers the possibility of nearly 300M bytes of on line storage. With a Challenger III, software can be designed or purchased for a floppy disc system and then run with hard discs as capacity grows.

Business application

Looking at the Challenger III as an end user system, one ought to be able to run 6800, 6502 and Z80 packages on it. Unfortunately Ohio Scientific supply virtually no system software for running 6800 code. So either 6800 system software must be purchased first to run 6800 application programs or those packages purchased must be written in machine code. In either case, as standard 6800 MIKBUG code will not execute under the Ohio Scientific monitor, it is a safe bet that 6800 programs will not run without the attention of a system programmer.

Moving on to the 6502, Ohio Scientific have written three application packages. The review machine was only supplied with their Data Base Mangement System. It is a comprehensive package with the nice feature of optional passwords for reading and/or writing protection from unauthorized users. Unfortunately I had no difficulty in crashing (both accidentally and intentionally) DMS so any potential buyer should expect to experience some problems when it is first installed. The other two packages that Ohio Scientific supply are a Word Processor and a Small Business Package. Bearing in mind that Ohio Scientific's software is of variable quality, I cannot recommend software I haven't seen. In any case, I have serious doubts about the usefulness of the Small Business Package. It was designed in



Inside the computer itself.

Benchmarks

	CP/M	65U
BM1	2.3	1.7
BM2	7.9	13.1
BM3	21	21.6
BM4	21	23.7
BM5	22.5	29.2
BM6	37.5	39.6
BM7	59.6	58.3
BM8	9.9	17.6

(processor timings in seconds)

America, for an American market where business jargon is different and VAT is unheard of.

Finally, turning to the Z80, the user should experience few problems. As CP/M runs on this microprocessor and most disc based British application packages run under CP/M, the situation is most satisfactory.

Development System

The Challenger III as a development system follows a similar pattern. Again the lack of systems software for the 6800 makes it difficult to use. On the other hand, the Z80 under CP/M gives access to a wide variety of system software. Translators for BASIC, FORTRAN, COBOL, PASCAL, Z80 Assembler and 8080 Assembler are on the market together with their corresponding debugging aids.

Looking at Ohio Scientific's own system software for the 6502 the kindest thing I can say is that it is of uneven quality. I have my doubts about the reliability of OS-65U. It crashed regularly throughout the period that I used it. Normally I would put this down to faulty hardware, but the system did not crash under CP/M and on the whole they use the same hardware. (I suppose there could have been something wrong with the 6502 chip itself). It has a few nice features such as a password system and good file handling facilities (including indexing) under OS-65U. On the whole Ohio Scientific's BASICs are less sophisticated than one would expect on a disc based system. It is also irritating to have three different BASICs each with its own advantages and disadvantages.

Summary

If a user either needs the large disc capacity or wants to run programs on more than one microprocessor, then the Challenger III has possibilities. If neither of these conditions apply, then the disadvantages inherent in the Challenger III probably outweigh its advantages. I cannot imagine the purchase of this machine for the developing or executing of 6800 programs. Ohio Scientific produce a less expensive range of computers, the Challenger II (6502 based only) for running their system software and Data Base Management System. There are a wide variety of other machines on the market that run under CP/M that are less expensive, more attractive and more robust.

Educational potential

I have my doubts whether the hardware is sufficiently rugged to withstand student users. Also the large number of cabinet vents might lead to objects, such as pencils, "falling into the computer". A rack mounted version would be more secure against such accidents.

I was told that it was a good machine for education because it allows students to use a variety of microprocessors. However, for the price of a C3 one can purchase a CP/M system, a PET and a single board 6800 computer. Although this collection doesn't provide identical facilities, it probably provides those features of the C3 that students would utilize, with scope for more "hands on" experience.

On the other hand, programming needs could well be met by the multi-user system with hard discs but again it is debatable, given the small BASIC, whether several stand alone computers would not provide a more reliable installation for the money.

Documentation

The documentation provided by Ohio Scientific Inc. was of variable quality. The OS-CP/M manuals (System, BASIC, FORTRAN, COBOL), written primarily by Microsoft Inc., are thorough, paginated, indexed and filled with examples.

The documentation that OSI

write themselves is more difficult to praise. Several of the manuals supplied were photocopies of preliminary versions, but even their final efforts are not impressive. Pages are only numbered within sections and there are no indices. The manuals are both repetitious and incomplete. There are very few programming examples and most of those are full-sized programs that are rather daunting to scan right through for a single question of syntax.

OSI seem to have difficulty in finding the appropriate level for each type of manual. For example in the documentation for OS-DMS (the Data Management System that is "immediately usable for the untrained small businessman") there is a glossary of terms with definitions such as: "Index — the index is the virtual field address of an entry field, record or file". In the midst of a technical discussion about the memory, the OSI technical writer, in an outburst of enthusiasm says, "520 memory is by far the finest semi-conductor memory available in computing, regardless of price, considering both its superb reliability and outstanding speed/power product".

On the whole, I feel a little tentative about reviewing a system whose characteristics risk being obscured by such documentation.

Expandability

Probably the largest personal computer system advertised is the Challenger III. A C3-S1 can be expanded to a full C3-B system with 768K bytes RAM, four 80M byte Winchester hard discs and 16 communication ports. Also announced is a multi-user version of 65U operating system.

Conclusion

When the Challenger III was designed, there was virtually no software on the market. At that time, people producing software had to program in machine code and so had a thorough knowledge of the operation of their microprocessor. It was a clever idea to place all the major microprocessors on one board, so that all available programs could be run. Unfortunately for the designers of the Challenger III system, software developments in the micro market have meant that programmers no longer need to learn machine code in order to use a personal computer. The overwhelming success of Microsoft BASIC, in which the majority of applications programs are coming to be written, means that the potential user who wants to fully exploit the C3 system will have to become more involved with the hardware than is necessary with other comparably priced systems.

My overall impression of the system was of a machine with some very clever ideas. However, I have the feeling that it was rushed into production, thus giving a rather ragged feel to the package. In particular neither the OS-65D and U system software nor the overall documentation are up to the standards currently available in systems priced upwards of £2000. The 74M byte hard disc system sounds promising but experience (with the Superboards) leads one to expect an element of delay between product announcement and eventual availability.

PCW would like to express its thanks to Computerland in Birmingham and the Byte shop in Ilford for the loan of equipment used in this test.

Prices

CS-S1	32K dual floppy in 2 cases with OS-65D	£2998
C3-OEM	32K dual floppies in 1 case	£2998
C3-A	48K dual floppies, 16 slot rack OS-65U	£4251
C3-B	C3-A with 74M byte hard disc	£9985
C3-C	C3-A with 29M byte hard disc	£7988
520	16K board	£ 385
	Centronics parallel interface board	£ 160
OS-65U	Single user	£ 200
OS-CP/M	With BASIC, FORTRAN and COBOL	£ 600
OS-DMS	Data Base Management System	£ 300
AMCAP	Small Business Package	£ 300
WP-2	Word Processing Software	£ 300

FIRST IMPRESSIONS	
Looks	**
Setting up	**
Ease of Use	***
HIGH LEVEL LANGUAGES	
BASIC (Ohio)	**
BASIC (CP/M)	****
COBOL	**
FORTRAN	****
PASCAL	n/a
System Software	***
PACKAGES	
Business	****
Education	n/a
Home	n/a
PERFORMANCE	
Processor	*****
Cassette	n/a
Disc	****
Peripherals	**
EXPANDABILITY	
Memory	****
Cassette	n/a
Discs	*****
Bus	***
COMPATIBILITY	
Hardware	**
Software	****
DOCUMENTATION	**
VALUE FOR MONEY	**

*****	excellent
****	very good
***	good
**	fair
*	poor

MEMORY MAP UNDER 65D

Address	
DFFF	
3178	Source File Work Space
2300	OS-65D
200	BASIC or Assembler
100	6502 Stack
0	6502 Page Zero

MEMORY MAP UNDER CP/M

Address	
	FDOS
B200	CCP
A900	TPA
0100	System Parameters & bootstrap
0000	

COMPUTER ANSWERS

Every month in PCW, Sheridan Williams assists readers with their hardware, software and systems difficulties. Some questions he deals with himself, other enquiries are directed towards members of his consultancy panel.

STOCK DISCS

I have been told that I should *not* look at cassette based micros for business purposes. I do not see why, as even a C60 cassette should in my estimation be capable of holding more than 50,000 characters. My application is for stock control, and I would be unlikely to have more than 1000 items on file at any one time.

There are many reasons for rejecting cassettes in favour of discs.

1 Even the 5¼ inch mini-floppy discs hold more on each side than the average cassette. I have seen figures from 70K to 350K quoted. This would save you having to change cassettes in order to swap between programs and files.

2 You can hold many (usually up to 40) programs and files on a single disc, and they will be instantly retrievable.

3 Cassettes can generally be read at between 30 and 300 characters per second. Discs can be read at around 10,000 ch/s upwards. To read a 50K file from cassette could take up to 20 minutes, and yet only around 10 seconds from disc.

4 Discs tend to be more reliable. . .this is because of the nature of the Philips cassette format and the cassette drives used. If digital cassettes with full logic control were used, this statement would not be true and also search times would be significantly improved.

5 Discs are a 'direct access' medium, whereas cassettes are 'serial access'. The advantage of direct access is that *any* record on a file is available for immediate use; in order to access the 1000th record from tape the previous 999 must be read and discarded. With disc the read/write head can be moved directly to the relevant track.

When discs are used, program packages may be written as a 'suite' of programs — one program calling the other when required. It is preferable to write many small programs rather than one large one as each can be worked on and developed separately.

Even if discs are used in a serial access mode (and there are many applications suited to serial access), they are considerably faster than cassettes.

I suggest that you follow this by reading more on the subject of files. I have only just brushed the surface on one absorbing aspect of programming.

Sheridan Williams

GIVE ME PROBLEMS

Is it possible for me to solve anything on my micro that no-one else has solved? I have found that I much prefer to program mathematical routines than ones related to data processing. Can you suggest any programs or ideas that I can look into?

You are obviously a person after my own heart. I agree that there is something very absorbing about manipulating numbers and expressions. However, I doubt whether you could do much with your micro as most of the pioneering work is carried out on machines that are a great deal faster. Don't let me put you off though; try and concentrate on finding better algorithms to solve common problems.

For example, to date I think that the highest known prime number is 2↑23209-1; it is known as the 26th Mersenne prime. It took 8 hours 40 minutes on a CDC Cyber 174 just to prove that it is prime! That's a good starting point. . .try to do it more quickly with a more efficient algorithm. Don't waste your time in BASIC, however, unless you have a BASIC *compiler*.

Look through past editions of PCW and find competitions set by myself. These will provide you with ideas on programs. In the meantime how about looking deeper into the Ackermann function. This can be stated very elegantly by the following recursive definition:—

A(0,n)=n+1
A(m,0)=A(m-1,1)
A(m,n)=A[m-1,A(m,n-1)]

Try building up a table for its values, and then try and find a formula for each row: ie. A(0,n)=n+1, A(1,n)=n+2, A(2,n)=2n+3. (You can do this one in BASIC.) Can you define a function recursively in your version of BASIC? In fact, better languages for this would be ALGOL or PASCAL.

You will uncover further reading on the above two problems in *Dr. Dobbs Journals* of June/July 1979 and August 1979. Good luck, and write back with your findings.

Sheridan Williams

RANDOM CONFUSION

What is the point of 'random access' files? If the files are random, how do you know where each record is stored?

I think the reason that you are confused is because of the word *random*. I prefer the term 'Direct access' to 'Random access'; the two terms are synonymous. I can only imagine that the term *random* access was coined because it does not matter in which order you access the records in the file. I much prefer to think of the file as a *direct* access file because you can access any record directly without first having to read all the previous records.

Your question about how do you find a record — this is answered fairly simply now. You only need a way of linking the 'key field' in the record to the disc address. This is known as the 'randomising algorithm' (there's that word again). The disc operating system usually takes care of the track and sector numbers, and all you have to do is work out the relative address (relative to the start of the file and record length). An example would be if you had a file of part numbers. For certain goods you could make your part numbers run from 0001 to 9999 say, and hence part number 1234 would be found at record number 1234.

Problems arise where the key field is a name. Where on a file of 26000 people would you place SMITH. Well, if the file is fairly well balanced, one idea is to start each letter of the alphabet at intervals of 1000 records, and each second letter in the name could start at 1000/26 intervals. Hence Smith would be placed at a record calculated by 19x1000+13x38=19494 (S=19th letter, M=13th). This is just one of many ideas, although obviously it can be wasteful of space.

Sheridan Williams

WHAT ARE THE PROSPECTS

I read your magazine regularly, and although I don't have my own system, I do intend to join the club one day. My immediate problem is my son. He is approaching 16 and will be leaving school next year with O levels (I hope). He expresses an interest in a computer career on the software side. What are his options as far as i) the course he should follow after school and ii) his choice of jobs within computing? And, as a matter of interest, how do the salaries compare?

Your son has a great deal to think about, and I would recommend that he talks to people, visits local colleges and libraries, and reads as much as possible.

As far as job choices are concerned, broadly the staff categories within a computer department are Systems Analyst, Programmer and Computer Operator. There are other categories and even subdivisions, but let us leave it at these three. Systems analyst is really only open to those with at least 5 years' experience and the approximate salary starts at £6500. Next comes a programmer; it's often from programming that people progress to systems analysis.

This is probably the best career to aim for; programmers' salaries start at £4500 for a trainee and can be as much as £250 per week for freelance contract work. A computer operator is the next category to aim for and it's worth saying too, that many companies give their operators time off from work to train as programmers. Computer operators often work shifts, and as such get shift allowances, but in general their salaries start at £4000.

The question of courses must really be dependent on whether your son wishes to follow a career in scientific computing or data processing. If he desires the former then the best course of action is probably A level computing science followed by a degree. If he seeks the latter then this reduces to a further question . . .degree or not degree (pun intended). You can gain some very valuable experience by stopping at the stage of A level, City & Guilds 746/747, Royal Society of Arts Computers in Data processing, a Threshold scheme, or a British Computer Society award, and getting a job immediately; in the three years that you would have been studying for a degree you could have become very knowledgeable in a purely practical environment. Please seek further advice as there are many points that I have not covered. However, I hope that I have given you a starting point on which to base further questions.
Sheridan Williams

(We feel that we should mention the almost universal misuse of the term Systems Analyst. Sheridan is quite right when he says that this is often the next step for a programmer — it is, but it often comes as a disappointment. To illustrate why let's pretend that there is another progression for a programmer — to Systems Designer. This job would involve designing a computer system based on a statement of the business requirements of that system. This statement of requirements would be produced by someone who had studied the existing system in detail — usually by thoroughly interviewing users of that system and documenting the results. This would by followed by an analysis of the findings in order to establish the precise requirements of the system.

You can see that the skills required for the two jobs described are quite different, yet they are frequently given the same description — systems analysis. The first, I imagine, would be a very satisfying next step for a programmer. The second may be extremely successful but, as well as a logical and analytical mind, it would also require a number of interpersonal skills which are not a natural adjunct to programming. Ed.)

BIG AND BEAUTIFUL

My computer has BASIC and it makes a great programmable calculator, but when I try to write large programs I always get tangled and can't get the program to work the way I want. How can I learn to write big programs that work?

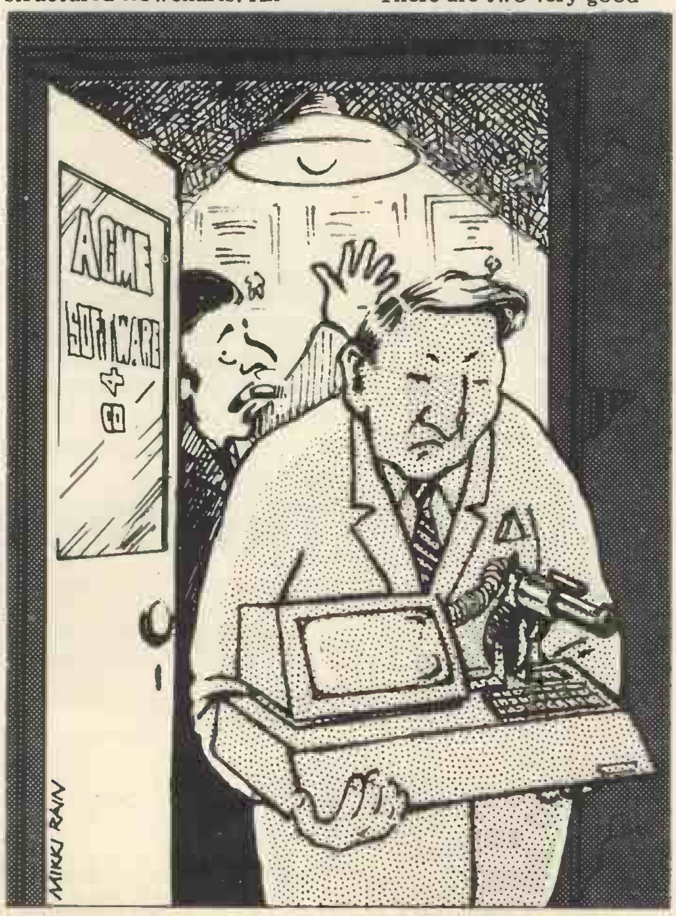
The mistake most beginners make is to jump straight into coding a program without designing it first. The problem with BASIC is that it makes coding very easy, but gives you no help at all in designing programs that are likely to do what was intended, or to be easy to test. Worse, beginners are usually taught flowcharting at the same time as BASIC, and though a well drawn flowchart is excellent for telling you which bit of the program is connected to which, it can still leave you clueless about the relationship between the program and the problem it was meant to solve. The art of programming is being able to go from a clear statement of the problem (a game, a calculation, handling a file), through several stages of refinement and definition, to a set of small, intelligible routines executed in the right order.

Many programmers use a structured programming technique to help them analyse a programming problem and record their stages of progress. There are several different forms. Some use pseudocode, a written problem definition language that looks like PL/1 or Pascal; some use a more pictorial technique such as a Warnier-Orr diagram, which brackets successive levels of the problem; some use structured flowcharts. All methods are based on the same theory — that correct programs have three components: a set of input data, a set of output data, and a process to convert one into the other. If the process is too complex to comprehend in one go, it can be decomposed into simpler processes by applying straight forward rules.

1 There are only three basic processes; sequence — input process output; decision — IF condition THEN sequence A ELSE sequence B; iteration — WHILE condition REPEAT sequence. 2 Any sequence block can be decomposed into two sequence blocks, or a decision block, or an iteration block. By using this technique the programmer is able to concentrate his efforts on one area of the program at a time (knowing the relationship with other parts) and push each area in turn towards more detailed definition, until he reaches a level from which he can code the final program. By this stage the design is complete and hopefully, most of the logic bugs have been discovered before a single statement is coded. The design stages have been recorded, so that the functions of the various sections of the program can be understood and tested, and there is a very good chance that the program will perform as intended as soon as the inevitable keying errors are eliminated. If it doesn't, then the design documentation enables you to backtrack and find out why.

There are two very good books for micro users:— *"Software Design for Microcomputers" by Ogdin (Prentice Hall, 1978)*; this will help you define program inputs and outputs, and design appropriate control structure for your applications. *"Microprocessor Programming for Computer Hobbyists" by Graham (Tab, 1977)*; this teaches pseudocode, plus a wealth of information on arithmetic, data structures, searching and sorting.

The American hobbyist magazines frequently have articles on program design and documentation:—
FLOWCHART — *Ellis* "Use of flowcharts to communicate" *Kilobaud, Feb 1979* (for basic use of flowchart symbols). *Dunn* "Structural Decomposition" *Interface Age June 1979* for structured flowcharting (strongly recommended).
WARNIER-ORR — *Higgins* various articles *BYTE Oct 77, Dec 77, Jan 78, Mar 79*.
PROGRAM DESIGN — *Hearn* "Top-Down Modular Programming" *Byte July 1978. Weems* "Designing Structured Programs" *Byte Aug 1978 Schwartz* "Pascal Versus BASIC: An exercise" *Byte Aug 1978*.
And may all your bugs be little ones.
L.S. Warner



"Sorry. . . but we already have 27 Russian roulette programs."

RAM DECODE

I found Mike Dennis' article "Practising a Little Microcontrol" most enlightening but I am not sure how to decode RAM. Is is the same as for ROM?

In a nutshell, yes! Remember, *any* device connected to the data bus must only respond to either a specific individual address or a specific band of addresses. Address decoding achieves this and *any* device can be decoded to respond to any address. However, since the address bus doesn't always contain a valid address, it would be foolish to decode the device from the address bus alone. For this reason, the CPU provides suitably timed control signals that are present only when the address bus contains a valid address — $\overline{\text{MREQ}}$ (Z80) and VMA (6800); These control signals must be gated with each uniquely decoded address to provide the unique Chip Selection that each device needs. Some micros discriminate further with control signals for either I/O or memory operations ($\overline{\text{IORQ}}$ and $\overline{\text{MREQ}}$ in the Z80). Other micros (6800 and 6502) do not and so any I/O port is simply treated as a specific memory address — all the devices are said to be "memory-mapped". There is no reason why the Z80 cannot be used in this mode as well.
Mike Dennis

SHAKESPEARE, BASIC AND THE CIA

Fingerprinting sentence structure

A few miles out of Washington, approaching Langley, Virginia, there is a sign over the highway. It reads 'C.I.A. Turn Right'. Shortly after making that turn a security barrier is encountered, and behind it a chain link fence. Identity documents are painstakingly checked against a list held by the guard, and your physical details verified with a computer housed in a large complex of buildings within the compound.
This is the first of a series of increasingly stringent checks that one meets on penetrating the heart of America's Intelligence machinery. And there, some six stories below ground, is a computer that plays with words. Exactly what this computer does, and indeed its very specification, remains a closely guarded secret. For at least part of the time, however, it is engaged in some fascinating literary detective work.
by Julian Allason.

Illustrations by Russell Mills

Literary detection is not a new science. Almost from the moment that Shakespeare was laid to rest, scholars have argued about the authenticity of various passages. In 1850 Spedding postulated that Shakespeare's disputed play, Henry VIII, was actually the result of a collaboration with Fletcher. This year Spedding's thesis was largely vindicated by Thomas Merriam — and a computer.

It was the Cold War, with its ceaseless propaganda battles, that generated the interest of the Intelligence community in computerised linguistics. Forgeries and plants abounded. They needed to know what was authentic — and what was not.

A celebrated case concerned the autobiography of Kim Philby, the Soviet double agent who had reached the top of the British Secret Intelligence Service. After his defection, a book entitled 'My Silent War' appeared, complete with foreword by Graham Greene. Philby claimed it was entirely his own work. SIS, knowing that it was a final attempt to smear them and damage Anglo-American relations, sent a copy to Langley. There, CIA specialists ran comparisons of 'My Silent War' with articles that Philby had written whilst operating as a Foreign Correspondent for the Observer. The tests showed that whole chapters had been written by others. The book is now regarded as highly suspect.

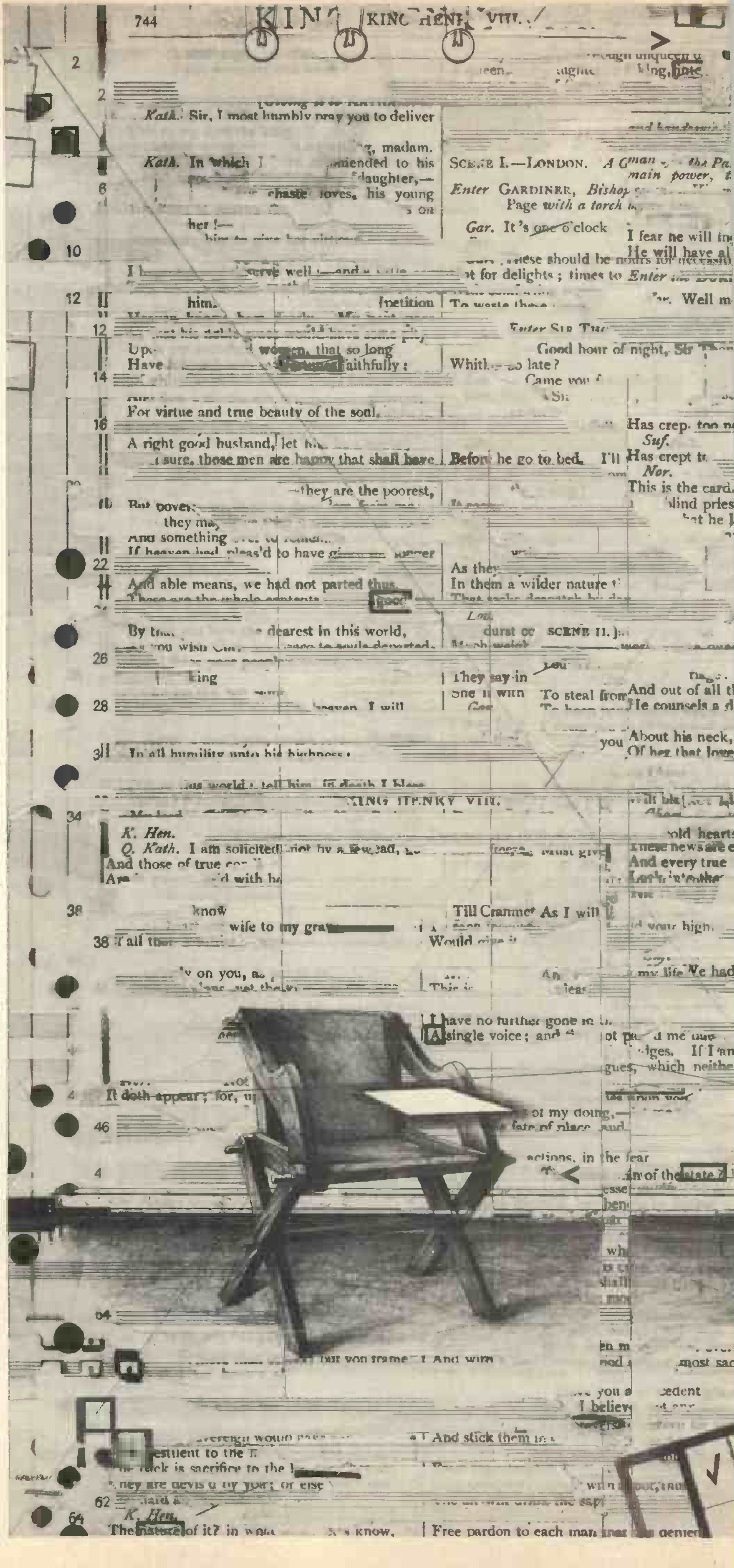
A similar thing happened when the memoirs of the West's top Kremlin agent appeared. Using similar methods, Soviet specialists swiftly "proved" the "Penkovsky Papers" a forgery. Penkovsky was shot and his book remaindered.

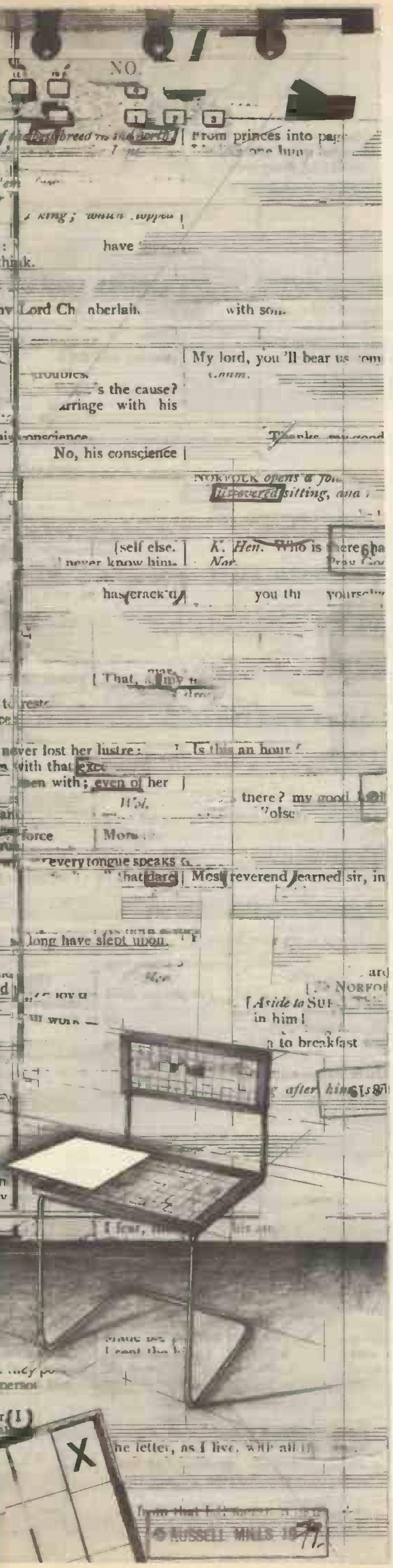
All methods of literary detection involve recognition techniques. The detective uses the computer to help establish an author's sub-conscious habits of speech or writing. Most Shakespearian scholars are capable of assembling a passable pastiche of the Bard's prose. To overcome vulnerability in this area, only very common "filler" words such as "and", or "it is" are tested. This is because use of these words is a matter of subconscious habit. Furthermore, they occur throughout written output whatever the mood or occasion. Position in the sentence is also held to be important.

Surprisingly, our syntactic habits are so ingrained that they show through even when an attempt is made to mimic the literary style of another. Usage of certain "filler" words remains fairly constant throughout a writer's work.

At the simplest level a literary detection program involves a series of string searches on text samples of established authorship. The incidence of certain strings, for example "such a", are noted. A profile of the author's literary style is then constructed. Similar tests are carried out on suspect text, and finally both profiles are compared. It should then be readily apparent whether or not all the samples were written by the same person.

Although fairly long samples of text are required for a definitive evaluation, it is possible to obtain reasonably accurate results from a short BASIC program. The routine I am working on for Petsoft uses less than 8K. The following simplified example illustrates a string search for the word "and".






```
100 DATA"HANDSOME ANDREW AND HIS WIFE"  : REM Text Sample
110 READ T$                             : REM Read Sample
120 FOR C=1 TO LEN(T$)                  : REM Set counter to no. of
                                          characters in string
130 IF MID$(T$,C,5)="  AND  " THEN S=S+1 : REM Tests next five characters
                                          (including spaces)
140 NEXT C                              : REM Increments Character Counter
150 PRINT " 'AND'APPEARED"S"TIMES"
160 PRINT"IN A STRING OF"LEN(T$)"CHARACTERS"
170 PRINT"ITS INCIDENCE WAS" S/(LEN[T$])*100 "%"
```

Note that five spaces are allowed for the string "AND" to avoid acceptance of "HANDSOME", "ANDREW" etc. An additional statement would be required to accept "AND" as the first word in a string.

In practice an expanded algorithm tests a much longer sample of text for the incidence and position of a number of such "filler" words and phrases.

At about the same time as the C.I.A. was trying to catch up on computerised literary detection they faced another problem. They needed English translations of all the scientific and technical information being published abroad. Their linguists could not keep up. Computers, it was argued, could provide the answer.

Early efforts at machine translation met with little success. The problem was the inadequacy of available syntactical analysis. Linguistics, the scientific study of language, was still in its infancy.

But in 1957, Professor Chomsky of the Massachussetts Institute of Technology, published a book called 'Syntactic Structures'. In it, he argued the existence of underlying or Deep Structures beneath the surface structure of the sentence. These defined and inter-related all the factors determining structural interpretation.

It is fair to say that not all linguisticians accept Chomsky's thesis. But it has given the machine translation and literary detection specialists a good deal to think about.

Computerised linguistics is now finding a much wider, and academically more respectable range of applications. In 1974 Dr. Andrew Morton created a legal precedent with his evidence that only 7 of 11 police statements submitted in a case had been written by the defendant. The result was an acquittal.

In a recent book *(Literary Detection, Bowker, £10.50)* Dr. Morton examined the difference between Jane Austin and The Other Lady, who in 1965 completed the novel which had lain unfinished since Jane's death. Although in literary terms the imitation is quite good. Morton demonstrated that the probability of Jane Austin having penned the 4,000 words that were written by The Other Lady to be more than one thousand million against (see chart).

With the continuing evolution of linguistics and the rapid pace of micro-processor development, it is reasonable to project not only considerably more accurate machine translation than my pocket Craig translator offers, but the prospect of an infallible literary detective.

Post script: Having run this article through my PET, the computer confirms that it is almost certainly not written by Shakespeare. . .

A comparison of Jane Austen and The Other Lady

Habit	Occurrences of the Habit in *Sense and Sensibility*	*Emma*	*Sandition* (Jane Austen)	*Sandition* (The Other Lady)	Chi squared (a)	(b)
an	25	26	11	29		
a + an	172	212	112	112	1.40	12.85
a	147	186	101	83		
P.B. *such*	14	16	8	2	0.20	3.92
and	253	299	151	154		
F.B.*I.*	12	14	12	1	2.45	6.84
the	270	271	229	221		
P.B. *on*	11	6	8	17	1.58	8.45
F.W.S.	22	26	19	8	0.43	6.34
this	32	39	15	15		
this + that	126	144	52	37	0.25	3.64
with	59	74	28	43		
with + without	77	84	38	47	5.02	3.71
very	37	68	26	27		
P.B. *the*	4	2	3	7	—	12.7

Notes: 1. The samples are: *Sense and Sensibility* — Chapters 1, 3, *Emma* — Chapters 1,2,3. *Sandition*, Jane Austen — Chapters 1,6. *Sandition*, The Other Lady — Chapters 12,24.
2. The figures for chi squared are for the comparison of the three genuine samples, (a), and then for the comparison of these samples taken together for the comparison with The Other Lady, (b).

If your Accounts are a problem- the solution could be on your desk.

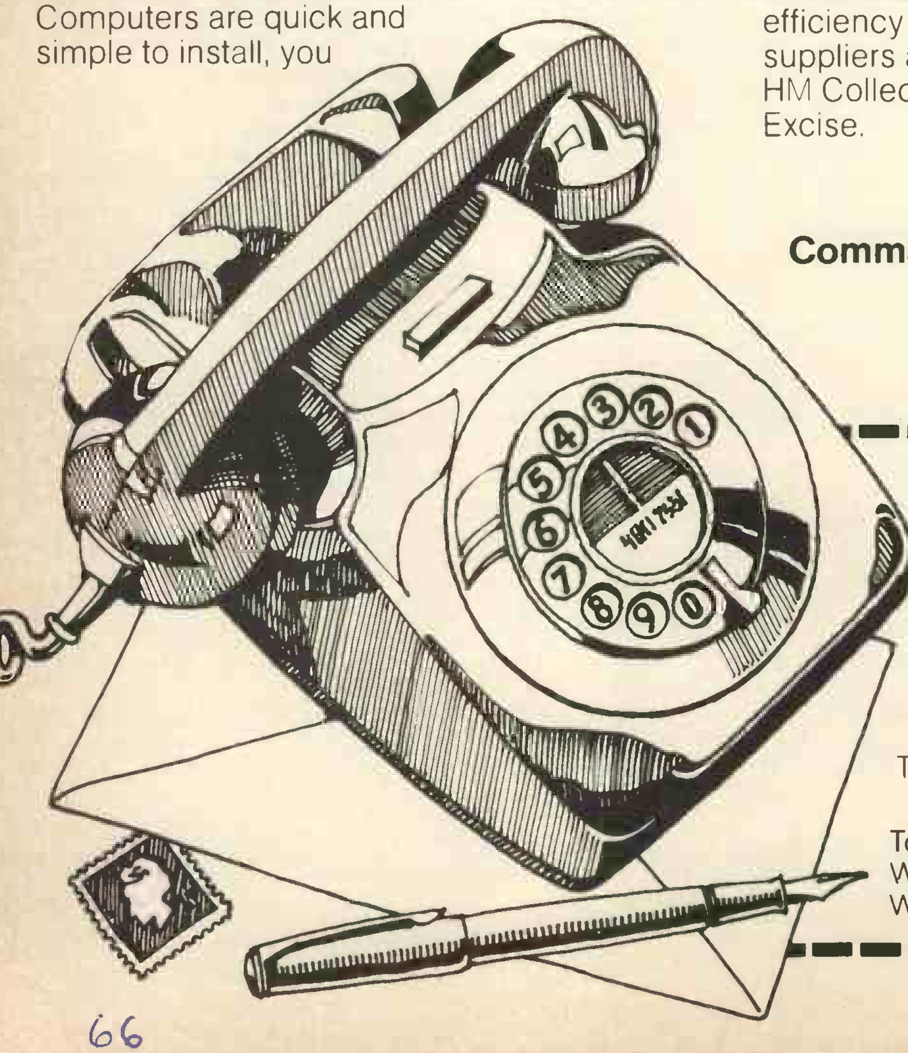
You could use your telephone to call Comma Computers on Brentwood (0277) 811131.

Or you could use a pen and envelope to complete and return the coupon to us.

Later on, you could have a free copy of our 'no jargon' brochure on your desk from which you could see how Comma Computers have used advanced micro-processor technology to make business computers easy to understand, use and afford!

Still later, but not much later because Comma Computers are quick and simple to install, you could have a Comma Aquarius, Aries or Leo business system on your desk – a complete system with computer, printer, keyboard, display, installation, 12 months maintenance and software to perform Payroll, Accounts, Invoicing and Credit Control applications and provide instant management information including Profit and Loss Statements and all from less than £6000.

Butchers, bakers, candlestick makers . . . can all, at last, enjoy the benefits brought to business by the silicon chip and increase the efficiency of dealings with customers and suppliers as well as accountants, auditors, HM Collectors of Taxes and Customs and Excise.



Comma Computers Ltd.
West Horndon
Industrial Park,
West Horndon,
Essex CM13 3XJ

Name ______________________

Position ______________________

Organisation ______________________

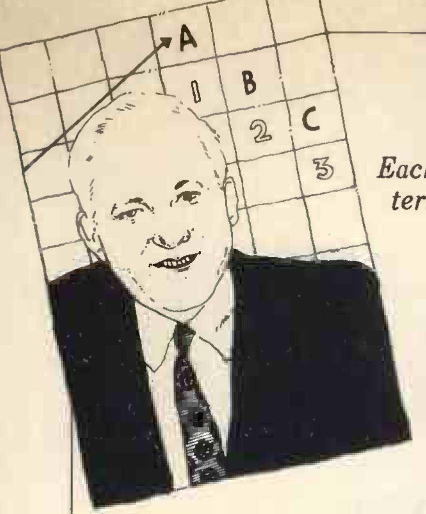
Address ______________________

____

Postcode ______________________

Telephone ______________________

To: Comma Computers Ltd,
West Horndon Industrial Park,
West Horndon, Essex CM13 3XJ



BUZZWORDS

Each month, Pete Reynolds takes us through the minefield of microcomputer terminology and jargon.

J

Jack Plug
A short rod-like connector whereby an electrical device (such as earphones) may be plugged into a jack.

JCL
Job Control Language — specifying (typically for some mainframe computer) the input/output devices and other environmental variables before a program or 'job' is run.

Jitter
Electrical instability, especially in the pattern of data displayed on a screen. Commonly due to inadequate voltage regulation.

Job
A set of data processing tasks, including running programs, for a specified application — for example, "weekly payroll".

Journal (tape)
A chronological listing, kept for backup, of all transactions and data entered to a computer system.

Joystick
A small control device whose knob may be moved in any direction within two dimensions in order, for example, to move a dot on a screen.

Jump
To depart from the normal sequence of program instructions. A jump instruction is often conditional. When a particular condition is satisfied (or not satisfied) the program instructions may be repeated in a loop, operating on each cycle on slightly modified data, until a predetermined count has been exhausted. The condition allowing the jump will then cease to obtain and program instructions will be followed sequentially.

Jumper
An electrical wire temporarily connecting two points on a circuit.

Justify (1)
To arrange printed (or typewritten) words so that the right-hand margin of each line forms a clear vertical line, as in most books and newspapers. There is usually no difficulty about achieving justification of the left-hand margin, where each line of type begins, but to achieve the same effect along the right-hand edge of every line requires that words and letters must, to some extent, be spaced out more than they need be; this calls for a count of the characters and spaces required in each line and a calculation of the number of extra spacing inserts required and where they can best be placed. This can be a minor computing problem in itself.

Justify (2)
To shift numeric characters to fill any spaces in the right-hand end of an accumulator or other area of computer storage, before an arithmetic operation.

K

Kilo (abbreviation)
Signifying 1000, as in kilometre or kilocycle. The internal storage of computers is commonly arranged by the manufacturer to hold a quantity of data which is some power of 2, for example, 4096 characters, bytes or words, which is 2^{12}. The convention is to refer to this number as 4K. 64K, sixteen times as great, actually amounts to 65536 (2^{16}). Note that the unit, bits, bytes or words is unstated. A '1K chip' probably holds 1K bits: to avoid ambiguity when referring to chips of greater capacity, the word size in bits may be shown after the K. Thus 1K8 means 1024 bytes, 16K1 means 16384 bits.

Kansas City Standard
A way of recording binary data on cassette tapes in which 1 is encoded as 8 cycles of 2400 hertz and 0 as 4 cycles of 1200 hertz.

Kb
1. Kilobytes, ie thousand bytes.
2. Kiloband, ie thousand cycles per second.
3. Kilobits, ie thousand bits.

K/c
Kilocycle (abbreviation). One thousand cycles per second; now known as KiloHertz (KHz).

KCS
Kansas City Standard.

Key
A pattern of digits used to identify an item or record.

Keyboard
The group of pushbuttons, as in a typewriter or calculator, whereby data or instructions may be input to a computer.

Key-to-disc/tape
A system for computer data entry on a commercial scale which was introduced to obviate the need for punched cards originally selected for data entry by companies already in the punched-card business. In a key-to-disc (or key-to-tape) system the data-entry operators write their data on the relevant magnetic media for subsequent processing in batch mode. A limited validation check may be made on the data at the time of entry.

Keyword
1. Same as key; a group of characters which identifies an item or record for data retrieval.
2. Same as password; a secret combination of characters which identifies an authorised user to a computer and may indicate which specific facilities are to be allowed or denied — eg to read data on the computer files but not to alter them in any way.

KHz
KiloHertz (abbreviation).

Kilo
Prefix signifying 1000 — but see entry under K.

KiloHertz.
A frequency of 1000 cycles per second.

KIPS
Kilo Instructions Per Second, describing the rate at which a processor can operate.

Kit
A set of parts for assembly by the user. It may not include case, power supply or connecting leads and may be more expensive than equivalent mass-manufactured systems (if such existing). But for those who enjoy assembly, a kit can be more instructive and satisfying and easier to modify to one's personal design.

Kludge
A local modification or patch in a computer program to overcome some error or design fault. Such patches make it difficult for others to follow the program or to deal with subsequent problems.

Knock-Off
A device for automatically inhibiting some machine activity in certain circumstances. For example, a paper knock-off will stop a printer when the paper supply is exhausted.

KSR
Keyboard Send Receive. Descriptive of a printing terminal, such as a teletype, having keyboard and printer but no other media (such as magnetic or paper tape) capable of sending or receiving messages without manual intervention (Automatic Send Receive, or ASR).

KVA
Kilo-Volt/Amps (abbreviation). A measure of electrical power, one KVA (or kilowatt) being equivalent, for example, to 4 amps at 250 volts or 5 amps at 200 volts. The consumption of one KVA for one hour (or 10 KVA for six minutes) is the familiar unit of the electricity bill.

L

L
1. Symbol for electrical inductance, eg of a coil, usually measured in Henries or Millihenries.
2. Low (state) in some bistable device.

Label
One or more characters used to identify the location of an instruction (when line numbers are not used) within a program. The process of compiling such a program will replace each label with an absolute address.

Lag
Delay between two successive events, such as reading a program instruction and completing its execution. The lag may be measured in clock cycles and knowledge of the interval used to advantage in advanced programming.

Land
An internal electrical connection, eg. between an LSI chip proper and one of the pins in the package inside which the chip is supported.

Language
Term used to describe a coding system by which instructions may be given, intelligible to a computer; for example, assembly language, BASIC, COBOL.

Large Scale Integration
The fabrication on a small silicon chip of a circuit embodying several hundreds of semiconductor devices (normally between 100 and 10,000)

10 PART PASCAL SERIES

THE COMPLETE PASCAL

BY SUE EISENBACH AND CHRIS SADLER

CHAPTER 3 CONTROL STRUCTURES: 1. LOOPS

In the last chapter, the procedure was presented as a means of performing the repetitive tasks so often required in computer programming. Thus program WALKING executed in "steps" LEFT and RIGHT alternately by successive calls to the procedures of those names. Some programs however have to repeat their procedures a large number of times, the precise figure often depending on conditions arising within the data or during the calculation, and hence not known in advance. In order to deal with these requirements, a programming device known as the loop *exists in almost all languages.*

The function of the loop is to cause the execution of certain lines of code (the *body)* a certain number of times. Different types of loop may be distinguished by the way in which they decide how many repetitions (or *iterations)* are required. The process of deciding whether to repeat the body of the loop one more time or to continue with the rest of the program is called a *test*. Every loop therefore consists of a body and a test and is known as a *control structure* because it causes the program control or "flow" to differ from the normal sequential execution of program statements.

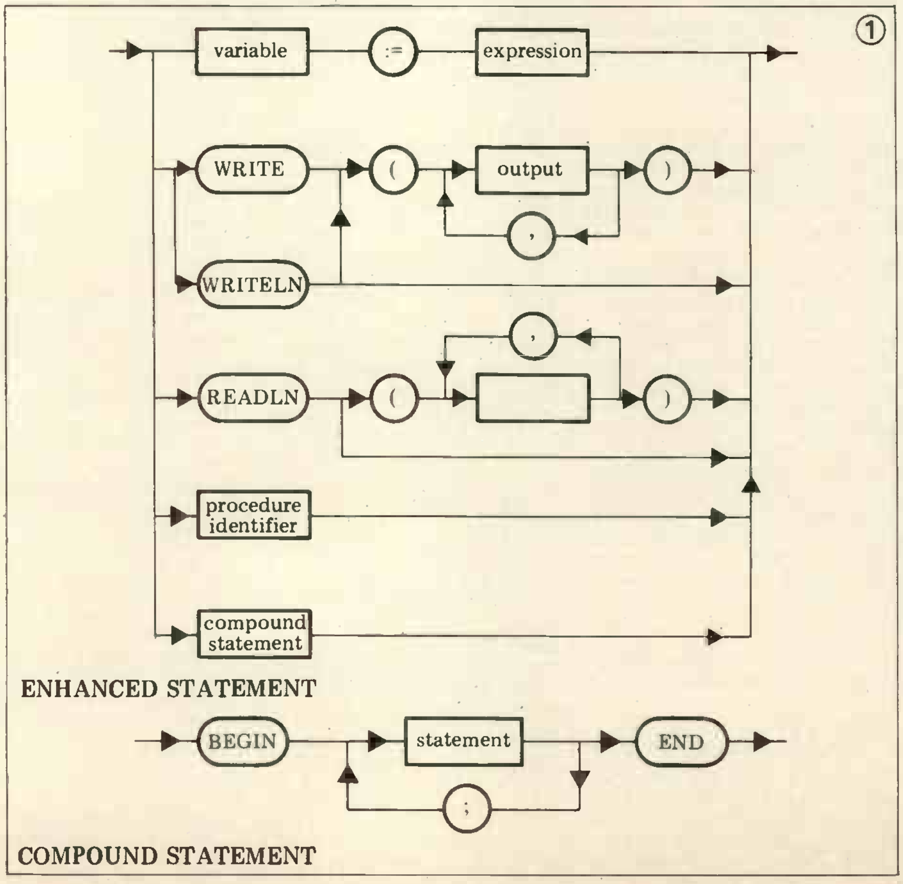
The most elementary type of loop is designed to execute the body a predefined number of times. This operation is controlled by an explicit *counter* variable and the test consists of comparing the value of the counter with the known finishing value. Depending on the outcome of the test, the counter is incremented (or sometimes decremented) and the body is repeated, or else program control passes to the code immediately beyond the loop.

In BASIC this structure is known as a FOR-NEXT loop and PASCAL has an equivalent called the FOR-DO loop. In addition, PASCAL has two loops for executing the body an unknown (or at least uncalculated) number of times. Here the test will depend on conditions arising within the body and a counter, if used at all, is not an explicit part of the loop. In the WHILE-DO loop, the test is made before the body is commenced whereas in the REPEAT-UNTIL loop, the test comes right at the end of the body. In the next few sections each of the above will be described, defined and exemplified in programs.

The body of a loop consists of either a single statement (now expanded to include the *compound statement*, as in the syntax diagram in Box 1) or in certain cases, a sequence of statements. When laying out a program it is normal to indent the code between every BEGIN-END pair. When the body of a loop does not contain a BEGIN-END pair, however, by convention it is indented anyway, to emphasize that it is controlled within a loop.

The FOR-DO Loop

Program ROLLOVER in Box 2 illustrates a FOR-DO loop in a fairly typical context. Procedure RESTOFVERSE contains the parts of the song which are repeated in each verse. The loop, set up in line 11, ensures that the part that changes (CROWDS) is correct for each verse. This requires the special DOWNTO reserved word to make the counter work backwards. Lines 13 and 14 actually produce each verse and line 15 sends the program control back to line 11 for the next verse — and so on. Line 16 finishes off the song. Lines 12

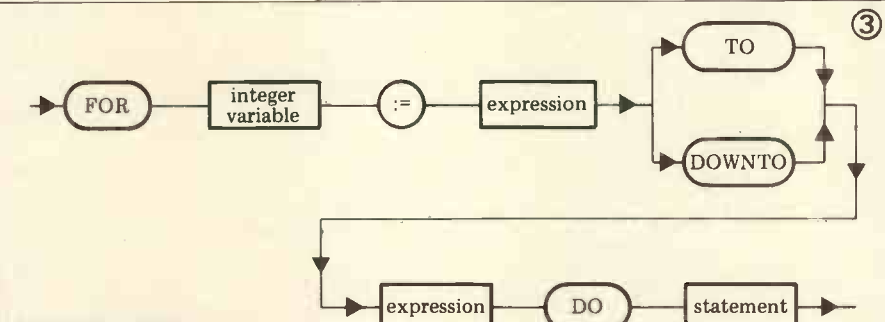


ENHANCED STATEMENT

COMPOUND STATEMENT

```
②
1   PROGRAM ROLLOVER ;
2   VAR CROWDS: INTEGER ;
3   PROCEDURE RESTOFVERSE ;
4   BEGIN
5     WRITELN(' IN A BED AND THE LITTLE ONE SAID') ;
6     WRITELN('''ROLLOVER, ROLLOVER''') ;
7     WRITELN('SO THEY ALL ROLLED OVER AND ONE FELL OUT.') ;
8     WRITELN
9   END ; (*RESTOFVERSE*)
10  BEGIN (*MAIN PROGRAM*)
11    FOR CROWDS:=10 DOWNTO 2 DO
12    BEGIN
13      WRITE('THERE WERE ', CROWDS) ;
14      RESTOFVERSE
15    END ;
16    WRITELN('THERE WAS 1 IN THE BED AND HE SAID ''GOODNIGHT''.')
17  END
-----------------------------------------------------------------
18  THERE WERE 10 IN A BED AND THE LITTLE ONE SAID
19  'ROLLOVER, ROLLOVER'
20  SO THEY ALL ROLLED OVER AND ONE FELL OUT.
21
22  THERE WERE 9 IN A BED AND THE LITTLE ONE SAID
23   ...
..   ...
..   ...
54  THERE WAS ONE IN A BED AND HE SAID 'GOODNIGHT'.
```

PROGRAM ROLLOVER



FOR-DO LOOP

to 15 provide an example of a compound statement. Finally, note PASCAL's solution to the problem of printing a ' mark. Since the quote (') is the text delimiter, the PASCAL compiler searches for pairs of quotes enclosing text. Two adjacent quotes will indicate that the text is not to be terminated but rather that a single quote is required for output.

The syntax diagram in Box 3 shows the precise structure of the FOR-DO loop. The different components appear as:

FOR (test) DO (body)

The counter is a variable (*not* a REAL) and must therefore, like any other variable, be declared explicitly in the declaration part. The starting and finishing expressions must be integer expressions. Because these expressions are evaluated before the loop commences, rather than during each iteration, there is no loss of efficiency in using quite complex expressions if required.

The counter increases or decreases (depending on whether TO or DOWNTO, respectively, is used) by 1 on each iteration. The restriction of the step size creates a loop-test requiring a minimal number of machine-code instructions. If a different step size is required, a "dummy" counter can be constructed within the body of the loop, but on no account should the value of the actual counter be changed inside the loop (for obvious reasons). The FOR-DO loop test will discontinue the loop when the value of the counter moves beyond the finishing value (in the indicated direction). This ensures not only that the body is executed the correct number of times, but also, if the counter is accidently set up to move away from the finishing value, the body of the loop will be skipped over entirely.

When the loop has finished the counter variable loses any value it had (i.e. it becomes *undefined)*. This feature is included in PASCAL as a safety measure to guard against the tendency of some programmers to re-use a loop counter at a later stage of the program, without assigning a new value to it.

EXERCISE:
Write a program to print out the song "Ten Green Bottles".

The Generalized Loop

Circumstances can often arise in programming where the use of a fixed-limit FOR-DO loop is too restrictive to allow for a fluent program style. As an example consider the problem of entering a list of numbers from a keyboard into a program. If you don't want to count how many numbers there are before you start, you need to have a way of telling the program when the list has come to an end. This is usually done with a "rogue" value — a number which couldn't possible be a part of the list (eg. -9999). When the program detects the rogue value, this is an indication that the input list is complete and further processing can continue.

It would be nice to place the item-by-item reading of such a list in a loop, but if the length of the list is unknown, then the only way of doing this with a FOR-DO loop leads to awkward and error-prone code. Because circumstances such as this arise quite frequently, PASCAL has a more generalized loop form.

The distinguishing feature of the generalized loop lies in the nature of its test. Instead of a steady incrementation of a counter, the test checks the validity of some relationship which is (presumably) affected by the body of the loop. When the relationship holds, one course of action is taken and when events within the loop cause the relationship to change, a different course of action is embarked upon. Quite clearly, only two possibilities exist — the relationship holds or it doesn't (i.e. it is *true* or *false*). Such a relationship is called a *Boolean expression* after the English mathematician George Boole who first studied the algebra of such expressions.

The syntax diagram in Box 4 fully defines the Boolean expression. Note that <> stands for "is not equal to". Consider a Boolean expression like A=B. This expresses the relationship "A is equal to B" and the = is known as a *relational operator* as are all the other symbols shown in Box 4. Compare this with the *assignment statement* A:=B which reads "A *becomes* equal to B". Here := is an *assignment operator* and it is this distinction which enables one to write X:=X+1 in a program where it would make no sense as an equation.

PASCAL provides two versions of the generalized loop. In the first, the WHILE-DO loop, the test is made *before* the body is commenced, and iteration occurs as long as the Boolean expression is *true*. If the expression is false when the program first encounters the loop, the entire loop will be skipped. The syntax diagram in Box 5 defines a WHILE-DO loop. As with a FOR-DO loop, the body is a single statement, generally compound.

The program in Box 6 illustrates the use of a WHILE-DO loop, which runs from lines 10 to 15, line 10 containing the test and the rest comprising the body. While this is not a very practical sort of guessing game, it does show the unlimited nature of the loop which will go on asking for new guesses until the right number turns up. It also shows the major danger of the generalized loop — suppose the test never fails? The program will stay in the loop forever. For instance, suppose TARGET was 16 while CORRECT and GUESS were REAL instead of INTEGER, and CORRECT became 3.99999 (as often happens). Any integer value guessed could *never* pass the test. This can happen quite easily especially when dealing with the mathematical functions with which rounding errors are associated. Consequently, it is good programming practice to check explicitly for realisable loop tests.

Examples of mathematical functions appear in line 6. SQRT(A) is a REAL value representing $\sqrt{A}$ while TRUNC(B) is the largest integer less than B (when B is positive). In line 6 the above functions are *nested* so that CORRECT is the square-root of the largest perfect square less than TARGET. A list of all mathematical or *standard functions* available in PASCAL appears in the Look-Up Table at the end of this chapter.

The second generalized loop in PASCAL is the REPEAT-UNTIL loop defined in Box 7. The test comes at the *end* of the body and iteration occurs as long as the condition is *false*. PASCAL has two complementary loops to allow for a fluent programming style. Sometimes it will seem more natural to use a WHILE-DO loop and sometimes a REPEAT-UNTIL will suggest itself. In the latter case however, the body will be executed at least once, whatever state the Boolean expression is in, because the test comes after the body. Program ANOTHERGO in Box 8 illustrates the use of a REPEAT-UNTIL loop running from lines 22 to 26. Line 26 contains the test and the body lies above it.

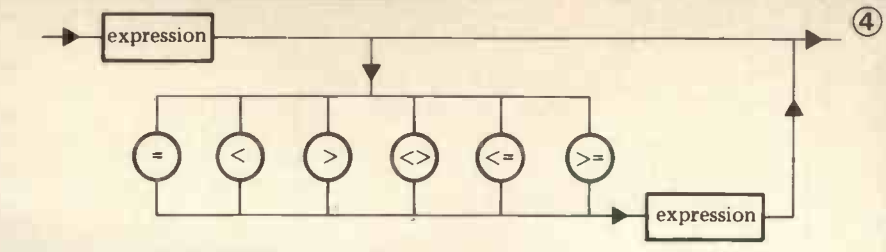
The REPEAT-UNTIL loop has

reserved words which effectively bracket the body of the loop. This is not the case with the other two loops where the reserved word DO merely leads up to the beginning of the body. The PASCAL compiler needs to know where the loop body stops and the rest of the program begins. It is for this reason that the two DO loops restrict the programmer to a body consisting of a single statement (usually compound). Without the DO keyword possessed by the other loops, the REPEAT-UNTIL loop can contain more than one statement in its body (cf. syntax diagrams for the different loops). This means that one tends not to find BEGIN-END pairs following a REPEAT although the indentation convention is observed nonetheless.

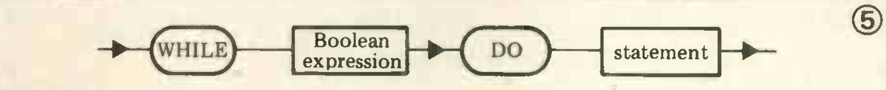
The program from Box 6 has been converted into a procedure for ANOTHERGO. This is a sensible way to develop programs — writing a small, self-contained section as a separate program, testing it, and then incorporating it as a procedure in some larger program. This theme will be developed in more detail in the next section. Finally, line 2 introduces a new *data type*, the character type CHAR which consists of a single letter of the alphabet, digit or normal keyboard punctuation mark. The variable ANSWER can contain any one of these characters and can be compared with actual characters enclosed in 'quotes' as in line 26. Variables therefore can be declared as INTEGER, REAL or CHAR.

Each of the three control structures defined above is an extension of the definition of a statement, since it appears in the action part of a program. Consequently a complete syntax diagram for the statement must incorporate all of these, and this is shown in Box 9.

EXERCISE:
Computers (and calculators) are often tested for accuracy by computing a range of nested mutually inverse functions [eg. exp (ln[x]) = x].
Write a program to input a sequence of (positive) numbers (rogue values could be 0 or less), in each case calculating exp (ln[x]) and outputting this value, together with x and the difference between them before reading in the next one.



BOOLEAN EXPRESSION

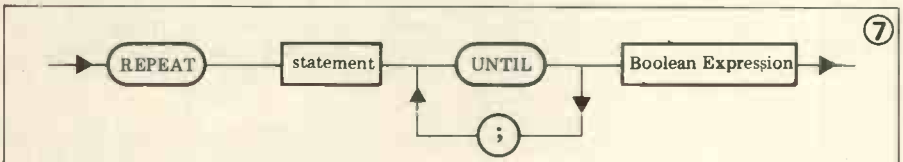


WHILE-DO LOOP

(6)

```
PROGRAM PERFECTSQUARE ;
VAR CORRECT, GUESS, TARGET: INTEGER ;
BEGIN
  WRITE('TYPE IN YOUR TARGET NUMBER:-') ;
  READLN(TARGET) ;
  CORRECT:=TRUNC(SQRT(TARGET)) ;
  WRITE('NOW GUESS THE LARGEST INTEGER YOU THINK HAS A SQUARE ',
        'NOT LARGER THAN ', TARGET, ':-') ;
  READLN(GUESS) ;
  WHILE GUESS<>CORRECT DO
  BEGIN
    WRITELN('NO THAT GIVES ', GUESS*GUESS) ;
    WRITE('SO GUESS AGAIN:- ') ;
    READLN(GUESS)
  END ;
  WRITELN('GOOD ', GUESS, ' HAS THE LARGEST PERFECT SQUARE ',
          ' NOT LARGER THAN ', TARGET,'.')
 END . (*PERFECTSQUARE*)
------------------------------------------------------------
 TYPE IN YOUR TARGET NUMBER:- 59
 NOW GUESS THE LARGEST INTEGER YOU THINK HAS A SQUARE NOT LARGER THAN 59:- 6
 NO THAT GIVES   36
 SO GUESS AGAIN:- 7
 GOOD   7 HAS THE LARGEST PERFECT SQUARE NOT LARGER THAN 59.
```

PROGRAM PERFECT SQUARE



REPEAT-UNTIL LOOP

PROGRAM ANOTHERGO (8)

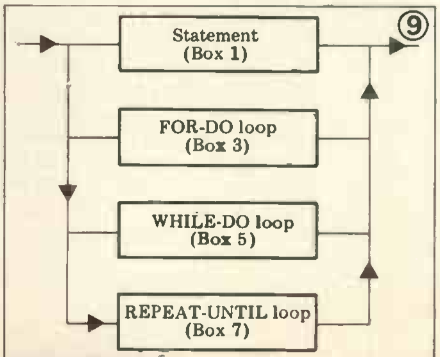
```
1   PROGRAM ANOTHERGO ;
2   VAR ANSWER: CHAR ;
3   PROCEDURE PERFECTSQUARE ;
4   VAR CORRECT, GUESS, TARGET: INTEGER ;
5   BEGIN
6     WRITE('TYPE IN YOUR TARGET NUMBER:-') ;
7     READLN(TARGET) ;
8     CORRECT:=TRUNC(SQRT(TARGET)) ;
9     WRITE('NOW GUESS THE LARGEST INTEGER YOU THINK HAS A SQUARE ',
10          'NOT LARGER THAN ', TARGET, ':-') ;
11    READLN(GUESS) ;
12    WHILE GUESS<>CORRECT DO
13    BEGIN
14      WRITELN('NO THAT GIVES ', GUESS*GUESS) ;
15      WRITE('SO GUESS AGAIN:- ') ;
16      READLN(GUESS)
17    END ;
18    WRITELN('GOOD ', GUESS, ' HAS THE LARGEST PERFECT SQUARE ',
19            ' NOT LARGER THAN ', TARGET,'.')
20   END ; (*PERFECTSQUARE*)
21   BEGIN (*MAIN PROGRAM*)
22     REPEAT
23       PERFECTSQUARE ;
24       WRITE('DO YOU WANT TO TRY ANOTHER TARGET?  ') ;
25       READLN(ANSWER)
26     UNTIL ANSWER='N'
27   END.
------------------------------------------------------------
28   TYPE IN YOUR TARGET NUMBER:- 44
29   NOW GUESS THE LARGEST INTEGER YOU THINK HAS A SQUARE NOT LARGER THAN 44:-
30  GOOD   6 HAS THE LARGEST PERFECT SQUARE NOT LARGER THAN  44.
31  DO YOU WANT TO TRY ANOTHER TARGET? N
```

Using Loops

As an everyday application of the use of loops, consider the construction of a mortgage repayment table. These are normally constructed by actuaries from formulae which give the monthly payment incurred by a loan assuming a fixed interest rate and where repayment occurs over a fixed time period.

This reputedly boring occupation seems ideally suited for rendering into machine soluble form, releasing the actuary for more valuable tasks (like estimating the insurance risks on a personal computer). Instead of employing the actuarial formula, however, the problem will be used to illustrate a common programming technique which consists of taking a guess at the likely value, working out the implications, comparing the results with the required outcome, improving the guess, working out the implications again, and repeating this process until an acceptable answer is reached. Clearly, the loop provides a means of programming such an iterative solution — although it's unlikely to tempt any actuaries away from their formulae!

The approach we shall take in programming this problem is known as "Top-Down Design". The Top-Down designer begins by explicitly defining the problem, stating what results are expected from what initial information. The task is then coded by calling several procedures, each a distinct subtask or *module* which contributes to the solution of the total problem. Any consideration of the detail of these



STATEMENT

```
1     PROGRAM REPAYMENTS ;
2     VAR MIN, MAX, LOAN, REPAY: INTEGER ;
3     PROCEDURE GETINPUTS ;
4               (*READ IN INTEREST RATE, NUMBER OF YEARS,
5                 MINIMUM AND MAXIMUM LOANS*)
6     PROCEDURE PRINTHEADINGS ;
7               (*PRINT OUT INTEREST RATE, NUMBER OF YEARS
8                 AND TABLE HEADINGS-I.E.LOAN & REPAYMENTS*)
9     PROCEDURE CALCULATEREPAY ;
10              (*WORK OUT MONTHLY REPAYMENTS*)
11    BEGIN (*MAIN PROGRAM*)
12      GETINPUTS ;
13      PRINTHEADING ;
14      LOAN:=MIN ;
15      WHILE LOAN<=MAX DO
16      BEGIN
17        CALCULATEREPAY ;
18        WRITELN(LOAN, '      ', REPAY) ;
19        LOAN:=LOAN + 1000
20      END
21    END. (*REPAYMENTS*)
```

(10) PROGRAM REPAYMENTS — FIRST ATTEMPT

```
PROCEDURE CALCULATEREPAY ;
VAR TOTALMONTHS: INTEGER ;
    MONTHLYINTERESTRATE, AMOUNTDUE: REAL ;
PROCEDURE TRYREPAY ;
          (*WORK OUT THE ACTUAL AMOUNT A GIVEN REPAYMENT
            WILL ACTUALLY PAY OFF*)
BEGIN
  MONTHLYINTERESTRATE:=INTERESTRATE/12 ;
  TOTALMONTHS:=12*YEARS ;
  REPAY:=LOAN DIV TOTALMONTHS ;
  REPEAT
    AMOUNTDUE:=LOAN ;
    REPAY:=REPAY + 1 ;
    TRYREPAY
  UNTIL AMOUNTDUE<=0
END ; (*CALCULATEREPAY*)
```

(11) PROCEDURE CALCULATEREPAY

```
PROCEDURE TRYREPAY ;
          (*WORK OUT THE ACTUAL AMOUNT A GIVEN REPAYMENT
            WILL ACTUALLY PAY OFF*)
VAR MONTH:INTEGER ;
BEGIN (*CALCULATEREPAY*)
  FOR MONTH:=1 TO TOTALMONTHS DO
    AMOUNTDUE:=(AMOUNTDUE-REPAY)*(1 + MONTHLYINTERESTRATE)
END ; (*TRYREPAY*)
```

(12) PROCEDURE TRYREPAY

```
PROCEDURE GETINPUTS ;
CONST IMIN=2 ; IMAX=50 ;
      YMIN=5 ; YMAX=35 ;
      LMIN=5 ; LMAX=200 ;
PROCEDURE GETINTEREST ;
          (*READS IN INTEREST RATE BETWEEN IMIN AND IMAX AND
            CONVERTS IT TO A DECIMAL*)
PROCEDURE GETYEARS ;
          (*READS IN DURATION OF LOAN BETWEEN YMIN AND YMAX YEARS*)
PROCEDURE GETMIN ;
          (*READS IN, IN THOUSANDS, THE MINIMUM LOAN VALUE BETWEEN
            LMIN AND LMAX AND CONVERTS IT TO POUNDS*)
PROCEDURE GETMAX ;
          (*LIKE GETMIN, BUT FOR THE MAXIMAL LOAN VALUE*)
63 BEGIN (*GETINPUTS*)
64   GETINTEREST ;
65   GETYEARS ;
66   GETMIN ;
67   GETMAX
68 END ; (*GETINPUTS*)
```

(13) PROCEDURE GETINPUTS

```
PROCEDURE GETINTEREST ;
          (*READS IN INTEREST RATE BETWEEN IMIN AND IMAX AND
            CONVERTS IT TO A DECIMAL*)
BEGIN
  WRITELN ('TYPE IN THE RATE OF INTEREST AS A PERCENTAGE.') ;
  REPEAT
    WRITE ('A NUMBER BETWEEN', IMIN, ' AND', IMAX, ':-' ) ;
    READLN (INTERESTRATE)
  UNTIL (INTERESTRATE>=IMIN) AND (INTERESTRATE<=IMAX) ;
  INTERESTRATE := INTERESTRATE/100 ; (* % -> DECIMAL *)
END (*GETINTEREST*) ;
---------------------------------------------------------------
PROCEDURE GETYEARS ;
          (*READS IN DURATION OF LOAN BETWEEN YMIN AND YMAX YEARS*)
BEGIN
  WRITELN ('TYPE IN NUMBER OF YEARS FOR WHICH MORTGAGE WILL RUN.')
  REPEAT
    WRITE ('A NUMBER BETWEEN', YMIN, ' AND', YMAX, ':-') ;
    READLN (YEARS)
  UNTIL (YEARS>=YMIN) AND (YEARS<=YMAX)
END; (*GETYEARS*)
---------------------------------------------------------------
PROCEDURE GETMIN ;
          (*READS IN, IN THOUSANDS, THE MINIMUM LOAN VALUE BETWEEN
            LMIN AND LMAX AND CONVERTS IT TO POUNDS*)
VAR LOANMIN : INTEGER ;
BEGIN
  WRITELN('TYPE IN THE SMALLEST MORTGAGE YOU ARE INTERESTED IN,',
          'IN THOUSANDS.') ;
  REPEAT
    WRITE ('A NUMBER BETWEEN', LMIN, ' AND', LMAX, ':-') ;
    READLN (LOANMIN)
  UNTIL (LOANMIN>=LMIN) AND (LOANMIN<=LMAX);
  MIN := LOANMIN*1000
END (*GETMIN*) ;
---------------------------------------------------------------
PROCEDURE GETMAX ;
          (*LIKE GETMIN, BUT FOR THE MAXIMAL LOAN VALUE*)
VAR LOANMAX : INTEGER ;
BEGIN
  WRITELN ('TYPE IN THE LARGEST MORTGAGE YOU ARE INTERESTED IN, ' ,
           'IN THOUSANDS.') ;
  REPEAT
    WRITE ('A NUMBER BETWEEN', MIN DIV 1000, ' AND', LMAX, ':-') ;
    READLN (LOANMAX)
  UNTIL (LOANMAX>MIN DIV 1000) AND (LOANMAX<=LMAX) ;
  MAX := LOANMAX*1000
END (*GETMAX*) ;
```

(14) PROCEDURES GETINTEREST ETC.

modules is deferred to a later stage of the design. In due course, each module will undergo the same treatment and thus the problem devolves into a hierarchy of more-or-less independent sub-problems until a level is reached at which only elementary programming functions are required. At this point the final coding can be done quickly and accurately, and the result should be a well-structured program.

Returning to the mortgage table program, the problem definition could be:

Given the interest rate and a time period for repayment, create a table showing the monthly payment due over a given range of loans.

The input data required is therefore:
1. interest rate (% p.a.)
2. repayment period (years)
3. maximum and minimum loans required (thousands of pounds).

The output should be a list of loans from minimum to maximum in steps of £1000, showing monthly repayments. The interest rate and repayment period should also be displayed.

The next stage is to decide on the method of solution in order to code the main program. At this level the tasks that must be accomplished include reading in the user's parameters, printing out the appropriate headings and, for each loan from the minimum to the maximum requested, calculating and printing the repayment amount. At this stage, the means by which the calculations are to be performed do not concern us and neither are we interested in the details of getting the input data or printing out the heading. The calculations will have to be performed in a loop which will stop when the maximum loan value is reached. In Box 10, we have called procedures named GETINPUTS and PRINTHEADINGS to handle the initial part of the problem, and introduced a WHILE-DO loop (lines 15 - 20) to control the calculation and output of the table. Procedure CALCULATEREPAY will actually perform the calculations.

The declaration part of this first attempt includes all identifiers used in the main program. These include the integer variables MIN, MAX, LOAN and REPAY, together with the procedures GETINPUTS, PRINTHEADINGS and CALCULATEREPAY. Notice that these procedures have not been fully defined at this stage but merely contain a comment indicating what each will eventually do.

EXERCISE:
Try re-writing this first attempt with a FOR-DO loop instead of a WHILE-DO loop .

We have now completed the highest level of the program design and are ready to proceed to the next level. The three procedures will be tackled in the same way that the whole problem REPAYMENTS was approached. The question arises as to which of the three should be dealt with first. We prefer to start with the "Heart" of the problem — CALCULATEREPAY (Box 11). The problem definition of CALCULATEREPAY could be:

Work out the monthly repayment as follows — first guess an obviously low value and calculate how much that would pay off over the given time period, taking into account the interest charges. If there is still a debt by the end, the repayment value was not enough, so increase it and try again. Continue until the repayment amount pays off the loan.

Input data
1. duration of loan
2. interest rate
3. amount of loan

Output data is the calculated monthly repayment amount.

In the declaration part, the variables required in the calculation will have to be declared only if they are local to the procedure, since the global variables will already have been declared. Thus a check should be made that the input and output variables, YEARS, INTERESTRATE, LOAN and REPAY appear in the variable declaration of the main program. Some of these may be missing in a "first attempt" version and so should be incorporated.

To start coding CALCULATEREPAY the first step is to generate the working data from the input data. The repayment period, for instance, is in years but is here required in months, as is the interest rate. Therefore two new (local) variables TOTALMONTHS and MONTHLYINTERESTRATE must be declared and calculated. Next, the initial estimate should be made, in order to start the whole process off. Since repayments will be *increased* to improve the "guess", it is important to start with an estimate *below* the likely value. A reasonable first estimate would be the amount one would pay back interest-free. This is simple enough to code at this stage as can be seen in line 10 of Box 11. (Note that DIV has been used since REPAY is an integer. This program could be changed to give pounds and pence if the user were willing to trade some speed for such accuracy). Since the initial estimate must be too low, the next step should be to add £1 to the repayment and test whether that will pay off the loan.

The process of incrementing the repayment amount and testing will be repeated until a figure is reached which actually *does* pay off the loan. This has been coded in the REPEAT-UNTIL loop, lines 11 to 15, Box 11, but, just as this calculation was put off in the main program, so the job of calculating how much a given value of REPAY would actually pay off over the time-period is deferred to procedure TRYREPAY, which is the next problem to be tackled (Box 12).

The problem definition of TRYREPAY could be:

Evaluate how much a given value of REPAY would pay off over the given duration of the mortgage using the given interest rate, assuming monthly payments and the compounding of interest.

Input data
1. monthly interest rate
2. duration of loan (months)
3. value of loan (£)
4. value of repayment (£ per month)

Output data — amount of debt remaining when time period has elapsed.

What is owing at the end of one month? Suppose AMOUNTDUE contains the amount due at the beginning of one month and an amount REPAY is paid back. At the end of that month, the amount due will be (AMOUNTDUE - REPAY) + interest accrued during the month. This figure will become the AMOUNTDUE for the next month; for N months, this calculation should pass through N iterations.

This is coded in the FOR-DO loop, Box 12, lines 6 and 7. The only variable needed that has not been previously declared is the loop counter, which is declared locally in line 4. This completes the definition of TRYREPAY which, in turn, completes the definition of procedure CALCULATEREPAY.

Having coded CALCULATEREPAY we now know exactly what information GETINPUTS must obtain. The problem definition could be:

Read in interest rate, duration of loan and maximum and minimum loans (in thousands of pounds). Convert interest rate to a decimal (instead of percentage) and loan values to pounds.

Output data
1. interest rate (decimal fraction)
2. duration of loan
3. minimum loan
4. maximum loan

An input procedure should usually check that the data it accepts is reasonable and unlikely to cause the program to crash. For instance, if the repayment period YEARS were zero, then TOTALMONTHS would also be zero. But we divide by TOTALMONTHS in CALCULATEREPAY, so that apart from zero being an unreasonable figure for years it will also crash the program.

Box 13 contains procedure GETINPUTS. In the action part the four procedures GETINTEREST, GETYEARS, GETMIN and GETMAX are called. The declaration part lays down limits within which the input data should fall (lines 2 - 4). If one of these should later on prove restricting, it will be easy to change the CONST declaration.

The four individual input procedures (Box 14) are so similar that only one, GETINTEREST, need be considered in detail. Its problem definition could be:

Output a message asking for the rate of interest. Check whether the response is within the range of reasonable values. Keep asking until an acceptable reply is received. Then convert this number from a percentage to a decimal fraction.

Input Data
IMIN and IMAX — limits of "reasonable" interest rates (as a percentage).

Output Data
INTERESTRATE — actual required interest rate as a decimal fraction.

A REPEAT-UNTIL loop (lines 6 to 9) is used to accept input. The program remains in the loop until an acceptable figure is entered.

The other three input procedures are developed in a similar fashion. Note that in procedure GETMAX, the minimum value for a loan is not LMIN but MIN DIV 1000 — the actual lower limit obtained from GETMIN (line 34).

Finally, PRINTHEADING is tackled (Box 15). Its problem definition could be:

Clear the screen, then print out a title followed by the required interest-rate and the duration of the loan. Skip several lines and print the headings MORTGAGE (for the loan) and MONTHLY REPAYMENTS.

Input Data
1. yearly interest rate (%)
2. duration of loan (years)

Output Data — none as this procedure simply produces the headings.

Cont. on Page 81

(15)

```
PROCEDURE PRINTHEADINGS ;
          (*PRINT OUT INTEREST RATE, NUMBER OF YEARS AND
            TABLE HEADINGS-I.E. LOAN AND REPAYMENTS*)
CONST SPACE='                          ' ;
BEGIN
  WRITELN(SPACE, '**MONTHLY MORTGAGE REPAYMENTS**') ;
  WRITELN(SPACE, '--------- -------- ------------') ;
  WRITELN ;
  WRITELN('INTEREST RATE =', 100*INTERESTRATE, '% OVER ',
          YEARS, 'YEARS') ;
  WRITELN('    LOAN      REPAYMENTS') ;
  WRITELN('    ----      ----------') ;
  WRITELN
END ; (*PRINTHEADINGS*)
```

PROCEDURE PRINTHEADINGS

Look up table

PASCAL STANDARD FUNCTIONS

Arithmetic:		
ABS(X)	Absolute Value	Real or Integer
SIN (X), COS (X), ARCTAN (X)	Trig functions	Answer Real
EXP (X), LN (X)	Exponential, Natural Logs	Answer Real
SQR (X)	Square	Real or Integer
SQRT (X)	Square Root	Answer Real
Transfer:		
TRUNC(X)	Truncate	X real, Answer is integer part
ROUND(X)	Round to closest integer	

[ROUND(X) = TRUNC(X + 0.5) when X is positive]

COMPUTING JARGON

Control Structure
Counter
Compound Statement
Rogue Value
Boolean Expression
Relational Operator
Module
Iteration

PASCAL RESERVED WORDS

FOR
DO
TO
DOWNTO
WHILE
REPEAT
UNTIL
CHAR

UCSD Exceptions

ATAN(X) instead of ARCTAN(X)
Also LOG(X) is log to base 10.

EXERCISE SUMMARY

1. Ten Green Bottles
2. Accuracy Test
3. Mortgage Table

COMPUCOLOR II

Now available from:

Abacus Computers Limited
62 New Cavendish St., London W1
01-580 8841

"One of the best small systems we have reviewed" — Practical Computing

Prices from £1058 ex. VAT

and from

Micro-Media Systems
14 Chepstow Road,
Newport, Gwent.
0633 50528

Tand V Johnson
165 London Road,
Camberley, Surrey.
0276 62506

Trans Am
12 Chapel Street,
London W2.
01-262 0814

Protech Instruments
224 Selbourne Road,
Luton, Beds.
0582 596181

Mektronic Consultants
116 Rectory Lane,
Prestwich,
Manchester.
061-798-0803

Microshade
1 Patford Street,
Calne, Wiltshire.
0249 815757

Kestrel Computing
195 Bennetts End Rd,
Hemel Hempstead,
Herts.
0442 50706

ABACUS PASCAL SYSTEM

£3850 Excluding VAT

COMPLETE

PURPOSE BUILT

PASCAL MAINFRAME FULLY CONFIGURED AND READY TO RUN. FEATURING AS STANDARD: 60K RAM; 3 SERIAL / 3 PARALLEL I/O PORTS; TWIN 8" DISC DRIVES; 12 SLOT S-100 FRAME, PLUS THE

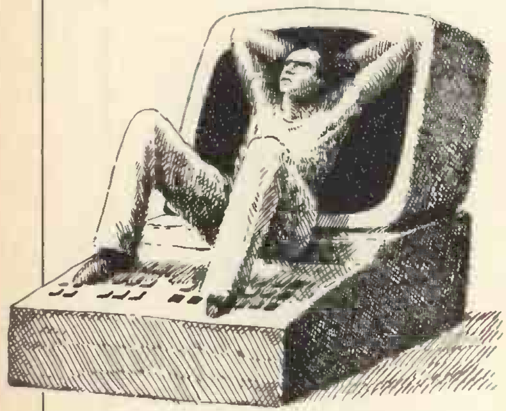
FULL UCSD PASCAL™

SOFTWARE PACKAGE, FULLY IMPLEMENTED BY ABACUS COMPUTERS LTD. IN THE UK.
FULL RANGE OF OPTIONS AVAILABLE.
MICROBASED EDITING TERMINAL AVAILABLE FOR £600.

Abacus Computers Limited
62, New Cavendish Street, London W1M 7LD
Telephone: 01-580 8841/2

Interrupt is the place in PCW where readers can unburden their grievances and air controversial views. New subjects are always welcome; the 'right of reply' shall be wielded at the discretion of the Editor. Please hold all contributions to within 800 words.

Future-What future?



I am beginning to have strong suspicions that our leaders and the media are actively conspiring to keep the real implications of information technology from the general public. Having always held BBC documentaries in high regard, I was disturbed to see the recent three part production "The Right To Work?" brilliantly obscuring the most important issues.

There was the predictable argument regarding the amount of unemployment which may occur, and how best to minimise it, and a tantalising, albeit superficial, glance at the role of leisure in utilising the spare man hours. Unfortunately, however, any good intentions there may have been collapsed into a familiar political squabble between Sir Keith Joseph and two TUC representatives. The problem was one of approach. "The Right To Work?" provided an excellent example of that little known but highly pervasive phenomenon, best described as 'temporal chauvinism'. This is manifest in the inability of members of a society to envisage any future society without imposing their own contemporary political and economic structures upon it. The best, latter-day example, would be the persistent attitude of unions and government alike towards unemployment; they see it as a social evil to be avoided at all costs.

The real question posed by the present technological revolution was summed up by Peter Large of the *Guardian*; "Do we want to work anyway, if our jobs are unsatisfying or unpleasant, if we are there just for the money? Can we rethink the work ethic and find another way, by whatever political means, of distributing wealth?"

In the present social climate, however, resistance to such ideas is very powerful (A Nation of Scroungers? reads the headline). Even aside from the indignant abhorrence of the average man in the street towards 'living off welfare', there are many respectable academics who would argue that, for the individual, unemployment causes depression and psychological instability. The latter is, of course, bound to be true in a society where a person's education and culture, together with the media and the inadequacy of many welfare payments, conspire to make the experience as miserable as possible.

Those of you familiar with the work of Ivan Illich will know that education can logically be considered as preparation for failure. The fact is that in national terms, educational institutions are designed to feed the needs of society. Thus, in a society where intellectual activities are accorded more status than manual, but where manual jobs greatly outnumber those requiring intellect, it is necessary to ensure that only a small minority of people finish their education feeling intellectually adequate. In order to achieve this end, examinations with a minority pass rate are held, and those who do not fall within the top twenty per cent or so are stamped 'failures'. Put simply, education is a subtle and apparently benign means of inducing people to accept their position within society.

"...it is necessary to ensure that only a small minority of people finish their education feeling intellectually adequate."

Many young adults thus enter both manual and clerical jobs accepting that they are unlikely to be satisfying in any real sense, but nevertheless with an intense fear of the ultimate failure — unemployment. The middle class college student, with some sense of intellectual adequacy and personal confidence, will find long periods of unemployment much more acceptable than the individual whose dignity relies heavily upon being able to run a car and buy a round in the pub. Thus we are brought up to think of our status in society almost entirely in terms of the job we do.

This is even reflected in linguistics; when asked "what do you do?" the words "for a living" are immediately implied, and one describes a job. If you were jobless you might say "I'm unemployed at present but. . . "; you would be unlikely to say "I look for a job one day a week, play basketball, read science fiction books and newspapers, go for walks and make love to my wife". Thus although many people are presently unable to accept unemployment — in the conventional sense — this is clearly a result of learning rather than any innate predisposition towards "employment" as such. It should therefore be possible, through changes in education and parental attitudes, to produce a situation where members of society are capable of achieving a high degree of personal satisfaction, regardless of their source of income.

But what about the profit motive, I hear you scream. Is not man really driven by greed, his career being merely an expression of an overwhelming desire to own more than his neighbour? How can we distribute wealth in a way which will satisfy this inner need without some kind of capitalist employment structure? Isn't the alternative a totalitarian state where nobody is really satisfied? True, one doesn't have to look far for confirmation of the intrinsic nastiness of human nature, and it would be foolish to suggest that this could change overnight, if at all.

Nevertheless, I cannot accept that mankind is incapable of achieving a higher degree of social awareness, or of maintaining a more egalitarian and humanistic social structure. Social evolution, which has long since superseded biological evolution as the major instrument of change in man, is difficult enough to understand — let alone control. The so called 'profit motive' is just one aspect of the complex relationship between the individual and his society which must be understood if we are to survive the hazardous future ahead of us.

I am not alone in the belief that we are entering a period of social change as dramatic and pervasive as the industrial revolution. This change will affect us all whether we choose to have any control over it or not. As individuals we can avoid the issue (and the headaches) by taking the view of Ron Condon (Editor of *Data Link)* that: ". . . as for the future, well, let it look after itself as it is so unpredictable anyway. . ."

I am sure, however, that many of you, feeling as I do, both excited and terrified by the social implications of information technology, will agree that we must at least attempt to direct the course of events if we are to avoid a catastrophic outcome. I am convinced that if we ignore the problems we will face inevitable social collapse and/or totalitarianism.

I have covered only a few of the most obvious points, and clearly there is much ground to be explored before any realistic plans can be made. Those of us already involved will have to carry the discussion across to the general public. Social change can only come from the people, since controlling bodies are, by their very nature, concerned only with social maintenance. Do you want your future to remain in their hands?

Dick Granby, Fitzrovia, London

Grow your own

From the way microcomputer hardware is sold these days you'd think that round every corner there were large forests of software trees tended by 'green fingered' programmers. In fact there's more likely to be giant sized briar patches tended by Weed Killer manufacturers.

"If we could only find skilled workers we could double our growth rate over the next four years".

"We've been looking for skilled workers for the past two years with no success".

These two comments were made by two exporters during recent television programs on the Malaise of British Industry.

It is, I suppose, a small comfort to see that the computer industry makes the same mistakes as the rest of British Industry. If you read the computer press, almost weekly you will see an article or letter bemoaning the 'lack of experienced programmers'. In fact, if you look at the job advertisements you see more and more companies offering larger and larger salaries to proportionally fewer and fewer programmers. On the other hand, one noticeable omission from the job advertisements is vacancies for trainee programmers. On those rare occasions when they do appear the response is normally overwhelming (one company reported 700 replies to one such advertisement).

As you can see our 'big' brothers in the mainframe business have already got a serious staff shortfall, (by the end of 1980 it is predicted that this shortfall will have reached 70% of the total requirements).

What are the prospects in micros?

One noticable omission from the job advertisements is vacancies for trainee programmers. On the rare occasions when they do appear, the response is normally overwhelming.

Let's look at the numbers first: in 1978 the average monthly sales volume of microcomputers was larger than the total worldwide number of all computers installed before that year. This fact alone seems to indicate that microcomputers are already in a disaster situation. So what can be done about it.

Solutions to the software problem, available now, are to either use standard packages or to custom build.

The package approach is the one which seems to have been adopted as 'standard'. Every month we see in the microcomputer press ever increasing numbers of software houses advertising ever increasing numbers of 'standard' packages. The problems associated with this are:

1. It gets increasingly difficult for small companies to evaluate these products. Many of the products on offer are poorly documented and little indication is given of their scope.
2. No joint standards have been agreed between software houses. So even if you get a package which meets your functional requirements it will need *customising* to interface with products from other software houses.
3. The products on offer are all *generalised* and in most cases, therefore, demand that you change your business to meet the requirements of the software rather than changing the software to meet your business needs.

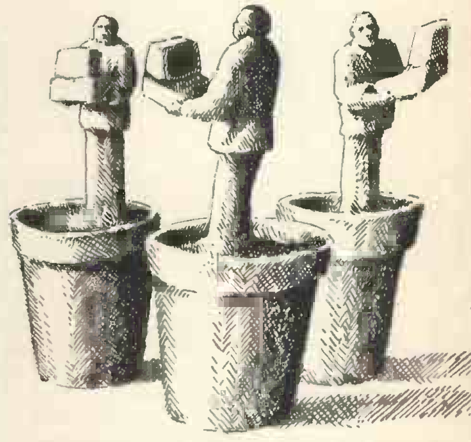
The custom build approach would seem to be ideal since you will get exactly what you ask for. The problems associated with this approach are:

1. The obvious problem of the small number of programmers available.
2. The high cost of programmers. They usually cost between £5 and £10 per hour, so a customised system could turn out as much as 6 or 8 times the cost of the hardware.
3. Programmers know a lot about programming but they don't necessarily know about your business.

In my view the only satisfactory answer to the growing software problem is to combine both of these approaches. What is needed is industry specialisation. You, the businessman, bring your knowledge of your industry and its problems; the programmer brings his knowledge of computers. Together you provide an ideal combination which will lead to a standard package approach, but, since the package is specific to one industry, it is likely to meet the requirements of most companies in that industry. Obviously no two companies are exactly the same so some customisation is always going to be needed, although, only to a small degree. In addition, the cost of producing an industry standard package is offset by the higher likely demand for that product in that particular industry.

This all sounds ideal, but some of you will have noticed the slight flaw in my argument. I started by saying that we have a major shortfall in programming resource; how can this be overcome?

Well, the answer is in your own hands. It takes two years to develop a good programmer but programmers can be productive after six months. The answer then is to train your 'industry' knowledgeable staff to be programmers. The better software houses usually run training schemes (e.g. 'BASIC' programming courses) usually lasting 3-5 days and costing approximately £50 per day (a lot of this 'expense' can be recouped from your own industry training board). These courses will enable you to provide your own programming, although they will need some direction. Once again, the better software houses should offer consultancy services so that the initial system design and program specification can be done by computer professionals with the bulk of the programming being done by your own staff. In addition, the same software house would probably undertake the marketing of the finished 'industry package' for you.



Illustrations by Ingram Pinn

In conclusion, one thing is certain — if you just sit there and do nothing the problem will not go away, it will just get bigger.

Mike Rose, Croydon

Micro-mania revisited

I would like to challenge Mr Smith's view of personal computing in PCW volume 2 Issue 5, Interrupt column.

Mr Smith — I think you missed the whole point of the hobby. I would like you to show us the magazine or newspaper in which such fantastic claims were made for it. Or did you make them up for rhetorical effect? Because there is no hobby which can truthfully profess to fulfil any of your 'claims'.

Are you addicted, bored, dehumanised? Moreover, are you unable to justify . . . etc? Surely you had some reason for buying yours in the first place, some motivation? Or did you get it because it was the latest executive toy? Anyone who buys anything for such a reason deserves to be bored out of his/her tiny mind.

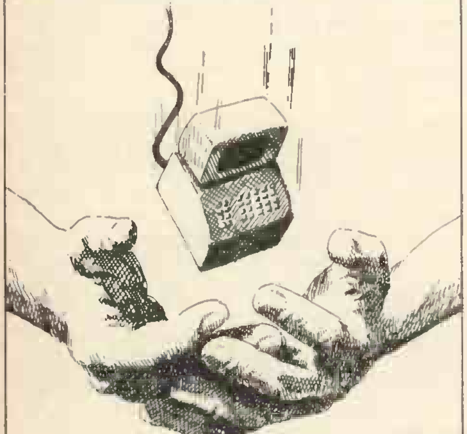
I am one of the genuinely poor people who cannot afford even the meanest computer kit. But already I have tens, even hundreds of ideas for my dream machine. None of them involves commercial equipment or software. The only reason I haven't tried to contribute to PCW is that I have not had the equipment to debug my ideas, and I doubt that the editor would enjoy reading any bug-riddled script, however valuable the core material.

As for your intro, we don't think of personal computers as liberating us in any sense. However the microprocessor *will* liberate a large proportion of the population, for a large proportion of their week, from the drudgery of work — whether in the typing pool or on a production line. Thus liberated there will have to be a cultural and educational renaissance, in which personal computers could play a great part.

The pocket computer you describe will follow the development of the plasma screen or else the ultra miniaturisation of existing LED/LCD screens. Whether Woolworths will sell them is a matter of speculation.

The only requirement for the 'mental annihilation' you talk about is a weak, even sick mind. No such mind could insinuate its owner into any position of responsibility. I must agree that computer art is boring. The reason is similar to the reason why musicians disdain 'disco' music, as being more or less speeded up Musak with heavy drums/bass overlaid. Computer art is constrained by:
Display and definition and colouring.
Computer speed and memory capacity.
Display medium; most have just a VDU and/or printer
Character graphics.
Could you produce a piece of art on a 25x40 grid with PET graphics which is not either totally random (sorry — pseudorandom) or documentatively symbolic? If so — you're a genius! In any case, judgement of art on any basis other than technical exactness has to be subjective. Similarly 'disco' music is constrained to a dominant beat surrounded in time by audible-but-not-too-distractingly-brilliant music.

Lastly to your question about defence spending. The only way this could be stopped is by multilateral agreement, which would in turn be followed by multilateral distrust, and by a surge in spending on surveillance techniques, and secretion techniques. In any case, suppose some goodly invention does come from defence research; that is, an invention which, had it been sought from any other point on the 'knowledge map', would have had minimal chance of being found without a huge amount of extra expenditure? The ultimate aim of a scientific explorer is not as significant as the route taken and what he finds on the way. Example: The whole 'space race' was sparked off by military interest. Why don't you send for a catalogue of the valuable spinoffs *that* produced?



If you ever decide to drop your machine from your window, inform me beforehand. I'll be there with a butterfly net, and I promise to decapitate the first new enthusiast I see who even nearly imitates your attitude.
T. Magee, Bradford.

SYSTEMS

PCW already has the most authoritative and comprehensive guides in the UK for hardware — namely our Benchtest and In Store contributions. Now, building on this success, we are pleased to introduce Systems. The brief for this new, monthly feature will be to take a different business application each issue and to report on some of the software packages available around it. PCW's David Tebbutt *and* Mike Knight *of* Mike Rose Micros *take up the explanation.*

Perhaps before looking at the fine detail of our approach for the future we should examine the reasons for introducing Systems.

You've probably seen or heard business packages described in glowing terms. They are said to be complete, comprehensive or total. Sometimes they are not described in any terms at all; sometimes they are described in terms which only the writer understands. Somehow the prospective buyer must decide from this morass of inadequate information, which packages to consider buying.

Nor do the problems end there. Having selected a few possible packages, the potential buyer needs to know quite a lot more before making any final decision.

Is it well documented, for example? We can barely believe some of the apologies produced in the name of documentation. It can be inadequate in a number of ways. First of all it may simply not exist. . .not even instructions for operating the machine! Secondly, yes, it may exist, but in such a form as to be totally unintelligible to mere mortals — not to mention the prospective buyer/user. Thirdly, it may exist, but only in parts. The missing sections are usually the ones you need when you're burning the midnight oil and all the 'experts' are fast asleep in bed.

An exaggeration? In many cases we think not, although we have to point out that some companies do produce quite excellent documentation.

And here's something else to think about — bugs. What are bugs? . . .well, in common parlance, they are errors existing in the application package which cause it to go wrong from time to time. Of course, ideally, one would like any problems resolved on the spot — time, after all, can be expensive. Here the difficulty may be that the firm from which you bought the package no longer exists. Perhaps (more likely) they aren't too interested, or don't have the staff to tackle any bugs. Again we don't want to paint an unduly miserable and pessimistic picture, but these are very serious matters and they need to be considered before any money is exchanged for software. For the businessman it could mean his business crashing down alongside the programs.

Okay, enough of the horror stories, time now to take a look at some constructive action.

Each month when we report on a particular application area, the feature will be divided into the following sections:
Objectives
Tasks and volumes
Evaluations
Comparisons
New products
Let's look at each of these in turn.

Objectives

In this section we shall define the objectives of the application. We shall also describe the application and explain any relationship with other applications. Failure to be very clear about objectives will lead any investigation to likely failure.

Taking 'payroll' as an example, we might describe the overall objective as 'to pay employees the amount due *on time* and to meet statutory requirements'. Then we might describe the application as follows:

1 Capturing information upon which payment will be based.
2 Using this information to calculate net payment.
3 Maintaining records of payments to each employee.
4 Producing appropriate documentation for company, employee and government records.

Finally, we might define the relationship with other applications as: 'information gathering — possibly the product of production hours recording. The payroll application will almost certainly create "transactions" for the accounting function'.

Tasks and volumes

In this section we shall select, say, three packages and match them against the tasks to be performed. Staying with our payroll example, we might say something like this:

"Not only will this give a guide to three particular packages, it will also offer a framework against

Tasks: Package	A	B	C
Create employee records	●	●	●
Delete (suspend) leavers		●	
Maintain existing records	●	●	●
Build up to gross	●		
Gross to nett		●	●
Print payslips	●	●	●
payroll	●	●	●
cash analysis		●	
cheques or credit transfers	●		
bank reconciliation	●		
NI stamp analysis	●	●	●
Update employee records	●	●	●
Prepare P60s			●
Produce accounting transactions	●		
etc.			
Maximum Volume/sizes:			
Employee records	400	250	600
Record size	180ch	360ch	200ch

which to measure other packages of your choosing."

Evaluations

In this section we shall again focus attention on the selected packages. This part of the feature will be written as a structured narrative, describing each package in turn. The main elements are as follows.

Availability
Documentation
System content
System maintenance
Costs
Hardware required
Support and training
User comments

Availability covers number of suppliers, their distribution and whether the product is available 'off the shelf'.

Documentation describes the scope, content and quality of the manuals and guides supplied.

System content will describe the programs involved in the package, their functions and certain aspects of their operation. For example, it may be that each program, on conclusion, automatically loads the next in sequence. On the other hand, there may be a need for a lot of disc or tape changing during the operation. We will try to give a picture of what will be involved in the day to day running of the system.

System maintenance. We shall be interested in whether the system has been designed to be changed easily. Examples which spring to mind are tax rates and discount terms. We shall also see if customisation is easy. Some packages are written with 'hooks' to enable customised routines to be added fairly simply. The language used is also important here. Finally, we shall check out who you have to go to to have these changes made.

Costs need little explanation. We shall give the costs for various versions of the package and, if applicable, the cost of any maintenance agreements.

Hardware required. We shall describe the different hardware configurations and relate these to the volumes which can be handled by each. We shall also give a guide to the hardware costs.

Support and training. If either of these areas are neglected, it's likely that you'll end up very disappointed with your new system. Training should, at the very least, teach you how to operate the system. Support is the on-going advice and guidance you will get from the supplier. It also covers their response to any problem you may encounter — a hardware fault, a software fault or perhaps an accident such as over-writing some important files. We shall assess the services offered.

User comments. We shall contact users of each system and summarise their opinions and experience of the package.

Comparisons

This section comprises a straightforward comparison chart showing all the packages notified to PCW, for the application in question. Each will be evaluated against the criteria discussed in this article. Because we cannot do an in-depth analysis of every package, this information will be based on that made available by the suppliers. If the publicity documentation fails to mention something, we shall not make assumptions and the column shall be marked N/A — not applicable.

New products

Finally, and quite separately to the above, we shall provide information on any packages newly notified, for application areas already covered.

We're sure that this structured approach to package evaluation will help readers in the selection of their business software. There are a lot of good and reliable suppliers of these products in the field, all of whom will give sound advice. But this series of articles, as much as anything, should help clarify your own thoughts on what can be a rather tricky subject.

From time to time, PCW publishes business case studies. This involves spending a few hours with a user of a computer and chatting through their experiences. Last month, you will recall, we featured a betting shop system. If you would like to tell the world about your system, be it a standard package or custom-built, then please get in touch — other people's successes (and failures) may offer invaluable information to businessmen working in similar areas.

PASCAL. . . continued from P.76

The coding for this procedure appears in Box 15. The entire program can now be gathered together, incorporating the extra global variables (INTERESTRATE and YEARS) into the declaration part of the first attempt (Box 10) and filling out the details of the different procedures as they have subsequently been designed.

EXERCISE:
Adapt REPAYMENTS to produce a table showing the 15 year, 20 year, 25 year and 30 year monthly repayment figures for a given range of loans. The input should be the interest rate and range of loans (and *not* the loan period) and the output should be a table with 5 columns — one for the amount of the loan and one each for each repayment period.

Conclusion

Loops control the repetition of a set of statements within a program. Every language needs a loop — PASCAL has three, which enriches the language and makes it versatile. Loops can be distinguished by the type and position of the loop test relative to the loop body.

Just as a program can be built up from basic blocks into an ordered structure, so can the data on which the program operates be organised into efficient and powerful *data structures*. The next chapter will serve as an introduction to these.

BUGGING THE 6800

This article presents a monitor program developed by John Moore for the 77-68 system (described in PCW vol. 1 issue 1). This is a very flexible computer system sponsored by the Amateur Computer Club and designed for home construction. It uses the Motorola MC6800 processor for which there is a wealth of readily available cheap and sometimes free software of good quality. The 77-68 hardware is usually configured with an interrupt driven parallel keyboard, and a memory mapped VDU with a Teletext compatible 24x40 format of upper and lower case characters. Users of other 6800 systems may also find ideas in this program that could be of use to them

For those unfamiliar with machine code programming and debugging, or the function of monitors, a little explanation may be in order. A monitor serves three main purposes:

1. It includes routines to give the computer access to the outside world through input/output devices such as keyboards, VDUs and printers.
2. It is used as the executive or operating system to allow the user to run his own applications programs. For example it might support a BASIC interpreter which in turn runs user programs written in the BASIC language. Some monitors can support more than one user program simultaneously.
3. Used for machine code program development, it allows the operator to examine and alter memory contents one by one, to run sample programs, and generally to get them into working order.

Usually monitors are held in ROM so that they are available and running as soon as the computer is started. This particular one is unusual in that after switch on it has to be loaded from tape into a hardware determined 1K byte block of memory (address FC00 - FFFF).

Although this is inconvenient it only takes 5 seconds at 2400 Baud, and it does allow for easy program modification to meet changing requirements. The 1K byte limitation meant there had to be some compromise between the three requirements outlined above so, if anything, the emphasis was in favour of program development capabilities. In particular MIKBUG, the original standard Motorola 512 byte monitor, and similar related variations do not support modern I/0 systems and have only slow and limited debugging facilities.

```
0000 00 00 00 00 00 00 00 00 00 00 00 00
000C 00 00 00 00 00 00 00 00 00 00 00 00
0018 00 00 00 00 00 00 00 00 00 00 00 00
0024 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00
003C 00 00 00 00 00 00 00 00 00 00 00 00
0048 00 00 00 00 00 00 00 00 00 00 00 00
0054 00 00 00 00 00 00 00 00 00 00 00 00
0060 00 00 00 00 00 00 00 00 00 00 00 00
006C 00 00 00 00 00 00 00 00 00 00 00 00
0078 8E F0 80 4F 06 5F→86 55 8D 03 7E FE
0084 85 36 86 04 4A 26 FD 32 39 00 00 00
0090 00 00 00 00 00 00 00 00 00 00 00 00
009C 00 00 00 00 00 00 00 00 00 00 00 00
00A8 00 00 00 00 00 00 00 00 00 00 00 00
00B4 00 00 00 00 00 00 00 00 00 00 00 00
00C0 00 00 00 00 00 00 00 00 00 00 00 00
00CC 00 00 00 00 00 00 00 00 00 00 00 00
00D8 00 00 00 00 00 00 00 00 00 00 00 00
00E4 00 00 00 00 00 00 00 00 00 00 00 00
00F0 00 00 00 00 00 00 00 00 00 00 00 00
00FC 00 00 00 00 00 00 00 00 00 00 00 00
0108 00 00 00 00 00 00 00 00 00 00 00 0r
 ^7FF █
```

1. Example of the display while using the Edit function. The simple program shown is as follows:

```
0078   8E   F080            LDS £$F080
007B   4F                   CLRA
007C   06                   TAP
007D   5F                   CLRB
007E   86   55              LDAA £$55
0080   8D   03              BSR DELAY
0082   7E   FE85            JMP CONTROL
0085   36         DELAY     PSHA
0086   86   04              LDAA £04
0088   4A         $1        DECA
0089   26   FD              BNE $1
008B   32                   PULA
008C   39                   RTS
```

The pointer is at location 007E.

MIKBUG™ Compatibility

The monitor described here retains a good number of MIKBUG compatible subroutines and although their coding and locations are different they do achieve the same end results. The following are included:

1. CONTROL, the normal re-entry point.
2. Output to VDU: OUT2H, OUTHL, OUTHR, OUT4HS, OUT2HS, OUTS, and OUTEEE. (Note that PDATA could not be squeezed in, but commercially available software usually has it built in).
3. Input: BADDR, BYTE, INHEX, INEEE. All these operate from keyboard or tape, so that a MIKBUG format tape can be loaded, if necessary, by use of a short routine relying heavily on the monitor.

Input/Output

Looking at the program

listing, the section from FEF8 to FFDD is the VDU driver. As most of it has been described before in the ACC news it will not be covered again in detail. The reason for it being so long (230 bytes) is that the 24x40 format is not simply divisible by binary numbers, and therefore some calculation is necessary. The HOME routine can be used at any time, even when a user program is running, because it is called by a keyboard interrupt (see FEDO) and and it is transparent, i.e. it saves and restores all registers that it uses so as not to disturb the user program.

The selections from FEB9 to FEBD and FEF4 to FEF7 handle the keyboard input. Its simplicity shows one of the advantages of using an interrupt driven keyboard. FEC8 to FEF3 handles the non-maskable interrupt itself, having been called by the vector at FFFC. The ability to break into a user program and escape from it (e.g. if stuck in a loop) is the other advantage of an interrupt driven system. The rather fiddly bit from FED9 to FEEB (19 bytes) is necessary to cater for the situation where a user program has stopped at a WAI instruction, to await a keyboard input. Press the Home key; after homing the screen, it re-establishes the WAI condition, (without that the user program would press on regardless).

Commands

To use the monitor for program development, commands are entered at the keyboard and decoded by FE8F to FEA5. In this case, the use of a keyed jump table is more efficient than successive comparison with a list of characters. It also allows easy modification and expansion; in fact, space has been left at FFF3 to FFF6 for the inclusion of your own defined command. This could even be a prefix to a whole group of commands, contained in another area of memory. The following are available in this monitor:

G XXXX Go to user program at address XXXX. The user program runs in a stack whose pointer is extracted from TGTSTK (F0FA in the scratchpad RAM). After a system reset the stack will be from F0E0 down to F0DA, but after escaping from a user program this value could be indeterminate. It is thus good practice to make sure that your own programs set their stack pointer at an early stage.

Continue from an escaped point in a program. This will only work properly if the target stack has been initialised to some area not used by the monitor (try LDS £$F080 for example).

L XXXX Loads a binary tape into memory from address XXXX. In this mode the test of the data register switches at FEB9 is bypassed and their position is immaterial. When the tape has finished it is necessary to use the Escape key to return to CONTROL.

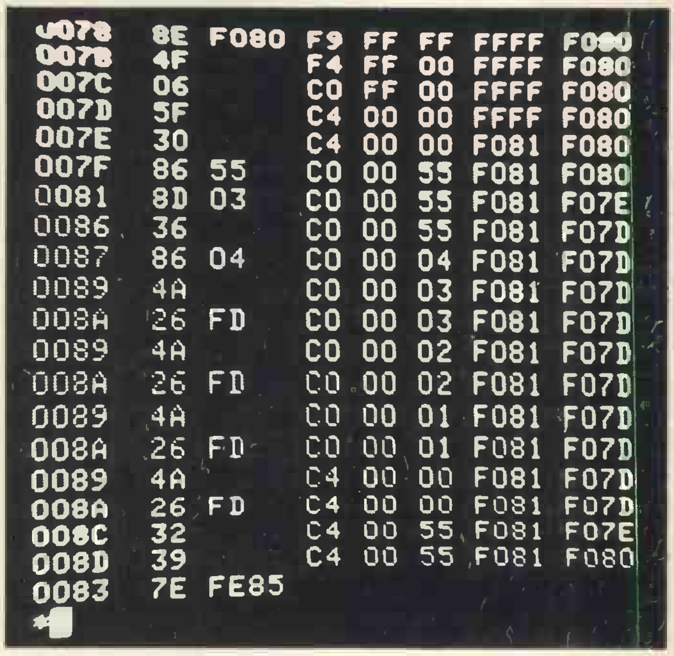
D XXXX YYYY Dump a binary tape from address XXXX to address YYYY. This routine can be used to dump any area of memory including the monitor itself. The program starts at FE13 by calling INADDS, a useful subroutine to get a pair of addresses from the keyboard. When the dump is finished it returns to CONTROL.

E XXXX YYYY Edit a block of memory from XXXX to YYYY. This routine is one of the central features of the monitor. It displays on the screen the first 276 bytes of memory contents starting from XXXX. The format is 23 lines and each contains the address at the start of line and 12 bytes of memory. A cursor, which is initialised to the first byte in the top left hand corner, points to one of the locations and may be moved left, right, up or down by the keys ←, →, ↑, and C/R. The byte pointed to may be overwritten simply by typing the new value. Bytes can be removed by typing "R" or inserted by typing "I" followed by the new value.

In both cases, the succeeding memory contents close up or move out as necessary up to the end address (initially YYYY). The new end address is displayed for reference at the bottom of the screen. This system allows fast and easy interactive editing of machine code in small or large chunks. In the process it does away with the need for a number of separate commands to manipulate memory contents. The edit routine runs from FC5B to FCE1; it uses the MEMPRINT subroutine at FC20 to FC5A to print the display. It should be exited by typing a non-hex character, such as a space.

M XXXX YYYY ZZZZ Block move of memory from XXXX to YYYY to a new area starting at ZZZZ. It is not necessary for the user to calculate the length of the block in advance. If desired, the effect of the move can be checked afterwards, with the Edit command. This routine (FDEA-FE04) is short and fast because it uses the stack pointer as a data counter in the absence of a second index register on the MC6800. You cannot, of course, move a block forwards to a new area which overlaps the old one. . . it will overwrite itself. In this case you have to move it first to a spare area and then to its destination.

T WWWW XXXX YYYY ZZZZ This is a software single step and trace routine that provides a powerful debugging tool. It traces a target program WWWW to XXXX, starting at instruction address ZZZZ. It needs a spare block of memory which you define to start at



0078	8E	F080	F9	FF	FF	FFFF	F080
007B	4F		F4	FF	00	FFFF	F080
007C	06		C0	FF	00	FFFF	F080
007D	5F		C4	00	00	FFFF	F080
007E	30		C4	00	00	F081	F080
007F	86	55	C0	00	55	F081	F080
0081	8D	03	C0	00	55	F081	F07E
0086	36		C0	00	55	F081	F07D
0087	86	04	C0	00	04	F081	F07D
0089	4A		C0	00	03	F081	F07D
008A	26	FD	C0	00	03	F081	F07D
0089	4A		C0	00	02	F081	F07D
008A	26	FD	C0	00	02	F081	F07D
0089	4A		C0	00	01	F081	F07D
008A	26	FD	C0	00	01	F081	F07D
0089	4A		C4	00	00	F081	F07D
008A	26	FD	C4	00	00	F081	F07D
008C	32		C4	00	55	F081	F07E
008D	39		C4	00	55	F081	F080
0083	7E	FE85					

2. An extra byte (30=TSX) has been inserted at 007E and then the program run in the single step and trace mode. The order of the columns is: address, opcode, operand if any, condition codes register (CCR), B,A,X,SP. Note the movement of the stack pointer on entering and leaving the subroutine and the further movement when pushing and pulling data onto the stack; also how the Z (=Zero) bit 2 of the CCR is set at 0089 when A has been finally decremented to zero. Details like these of the internal workings of the processor are clearly demonstrated by this sort of display.

YYYY and must be of the same length as the target. Any areas of data must be excluded from both and left intact as they are used by both the target program and the trace routine (or "host" program).

To understand how this routine works you should know that manual debugging of a user program uses the software interrupt (SWI, opcode 3F) as a breakpoint. The SWI is placed at a strategic point in the target so that, when the program gets to it, it responds to the artificial interrupt by dumping the processor registers on the stack and jumping to the address contained at FFFA. It is then possible manually to examine the stack to see what was happening at this point. There are some problems with this approach:

1. The program may never get to the SWI. You can counter this by placing several SWIs in different likely places in the hope of hitting one of them.
2. The process of substituting bytes of target with SWIs, remembering them, and replacing them afterwards is tedious and prone to error.
3. The whole business takes a long time and a lot of mental effort.

The solution adopted here is to extend this method to its logical conclusion by filling the whole of the target program, except for the instruction being executed, with SWIs. Whatever the instruction does it should now hit an SWI next. If it should jump right out of the target program this will be obvious from the display, so you will at least have located the problem instruction.

The trace, starting at FD0F, sets its own SP at F0CF, and the target's at F0B8 (to allow the target to run in an independent stack right from the start, and both target and host to be independent of the monitor stack). As soon as (or if) the target sets its own SP, this is used from then on. At FD32 the host calls the Transfer routine to save a copy of the target in the area from YYYY, and to fill the original with SWIs. At FD37 it starts a line by line print by displaying the first instruction address. From FD48 to FD74 it measures the length of the instruction (1, 2 or 3 bytes), then displays it. At FDA7 the program waits for the single step command from the keyboard (→) and at FDB2 it executes the one instruction. Usually an SWI will be encountered next which will vector to FDB9 and fill the rest of the display line with the resulting register contents for your information. At FDE1 Transfer again prepares the target area and the program then loops round to the start again and waits for the next single step command.

The result is a line by line trace of the program flow that shows exactly what happens at each step. Apart from its diagnostic use this is a first rate educational tool for showing the internal workings of the microprocessor.

If the target gets into a long loop (e.g. a timing loop) you can skip to the end as follows:

1. Press the Escape key once. Do not press it again while you are temporarily out of the trace routine.
2. Knowing from the display where the target SP is, use the Edit command to examine the 7 bytes below it. These correspond to the CCR, B, A, X, and PC registers. You can then modify the CCR, B, A, and X registers (but not the PC) as necessary to shorten the loop. Press the space bar to leave Edit.
3. Alternatively you can modify areas of data in memory in the same way.
4. Press the continue key (←) to return to the Trace routine.

The only types of program that cannot be handled by the Trace are those with self modifying code or those with areas of data and program intermingled.

When you have finished, it is necessary to use the Move command to shift the program copy from YYYY back to WWWW.

Conclusion

This monitor has been in use for about six months now and has made it possible to tackle programs like 8K BASIC, study them and modify them to run on the 77-68 without too much difficulty. If you have more than 1K available for your monitor I suggest the following inclusions:

1. Automatic return of a traced program to its original location when finished.
2. Automatic decoding and display of the individual CCR flags during Trace.
3. Addition of the MIKBUG PDATA routine.
4. Inclusion of more sophisticated tape dump and load formats with file names and error checking *before* transfer from a buffer to storage, unlike MIKBUG which loads each block first and then checks the address and data to see if it got it right!
5. An even bigger Edit display if you have a bigger VDU format.
6. Software vectoring of interrupts to allow more than one interrupt driven peripheral at a time, and also multi-programming.

Finally, may I add that suggestions for improvement of the existing monitor will always be welcome.

Copies of the monitor are available on 2708 EPROM at £14.10 + VAT. Contact John at 1 The Spinney, Fleet, Hants. Telephone: 02514 29553.

Program

```
FC00  A6 00     OUT2H     LDAA X
FC02  8D 05               BSR OUTHL
FC04  A6 00               LDAA X
FC06  08                  INX
FC07  20 04               BRA OUTHR
FC09  44 44     OUTHL     LSRA x 2
FC0B  44 44               LSRA x 2
FC0D  84 0F     OUTHR     ANDA £%00001111
FC0F  8B 90               ADDA £$90         Allison's algorithm - saves 2 bytes &
FC11  19                  DAA               average of 2¼ cycles per character
FC12  89 40               ADCA £$40
FC14  19                  DAA
FC15  7E FE F8  OUTCH     JMP OUTEEE
FC18  80 E6     OUT4HS    BSR OUT2H
FC1A  8D E4     OUT2HS    BSR OUT2H
FC1C  86 20     OUTS      LDAA £'space
FC1E  20 F5               BRA OUTCH
FC20  BD FF 1C  MEMPRINT  JSR HOME
FC23  FE F0 F1            LDX STARTING
FC26  FF F0 F6            STX TEMPX
FC29  C6 17               LDAB £$17         page length
FC2B  37        NEWLINE   PSHB
FC2C  CE F0 F6            LDX £TEMPX
FC2F  C6 0C               LDAB £$0C         line length
FC31  8D C0               BSR OUT2H
FC33  8D CB               BSR OUT2H         Print address
FC35  FE F0 F6            LDX TEMPX
FC38  8C F0 EF  NEWBYTE   CPX MEMLOC
FC3B  26 06               BNE MEM1
FC3D  86 50               LDAA £'→          Print pointer against the byte
FC3F  8D D4               BSR OUTCH
FC41  20 02               BRA MEM2
FC43  8D D7     MEM1      BSR OUTS
FC45  8D B9     MEM2      BSR OUT2H
FC47  5A                  DEC B
FC48  26 EE               BNE NEWBYTE
FC4A  FF F0 F6            STX TEMPX
FC4D  33                  PULB
FC4E  5A                  DEC B
FC4F  26 DA               BNE NEWLINE
FC51  86 3C               LDAA £'>
FC53  8D C0               BSR OUTCH
FC55  CE F0 F3            LDX £ENDING
FC58  8D BE               BSR OUT4HS        Print end address
FC5A  39                  RTS               End of MEMPRINT
FC5B  BD FE 05  EDIT      JSR INADDS
FC5E  8D C0     ED6       BSR MEMPRINT
FC60  3E                  WAI               Get keyboard command
FC61  B6 F0 FE            LDAA KBUFF2
FC64  FE F0 EF            LDX MEMLOC
FC67  81 5D               CMPA £'→          Cursor right?
FC69  26 03               BNE ED1
FC6B  08                  INX
FC6C  20 20               BRA ED2
FC6E  81 5B     ED1       CMPA £'←          Cursor left?
FC70  26 03               BNE ED3
FC72  09                  DEX
FC73  20 26               BRA ED2
FC75  81 00     ED3       CMPA £'↓          Cursor down?
FC77  26 0F               BNE ED4
FC79  B6 F0 F0            LDAA MEMLOC LOW
FC7C  8B 0C               ADDA £$0C
FC7E  B7 F0 F0            STAA MEMLOC LOW
FC81  24 03               BCC ED5
FC83  7C F0 EF            INC MEMLOC HIGH
FC86  20 D6     ED5       BRA ED6
FC88  81 5E     ED4       CMPA £'↑          Cursor up?
FC8A  26 14               BNE ED7
FC8C  B6 F0 F0            LDAA MEMLOC LOW
FC8F  80 0C               SUBA £$0C
FC91  B7 F0 F0            STAA MEMLOC LOW
FC94  24 03               BCC ED8
FC96  7A F0 EF            DEC MEMLOC HIGH
FC99  20 C3     ED8       BRA ED6
FC9B  FF F0 EF  ED2       STX MEMLOC
FC9E  20 BE               BRA ED6
FCA0  81 49     ED7       CMPA £'I          Insert?
FCA2  26 19               BNE ED9
FCA4  BD FE 55            JSR BYTE
FCA7  FE F0 F3            LDX ENDING
FCAA  08                  INX
FCAB  FF F0 F3            STX ENDING
FCAE  09        ED10      DEX
```

```
FCAF E6 00          LDAB X
FCB1 E7 01          STAB 1,X
FCB3 BC F0 EF       CPX MEMLOC
FCB6 26 F6          BNE ED10
FCB8 A7 00          STAA X
FCBA 08             INX
FCBB 20 DE          BRA ED2
FCBD 81 52    ED9   CMPA £'R          Remove?
FCBF 26 13          BNE ED11
FCC1 FE F0 EF       LDX MEMLOC
FCC4 E6 01    ED12  LDAB 1,X
FCC6 E7 00          STAB X
FCC8 08             INX
FCC9 BC F0 F3       CPX ENDING
FCCC 26 F6          BNE ED12
FCCE 09             DEX
FCCF FF F0 F3       STX ENDING
FCD2 20 8A          BRA ED6
FCD4 BD FE 69 ED11  JSR INHEX1
FCD7 BD FE 57       JSR BYTE1
FCDA A7 00          STAA X
FCDC 08             INX
FCDD FF F0 EF       STX MEMLOC
FCE0 20 F0          BRA ED6           End of Edit routine
FCE2 FE F0 E2 TRANSFER LDX TGT START
FCE5 BE F0 EC       LDS M.S.A.
FCE8 C6 3F          LDAB £$3F
FCEA B6 F0 FD TRAN4 LDAA VDULOW       Used as a flag
FCED 84 0F          ANDA £%00001111
FCEF 26 04          BNE TRAN1
FCF1 A6 00          LDAA X
FCF3 36             PSHA              Included if flag = 0
FCF4 31             INS
FCF5 E7 00    TRAN1 STABx
FCF7 BC F0 EA       CPX I.A.
FCFA 26 03          BNE TRAN2
FCFC BF F0 E8       STS M.I.A.        Sets Mirror Instruction Address
FCFF BC F0 E4 TRAN2 CPX TGT END
FD02 27 04          BEQ TRAN3
FD04 08             INX
FD05 31             INS
FD06 20 E2          BRA TRAN4
FD08 8E F0 CD TRAN3 LDS £HOSTSTK-2    To enable correct RTS
FD0B 39             RTS
FD0C 7E FE 44 BAD1  JMP BADDR
FD0F 8E F0 CF TRACE LDS £HOSTSTK      Beginning of single step & trace routine
FD12 CE F0 B8       LDX £TGTSTK       Trace target stack
FD15 FF F0 E6       STX TEMPSTK       To allow Continue command
FD18 8D F2          BSR BAD1
FD1A FF F0 E2       STX TGT START     Start of target program area
FD1D 8D ED          BSR BAD1
FD1F FF F0 E4       STX TGT END       End of target program area
FD22 8D E8          BSR BAD1
FD24 FF F0 EC       STX M.S.A.        Mirror program start address
FD27 8D E3          BSR BAD1
FD29 FF F0 EA       STX I.A.          Instruction address
FD2C FF F0 BE       STX P.C.          Sets P.C. in target stack
FD2F BD FF 1C       JSR HOME          Clear screen
FD32 8D AE          BSR TRANSFER
FD34 CE F0 EA TRACE1 LDX £I.A.
FD37 BD FC 18       JSR OUT4HS        Print Instruction Address
FD3A 8D 77          BSR OUTS1
FD3C FE F0 E8       LDX M.I.A.
FD3F A6 00          LDAA X
FD41 81 3F          CMPA £$3F         Is it a SWI?
FD43 26 03          BNE TRACE2
FD45 7E FE 82       JMP ESCAPE        If so
FD48 81 8C    TRACE2 CMPA £$8C        Start disassembly of Opcode
FD4A 27 22          BEQ THREE
FD4C 81 8E          CMPA £$8E
FD4E 27 1E          BEQ THREE
FD50 81 CE          CMPA £$CE
FD52 27 1A          BEQ THREE
FD54 84 F0          ANDA £%11110000
FD56 81 50          CMPA £$50
FD58 22 0C          BHI MORE
FD5A 81 20          CMPA £$20
FD5C 26 04          BNE ONE
FD5E C6 02    TWO   LDAB £2
FD60 20 0E          BRA TRACE3
FD62 C6 01    ONE   LDAB £1
FD64 20 0A          BRA TRACE3
FD66 85 10    MORE  BITA £$10
FD68 27 F4          BEQ TWO
FD6A 85 20          BITA £$20
FD6C 27 F0          BEQ TWO
FD6E C6 03    THREE LDAB £3
FD70 86 03    TRACE3 LDAA £3          Instruction length now in B
FD72 B7 F0 F5       STAA CKSM         Length of opcode display
FD75 8D 3F          BSR OUT2HS1       Print first byte of opcode
FD77 09       PREP1 DEX
FD78 A6 00          LDAA X            Get instruction again
FD7A FE F0 EA       LDX I.A.
FD7D A7 00          STAA X
FD7F 7A F0 F5       DEC CKSM
FD82 5A             DECB
FD83 27 10          BEQ PREP2
FD85 08             INX
FD86 FF F0 EA       STX I.A.
FD89 FE F0 E8       LDX M.I.A.
FD8C 08             INX
FD8D FF F0 E8       STX M.I.A.
FD90 BD FC 00       JSR OUT2H         Print operand
FD93 20 E2          BRA PREP1
FD95 7D F0 F5 PREP2 TST CKSM
FD98 27 08          BEQ PREP3
FD9A 8D 17          BSR OUTS1
FD9C 8D 15          BSR OUTS1
FD9E 7A F0 F5       DEC CKSM
FDA1 20 F2          BRA PREP2
FDA3 8D 0E    PREP3 BSR OUTS1
FDA5 8D 0C          BSR OUTS1
FDA7 3E       PREP4 WAI               Get command from keyboard
FDA8 B6 F0 FE       LDAA KBUFF2
FDAB 81 5D          CMPA £'->         Wait for single step command
FDAD 26 F8          BNE PREP4
FDAF BE F0 E6       LDS TEMPSTK
FDB2 3B             RTI               Run next instruction
FDB3 7E FC 1C OUTS1 JMP OUTS
FDB6 7E FC 1A OUT2HS1 JMP OUT2HS
FDB9 BF F0 E6 SWINT STS TEMPSTK       Start of SWI service routine
FDBC 8E F0 CF       LDS £HOSTSTK
FDBF FE F0 E6       LDX TEMPSTK
FDC2 08             INX
FDC3 8D F1          BSR OUT2HS1       Print CCR
FDC5 8D EF          BSR OUT2HS1       Print B
FDC7 8D ED          BSR OUT2HS1       Print A
FDC9 BD FC 18       JSR OUT4HS        Print X
FDCC AE 00          LDS, X            Get P.C.+1
FDCE 34             DES               Correct it
FDCF BF F0 EA       STS I.A.          Update Instruction Address,
FDD2 AF 00          STS X             and P.C. on target stack.
FDD4 8E F0 CF       LDS £HOSTSTK
FDD7 08             INX               Correct S.P.
FDD8 FF F0 F6       STX TEMPX
FDDB CE F0 F6       LDX £TEMPX
FDDE BD FC 18       JSR OUT4HS        Print corrected S.P.
FDE1 BD FC E2       JSR TRANSFER      Refill with SWIs & update M.I.A.
FDE4 BD FE FC       JSR OUTC/R
FDE7 7E FD 34       JMP TRACE1        and round again
FDEA 8D 19    MOVE  BSR INADDS        Block Move routine
FDEC 08             INX
```

```
FDED FF F0 F3       STX ENDING+1
FDF0 8D 52          BSR BADDR
FDF2 35             TXS
FDF3 31             INS
FDF4 FE F0 F1       LDX STARTING
FDF7 A6 00    MOVE1 LDAA X
FDF9 08             INX
FDFA 36             PSHA
FDFB 31 31          INS x 2
FDFD BC F0 F3       CPX ENDING+1
FE00 26 F5          BNE MOVE1
FE02 7E FE 85 CONT  JMP CONTROL
FE05 8D 3D    INADDS BSR BADDR        Get start & end addresses
FE07 FF F0 F1       STX STARTING
FE0A FF F0 EF       STX MEMLOC        For MEMPRINT routine
FE0D 8D 35          BSR BADDR
FE0F FF F0 F3       STX ENDING
FE12 39             RTS
FE13 8D F0    DUMP  BSR INADDS        To tape in binary format
FE15 FE FD F1       LDX STARTING
FE18 8D 14          BSR INITZE
FE1A B6 F4 01 DUMP1 LDAA ACIA S
FE1D 85 02          BITA £%00000010   Tx busy?
FE1F 27 F9          BEQ DUMP1
FE21 A6 00          LDAA X
FE23 B7 F4 00       STAA ACIA D       Dump a byte
FE26 BC F0 F3       CPX ENDING
FE29 27 5A          BEQ CONTROL
FE2B 08             INX
FE2C 20 EC          BRA DUMP1
FE2E 86 23    INITZE LDAA £$23        Subroutine to prepare ACIA
FE30 B7 F4 D1       STAA ACIA C
FE33 47             ASRA
FE34 B7 F4 01       STAA ACIA C       No parity, 2 stop bits
FE37 39             RTS
FE38 8D 0A    LOAD  BSR BADDR         Load a binary tape until stopped
FE3A 8D F2          BSR INITZE
FE3C BD FE BE LOAD1 JSR STATUS        Get a byte
FE3F A7 00          STAA X
FE41 08             INX
FE42 20 F8          BRA LOAD1         round again
FE44 8D FC 1C BADDR JSR OUTS          Mikbug routine, from keyboard or tape.
FE47 8D 0C          BSR BYTE
FE49 B7 F0 F6       STAA TEMPX HI
FE4C 8D 07          BSR BYTE
FE4E B7 F0 F7       STAA TEMPX LO
FE51 FE F0 F6       LDX TEMPX
FE54 39             RTS
FE55 8D 10    BYTE  BSR INHEX
FE57 48 48          ASLA x 2
FE59 48 48          ASLA x 2
FE5B 16             TAB
FE5C 8D 09          BSR INHEX
FE5E 1B             ABA
FE5F 16             TAB
FE60 FB F0 F5       ADDB CKSM
FE63 F7 F0 F5       STAB CKSM         Update Checksum
FE66 39             RTS
FE67 8D 50    INHEX BSR INEEE
FE69 80 30    INHEX1 SUBA £$30
FE6B 2B 18          BMI CONTROL
FE6D 81 09          CMPA £$09
FE6F 2F 0A          BLE INTHG
FE71 81 11          CMPA £$11
FE73 2B 10          BMI CONTROL
FE75 81 16          CMPA £$16
FE77 2E 0C          BGT CONTROL
FE79 80 07          SUBA £$07
FE7B 39       INTHG RTS
FE7C 8E F0 DA RESET LDS £CTRLSTK-7    Initialises target stack if required
FE7F BD FF 1C       JSR HOME          Clears screen
FE82 BF F0 FA ESCAPE STS TGTSTK
FE85 8E F0 E1 CONTROL LDS £CTRLSTK
FE88 BD FE FC       JSR OUTC/R        Carriage return
FE8B 86 2A          LDAA £'*
FE8D 8D 69          BSR OUTEEE
FE8F 8D 28          BSR INEEE         Get keyboard command
FE91 CE FF 0E       LDX £JTABLE       Point at keyed jump table
FE94 E6 00    CONT1 LDAB X            Get key
FE96 C1 00          CMPB £00          End of table?
FE98 27 E8          BEQ CONTROL       If character not recognised
FE9A 11             CBA
FE9B 27 05          BEQ FOUNDIT
FE9D 08 08 08       INX x 3           Point to next key
FEA0 20 F2          BRA CONT1
FEA2 EE 01    FOUNDIT LDX 1,X
FEA4 6E 00          JMP X             Jump to required routine
FEA6 8D 9C    GO    BSR BADDR
FEA8 FE F0 FA       LDX TGTSTK
FEAB B6 F0 F6       LDAA TEMPX HI
FEAE A7 06          STAA 6,X
FEB0 B6 F0 F7       LDAA TEMPX LO
FEB3 A7 07          STAA 7,X          Target P.C. prepared
FEB5 BE F0 FA CONTINUE LDS TGTSTK     Target S.P. prepared
FEB8 3B             RTI               and go
FEB9 B6 F0 FF INEEE LDAA SWITCHES     Mikbug compatible routine.
FEBC 27 36          BEQ KBD           If parallel input required (sws.=0)
FEBE B6 F4 01 STATUS LDAA ACIA S      If serial input required.
FEC1 44             LSRA              Rx ready?
FEC2 24 FA          BCC STATUS
FEC4 B6 F4 00       LDAA ACIA D
FEC7 39             RTS
FEC8 B6 FB FF NMI   LDAA KBUFF        NMI handler
FECB 81 03          CMPA £$03         Escape?
FECD 27 B3          BEQ ESCAPE
FECF 36             PSHA
FED0 81 04          CMPA £$04         Home?
FED2 26 1B          BNE NMI1
FED4 FF F0 F8       STX TEMPX2
FED7 8D 43          BSR HOME
FED9 30             TSX
FEDA EE 06          LDX 6,X
FEDC 09             DEX               Get last byte before NMI occurred.
FEDD A6 00          LDAA X
FEDF 81 3E          CMPA £$3E         Was it WAI?
FEE1 26 09          BNE NMI2
FEE3 30             TSX
FEE4 6D 07          TST 7,X
FEE6 26 02          BNE NMI3
FEE8 6A 06          DEC 6,X
FEEA 6A 07    NMI3  DEC 7,X           Re-establish WAI after RTI
FEEC FE F0 F8 NMI2  LDX TEMPX2
FEEF 32       NMI1  PULA
FEF0 B7 F0 FE       STAA KBUFF2
FEF3 3B             RTI
FEF4 3E       KBD   WAI               Get keyboard input
FEF5 B6 F0 FE       LDAA KBUFF2
FEF8 81 DD    OUTEEE CMPA £'C/R       Mikbug compatible, echoes INEEE
FEFA 26 39          BNE PUTVDU
FEFC 37       OUTC/R PSHB             Entry to print carriage return
FEFD 36             PSHA
FEFE 86 20          LDAA £'space
FF00 F6 F0 FC OUT1  LDAB VDU HI
FF03 C1 FB          CMPB £$FB
FF05 27 04          BEQ OUT2
FF07 8D 2C          BSR PUTVDU        Print space
FF09 20 F5          BRA OUT1
FF0B F6 F0 FD OUT2  LDAB VDU LO
FF0E 53             COMB
FF0F C5 07          BITB £07          End of line?
FF11 27 04          BEQ ENDOUT
FF13 8D 20          BSR PUTVDU        Print space
```

Dick Pountain *analyses and reports on the micro-associated world of programmable calculators.*

GETTING IT TAPED

IN My glowing review of the Casio FX501/502P last month ("This one will run and run . . ."), I promised a follow up on the FA-1 adaptor. Here it is.

Tempus of Cambridge kindly supplied the adaptor and also exchanged the 501 for a 502 (which, as I suspected, is just like the 501, but more so).

The FA-1 is a small cradle into which the calculator slips, connecting via a gold plated, 7-pin socket. The cradle has a lead ending in two mini jack plugs which fit into the microphone and earphone sockets of a standard cassette recorder. Cassette radios, mono and stereo cassette decks and even mini cassette dictation machines will work. Some hi-fi cassette decks will require a 5-pin DIN plug to be substituted for the mini jacks.

Saving and loading are performed in the same manner as on a micro computer. The instruction SAVE is followed by a three digit file number, and is executed from the keyboard with the recorder running. Loading is by LOAD and the same file number. The calculator searches for the named file and displays, say, FP005 when a program file is found or F005 when a data file is found.

For speed, it is best to first roughly locate the file using tape counter readings, but if necessary, the calculator will search a whole tape – displaying each file name as it passes, but loading only the designated one.

The maximum time for a LOAD is about 15 seconds. . . correspondingly shorter for smaller programs. Therefore a C-60 cassette can hold over 200 programs.

As alluded to earlier, program and data are stored in separate files, unlike most magnetic card calculators which store the whole program and data register contents on one card. The separate way is of much more use as the same program can be used with any number of different data sets. Also, execution of a program may be halted partway through and a fresh data file loaded (manually, of course) which greatly increases the data handling facilities of the 502. In addition, the load instruction will fill all or any of the ten program registers. Therefore several different programs may be loaded into the machine simultaneously (size permitting), or parts of a large program may be loaded independently, or subroutines may be stored on tape and added to existing programs in the machine.

I have used the adaptor with three different recorders including an 'electronic memo-pad' (which provides great portability) and found loading and saving very easy and reliable on all of them — provided the output volume is set as high as possible.

The FA-1 is also necessary for the music synthesizing function of the calculator, about which the less said the better. The world didn't need Rolf Harris and the Stylophone; it needs a robot Rolf Harris even less.

To summarise then, the Casio FX502P with the FA-1 and a mini cassette recorder provides a pocketable computing system which is in some ways unrivalled, even by the £150 plus Hewlett Packard and Texas machines, particularly in regard of ease of learning.

As a footnote, I must clarify a wrong impression my review last month may have given. I said that the 501/502 have only ten labels available for use in programs. Of course, since there are ten independant program registers, all ten labels may be used ten time over, if the program is broken down into modules (which is a habit the machine rapidly encourages). This gives a potential of 100 labels, which is quite sufficient.

Master pack

Following on from the above, I have been informed by Premier Publications that they are launching the Master Pack, a software package for the Casios. It takes the form of a cassette containing over 160 programs and a 60 page manual which includes an introduction to programming, advanced programming, plus full program documentation. At press time I hadn't seen the whole package, though extracts I have looked at from the manual suggest that it will be far superior to Casio's own User Manual. The programs include all of Casio's own library programs ready to load, plus a variety of educational, games and personal finance routines and general purpose subroutines for advanced programmers.

The pre-production sample contained some quite sophisticated games with ingenious use of the Casio's display formats, including one which scrolls a 10 by 10 field, line by line over the display. Another routine provides, via data packing techniques, the equivalent of 100 independent memories — each to store a single digit variable.

The Master Pack will be available from dealers after the middle of October and I shall report more fully when a production sample is available.

Look sharp

A new pocket calculator from Sharp, the EL-5100, has reached me; although it won't interest PCW readers particularly, since it is not truly programmable, nevertheless it has some clever features which may give pointers to the future.

Immaculately presented in the inimitable Sharp fashion, it's distinguished from an ordinary scientific calculator by its unusually large LCD display. This display is alphanumeric (though only a part alphabet is provided).

When you enter a simple arithmetic calculation problem, the whole calculation is displayed, e.g. 5.7 + 3.8 x 6.4. On pressing = the answer is displayed. But more intriguing is the Algebraic Expression Mode. In this mode, algebraic expressions of any complexity may be written in stan-

Cont. on Page 93

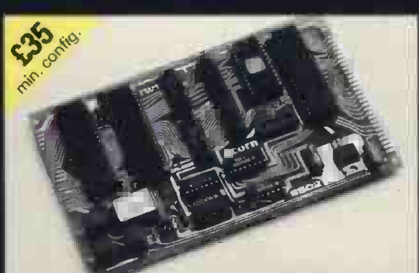
Four of a kind!

Acorn Computers announce with pride the fourth module in the series—a VDU interface on a Eurocard. This unit uses two very powerful devices, the MC 6845 and the SAA5050. The 6845 programmable controller provides all the signals to drive a 625 line 50 frames per second VDU together with read addresses for the character RAM, the SAA5050 character generator then produces the necessary dot patterns to refresh the VDU. The SAA5050 produces standard teletext characters and graphics and has Red, Green and Blue outputs. This means that the Acorn system will be compatible with CEEFAX, ORACLE and PRESTEL transmissions.

The Acorn VDU module in kit form is complete with sockets and is supplied with listings for programs which set up the 6845, a miniature dissassembler which displays 25 hex instructions (double or treble byte) and graphics programs. All these may be loaded and run using the Acorn system 1 monitor.

Options include:- VHF modulator for B.W. domestic T.V. and PAL colour encoder for domestic colour T.V.

- 40 characters per line
- 25 lines per page
- 5 colour graphics and characters
- Upper and lower ASC11
- Teletext graphics font
- Programmable cursor
- Hardware scroll
- Light pen facility
- Memory mapped
- Transparent access
- Single 5V supply



Acorn CPU

Acorn System 1

Acorn 8K Memory

Order form

Pcw 11

Send to: Acorn Computers Ltd., 4a Market Hill, Cambridge, Cambs.

☐ Acorn controller	@	£35.00 plus VAT	5.25
☐ Microcomputer	@	£65.00 plus VAT	9.75
☐ Assembled Microcomputer	@	£79.00 plus VAT	11.85
☐ Memory	@	£95.00 plus VAT	14.25
☐ Memory assembled	@	£100.00 plus VAT	15.00
☐ V.D.U.	@	£88.00 plus VAT	13.20
☐ V.D.U. assembled	@	£98.00 plus VAT	14.70
V.H.F. Modulator	to be announced		
Colour Encoder	to be announced		

Name

Address

87

ACORN COMPUTER

Acorn Computers Ltd., 4a Market Hill, Cambridge.
Telephone 0223 312772

FROM: KINGSTON COMPUTERS LIMITED.

TO: ALL PET OWNERS.

INFO: H.B. COMPUTERS
ROBOX OFFICE EQUIPMENT
STACK COMPUTER SERVICES
TAYLOR WILSON SYSTEMS
WEGO COMPUTERS

DATE: 15.10.79

REF: KC/PCW/9/1

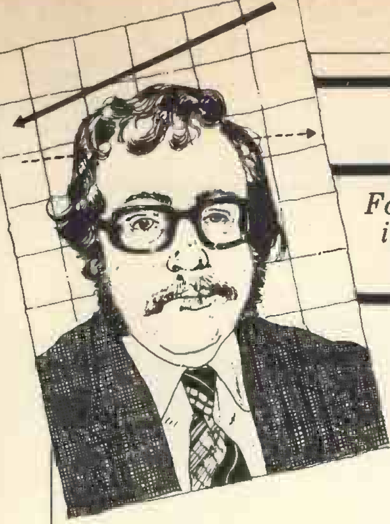
KETTERING	0536-83922.
GLASGOW	041-221-5401.
LIVERPOOL	051-924-1125.
SOLIHULL	05645-6192.
SURREY	0883-49235.

HAVE PURCHASED 500 ADA 1200 RS232/IEEE INTERFACES AT UNBELIEVABLE PRICE — STOP — SUGGEST YOU RETAIL AT 65 POUNDS — STOP — WILL SUPPORT FULL ONE YEAR GUARANTEE — STOP — REGRET CANNOT REPEAT OFFER ONCE STOCK CLEARED.

REGARDS,

SANDRA V. OLIVER.
SALES MANAGER.

KINGSTON COMPUTERS LTD.
SCARBOROUGH HOUSE
SCARBOROUGH ROAD
BRIDLINGTON
YO16 5NS
0262-73036
ONE OF THE DALE GROUP OF COMPANIES



ON THE LINE

For almost a year now, David Hebditch in his 'On the Line' series has been expounding the basic concepts of using your personal computer to communicate with other systems over the public telephone network.

HIGH-LEVEL PROTOCOLS

By the way of review, we have now explored the basic hardware and software mechanisms for interfacing personal computers to the public telephone network and for moving blocks of data between processors with a minimal level of error control.

This may well be adequate for most users to be able to establish simple point-to-point 'conversations'. However, in order to determine what to do with this capability it may be useful to return to base and consider our original ideas for the practical application of personal computer networking.

Let's forget (for the time being) the medium-term possibilities of using Prestel, Teletext and teleconferencing etc. The most practical (and useful) applications in the short-term are listed below.

1. **Conversations:** Simply sending messages between systems. The benefits of this are:
— a fast (and relatively cheap) way of sending someone a message — electronic mail?
— an effective way of using the telephone for deaf people.
— a means of setting-up calls for other purposes (see below)
— given the correct data link control, a basis for emulating a terminal for linking to another, larger computer (eg. a time-sharing service or database provider).

2. **Program Transfer**
— a means of swapping programs with another user.
— a means of sending the fixes to make the previous version of the program work.

It is technically feasible for software companies to use this as a method of distributing programs. Of course, the number of prospective customers with a communications capability needs to be big enough to make the investment on their part worthwhile and it may be some time before this is achieved.

3. **File Transfer** is functionally very similar to program transfer but involves the shifting of *data* from one processor to another. This is more likely to be of use in a business system than in a domestic environment. For example, details of goods received at a warehouse could be transmitted to the order-processing computer for addition to the stock-on-hand file. The major difference between program transfer and file transfer is one of data format and this will be discussed later in the article.

I did consider including a category for interactive game-playing. The protocol required is, however, a function of the type of game involved. In the case of video games very little information needs to be sent (eg. the eight-bit value of a game-control) but it must be sent quickly (to avoid missing the ball!) In view of this and the relatively unimportant nature of the application, the use of a data link control with error handling would probably be too cumbersome. Other games like

ANNOUNCING...THE PERSONAL COMPUTER NETWORK

The number of users with some form of communications capability has now reached a high enough level to justify the introduction of the 'Personal Computer Network'. The network will comprise a directory published in Personal Computer World (and periodically updated). The directory will include an entry for each reader who is interested in linking up to other enthusiasts and will list his name, telephone number, type of system, times available and applications. If you wish to participate, please complete the form below and send it to Personal Computer World.

During the Personal Computer World show, David Hebditch will be demonstrating data communications on a number of popular systems, including the Apple II, Pet and Nascom I. He will be available to provide assistance and answer any questions you may have about networking in general and the Personal Computer Network in particular.

Please register me as being interested in Personal Computer Networking.

Name: ______________________

Address: ______________________

____

____

Telephone No: ______________________

Computer System: ______________________

____

I do not yet have a communications interface, but would like to be kept informed of developments: Tick ☐

I have the following type of communications interface: Tick

I can transmit at the following speeds 110 bit/s ☐
300 bit/s ☐
1200 bit/s ☐

I have a Post Office modem: ☐
I use an acoustic coupler: ☐
I can act as an originating station: ☐
a receiving station: ☐
both: ☐

Other comments: ______________________

____

____

Date: ______________________

Signature: ______________________

'We stock 20 different makes of computer. So our only vested interest is customer satisfaction'

Bill Cannings, Managing Director

Businesses can only improve their efficiency with computers if they buy exactly the right kind of equipment and software for their needs. The problem can be in matching their needs with what the market can supply at any one time.

At the Byte Shop and Computerland we have the widest range of computers available from any single source. We have deep-rooted systems know how going back over 10 years, so you get not only a refreshing breadth of choice, but also high level impartial advice from computer specialists on what to buy.

Our business is computers and only computers, so you will be talking to people who really understand their subject. Once you have taken the decision to visit us, you are already a good way towards choosing the right computer for your needs.

Branches at:

Ilford
426 428 Cranbrook Road,
Gants Hill, Ilford, Essex IG2 6HW
Tel. 01-554 2177

London
48 Tottenham Court Road,
London W185 4TD
Tel. 01-636 0647

Birmingham
94 96 Hurst Street,
Birmingham 85 4TD
Tel. 021-622 7149

Nottingham
92a Upper Parliament Street
Nottingham NG1 6LF
Tel. 0602 40576

Manchester
11 Gateway House,
Piccadilly Station Approach,
Manchester
Tel. 061-236 4737

Glasgow
Magnet House
Waterloo Street
Glasgow
Glasgow Tel. 041-332 2468

THE BYTE SHOP LTD

COMPUTERLAND LTD

chess and simulations could be handled using the conversational mode. To cut a long story short, I don't think that any special protocol is needed for game-playing.

A major problem with the three modes identified above is that of transparency. This is caused by the transmission of characters in the text of a message which could be mistakenly identified at the receiving end as a data-link control character. For example if you sent ASCII ETX (3) in a message the receiver will treat this as the last character in the transmission and lose all the subsequent characters. The same thing could happen with EOT (4).

"Well", you might ask, "why do such a silly thing?" But it may be difficult to avoid. For example, in conversational mode, the user may inadvertently type a control-shift key which generates a protocol character. The solution here is relatively straight-forward; impose a rule which says that only displayable characters may be included in messages. This means that your dialogue control program must 'filter out' any illegal characters.

But what if you have to transmit the equivalent of a control character in a message? This could happen during the transmission of a program in object form, or of a data file containing integer values, or of a program in source form with 'funny' characters between the quotes in a PRINT statement.

The standard way of solving this problem is to employ some form of 'escape logic'. This involves the prefixing of each dubious character with ASCII ESC (27). This has to be inserted by the transmitting program and tells the receiving software that the next character is not really ETX (or whatever). The receiver will delete the ESC.

Now the smart guys amongst you are already asking, "What happens if you want to send 'ESC'?" Clearly a spurious ESC immediately prior to the real ETX will cause the ETX to be ignored. More problems. In this case, a further ESC prefix is also required and will cause the following ESC to be treated as a regular data character.

Phew! That's enough of escape logic. Now let's move on to the high level protocols (HLP) needed for the three application modes.

(1)

Byte No.	Name	Comments
0	Type	H: Handshake message D: Dialogue P: Program Transfer F: File Transfer etc.
1 - 3	Transmission Number	Sequential message number (incremented automatically by transmitting program)
4	Action code (Command/ Response)	I: Initial transmission block S: Subsequent transmission block F: Final transmission block etc.
5 - 9	SPARE	(And anything else we can think of).

First of all, I have to make a disclaimer; there are no internationally agreed standards for HLP. Indeed there are no national standards, either. Whilst writing this article, I have a three-foot molehill on my desk of working papers from the various standards organizations (BSI, ISO, ECMA, CCITT, ANSI et al) as well as the manuals for many proprietary networking standards (IBM's SNA, Digital Equipment's DECNet and so on). I even have articles from BYTE and INTERFACE AGE describing the procedures used in the various US personal computer networks. And I plan to ignore them all.

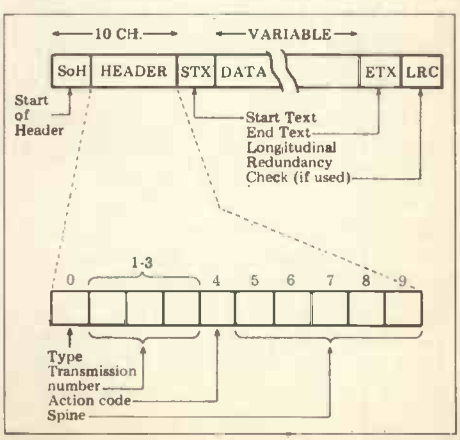
The reasons are as follows:
1. They are too complicated.
2. Although we are talking about the establishment of a Personal Computer *Network* the network we are employing is the plain old telephone system rather than any sophisticated multi node grid, permanently interconnecting a large number of users. Only two processors will be connected at any one time.
3. Implementation of the HLP should be possible by the average home user.
4. Costs must be kept to a minimum.
5. Speed and reliability concerns are not so serious.
6. Which proposed 'standard' do we choose anyway?

Now having said all that, I would need an ego of enormous proportions to even consider that I might be able to come up with the definitive HLP for all future requirements. But as my ego is only of moderately large proportions (he says modestly) I am only going to *suggest* a possible protocol and then throw the floor open for discussion. In other words, we need a simple, minimal, easy to implement protocol with which we can play and experiment and develop. HLP's are all based upon the use of a 'header' in each message which will contain all the necessary control information. A possible format is shown in Figure 1. The header occurs in every message of the exchange and comprises the following items: (Box 1). It is now necessary to consider how this header might be used for each mode. For example, I would suggest that the first message sent by each processor be a handshake message (Type 'H'). The action code would be set to 'I' and the transmission number to '001'. The data area might consist of:
— User name (20 bytes)
— Telephone number (15 bytes)
— System type (e.g. APPLE II 24 K) (20 bytes)

A handshake message must be received before any further communications can take place. At a later stage, a password might be incorporated in the message.

Immediately handshakes have been exchanged the system can enter dialogue mode to facilitate a conversation between the users. Subsequently, file or program transfer can be initiated.

We will look at these in more detail in another article. In the meantime, please send me your ideas and comments c/o The Editor, Personal Computer World.



HARD TIMES

Winchester discs – secure or not?

Comart Ltd. have just announced the Cromemco Z2H microcomputer system, incorporating a Winchester disc drive. David A. Broad, Managing Director and Chairman of Comart, gives a description of the device and presents some solutions to the data security problems imposed by it.

The emergence of high capacity low-cost disc storage units housed in the physical space previously occupied by first generation floppy disc drives has opened up a whole new spectrum of applications for the small computer and intelligent controller. The technology of "Winchester" drives was first pioneered by IBM with their Piccolo fixed drive systems; they were designed to incorporate high reliability with high capacity. The essence of the concept is a disc pack fixed and spinning inside a sealed enclosure. Air within this enclosure is internally filtered by convection through a micro filter, with the effect that the disc drive has its own environment.

Another aspect of the Winchester concept is that the disc heads and carriage are of very low mass and inertia, allowing them to come within a very close proximity of the spinning disc surfaces. This results in a high sensitivity to flux changes which, in turn, enables a high storage capacity. Electronics are normally mounted outside the enclosure itself, to minimise dissipation and the necessity for access into it.

The concept of the fixed disc is particularly relevant to the microcomputer market. This is a market where the owner, operator and programmer of a computer system is often one and the same person and the environment that the system is intended to work in is that of a normal office, workshop or laboratory. Exchangeable disc systems should ideally only be operated in environments which are controlled in terms of the dust and temperature.

The second important benefit relates to one of the prime aspects of all microcomputer devices..... their very low cost. Systems are being brought to the marketplace now which enclose 11M bytes of hard disc storage, 64K bytes of main memory processors, and interfaces for a VDU and printer – all for less than £5,000; it's the price that the micro user wants to pay.

The other aspect is the very small size of the Winchester disc drives. They can be inserted in physical replacement of standard floppy drives and with very similar power supply requirements. Indeed their ability to work from DC supplies makes them not only very suitable for microcomputer applications, but also ideal for international use where there may be variations in mains frequencies and voltage. Of course the drives will find applications in the minicomputer industry and other types of small computer where low cost and compactness are desirable. But application areas will also open up in communications controllers, word processors and in other dedicated but intelligent devices where high capacity is required.

Security aspects

One of the regular comments made on the fixed disc Winchester drives relates to their use in business applications. Here, the necessity exists for protecting and backing up the data in the event of a catastrophic failure and, indeed, advice is often sought from the suppliers on the best way for a customer to approach this problem.

"..the necessity exists for protecting and backing up the data in the event of a catastrophic failure.."

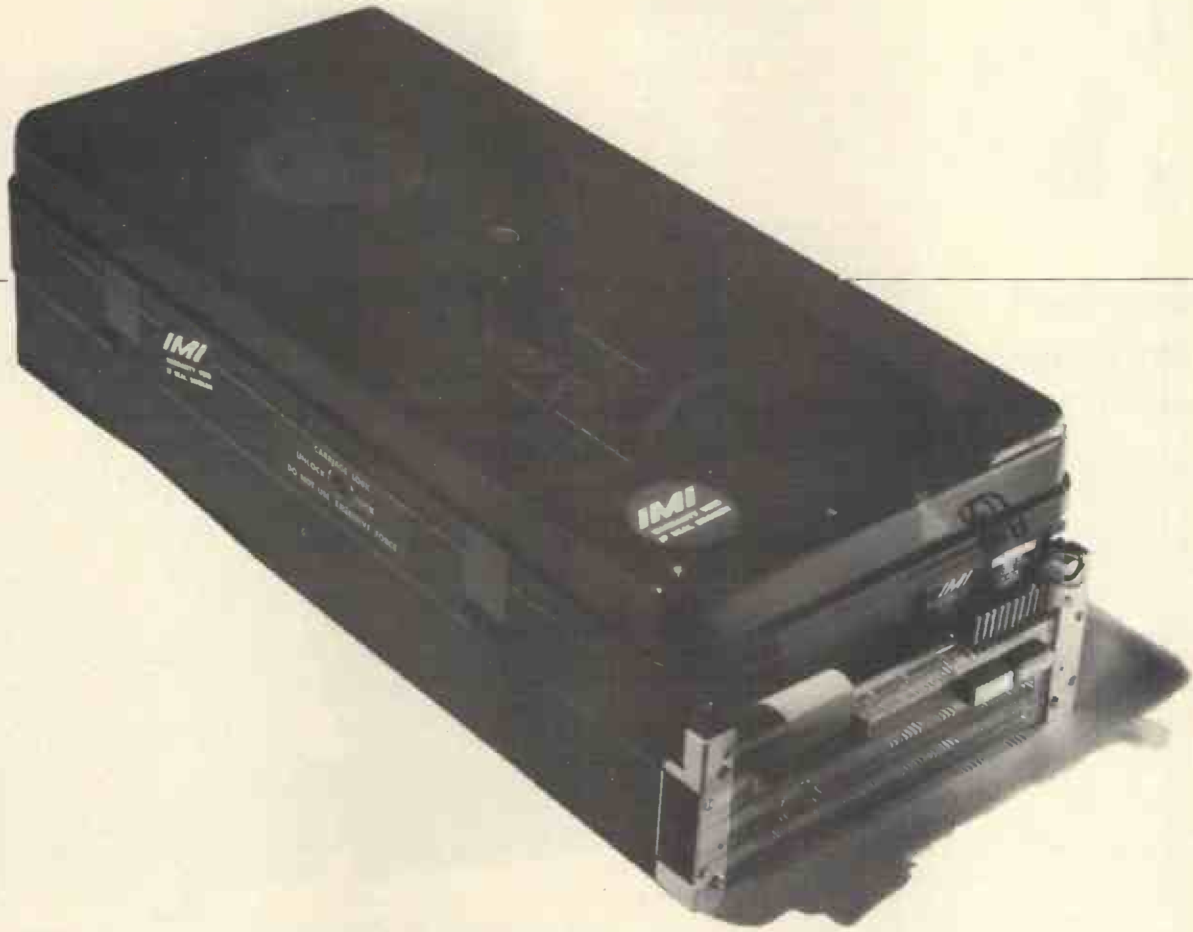
Let us first, however, consider the nature of the problem in relation to the design of the drives. Firstly, because of the light head mechanism and loading techniques, consideration of 'head crashes' is of lesser concern. The media itself is lubricated and in certain circumstances the heads will come in contact with the media (for example during powering up or powering down of the disc drive). Secondly, because of the enclosed environment, the ingress of dust and other foreign bodies is almost totally restricted and build-ups of material on the head is eliminated.

Electronic controller design also allows individual surfaces to be write protected and operating system design further allows faulty tracks to be interchanged in the event of corruption. So the problem of disc failures is greatly reduced. Write circuitry in the controllers is also normally designed for fail safe operation so that the incorrect combination of conditioned signals will result in no current passing through the disc head.

But of course failures can occur and the MTBF of the drives are commonly quoted at 10 thousand hours or so. To back up these disc drives several methods are often proposed by the manufacturers. Firstly, individual files or transaction records can be backed up to floppy discs. Careful systems design can enable the history of changes to the 10M byte data base to be recorded in concise form in transaction records. It is not necessary therefore for the entire disc to be saved in a back up procedure – the latest transactions are merely re-run to a different disc file.

Secondly, many suppliers propose the attachment of an auxiliary cartridge tape system. These systems often use high speed slewing of the tape across the head in order to record a very large amount of data in the shortest possible time. These devices, however, would still take some 15 minutes or so to back up a drive, and current deliverable versions of cartridge units average in the order or 4-5 M bytes total capacity. Also, the data rates proposed often exceed the design specifications of the cartridge media.

The conclusion on tape cartridge back-up media may therefore be



that, because of these limitations, it is not the best method of protecting valuable data. Indeed, it is the very occasion when you need to recover using back up media that the best possible reliability is required.

A third method of data protection is perhaps more practicable and certainly most reliable. That is the provision of a second disc drive which can often be run from the same controller in a daisy chain fashion and which may well have write protect key lock switches for operator protection. With the very high data rates that these drives possess, it is often possible to do a complete back up of data in a minute or so.

It's not surprising that back up procedures are often only followed where the operation is quick and easy. Most will be content to wait a minute or two to undertake a back up, whereas 15 to 30 minutes is unpopular and hence often avoided.

Conclusions

Winchester technology has brought a new impetus to the microcomputer revolution. Few people would have envisaged with the introduction of the floppy disc drive that it would become an essential part of the standard microcomputer system of today. Now, many anticipate that the microcomputer system of tomorrow will include a Winchester drive as a standard feature. There is no doubt that whole new ranges of applications can be brought within the capabilities of the microcomputer and that many who up to now have been using a mini, on an OEM basis, are starting to consider the micro as an alternative tool.

Calculated Corner
Cont. from P. 86

dard algebraic form, e.g. $A^2 + B^2 + SIN\ C^2$ ...... up to 88 characters. On returning to the Computation Mode, the calculator asks you for the variable values (A = ?) and evaluates the expression! Editing is via a flashing cursor, and expressions longer than the display scroll off to the left. The eleven memories are non-volatile and a full set of scientific and statistics functions are provided.

The sophistication of this display is such that, surely soon, once large scale LCD displays are available, we will have a hand held micro which is programmed in BASIC!

For a user who needs to evaluate many algebraic functions and doesn't want to trouble to learn a calculator "language" for programmed solutions, this calculator will be useful; anyway, it's certainly a status symbol. But the impossibility of any sort of recursive operation limits its flexibility drastically and at a price of £69.90, it cannot compare well with various programmables in this cost range. Perhaps worse is that it takes away all the fun of playing computers!

BOOKFARE

Malcolm Peltu has made his name writing and lecturing about the nature and impact of computer-related technologies.

In the grip of the systems monster

Why is it that a personal computing "amateur" can produce a multiprogramming operating system, compiler and utilities in a couple of years part-time work, whereas "professional" software suppliers can invest decades of man years to less avail?

"Perhaps it was because I didn't know that some problems existed, so I never encountered them", was the answer given to me by Tom Aschenbrenner who won the 1977 Personal Computer Fair competition at the US National Computing Conference (he had developed a message switching system for fellow radio hams in the Dallas area). As, another reason he added, "I did it because I enjoyed it — not because I had to, in order to earn a living." Aschenbrenner's comments would fit very pertinently into *Systemantics* by *John Gall* which tries humourously to analyse the behaviour of a beast which seems to run amok through so many aspects of modern life — The System. Written in a sometimes irritating, too-clever-by-half style, *Systemantics* offers a number of "laws" relating to systems behaviour, often paying homage to folk-lorish inventions like Murphy's Law that "If anything can go wrong, it will go wrong."

The Fundamental Theorem of *Systemantics* is that "New systems create new problems". According to Gall, one starts with a problem, like getting rid of rubbish. Then a system is set up to organise garbage collection and the main objective of that organisation is to *manage the system* rather than to solve the original garbage problem. In fact, Gall says, "for the practising systems-manager, the greatest pitfall lies in the realm of problems and problem-solving. Systems can do many things, but one thing they emphatically cannot do is to solve problems. This is because problem-solving is not a systems-function, and there is no satisfactory systems-approximation to the solution of a problem. A system represents someone's solution to a problem but does not *solve* a problem.

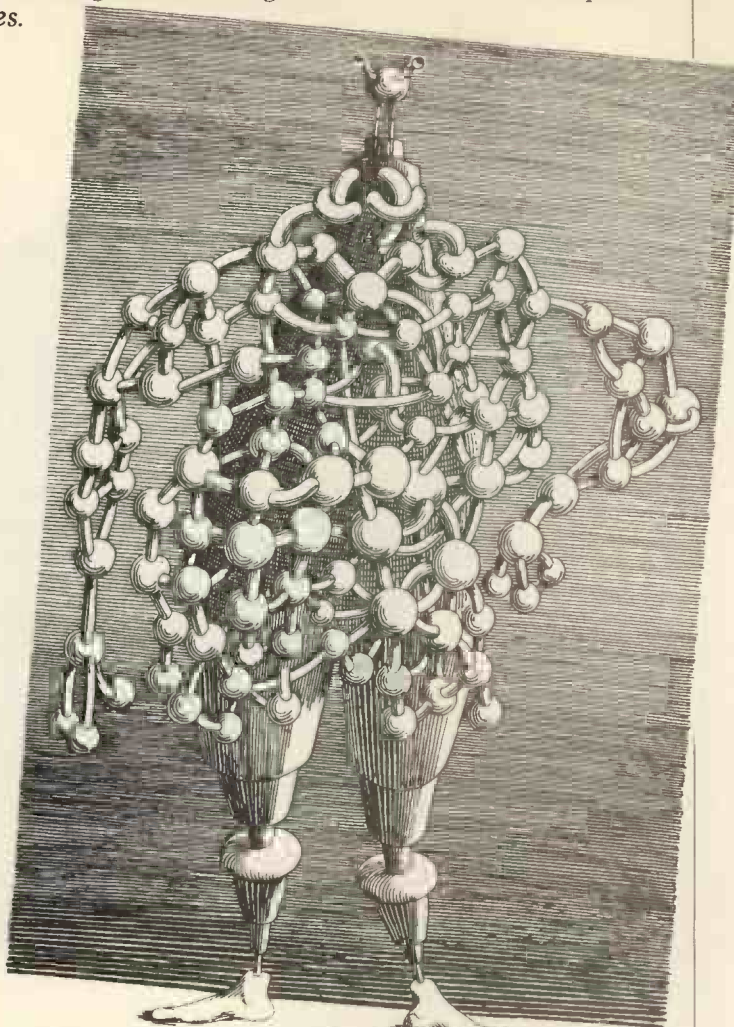
"Solutions," he continues, "usually come from people who see in the problem only an interesting puzzle and whose qualifications would never satisfy a select committee." There could be no better proof of Gall's pudding than in the exciting tang of the home-brewing personal computer world. Aschenbrenner's remark about not being aware of the problems was true because many of the problems of software development are concerned with the complexities of managing large projects. A programmer in a Data Processing department or large software development team is likely to be less productive and less creative than a hobbyist.

Gall's most biting comments are lashed out at the dangers of trying to control complex systems. "Any large system is going to be operating most of the time in failure mode," he says, putting the boot in further with the Fundamental Failure Theorem that "a system can fail in an infinite number of ways" and that "the mode of failure of a complex system cannot ordinarily be predicted from its structure."

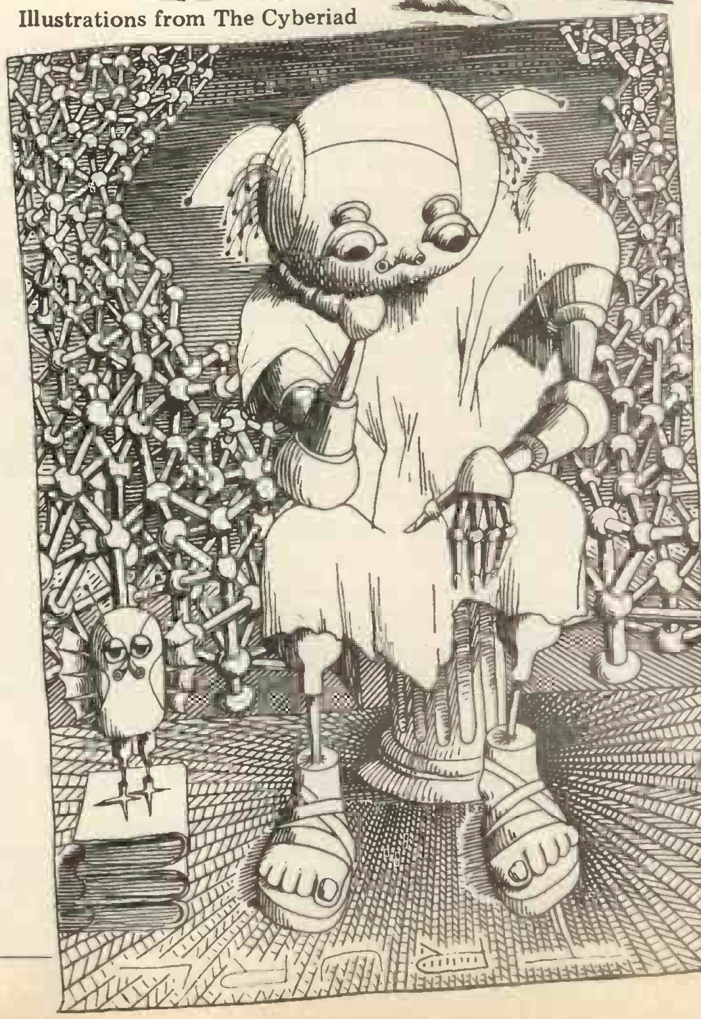
For programmers he has two special axioms: programs never run the first time, and complex programs never run. In a more general context, these axioms are summarised by his belief that complex systems designed from scratch never work; the only complex systems that do work are those which have evolved from successful simple systems.

I believe that computers are ideal models of Gall's system world. Operating systems which hog machine resources in order to sort out machine rather than user problems are perfect examples of how complex systems fail to tackle the problems for which they were initially created. The way in which the personal computing market has focused on the development of more human interfaces, such as colour graphics, has also shown that the computer industry as a whole proves another Gallism, "To those within a system, the outside reality tends to pale and disappear."

At the last National Computing Conference in New York, the hundreds of stands from the traditional computer industry paid lip-service to "user needs" but were still essentially displaying evolutions from the grey elephants which form



Illustrations from The Cyberiad



their "user base". The personal computing show at the same event was filled with colour graphics, voice synthesisers and other devices that offer a human window to the computer. And the reason is that the personal computer user is also a systems developer who focuses attention to the main goal of using the system.

For larger, established computer companies, the System has other goals, like growing and extending its administrative machine, its sales targets and all those other factors that have little to do with the user.

Systemantics is a book with a serious message and you'll find it if you scratch beneath its glossy, over-jokey veneer. Anyone working a bureaucracy whose purpose is to deal with people in need should, for example, spread the message "The dossier is not the person" which is Gall's extension of a sign he saw in a smallish hospital which said "The chart is not the patient".

The underlying strength of Gall's book is that it is based on a good appreciation of General Systems Theory and cybernetics. With tongue in cheek, Gall does in fact admit that the science of General Systemantics is a spoof of General Systems Theory, an idea inspired by one Ludwig von Bertalanffy (who coined the phrase *system* to describe the entity concerned with the organisation of a function rather than the function itself).

In *An Approach to Cybernetics* by *Professor Gordon Pask*, von Bertalanffy's work is cited as one of the sources that lead to the study of cybernetics. Where Gall provides some glib but perceptive insights into the complexities of Systems thinking, Pask, who is a professor of cybernetics at Brunel University, takes a more scientific and mathematical approach in trying to explain the background and scope of one of the major Systems "sciences".

For example, Gall and Pask both quote Le Chatelier's Principle, derived from chemistry, which states that any natural process tends to set up conditions opposing the further operation of that process; thus equilibrium can be maintained when various forces, such as chemical reactants, are mixed in certain concentrations in a closed vessel.

Gall turns this Principle into a corner stone of General Systemantics, that "Systems get in the way and Systems tend to oppose their own proper functions". He clarifies this by the example of a research worker who is asked to define his aims and objectives to satisfy various organisational Systems needs — like touting for research money. So he makes up objectives that look good to the System, such as writing *x* papers in a year, even though his real objectives are different. But he then has to waste time meeting his Systems-inspired objectives. The System has therefore got in the way of real objectives.

For Pask, Le Chatelier provides a simplified analysis of what he regards as the crux of organisational Systems study — stability. "That which is stable can be described, either as the organisation itself or some characteristic which the organisation preserves intact". He writes "The trouble with cybernetics is that the very substance of its study is an entity as amorphous and generalised as the words 'organisation' and 'systems'."

Pask, however, makes a brave attempt at trying to explain in relatively simple terms the unique characteristics of a science which, as he says, "cuts across the entrenched departments of natural science; the sky, the earth, the animals, and the plants." The book is well worth reading as a first step towards a deeper understanding and involvement in a subject which both fascinates and confuses by its general applicability to anything — from running a company to developing a computer to studying the brain.

A mad and magnificent book which puts the whole Systems and cybernetics approach into an imaginative galactic context is *Stanislaw Lem's The Cyberiad*. A combination of science fantasy, political satire and mathematical impishness, *The Cyberiad* consists of a number of short fables, most of them loosely linked by the journeys of the 'constructors' Trurl and Klapaucius.

The starting point of each fable is often a superbly illogical logical idea like a machine that can create anything that begins with *n*, then causes havoc when asked to create Nothing. Or the stupidest eight storey thinking machine in the world which terrorises the constructors because they challenge its belief that 2+2=7.

My favourite is Trurl's Electronic Bard, the poetry machine. In Lem's words "Whenever Trurl felt he just couldn't take another chart or equation, he would switch over to verse, and vice versa. After a while it became clear to him that the construction of the poetry machine itself was child's play in comparison with writing the program. The program found in the head of an average poet, after all, was written by the poet's civilisation, and that civilisation was in turn programmed by the civilisation that preceded it and so on to the very Dawn of Time. Hence, in order to program a poetry machine, one would first have to repeat the entire Universe from the beginning. . ." and that is what Trurl does to Universe-shattering effect.

A trip on Lem's Cyberiad machine gives a whole new perspective to the real world of machines, people and organisations and helps to point to the farcical pimples on the bum of the Systems beast.

Learning the lingo

One day Grace Hopper, one of the founders of the Cobol programming language, found herself lost in Tokio. And she managed to get back to her hotel merely by speaking Cobol words such as MOVE and GOTO because, she says, Cobol uses such basic universal commands.

It would be nice if a stranger lost in computerland could rely on a similar simple language. (In parentheses it is worth noting that Grace Hopper is said to have originated that descriptive computer jargon word 'bug', meaning an error. According to the story, one of the early computers with which she was working was giving a lot of trouble, until one day she opened a processor cabinet and a moth flew out. Hence the 'bug' came into being).

Meanwhile, back with the stranger in computerland, it is necessary to provide him/her with two forms of route finding assistance — firstly some guidance through the jargon used to describe the technology, and then help with learning the programming languages that get the machines doing something useful.

The problem with introductory books in computing is that they tend to be either too general, and therefore of little use in finding out about one system, or else they are far too specific to give a good perspective on the intrinsic points of the technology.

Introduction to Microprocessors by G. L. Simons offers a general overview of the hardware and software technology together with sufficient detail of some popular processors, languages and microcomputers to give the stranger some confidence and sense of direction; however it can in no way be taken as a training or reference book.

In addition to the systems, Simons provides a useful overview of design needs and the range of applications for micros, as well as a summary of some contemporary views on the social consequences of microcomputing.

When it comes to learning a computer language, one's choice is usually limited to those available on the machine at hand. With microcomputers, the most widely available language is, of course, BASIC.

The trouble is each machine has its own restrictions and dialects for any given language. And this is the major drawback with the otherwise excellent *The BASIC Handbook by David A Lien*

The Handbook is aptly described by its subtitle as being an "encyclopedia". It clearly admits in the introduction, however, that it is not intended to replace the manufacturer's handbook which describes the language facilities for that machine. Instead it concentrates on simple, clear descriptions of the fundamental core words that are common to most machines using BASIC.

The aim is to provide some help to those who wish to adapt programs in a magazine like Personal Computer World into suitable forms to run on particular machines. Each BASIC word discussed starts at the top of page. They are listed alphabetically and an indication is clearly given whether the word is part of the American National Standards Institute (ANSI) BASIC standard.

Then there is a standard list of topics dealt with for each word, such as its word category, general description and variations that might be encountered. Test programs for the word and sample runs are also given as well as some very useful hints, including what to do if your computer does not have a particular word. Used as an encyclopedia, the Handbook will be exceedingly helpful in a variety of ways.

But, as Lien says, "like the expanding universe theory, BASIC keeps expanding; we can only chase it — but never catch it all."

So, although the BASIC Handbook will shine a guiding light through some unknown territory, it will still be necessary to get a more detailed and updated A to Z of any real system you want to use.

Books discussed in this month's Bookfare have been:
Systemantics by John Gall (Fontana, 85p)
An Approach to Cybernetics by Professor Gordon Pask (Hutchinson's Radius Books, £1.00)
The Cyberiad by Stanislaw Lem (Secker & Warburg, £3.90)
Introduction to Microprocessors by G. L. Simons (National Computing Centre, £6.50)
The BASIC Handbook by David Lien (Compusoft, available through Rostronics, 118 Wandsworth High Street, London SW18 — £11.00)

THE ALPHA MICRO COMPUTER

Multi-User, Multi-tasking, Timesharing, Memory Management

Basic 64K RAM, 2·4Mb Floppy Disk System: £6,496.00
Basic 64K RAM, 10Mb Hard Disk System: £9,965.00
(Terminals & Printers to be added to user specification.)

ALPHA MICRO gives a new meaning to the words "Cost Effective." It combines a powerful 16 Bit processor with a proven time-sharing disk operating system to give you data handling and software sophistication parallel to that of high performance commercial minicomputers. It can be upgraded from a simple 64K single terminal floppy disk system up to a 24 terminal, multi-printer, system with 2400 Megabytes of disk storage and 1·02 Megabytes of Random Access Memory without any hardware redundancy.

ALPHA MICRO in Business

A fully integrated Accounting System is available "off the shelf." It includes Order Processing, Automatic Invoicing, Stock Control, Accounts Receivable, Accounts Payable, Nominal Ledger, Payroll, and Sales Analysis by Customer, Product or Salesman.

ALPHA MICRO Word Processing

Comprehensive word processing software is available which can run simultaneously with the accounting system (or any other program for that matter). It will handle anything from standard letters up to large and voluminous documents with automatic Index/Table of Contents generation.

ALPHA MICRO in Research & Education

Since the system can handle up to 24 terminals, where each user terminal has its own 32 or 48K memory partition, it is ideal in education or research since each user can do his own application, i.e. one can be running the BASIC Compiler whilst another runs LISP; again another can do programming in PASCAL or ASSEMBLER etc.

ALPHA MICRO Standard Features

* Multi-User, Multi-Tasking, Time-sharing Disk Operating System
* Memory Management from 64Kb-1·02Mb
* Disk storage from 2·4Mb-2400Mb
* Powerful WD16 16-Bit Processor
* S100 Bus Compatible
* Expands from 6-24 terminal ports
* Multi-Printer Spooler
* Adaptable to most RS232 peripherals
* Sequential, Index Sequential and Random Access files supported
* Comprehensive disk file management system and utilities
* Multi-User structured file system with programmer/project number and password protection
* Command file interpreter with parameter substitution
* Multiple level DMA and vectored interrupt system
* Multiple pass assembly programming system with linking loader
* ALPHABASIC Extended compiler and re-entrant runtime package
* Index sequential files supported in both Assembler and ALPHABASIC
* File management system with logical file I/O calls
* ALPHAPASCAL, one of the best UCSD implementations
* ALPHALISP, a textual data manipulation language

THE ALPHA MICRO COMPUTER

newtons LABORATORIES
PO Box 789
123 Wandsworth High Street
London SW18 4JB

Tel: 01-870 4248 Telex: 929222 (SLOTS G)

Request for ALPHA MICRO brochure PCW 11
Name
Title
Company
Address

Postcode Tel:

DIARY DATA

London, England	Business Computing, Word Processing & Information Mgt., Exhibition & Conference. BED Exhibitions Ltd., Bridge House, Restmor Way, Wallington, Surrey. SM6 7BZ. Tel: 01-647 1001	Feb 12 - Feb 15
Wembley, England	INFO EUROPE — European Information Management Exhibition & Conference. Clapp & Poliak Europe Ltd., 232 Acton Lane, London W4 5DL. Tel: 01-995 4806	Feb 18 - Feb 21
Bournemouth, England	BEX — Business Equipment Exhibition. Douglas Temple Studios Ltd., 104b Old Christchurch Rd., Bournemouth, Dorset, Tel: 0202 20533	Feb 20 - Feb 21
Swansea, Wales	OFFEX — Office Equipment Exhibition. Phoenix Exhibitions Ltd., 1st Floor, Burrows Crambers, East Burrows Rd., Swansea. Tel: 0792 460364	Feb 20 - Feb 22
Dortmund, W, Germany	HOBBYTRONIC — Electronic Hobby Exhibition. Westfalenhalle GmbH, Postfach 1130, Reinlanddamm 200, 4600, Dortmund, W. Germany	Feb 20 - Feb 24
Birmingham, England	IEA — International Instruments, Electronics & Automation Exhibition. Industrial & Trade Fairs Ltd., Radcliffe House, Blenheim Court, Solihull, West Midlands, B91 2BD. Tel: 021 705 6707	Feb 25 - Feb 29
Copenhagen, Denmark	TECHEX — World Fair of Technology Exchange. Dr Dvorkovitz & Associates, P.O. Box 1748, Ormond Beach, Florida 32074 U.S.A.	Feb 26 - Feb 29
Birmingham, England	Computermarket '80, Couchmead Ltd, 42 Great Windmill Street, London W1V 7PA. Tel: 01-437 4187	Mar 4 - Mar 6
Liverpool, England	Merseyside Business Efficiency & Office Equipment Exhibition. Gwen Shillaber Design, 81 Whiteladies Rd., Clifton, Bristol BS8 2NT. Tel: 0272 312850	Mar 4 - Mar 7
London England	Microforum Europe. Business Equipment Trade Association, 109 Kingsway, London WC2B 6PU. Tel: 01-405 6233	Mar 11 - Mar 13
Sheffield, England	Business Efficiency & Office Equipment Exhibition. Gwen Shillabar Design, 81 Whiteladies Rd., Clifton, Bristol BS8 2NT. Tel: 0272 312850	Mar 11 - Mar 13
Manchester, England	Computermarket '80. Couchmead Ltd., 42 Great Windmill St., London W1V 7PA. Tel: 01-437 4187	Mar 11 - Mar 13
Bahrain, UAE	Middle East Business Equipment Show. Arabian Exhibition Management 11 Manchester Sq., London W1M 5AB. Tel: 01-486 1951	Mar 16 - Mar 20
Glasgow, Scotland	Computermarket '80. Couchmead Ltd., 42 Great Windmill St., London W1V 7PA. Tel: 01-437 4187	Mar 18 - Mar 20
London, England	Computermarket '80. Couchmead Ltd., 42 Great Windmill St., London W1V 7PA. Tel: 01-437 4187	Mar 25 - Mar 27
London, England	Viewdata '80. Online Conferences Ltd., Cleveland Road, Uxbridge, Middx UB8 2DD. Tel: 0895 39262	March 26 - March 28
Paris France	International Exhibition of Electronic Components. French Trade Exhibitions, 54 Conduit Street, London W1R 9SD. Tel: 01-439 3694	Mar 27 - Apr 2

COMPETITIONS ROUND-UP

One of the less appreciated inheritances for the new team at PCW was a collection of hitherto unresolved competitions. We believe, after some hours of research, that five (and possibly six) sets of results are outstanding (!) — please let us know if you spot others — and of those, at least two still require their instigators to pass judgement. They are: "Puzzle Dazzle 2", set in the February '79 issue; "Alphametics" set in the May '79 issue. Others, which we can deal with now, are: "Magic Squares", set by Sheridan Williams in the June '79 issue: "Knight's Tour", again set by Sheridan Williams, this time in the January '79 issue; finally, "Witbit 1" set by David Parkinson and Graham Trott in the June '79 issue.

It's obviously most unfair that the winners be kept waiting any longer. However, it'll come as no surprise to everyone to learn that the outlining of all these reports would take up far more space than any one PCW issue could possibly donate (not to mention the possibility of our readers going down with a nasty bout of 'competition overkill').

Therefore, this month we are restricting ourselves to little more than a round-up of results although, through later issues, we hope eventually to publish a much fuller analysis.

KNIGHT'S TOUR

The problem was to find a complete tour of the chessboard for a knight, so that the piece visits, in turn, every square on the board once, and once only.

Sheridan had purposely set a difficult competition. . . and yet the entries still came flooding in. Joint winners (£10 each) are Philip Crane of Romford in Essex and Brian Legg of Bishops Stortford in Hertfordshire. The run times of the two programs were 3.87 seconds and 2.36 seconds, respectively.

MAGIC SQUARES

The problem here was to find a magic square that satisfies the following conditions: (1) It comprises of 2-digit numbers (zeroes not allowed) (2) It's a 3x3 square (3) When the digits are reversed, another magic square is produced with none of the original numbers reappearing (4) The sum of the *two* magic constants is less than 200.

A prize was offered for the first correct entry supported by BASIC program. The solution was as follows:

13	34	25		31	43	52
36	24	12	reversed	63	42	21
23	14	35		32	41	53
MC=72				MC=126		

The winner is: Mr O. M. Dixon of Alverstoke in Hampshire, who receives £10. A consolation prize of £5 goes to Mr C. Palmer of Bradford, Yorkshire who, although not first out of the 'hat', submitted the best entry.

WITBIT

The problem set was to write a short subroutine for an editor to execute a "Find string" command.

Solutions were accepted in Z80 and 6800 code, prizes to be awarded to the winners of each section. **Z80.** First prize (£10) goes to Mr J. Robertson of East Kilbride, Scotland for his neat solution which uses the minimum of temporary storage. Second prize (£5) goes to David Medland-Slater of Farnborough in Hampshire. **6800.** Only four entries received in this section! Winner (£10) is Martin Bond of Didsbury, Lancashire and runner-up (£5), John Phillips of Saltash, Cornwall.

Cryptic clue

Re September Issue 1979, Spaceship FX201-P . I have an entry for your diversions and puzzles page.

With reference to the above article:-

A Try and figure out how the list of step numbers applies to the program.

B Make a list of incorrect function signs.

C Fill in the missing line which would give answer 4 — your new radial distance.

I would be most grateful if you could send me the same copy that Dick Pountain wrote his review from.

E. Fernie, Enfield, Middx

Okay, pax. . . Corrections to Spaceship FX201-P in Blunders at the end of our Programs section.

Faith healing

Further to my letter of 1st June, 1979, concerning difficulties I have experienced in obtaining a MPS 6550 memory chip for my Commodore PET. I now have the greatest pleasure to inform you of the remedy. . . a small shop in the town of Luton, Bedfordshire, called Isher-Woods. I rang them and explained my problem and was told that they had the devices in stock. I was invited to take my PET over to them where they could test the device in situ.

Once at the shop I was received by their Wizard, John Rees, who operates in a well-organised (you should see mine) workshop with an air of calm confidence and rather like a slow-motion Magnus Pyke. That he knows what he is doing is evident from the constant flow of people seeking his advice and leaving satisfied. I was invited to participate in the operation (painless) but the best part was enjoying the interesting chat seeded with snippets of valuable information.

The Wizard introduced me to the Vizier, Ian Wade who is their Divisional Controller and obviously knows a lot more about the aspects of hard and software than he is telling! If you wish to discuss the purchase of either he is in an ideal position to give you the "low down" on it. I wonder how many persons in his position can say the same.

Throughout my visit I was impressed with the atmosphere of friendly co-operation and enthusiasm and I strongly urge anyone in the vicinity, to drop in and say hello. This really is an unsolicited testimonial; unless Isher-Woods reads your worthy publication they will never know about it.

S.R. Somers, Aylesbury

Well earned "plugs" we never mind repeating — Ed.

Punter postscript 1

I read with interest "The Programmed Punter" by Dr. M.R.J. Morgan in the July issue of PCW. I was surprised at the low limits on the value of the permutations he could calculate until I realised that he calculated them from three factorials.

This is a very long winded and restricting method as many terms in the fraction always cancel out. To take his example:

$$\frac{8!}{3!\times 5!} = \frac{8\times7\times6\times5\times4\times3\times2\times1}{(3\times2\times1)\times(5\times4\times3\times2\times1)} = \frac{8\times7\times6}{3\times2\times1}$$

The subroutine at line 1000 in the program below calculates the permutations this way cancelling the larger factorial on the bottom into the top.

```
10 REM *** BINOMIAL CO-EFFICIENTS ***
20 INPUT "M, N"; M, N
30 IF M=0 THEN END
40 GOSUB 1000
50 PRINT C
60 GOTO 20
1000 REM $1$ BIN CO $1$
1010 IF M-N>=N GOTO 1030
1020 N=M-N
1030 C=M
1040 IF N=1 GOTO 1090
1050 M1=M+1
1060 FOR I=2 TO N
1070 C=C*(M1-I)/I
1080 NEXT I
1090 RETURN
```

This program will allow your readers to see how large they can get M and N on their systems. With M=122 I can do all values of N although my RML380Z overflows at about 10^{38}.

Hugh Williams (Past Chairman MUSE) West Bridgeford, Nottingham

Punter postscript 2

In Dr Morgan's short article "The Programmed Punter" (July PCW) he used the formula M!/N! (M-N)! which gives the number of combinations of N objects that can be chosen from M unlike objects. ABC and ACB are different permutations of the same combination of letters.

An alternative way of calculating the number of possible combinations, other than evaluating the three factorials as that program does, uses the fact that the coefficients of the binomial expansion $(x+y)^M$, are the number of combinations of 0,1,2,...,M objects from M objects. For example, the coefficients of the terms of $(x+y)^7$ are 1,7,21,35,35,21,7,1 so there are 21 possible combinations of 2 objects from 7 objects.

Successive coefficients are related by:

$$^{m}C_{n}/^{m}C_{n-1}$$

$$= m!/n!(m-n)! \times (n-1)!(m-n+1)!/m!$$

$$= (m-n+1)/n$$

$$^{7}C_{3}/^{7}C_{2} = 35/21 = 5/3 = (7-3+1)/3$$

$$\text{so } ^{m}C_{n} = {}^{m}C_{n-1} \times (m-n+1)/n$$

As $^{m}C_{0} = 1$, the number of combinations can be evaluated by successively multiplying by (M-I+1)/I where I=1,2, ...,N. A program, assuming that the data is always correct being:

```
10 INPUT M,N
20 C = 1
30 FOR I=1 TO N
40 C= C*(M-I+1)/I
50 NEXT I
60 PRINT C
```

As $^{m}C_{n} = {}^{m}C_{m-n}$ another line could be added to increase the efficiency of the program

```
15 IF M-N <N THEN N=
M-N
```

Using this algorithm, our RM 380Z, using DBAS12, can evaluate the number of combinations of N objects from 123 objects to an accuracy of 10 significant figures. Some results for values of M greater than this can be obtained, but the greater M is, the smaller N has to be. When M= 300,N can only be 26 or less. Using the formula for the number of combinations directly, the greatest value of M is 33.

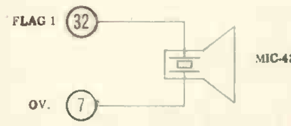
The formula for the number of permutations of N objects chosen from M unlike objects is M!/(M-N)! and a similar method of evaluation can be used that does not involve working out factorials.

As the order in which multiplication and division are executed does not matter, it is worth investigating to see if the equivalent statement C=C/I*(N-I+1) will enable greater values of M to be evaluated.

Peter Butt, Chadwell Heath, Essex.

MK14 sound out

In musical or noise generating applications of the Science of Cambridge SC/MP based Mk.14 microcomputer, a crystal microphone insert may be used as a high impedance loudspeaker, driven directly by the logic levels at the flag outputs of SC/MP. The prototype used an ACOS type MIC-43 connected as shown below.



The brief program listed here may be used to generate a sound to test the set up. It operates by incrementing a store, loading the result to the Status (flag) Register, and also using the result as a parameter for a variable delay before jumping back to the start. The program is written in locations 0F20 to 0F26 inclusive with 0F1F as a store but it is relocatable to any eight contiguous locations in RAM. The type of output sound may be varied by altering the delay base number in location 0F24.

```
0F1F  00     Store:
0F20  A8FE   Start:  ILD   Store
0F22  07             CAS
0F23  8F00           DLY
0F25  90F9           JMP   Start
      0000           .END
```

See the Mk.14 User Manual Music section for more ambitious programs using this circuit.

T. J. Spriggs, Havant

Dodgy delays

The SC/MP micro-processor has a DELAY instruction (opcode 8F) which provides a pause, corresponding to a specified number of microcycles, with very simple software. (In the 6800 MPU, we have to write a short subroutine to achieve the same effect).

In the manual for the Mk.14 (which uses the SC/MP chip) there are programs (on Pages 65 and 66) for Serial Data Input and Serial Data Output. These programs use the DELAY instruction. In a program I wrote, similar to the Serial Input program, I found that the constants for the DELAY operation suggested in the Mk. 14 programs, were not correct. My program worked correctly when the constants were changed to values which were found after some experiments guided by theory.

Page 64 of the manual gives a table for delay constants based on a frequency of 4 MHz. These constants are used in the Mk. 14 program for Serial Data Input. But the Mk. 14 works in association with a crystal which has a frequency of 4.433618 MHz (this is the value stamped on the casing of the crystal). It would therefore seem to be wrong to use the constants in the table.

In my project, I was working at 300 baud, the "bit time" is 3.333 milliseconds. For this condition, the table suggests

"C4 5E
8F 03"

This means "Load 5E into the accumulator and set the displacement in the DELAY instruction to 06".

This leads to a delay of n microcycles, where n is given by 13 + 2 x (accumulator) + 2 x displacement + 2^9 x displacement; i.e.
13 + 2 x 5E + 2 x 6 + 512 x 6.

This statement is somewhat confusing in that decimal numbers are mixed with hexadecimal. "5E" in hex means "94" in decimal. Thus n =
13 + 188 + 12 + 3072
= 3285 microcycles.

At a frequency of 4 MHz, one microcycle lasts 1 microsecond. The delay is thus 3.285 milliseconds.

We require 3.333 milliseconds. This leaves 48 microcycles for the SC/MP instructions — a reasonable figure.

However, when SC/MP is working at 4.433618 MHz, the delay corresponding to 3285 microcycles is

$$3285 \times \frac{4}{4.433618} \text{ microsecs}$$

i.e. 2.9637 milliseconds.

For my project, I found that the program would work with

"C4 22
8F 07"

The constants were chosen by taking the mean of the upper and lower limits of the delays found to be satisfactory.

Because of the synchronising action of the START and STOP bits in the program, there is a certain range within which operation is satisfactory. With "8F 07" the program would work with values stored in the accumulator varying from 00 to 44 (in hex).

In addition to changing the constants for "bit time", I changed those required for "half bit time". To save space, those changes will not be mentioned here. The purpose of this letter is to suggest that values in a table intended for a frequency of 4 MHz should not be applied to an MPU working at 4.433618 MHz.

Tom Palmer, Kew

We rang Science of Cambridge who confirm your observations. They did point out, however, that since January they have incorporated a genuine 4MHz crystal. They also mentioned that their manual page numbers have changed following a recent update so you may find Tom's references different to your own. One last thing — they also told us that because the input and output routines use the same byte in memory, spurious data can find its way onto the output line as it is displaced by incoming data. You can work out your own solution to this by either dealing with the content of this byte before a read or by 'gate'-ing the output. — Ed.

Pascal possibilities

Alex Cawley's letter in your September Issue gives incorrect information concerning the availability of PASCAL Compiler RAM requirements.

Our company has a 3 Pass PASCAL Compiler designed for the RCA 1802 Microprocessor which runs in a 20K RAM System with Floppy Discs. This Compiler, whilst designed for the 1802 family, can be adapted to other microprocessors by alteration of the 2K run time kernel to suit the required instruction set.

The 2K interpreter makes application programs as small as 3K a practical possibility; the package is designed to appeal to the professional and industrial user looking for minimum read only memory costs.

M. J. Dalgleish, Golden River Company, Bicester

Routine business

I read an aritcle in PCW recently describing Dr. Roger Quy's 380Z system at the National Hospital's Institute of Neurology. In it, Dr. Quy was quoted as saying that he had found PCW to be a useful source of assembler multiplication and division routines. As a fairly recent convert to PCW and a new user of a 380Z, I should like to track down these routines. I wonder if you can quote me chapter and verse? I'd be very grateful.

Mrs A M Guenault, Lancaster

We rang Dr. Quy and, with his help, tracked down an article by Neil Harrison in volume 1 number 2. It's called 'Four Easy Pieces' and in it, among other things, he describes a multiplication routine. He thinks that his division routines came out of a hardware manual — Ed.

Was ist das?

I would be grateful if you could kindly inform me of any computer that translates German into English, it would also be a great help if you could supply the companies' names and addresses.

PS I do take your magazine.

H. Thomas, Shirehampton, Bristol.

Nice to hear from another discerning reader! The company distributing translators in the UK is Lexicon. Their head office is in Parliament Street, London (Tel: 01-930 3030). They supply to shops all over the country — the nearest one to you is probably Communications Imports in Cheltenham. The phone number there is 0242 41173. It is probably worth noting that the translator has a repertoire of some 1500 words and translates word for word in the present tense, first person singular. Therefore, although it's no replacement for a human interpreter, it does provide a very useful means of communication. One last thing — price; a Lexicon 3000 with one language module of your choice costs £148 + 15% VAT. Each additional module costs £32.95 + 15% VAT. Each module plugs in and allows translation in either direction.

Stop Press: Lexicon have just announced that they are selling 'personal program' modules — you can store recipes, 'phone numbers jokes etc. — Ed.

Sorcerer tips

Despite claims to the contrary, there is no GET statement in Sorcerer BASIC. It is, however, possible to simulate a GET statement using a machine code routine that is POKE'd in from BASIC.

The statements are as follows:

firstly,

```
FOR X = 1 to 14
READ W
POKE 223 + X, W
NEXT X
DATA 62, 0, 50, 240, 0, 205,
9, 224, 200, 50, 240, 0, 201,
0.
```

then,

```
POKE 260, 224
POKE 261, 0
```

To use this routine:

V = USR (0) : A = PEEK (240)

A now has the value of the ASC11 code of the last key to be pressed.

If no key is pressed then A = 0.

An example is shown below:

```
10  PRINT "DO YOU WANT TO CONTINUE"
20  V = USR (0) : A = PEEK (240)
30  IF A = 0 THEN 20
40  IF A = ASC ("Y") THEN 80
50  IF A = ASC ("N") THEN 100
60  PRINT "RESPONSE INVALID"
70  GO TO 20
80  REM Do something
90  GO TO 10
100 END
```

To control a printer from BASIC, rather than from the monitor, USE the output vectors:

7FD0 H 32720 D
and 7FD1 H 32721 D

The contents of these locations will change depending on the output option selected. To turn on the line printer (Centronics) POKE 32720, 147 and to switch it off again POKE 32720, 240. (A word of warning, we've found that if we mess about in the monitor before executing POKE 32720, 147 we lose our program entirely — some caution is needed).

Rob Beynon, Liverpool University.

SP

SIRTON PRODUCTS

13 Warwick Road, Coulsdon, Surrey, CR3 2EF
Telephone : 01 - 660 5617

Professional versatile computer system with comprehensive front panel facilities and 20-slot motherboard. Units have substantial power supply etc. and come with 2 or 4 MHz Z80 CPU. BUS conforms to the IEEE S100 standard.

DPS.1 from £695

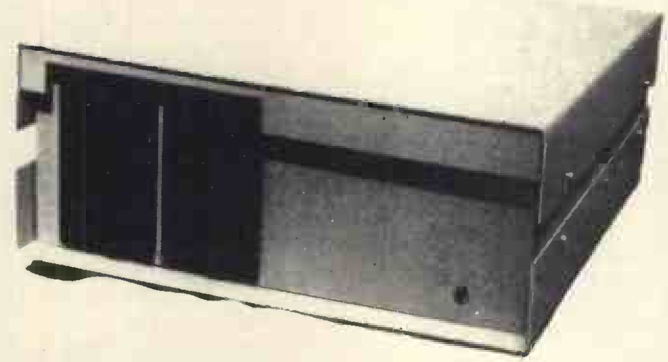
Available with K2 operating systems & PASCAL/Z. Companion Disc Drive Enclosure for on-line storage for 250K Bytes to 2 M Bytes.

MIDAS

MIDAS S.100 SYSTEMS

Substantial Mainframe to house your S100 system, with optional 5" or 8" disc drives. Special systems built to your requirements from Z80 CPU and other S100 boards held in stock

Mainframes from £228
MIDAS 1 : Z80 System from £625 (built)
MIDAS 2 : Z80 Disc System from £1100 (built)

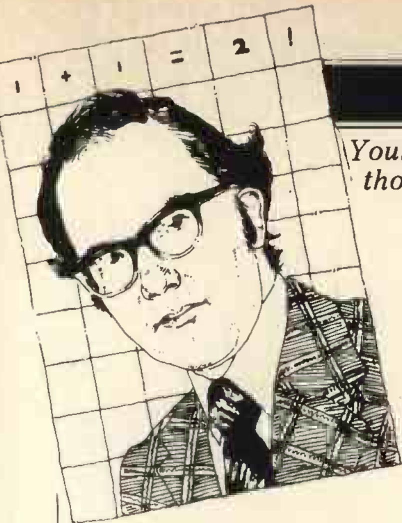


DOUBLE DENSITY recording available on MIDAS giving up to 2M BYTES of on-line storage. Software for MIDAS includes CP/M, FORTRAN, COBOL, PASCAL and several BASICS including XYBASIC for control applicaitons.

BOARD KITS — many also as bare boards	
Z80 Starter Kit, featuring on board Keyboard, 2K Monitor, 1K RAM, 2x8 bit I/O Ports, Prom Programmer etc.	£145.00
Z80 CPU, 2 MHz, 1K RAM, up to 8K EPROM, Serial/Parallel I/O port, four channel counter timer – SBC 100	£155.00
Z80 CPU Board 4 MHz, jump-on-reset	£116.00
8080 CPU Board, jump-on-reset	£87.00
8K RAM Board, low power 450 n Sec (21L02-1)	£79.00
16K RAM Board, low power 250 n Sec. Static	£195.00
Dynamic RAM Board for 16-64K RAM (4116)	£91.00
2708 EPROM (16K) for 2708 or 2716 EPROMS	£63.75
2708/2716 EPROM Prog. Board, 2 Textool A/T sockets	£86.50
I/O Board, 2/2, DIP switch selection	£95.00
I/O Board, 2/4 provision 4K RAM/4K ROM	£135.00
Video Interface, 16 lines, 32 or 64 ch/line (60 frames)	£91.00
Video Interface, 16 lines, 64 ch/line (50 frames) A/T	£108.75
Video Display, 80 ch x 24 l, keyboard interface, 2K RAM, 256 available chars etc	£192.00
Motherboard (13 slot with four edge connectors)	£48.50
Motherboard (9 slot with four edge connectors)	£40.00
FLOPPY DISCS	
SHUGART SA400 Mini Floppy 5¼"	£190.00
SIEMENS 120 8" Disc Drive	£325.00
Disc Controller, 5" or 8"	£98.00
Double-Density Disc Controller	£280.00
HARDWARE	
EPROM Eraser (240V) : Erases up to 12 EPROMS at a time	£40.00
S100 Edge Connectors, solder tail	£2.45
S100 Edge Connectors, gold plated wire wrap	£3.50
Transformer Pri 110/240V; sec. 8V @ 10 amp & 25V CT @ 2 amp	£12.75
Bridge Rectifier 25 amp @ 50V	£3.75

SOFTWARE	
CP/M on 8" or 5" Floppy Disc	£76.00
C-BASIC 2	£74.00
Z80 Disc Based Assembler, Linker & Editor	£60.00
Microsoft BASIC V.5.	£160.00
XYBASIC – Extended	£220.00
FORTRAN 80	£210.00
COBOL 80	£325.00
PASCAL	£145.00
PASCAL/Z	£131.25
DISZILOG Z80 Disassembler	£37.00
PERIPHERALS	
12" VIDEO MONITOR, green phosphor bonded tube, displays up to 80 ch/24 lines, 50/60 Hz operation	£215.00
PRINTER, Bi-Directional Dot Matrix; 112 ch/sec: 96 ch ASCII set, 80 ch/line; 900 ch buffer; RS232 or parallel input	£595.00
KEYBOAR set, 80 ch/line; 900 ch buffer;; PERI	
PERIPHERALS	
12" VIDEO MONITOR, green phosphor bonded tube, displays up to 80 ch/24 lines, 50/60 Hz operation	£215.00
PRINTER, Bi-Directional Dot Matrix; 112 ch/sec: 96 ch ASCII set, 80 ch/line; 900 ch buffer; RS232 or parallel input	£595.00
KEYBOARD, 56-key Tri-mode ASCII output, with case & edge connector	£68.00
12V DC/DC Convertor	£6.00

WRITE OR PHONE FOR CATALOGUE



YOUNG COMPUTER WORLD

Young Computer World is the place where, each month, John Coll highlights the thoughts, ideas and contributions of PCW's younger readers.

Reactions

I can see that the major problem about this page is going to be finding space to print all the good stuff that comes in. We have given some thought to this problem and perhaps there is a place for publishing a whole selection of programs in book form in addition to the regular page in PCW. Anyway that remains to be seen. Also, of course, we are able to "overflow" into the Programs section.

However, I have only had one reply to my request for an idiot proof input subroutine, but I guess that may be because of the misprints which made the idea a little difficult to follow! I will leave that topic open for a while in the hope that others will try.

Calculator programmes

I've been surprised at the number of calculator programs sent in. S.P Tait (17) is an apprentice with Marconi Communications Systems in Chelmsford and he has submitted five programs for the T157. One uses Kirchoff's and Ohm's laws, one plays pontoon. The other three deal with Matrix Multiplication, Number Base Conversion and a version of Mastermind. One of his programs is printed below.

Number base conversion

```
STO 2          32 2
R/S            81
STO 0          32 0
R/S            81
STO 5          32 5
RCL 2          33 2
LBL 1          86 1
STO 2          32 2
1              01
STO 3          32 3
LBL 6          86 6
RCL 5          33 5
INV PROD 2     -39 2
X              55
RCL 2          33 2
INV INT        -49
INV SUM 2      -34 2
X              55
RCL 3          33 3
=              85
SUM 1          34 1
RCL 0          33 0
PROD 3         39 0
RCL 2          33 2
X=T            -66
GOTO 6         51 6
RCL T          33 1
INV SUM 1      -34 1
R/S            81
RST            71
```

The program converts any integer in any base 1 to 10 to decimal or any decimal integer to any base 1 to 10
To use the program enter the number then R/S. Enter the base of the first number then R/S, then enter the base of the result followed by R/S.

CESIL

Undoubtedly the most interesting letter this month came from Richard Clyne (15) of London SW11. He has written a CESIL interpreter in BASIC. CESIL is a language which makes the computer behave like a very simple machine and illustrates how an assembler works. Space does not permit a detailed explanation of how to work the program but it's fairly obvious. It was not the length of the program that was impressive but rather the fact that it was so clearly set out and easy to use. Richard's program was written to run on the ILEA RSTS Systime 6000 but it will be easy to alter the file handling for other systems. A fine piece of work.

See you at the show and in the meantime keep sending me useful bits and pieces. My address is Laxton House, Oundle, Peterborough PE8 4AQ. Thanks.

Program listing

```
CESIL    13:37              13-SEP-79
10 RANDOMIZE
20 PRINT " MODES AVAILABLE :"
30 PRINT TAB(10),"  (1) INPUTING A PROGRAM"
40 PRINT TAB(10),"  (2) LISTING A PROGRAM"
50 .PRINT TAB(10),"  (3) EDITING A PROGRAM"
60 PRINT TAB(10),"  (4) RUNNING A PROGRAM"
70 PRINT TAB(10),"  (5) RECALLS A SAVED PROGRAM"
80 PRINT TAB(10),"  (6) INDEX OF ALL SAVED PROGRAMS"
90 PRINT TAB(10),"  (7) SAVE A PROGRAM"
100 PRINT TAB(10),"  (8) LIST OF VARIABLES"
110 PRINT TAB(10),"  (9) LIST OF LABELS"
120 PRINT TAB(10),"  (10) DELETING A PROGRAM"
130 DIM C$[3,200],L[200],L$[200],V$[200],V[200],P$[100]
131 DIM D[100]
140 PRINT:INPUT "MODE";M
150 IF M>10 GOTO 20
160 M= INT (M)
170 IF A9>A6 THEN A6=A9
180 IF M>7 GOTO 1940
190 ON M GOTO 200,310,410,530,1530,1610,1710,1950
200 OPEN 'KB:'FOR INPUT AS FILE 9%
210 A9=1
220 INPUT #9,"L<";C$(1,A9)
230 IF C$(1,A9)="END"GOTO 290
240 INPUT #9,"C<";C$(2,A9)
245 IF C$(2,A9)="DATA" GOTO 301
250 INPUT #9,"A<";C$(3,A9)
260 PRINT
270 A9=A9+1
280 GOTO 220
290 CLOSE #9
300 GOTO 140
301  INPUT "HOW MANY DATA ITEMS";D:FOR D1=1 TO D:INPUT #9,D(D1):NEXT D1
302 D2=D
303 GOTO 140
310 !          LISTING PROGRAM (CESIL)
320 PRINT:PRINT
330 PRINT "CARD NUMBER",
340 PRINT "LABEL","COMMAND","LABEL/VARIABLE"
350 A9 =1
360 IF C$(1,A9)="END" GOTO 140
375 IF C$(2,A9)="DATA" GOTO 401
377 PRINT A9,
380 PRINT C$(1,A9),C$(2,A9),C$(3,A9)
390 A9=A9+1
400 GOTO 360
401 PRINT "DATA",:PRINT D(D); FOR D= 1 TO D2
402 GOTO 140
410 ! ************************************************************
420 A9=1
430 IF C$(1,A9)="END" GOTO 440 ELSE GOTO 460
440 INPUT "ADD MORE";Y$
450 IF LEFT(Y$,1)="Y" GOTO 220 ELSE GOTO 140
460 PRINT C$(1,A9),C$(2,A9),C$(3,A9)
470 INPUT "KEEP" ; E$
480 IF E$="E" GOTO 140
490 IF E$="C" GOTO 500 ELSE A9=A9 +1:GOTO430
500 INPUT C$(1,A9),C$(2,A9),C$(3,A9)
510 A9=A9+1
520 GOTO 430
```

```
530 L=1:D=1
540 FOR A8=1 TO A6-1
550 IF C$(1,A8)=""GOTO570
560 L$(L)=C$(1,A8):L(L)=A8:L=L+1
570 NEXT A8
580 L(0)=L
590 V=1
600 FOR A8=1 TO A6-1
610 V9$ =C$(2,A8)
620 IF V9$="STORE" GOTO 650
630 NEXTA8
640 GOTO 680
650 V9$=LEFT(C$(3,A8),1)
660 IF V9$="+" OR V9$="-" GOTO 630
670  V$(V)=C$(3,A8):V=V+1:GOTO 630
680 V(0)=V
690 P=1
700 A=INT(A)
710 X$=C$(2,P)
720 IF X$="IN" GOTO 890
730 IF X$="OUT" GOTO 920
740 IF X$="HALT" GOTO 950
750 IF X$="LOAD" GOTO 980
760 IF X$="JIZERO" GOTO 1090
770 IF X$="JINEG" GOTO 1120
780 IF X$="JUMP" GOTO 1140
790 IF X$="STORE" GOTO 1210
800 IF X$="PRINT" GOTO 1270
810 IF X$="LINE" GOTO 1290
820 IF X$="ADD" GOTO 1310
830 IF X$="SUBTRACT" GOTO 1340
840 IF X$="MULTIPLY" GOTO 1370
850 IF X$="DIVIDE" GOTO 1400
860 IF C$(2,P)="" GOTO140
870 PRINT C$(2,P);" IS NOT A LEGAL COMAND
880 PRINT"EDIT IT OUT!":GOTO140
890 A=D(D):D=D+1
900 P=P+1
910 GOTO 700
920 PRINT A;
930 P=P+1
940 GOTO 700
950 GOTO 130
960 P=P+1
970 GOTO 700
980 J$=C$(3,P)
990 IF LEFT (J$,1)="+" OR LEFT(J$,1)="-" GOTO1060
1000 FOR A9=1 TO V(0)
1010 IF C$(3,P)=V$(A9) GOTO 1040
1020 NEXT A9:P=P+1
1030 GOTO 700!NEVER REACHED
1040 A=V(A9)
1050 P=P+1:GOTO 700
1060 A= VAL(RIGHT(J$,LEN(J$)-1))
1070 IF LEFT(J$,1)="-" THEN A=-(A)
1080 P=P+1:GOTO 700
1090 IF A=0 GOTO 1140
1100 P=P+1
1110 GOTO 700
1120 IF A<0 GOTO 1140
1130 P=P+1:GOTO 700
1140  FOR  A8=1 TO L(0)
1150 IF C$(3,P)=L$(A8) GOTO 1190
1160 NEXT A8
1170 PRINT"LABEL ERROR CARD";P
1180 GOTO 140
1190 P=L(A8)
1200 GOTO 700
1210 FOR A7= 1 TO V(0)
1220 IF V$(A7)=C$(3,P) GOTO 1260
1230 NEXT A7
1240 PRINT "VARIABLE ERROR! CARD";P
1250 GOTO 140
1260  V(A7)=A:P=P+1:GOTO700
1270 PRINT C$(3,P);
1280 P=P+1 :GOTO 700
1290 PRINT
1300 P=P+1:GOTO 700
1310 GOSUB 1430
1320 A=A+T
1330 P=P+1:GOTO 700
1340 GOSUB 1430
1350 A=A-T
1360 P=P+1:GOTO 700
1370 GOSUB 1430
1380 A=A*T
1390 P=P+1:GOTO 700
1400 GOSUB 1430
1410 A=INT(A/T)
1420 P=P+1:GOTO 700
1430 J$=C$(3,P)
1440 IF LEFT (J$,1)="+" OR LEFT (J$,1)="-" GOTO 1500
1450 FOR A7= 1 TO V(0)
1460 IF V$(A7)=J$ GOTO 1490
1470 NEXT A7
1480 PRINT" VARIABLE ERROR!!!(MATH FUNCTION) CARD";P:GOTO140
1490T=V(A7):RETURN
1500 T=VAL(RIGHT(J$,LEN(J$)-1))
1510 IF LEFT (J$,1)="-" THEN T=-(T)
1520 RETURN
1530 INPUT "PROGRAM NAME";P$
1540 OPEN P$ FOR INPUT AS FILE 1%
1550 V=1
1560 INPUT #1,C$(1,V)
1570 IF C$(1,V)="END" GOTO 1590
1580 INPUT#1,C$(2,V):INPUT #1, C$(3,V):V=V+1:GOTO 1560
1590 CLOSE#1
```

PROGRAMS

```
1600 GOTO130
1610 OPEN 'INDEX' FOR INPUT AS FILE 1%
1620 A=INT (RND*6)+7
1630 PRINT TAB(A),"CESIL PROGRAMS"
1640INPUT #1,J
1650 FOR 05 = 1 TO J
1660 INPUT #1,P$
1670 PRINT P$
1680 NEXT 05
1690 CLOSE #1
1700 GOTO 130
1710 INPUT"PROGRAM NAME";P$
1720 OPEN P$ FOR OUTPUT AS FILE 1%
1730 FOR A7=1 TO A6
1740 PRINT #1,C$(1,A7)
1750 PRINT #1,C$(2,A7)
1760 PRINT #1,C$(3,A7)
1770 NEXT A7
1780 PRINT #1,"END"
1790 CLOSE #1
1800 OPEN'INDEX' FOR INPUT AS FILE 1%
1810 INPUT #1,J
1820 FOR A=1 TO J
1830 INPUT #1,P$(A)
1840 NEXT A
1850 CLOSE #1
1860 P$(J+1)=P$
1870 OPEN 'INDEX' FOR OUTPUT AS FILE 1%
1880 PRINT #1,J+1
1890 FOR A=1 TO J+1
1900 PRINT #1,P$(A)
1910 NEXT A
1920 CLOSE #1
1930 GOTO 130
1940 ON M-7 GOTO 1950 ,2070,2110
1950 ! LIST OF VARIABLES
1960 Z=1
1970 FOR X=1 TO V(0)
1980 FOR Y = 1 TO X-1
1990 IF V$(X)=V9$(X) GOTO 2030
2000 NEXT Y
2010 V9$(Z)=V$(X):V9(Z)=V(X)
2020 Z=Z+1
2030 NEXT X
2040 PRINT "VARIABLE","CONTENTS"
2050 PRINT V9$(X),V9(X) FOR X= 1 TO Z-1
2060 GOTO 130
2070 ! LIST OF LABELS
2080 PRINT"CARD","LABEL"
2090 PRINT L(Z),L$(Z)FOR Z=1 TO L(0)-1
2100 GOTO 130
2110    !DELETEING A FILE
2120 INPUT"PROGRAM TO DELETE";P$
2130 OPEN'INDEX'FOR OUTPUT  AS FILE 1%
2140  INPUT #1,J
2150 FOR X=1 TO J
2160 INPUT #1,P$(X)
2170 IF  P$=P$(X) THEN 2210
2180 NEXT X
2190 PRINT"NO SUCH PROGRAM"
2200 GOTO 130
2210 INPUT #1,P$(X)
2220 FOR Y=X+1 TO J
2230 INPUT #1,P$(Y)
2240 NEXT Y
2250 CLOSE #1
2260 KILL P$
2270 OPEN'INDEX'FOR OUTPUT AS FILE1%
2280 PRINT #1,J-1
2290 FOR X=1 TO J-1
2300 PRINT #1,P$(X)
2310 NEXT X
2320 CLOSE #1
2330 GOTO 130
32627 END
```

BELLS & WHISTLES

Recently PCW has received several cassette handling programs and subroutines. Here are two which should prove particularly useful.

READ/WRITE ROUTINES

Thomas Turnbull, PETSOFT consultant presents a method for reading and writing PET data files without error.

This method gives close to 100% reliability. It involves two subroutines to increase the gap between data blocks written to tape, thus allowing the machine to read back all the data without dropping a single block. Remember, if a block that is lost contains an EOT or EOF the computer will crash with hardly any hope of recovery.

My subroutine starts at line 5000 for tape 1 and line 6000 for tape 2. These subroutines need only be used on PRINT files (not READ files). Before opening a print file to CASSETTE 1 have the following POKE commands:

```
10 POKE 244,2:POKE 243,122:open 1,1,1
```

This is the POKE command for CASSETTE 2:

```
20 POKE 244,3:POKE 243, 58:OPEN 3,2,1
```

These POKE commands tell the PET which buffer it is to use and make sure that a proper tape header is written. If this is not done you will be unable to open that file for read operations. These POKE commands need only be put before the open statements and nowhere else in the print file used.

To use these subroutines you must GOSUB 5000 for tape 1 or GOSUB 6000 for tape 2 after every print to the file.

Here is an example:

```
40 PRINT*1,A$:GOSUB5000:REM THIS IS FOR TAPE 1
     TAPE 1
5000 IF PEEK(625)<180 THEN RETURN:REM LOCATION
     625 IS THE TAPE 1 BUFFER COUNTER
5010 POKE59411,53:T=TI:REM POKEING LOCATION
     59411 WITH 53 STARTS TAPE 1 MOTOR RUNNING
5020 IF TI-T<6 THEN 5020:REM THIS SETS TAPE
     RUNNING FOR 1/10TH SECOND INCREASING GAP
5030 POKE59411,61:RETURN:REM THIS POKE COMMAND
     SWITCHES CASSETTE 1 OFF
     TAPE 2
6000 IF PEEK(626)<180 THEN RETURN:REM LOCATION
     626 IS BUFFER FOR TAPE 2
6010 POKE 59456,207:T=TI:REM THIS STARTS CASSETTE
     2 MOTOR
6030 IF TI-T<6 THEN 6030
6040 POKE 59456,223:RETURN
```

All PETSOFT programs that use files have this subroutine included and they are very reliable in use.

The reason that the buffer is made to check the number 180, and not 191 as you would expect, is because this keeps the motor running in small starts until the buffer is finally emptied. Once empty, there is no need for the tape recorder to build up to writing speed as it will already be at the right speed and the data will be written at the correct rate.

GLITCH FREE LOADING

by J. Luxford

This is written for NASCOM 1 users but the principles described may be easily applied to other micros.

Problem:
You have a cassette written on another recorder which (due to incompatible head alignment, speed differences or poor tape quality) will not give error free program loading. You do not have listings in order to make manual corrections and, anyway, even if you did there may be too many. What to do?

Solution:
1. Load the corrector program in a disused location.
2 Clean the tape recorder heads.
3 Load as normal, (keep a note of errors). We will call the memory block just loaded block 1.
4 Copy block 1 to a free memory area. Call this area block 2.
5 Re-run the tape, reloading block 1 to free memory area. Call this block 3.
6 Re-run the tape, reloading block 1
7 Execute the corrector program. If there are any remaining errors the faulty locations will be listed. If none are listed the program is loaded.
8 If errors still exist copy block 1 to block 2, reload block 1. Execute correc-

PROGRAM: Data error corrector for Nascom 1 (Z.80)

CP	Machine Code	Label	Mn	Op1	Op2	Comments
0D00	21000E	START	LD	HL	#0E00	Initialise pointers to start of mem. blocks 1,2& 3.
03	DD210016		LD	1X	#1600	
07	FD21001E		LD	1Y	#1E00	
0B	7E	NEXT	LD	A	(HL)	Get the bytes for comparison
0C	DD5600		LD	D	(1X+d)	
0F	FD5E00		LD	E	(1Y+d)	
12	BA		CP	D		Are blocks 1,2 same ?
13	2812		JR	Z	GOOD	If so, good
15	BB		CP	E		Else compare blocks 1,3
16	280F		JR	Z	GOOD	If 1,3 same good
18	7A		LD	A	D	Else compare blocks 2 & 3
19	BB		CP	E		
1A	77		LD	(HL)	A	Upgrade block 1
1B	280A		JR	Z	GOOD	
1D	E5D5		PUSH	HL	DE	If data bad print bad addr. and scroll
IF	CD3202		CALL	TBCD3		
22	CD4002		CALL	CRLF		
25	D1E1		POP	DE	HL	
27	23	GOOD	INC	HL		Set pointers to next byte
28	DD23		INC	1X		
2A	FD23		INC	1Y		
2C	010016		LD	BC	(#END BLK1+1)	Check to see if finished
2F	B7		OR	A		
30	ED42		SBC	HL	BC	
32	09		ADD	HL	BC	
33	CA8602		JP	Z	PARSE	Exit to monitor
36	1BD3		JR	NEXT		Else get next byte

tor program.
9 If errors are still listed repeat step 8.
Note: If insufficient memory is available to load the whole program in one go, split the program into segments. When each segment is cleaned up, DUMP on to scratch tapes, then assemble the individual good tapes to re-form the complete program.

Example:
1 The corrector is loaded at 0D00 — 0D37. (This may be relocated as only relative jumps are used). Our program to be loaded resides in 0E00 to 15FF so we define block 1 as 0E00 to 15FF, block 2 as 1600 to 1DFF and block 3 to 1E00 to 25FF.
Note: corrector lines 0D00, 0D03 and 0D07 are set to point at the start of these blocks and line 0D2C is a terminator set at [(END OF BLOCK 1) + 1]. Notice Z80 practice of putting Lo order byte first.
2 Load block 1. Copy to block 2.
>CE00 1600 7FF NL
3. Reload block 1. Copy to block 3.
>CE00 1E00 7FF NL
4. Reload block 1
5. Execute corrector program, but because block 1 overlaps page 0 — 1 first modify R.SP. to 0C33 to prevent corruption of block 1. (see PCW March 1979 letters).
>M0C3D NL 33 0C. NL
>E0D00 NL
The monitor will now list any remaining errors. If none, the monitor will return a prompt (>) and the program is loaded.
Final note: This represents a very simple process of choosing any two from three, more sophisticated combinations may be used but it is doubtful if more complex and hence longer programs are justified on this application.

FUN & GAMES

APPLE WORMS

by Ray West, freelance programmer

TAPEWORMS: A KEYBOARD VIDEO GAME FOR THE APPLE

'Tapeworms' is a game for two players which uses the keyboard interactively. Each player has four allocated keys, which are identified by the keyboard PEEK function. Two shape tables are loaded by the program, and these give each 'worm' a different appearance. To improve the appearance of the display, the rotation feature of the shape table is used so that the direction of movement of the worm is matched by the rotation of the shape. A game ends either when a player crosses the rectangular border of the playing area, or collides with a previously plotted shape; the collision counter provides a way of checking for this event. For a detailed explanation of the listing, now read on. . .

Lines 510-640 are the main program control. There are essentially six subroutines which are called.

SUB 20000. This sets up a shape table of two shapes. Line 20000 sets up pointers in locations 232 and 233, the low and high bytes respectively. Since 117*256+48=30000, the Apple expects the shapes to start at 30000. This works for a 48K or 32K machine. Line 20001 tells the machine there are two shapes in the table, and lines 20005 and 20010 give their addresses, offset from 30000. So shape 1 begins at 30000+256*0+159=30159, for example. The two shapes are 'A' and 'V', and were used because they happened to be available. If you don't like them, try adjusting the table!

SUB 10000. This prints the title page onto the screen, enabling one of three playing speeds to be selected. In addition, variables are stored just after the program; line 10010 ensures that the coordinates and directions of each 'worm' are stored where retrieval time is minimised. Random start points and directions are generated for each player; in line 10220 they are checked to avoid starting too close to each other. Directions 1 to 4 are converted into the relevant keystroke equivalent for each player.

SUB 1000 & SUB 1400. This symmetrical pair of routines reads the keyboard. The point of the last statements of lines 1000 and 1400 is that the Apple seems sometimes to admit a low ASCII value. If on A's turn his part of the keyboard registers an input, its ASCII value is saved; and similarly on B's turn. In a fast game, only one peek at the keyboard is allowed.

SUB 2110 & SUB 2510. The x or y coordinate is incremented/decremented as required, and the direction indicator AD or BD set to correspond. Lines 2145 and 2545 test the new plot. If it is an acceptable move, the other person's score is increased by 1 and exit to the end-of-game routine occurs. The POP instruction removes the subroutine's return address from the stack: were this instruction omitted, after about 24 games the stack would fill up and an OUT OF MEMORY message appear. The formulae for ROT need to introduce multiples of 16, for which the values differ for the shapes plotted, so that lines 2147 and 2547 use different calculations. The direction is coded as 1 for north, 2 for east, and so on.

SUB 25000. This is entered only in a slow or medium speed game. It uses simple delay loops, which, however, have diminishing effect as the game proceeds. So the tempo accelerates towards the end.

SUB 26000. This routine displays the aggregate scores to date, the player sitting on the left having his score shown at the left of the screen and vice versa. The set of games can be terminated in order to change speed, or start afresh, by entering 'N'. Since some characters may remain in the buffer, line 26040 checks for the presence of an 'N' within it. If the set of games continues, line 26040 loops back to reset new starting positions and directions, before returning to the program's main control.

PROGRAMS

```
0  REM 'TAPEWORMS' GAME (C) RAY WEST MARCH 79
11  GOSUB 20000: GOSUB 10000
510  GOSUB 1000: REM  SEE IF A WANTS TO MOVE.
520  GOSUB 2110: REM    PLOT A'S NEW POSITION
530  IF  PEEK (234) (   ) 2 THEN SB% = SB% + 1: GOTO 26000: REM END OF A GAME INDICATED BY COLL
     ISIDN COUNTER
540  IF R (  ) 1 THEN  GOSUB 25000
610  GOSUB 1400: REM   SEE IF B WANTS TO MOVE
620  GOSUB 2510: REM    PLOT B'S NEXT SEGMENT
630  IF  PEEK (234) (   ) 0 THEN SA% = SA% + 1: GOTO 26000: REM  END OF A GAME
632  IF R = 1 THEN 510
635  GOSUB 25000
640  GOTO 510
1000 Z =  PEEK ( - 16384): POKE  - 16368,0: IF Z ( CT% THEN Z = Z + CT%
1002  IF Z = WK% OR Z = SK% OR Z = ZK% OR Z = AK% THEN AD% = Z: RETURN
1004  IF R = 1 THEN  RETURN
1006 Z =  PEEK ( - 16384): POKE  - 16368,0: IF Z ( CT% THEN Z = Z + CT%
1008  IF Z = WK% OR Z = SK% OR Z = ZK% OR Z = AK% THEN AD% = Z: RETURN
1020  RETURN
1400 Z =  PEEK ( - 16384): POKE  - 16368,0: IF Z ( CT% THEN Z = Z + CT%
1402  IF Z = OK% OR Z = LK% OR Z = CK% OR Z = KK% THEN BD% = Z: RETURN
1404  IF R = 1 THEN  RETURN
1406 Z =  PEEK ( - 16384): POKE  - 16368,0: IF Z ( CT% THEN Z = Z + CT%
1408  IF Z = OK% OR Z = LK% DR Z = CK% OR Z = KK% THEN BD% = Z: RETURN
1510  RETURN
2110  IF AD% = WK% THEN AY% = AY% - C7%:AD = C1%
2120  IF AD% = SK% THEN AX% = AX% + C7%:AD = C2%
2130  IF AD% = ZK% THEN AY% = AY% + C7%:AD = C3%
2140  IF AD% = AK% THEN AX% = AX% - C7%:AD = C4%
2145  IF AX% ( 2 OR AX% ) 277 OR AY% ( 3 OR AY% ) 152 THEN  POP :SB% = SB% + 1: GOTO 26000
2147  ROT= CS% * (AD - C1%)
2150  DRAW 1 AT AX%,AY%: RETURN
2510  IF BD% = OK% THEN BY% = BY% - C7%:BD = C1%
2520  IF BD% = LK% THEN BX% = BX% + C7%:BD = C2%
2530  IF BD% = CK% THEN BY% = BY% + C7%:BD = C3%
2540  IF BD% = KK% THEN BX% = BX% - C7%:BD = C4%
2545  IF BX% ( 2 OR BX% ) 277 OR BY% ( 3 OR BY% ) 152 THEN  POP :SA% = SA% + 1: GOTO 26000
2547  ROT= CS% * (C1% + BD)
2550  DRAW 2 AT BX%,BY%: RETURN
10000  TEXT : HOME : FLASH : PRINT '****          TAPEWORMS          ****': NORMAL
10010 AX% = 0:AY% = 0:BX% = 0:BY% = 0:AD% = 0:BD% = 0
10015 WK% = 215:SK% = 211:ZK% = 218:AK% = 193:OK% = 207:LK% = 204:CK% = 172: REM
10016 KK% = 203: REM  K=KEY. STORED AT START OF VARIABLE SPACE
10017 C1% = 1:C2% = 2:C3% = 3:C4% = 4:C7% = 7:CS% = 16:CT% = 128
10030  PRINT : PRINT : PRINT 'A GAME FOR 2 PLAYERS': PRINT : PRINT 'PREFERABLY ONE LEFTHANDER!
    ': PRINT : PRINT : PRINT 'FIRST PLAYER'S WORM CONTROLLED': PRINT 'WITH KEYS W,S,Z,A.'
10040  PRINT : PRINT 'SECOND PLAYER USES O,L,COMMA,K.'
10045  PRINT : PRINT : PRINT 'A RAY WEST SUPERIOR': PRINT 'PROGRAM PRODUCT': PRINT
10046  INPUT 'SLOW, MEDIUM, OR FAST GAME?';K$
10047  IF  LEFT$ (K$,1) = 'F' THEN R = 1:S = 1:T = 1: GOTO 10050
10048  IF  LEFT$ (K$,1) = 'M' THEN R = 100:S = 50: GDTO I0050
10049 R = 200:S = 150:T = 150
10050  PRINT : PRINT 'HIT RETURN TO PLAY': GET K$
10200 AX% = 7 *  INT (( RND (1) * 34 + 3)):AY% = 7 *  INT (( RND (1) * 17 + 3)):AD =  INT (1 +
     4 *  RND (1))
10201  IF AD = 1 THEN AD% = 215
10202  IF AD = 2 THEN AD% = 211
10203  IF AD = 3 THEN AD% = 218
10204  IF AD = 4 THEN AD% = 193
10210 BX% = 7 *  INT (( RND (1) * 34 + 3)):BY% = 7 *  INT (( RND (1) * 17 + 3)):BD =  INT (1 +
     4 *  RND (1))
10211  IF BD = 1 THEN BD% = 207
10212  IF BD = 2 THEN BD% = 204
10213  IF BD = 3 THEN BD% = 172
10214  IF BD = 4 THEN BD% = 203
10220  IF  ABS (AX% - BX%) ( 20 AND  ABS (AY% - BY%) ( 20 THEN 10200
10300  HCOLOR= 3: SCALE= 1: HGR
10305 SP = 0
10310  HPLOT 3,3 TO 277,3 TO 277,152 TO 3,152 TO 3,3
10320  RETURN
20000  POKE 232,48: POKE 233,117
20001  POKE 30000,2: POKE 30001,0
20005  POKE 30002,159: POKE 30003,0
20010  POKE 30004,21: POKE 30005,3: REM  V
20015  POKE 30159,146: POKE 30160,27
20020  POKE 30161,4: POKE 30162,36
20025  POKE 30163,36: POKE 30164,33
20030  POKE 30165,49: POKE 30166,49
20035  POKE 30167,49: POKE 30168,54
20040  POKE 30169,6: POKE 30170,36
20045  POKE 30171,63: POKE 30172,63
20047  POKE 30173,0
20050  POKE 30789,146: POKE 30790,36
20055  POKE 30791,39: POKE 30792,60
20060  POKE 30793,36: POKE 30794,140
20065  POKE 30795,73: POKE 30796,54
20070  POKE 30797,55: POKE 30798,62
20075  POKE 30799,0
20080  RETURN
25000 SP = SP + 1: IF SP ) 80 THEN 25040
25010  IF SP ) 40 THEN 25030
25020  FOR I = 1 TO R: NEXT
25030  FOR I = 1 TO S: NEXT
25040  FOR I = 1 TO T: NEXT
25050  RETURN
26000  HOME : FOR I = 1 TO 7: PRINT  CHR$ (7);: NEXT : VTAB 23: PRINT  SPC( 11);: INVERSE : PRINT
     '****  SCORE  ****': NORMAL : PRINT SA%; TAB( 41 -  LEN ( STR$ (SB%)));SB%
26010  INPUT 'HIT RETURN FOR NEXT GAME OR N TO EXIT';K$
26020  VTAB 22: CALL  - 956
26030  IF K$ = '' THEN  GOSUB 10200: GOTO 510
26040 L =  LEN (K$): FOR I = 1 TO L: IF  MID$ (K$,L,1) (   ) 'N' THEN  GOSUB 10200: GOTO 510
26050  NEXT I
26060  END
```

Here, by popular demand, is the continuation of David Parkinson's Revas. We apologise for the delay — it disappeared during the recent move.

```
FC24  CD 16 FC       0680          CALL  ST1         ;"A"
FC27  CD 31 F9       0681          CALL  ¬(*:¬¬¬¬¬¬¬H'¬¬¬¬1¬(¬¬¬¬¬1            0682
FC2B  FE 22          0683 LD1:     CP    $22         ;REGPR OR EXTENDED?
                                                                FC2D  EB
FC2E  36 28          0685          LD    (HL),'('
FC30  23             0686          INC   HL
FC31  EB             0687          EX    DE,HL
FC32  3F             0688          CCF               ;REVERSE RESULT OF COMPARE
FC33  D4 7E F9       0689          CALL  NC,REGPR    ;(RETURNS WITH NC)
FC36  DC 57 F9       0690          CALL  C,LD16A     ;EXTENDED ADDRESS
FC39  C3 04 FA       0691          JP    NOTIXY      ;CLOSE BRACKETS
FC3C                 0692 ;
FC3C                 0693 ; 16-BIT INDIRECT STORE  LD (NNNN),PP
FC3C                 0694 ;
FC3C  CD 2B FC       0695 ST16I:   CALL  LD1         ;DO INDIRECT BIT
FC3F  CD 11 F9       0696          CALL  COMMA       ;","
FC42  C3 7D F9       0697          JP    REGPR-1     ;GET OP & PR. REG PAIR
FC45                 0698 ;
FC45                 0699 ; 16-BIT INDIRECT LOAD  LD PP,(NNNN)
FC45                 0700 ;
FC45  F1             0701 LD16I:   POP   AF          ;GET OPCODE
FC46  CD 7E F9       0702          CALL  REGPR       ;PRINT REG PR.
```

PROGRAMS

```
FC49  CD 11 F9          CALL  COMMA     ;","
FC4C  A7                AND   A         ;CLEAR CARRY
FC4D  18 DE             JR    LD1+2     ;DO INDIRECT BIT
FC4F                ;
FC4F                ; ROTATE/SHIFT/BIT/SET/RESET
FC4F                ;  NB. IF INDEXED THEN OFFSET PRECEDES OPCODE
FC4F                ;
FC4F  3A 0B 10      CB:     LD    A,(HXYFLG)
FC52  A7                    AND   A         ;INDEXED?
FC53  F5                    PUSH  AF        ;SAVE FLAG
FC54  28 0B                 JR    Z,NOTXY   ;NO,SKIP
FC56  11 34 10              LD    DE,BUFFER+34;YES,WRITE..
FC59  3E 06                 LD    A,6       ;...REG. FIRST.
FC5B  CD C4 F9              CALL  SREG
FC5E  11 2D 10              LD    DE,BUFFER+27;RESET POINTER FOR MNEMONIC.
FC61  CD DB F8      NOTXY:  CALL  BYTE      ;GET OPCODE
FC64  F5                    PUSH  AF        ;SAVE IT
FC65  FE 40                 CP    $40       ;<$40?
FC67  38 27                 JR    C,ROTATE  ;YES,JUMP
FC69  21 A8 FC              LD    HL,BRSTAB-3;LOAD POINTER
FC6C  07                    RLCA            ;SHIFT OPCODE DOWN
FC6D  07                    RLCA
FC6E  E6 03                 AND   3         ;ISOLATE ID
FC70  47                    LD    B,A       ;MAKE  3,6,OR 9
FC71  07                    RLCA
FC72  80                    ADD   B
FC73  CD 2A F9              CALL  FTADR     ;FORM ADDRESS
FC76  CD 23 F9              CALL  COPY3     ;WRITE MNEMONIC
FC79  13                    INC   DE        ;SPACE
FC7A  13                    INC   DE
FC7B  F1                    POP   AF        ;GET OPCODE
FC7C  F5                    PUSH  AF        ;SAVE AGAIN
FC7D  0F                    RRCA            ;PRINT BIT NUMBER
FC7E  0F                    RRCA
FC7F  0F                    RRCA
FC80  E6 07                 AND   7         ;ISOLATE BIT ID
FC82  F6 30                 OR    $30       ;MAKE ASCII
FC84  12                    LD    (DE),A    ;WRITE IT
FC85  13                    INC   DE
FC86  CD 11 F9              CALL  COMMA     ;","
FC89  C1            TESTXY: POP   BC        ;RECOVER OPCODE
FC8A  F1                    POP   AF        ;RECOVER HXY FLAG
FC8B  78                    LD    A,B       ;LOAD OPCODE
FC8C  C0                    RET   NZ        ;YES,RETURN
FC8D  C3 C4 F9              JP    SREG      ;NO,GO WRITE.
FC90  0F            ROTATE: RRCA            ;SHIFT DOWN
FC91  0F                    RRCA
FC92  C6 02                 ADD   2         ;ROLL CODING ROUND
FC94  E6 0E                 AND   $E        ;ISOLATE ID
FC96  FE 0E                 CP    $E        ;IS IT 0E?
FC98  CA D6 FD              JP    Z,NTVL    ;YES,INVALID CODE
FC9B  47                    LD    B,A       ;DO ID*3 AGAIN
FC9C  0F                    RRCA
FC9D  80                    ADD   B
FC9E  21 B4 FC              LD    HL,ROTTAB ;LOAD BASE ADDRESS
FCA1  CD 2A F9              CALL  FTADR     ;FORM ADDRESS
FCA4  CD 23 F9              CALL  COPY3     ;WRITE MNEMONIC
FCA7  13                    INC   DE
FCA8  13                    INC   DE
FCA9  18 DE                 JR    TESTXY    ;EXIT WRITING REGISTER
FCAB                ;
FCAB  42 49 54 52   BRSTAB: DB    'BITRESSET'
      45 53 53 45
      54
FCB4  53 52 4C 52   ROTTAB: DB    'SRLRLCRRCRL RR SLASRA'
      4C 43 52 52
      43 52 4C 20
      52 52 20 53
      4C 41 53 52
      41
FCC9                ;
FCC9                ; AUTO CP LD IN OUT
FCC9                ;
FCC9  CB 57         AUTO:   BIT   2,A       ;TEST FOR VALIDITY
FCCB  C2 D8 FD              JP    NZ,NOTVAL ;JUMP IF NOT
FCCE  F5                    PUSH  AF        ;SAVE OPCODE
FCCF  E6 03                 AND   3         ;ISOLATE OP ID
FCD1  07                    RLCA            ;*2
FCD2  21 E9 FC              LD    HL,OPTAB  ;LOAD BASE ADDRESS
FCD5  CD 2A F9              CALL  FTADR     ;FORM ADDRESS
FCD8  CD 25 F9              CALL  COPY2     ;WRITE PART OF MNEMONIC
FCDB  F1                    POP   AF        ;RECOVER OPCODE
FCDC  21 F1 FC              LD    HL,OPTAB+8;LOAD BASE ADDRESS
```

To be continued

BLUDNERS

Basic Problem
You all spotted the $s coming out as Ss in Bench Test and ESP didn't you? If not, why not!

Puzzle
We've decided that Pythagoras was right after all — the area of a right-angled triangle is (once again) ½B x H.

Spaceship
We think that we've had phone calls from every Fx 201-P owner! Just in case we haven't, ÷ came out as – in the following steps: 40,59,74, and 86 (the second one). Step 98 reads 4 = 7 ÷ K2 + 9 + 4:.

PUZZLES

LEISURE LINES

With J. J. Clessa

Most of you spotted the deliberate (?) mistake in our first Leisure Lines – gulp!

Puzzle 1A involves some logical reasoning, and should not have proved much of an obstacle to our readers. The solution is that the pilot's name is SMITH.

Puzzle 1B was a bit tougher, and although it can be solved analytically, by anyone who's familiar with Diophantine analysis, it's a much simpler task to write a small program for desk calculator or microcomputer.

Since we made the error in defining the area of a triangle we decided we would accept either of the two possible solutions — many entries included both anyway.

Using our formula for triangle area (area - axb), the smallest solution possible is a triangle with sides 36,48 and 60 units, which has a perimeter of 12^2 and an area (?) of 12^3.

However, using the correct formula for area (axb/2), the smallest solution is a triangle with sides 144,192,240, with a perimeter of 12^2 and an area of 24^3.

Since there was no outright winner, we made a draw and the two lucky readers are: Puzzle 1A: D. E. Arnett of Grimsby. Puzzle 1B: Paul Durrant of Norwich.

Congratulations to both and stand by for a shower of chocolate bars (not to mention the subsequent visit to the dentist).

Just one puzzle for this month, but it's really a rather interesting one. Three friends, Alan, Bert and Colin each possess vehicles. Alan owns a big foreign car, Bert a small English car and Colin, a moped.

One day while discussing mileages, Alan reports that his mileometer, which gives 6-figure mileage readings, is currently showing a palindromic reading of 006600 miles (for those that know not, a palindromic number is one which reads the same from right to left as it does from left to right).

"What a coincidence", explains Bert, "So is mine. The 5-figure reading at the moment is 18981 miles".

"Well I never", says Colin, "although the mileometer on my moped only shows 4-figures, it's reading 5335 miles, which is also palindromic".

"I wonder if we're ever likely to get such a coincidence again," says Alan.

Well, of course, since each vehicle does a different weekly mileage from the others, there's no way that the question could be answered. But, supposing all three mileometers were connected to just one vehicle, and also supposing that they were equally accurate, then what is the least number of miles that would elapse before a) Alan's and Bert's mileometers are both showing palindromic readings again? b) Alan's and Colin's mileometers are both showing palindromic readings again? c) Bert's and Colin's mileometers are both showing palindromic readings again? and d) all three mileometers are mutually palindromic?

Answers please on a postcard to Puzzle No. 3, Personal Computer World, 14 Rathbone Place. London W1P 1DE. Entries must reach our offices by November 30th.

PRIZES FOR THIS MONTH

This month's prize is really cunning. In order to make sure the winner continues to send in entries to Leisure Lines, we intend presenting him/her with a hundred 10p stamps.

Happy Birthday PETDISK!

The U.K.-designed and manufactured Novapak disk system for Commodore's PET*, first seen at Compec '78, has now entered its 2nd year of production. It's unique saddle configuration continues the integrated design concept of your PET, with no trailing wires or bulky desk-top modules.

* Novapac may be used with any available RAM plane.
* May be used with both types of PET
* Data transfer takes place at 15,000 char/sec – effectively 1,000 times faster than cassette!
* Storage capacity is 125 K/bytes (unformatted) on 40 tracks per diskette side.
* Dual index sensors permit dual-side recording for 250 K/bytes per diskette.
* Easy operation full-width doors prevent media damage.
* System expandable to ½ Mbyte on-line storage (4 drives).
* Industry Standard IBM 3740 recording format for industry-wide media compatibility offered only by NOVAPAK.
* Dedicated Intel 8048 microprocessor and 1771 FDC minimise PET software overhead.
* Nationwide maintenance available.

Novapac Dual – disk system complete with PDOS and Utilities on disc £899 + VAT
Single-disk (stand alone) £499 + VAT
Pack of 10 mini-diskettes £30
Full cash with order is subject to 5% discount.

Clear documentation and a range of Demonstration programs assist the first-time user, while for the experienced programmer full utilities are supplied to aid concise program development. The sophisticated, easy to use Disk Operating System supports multiple file handling and incorporates extensive error-recovery software for maximum data integrity. Optional password security for any file or any disk is provided.

Several standard software packages are available, and we will tailor any of these to match your specific requirement. Call now for details or a custom software quotation.

analog electronics

47 Ridgeway Ave,
Coventry
Tel: (0203) 417761

APPLE II comes to Glasgow

Why not call and see the fantastic Apple II the finest micro currently available.
Demonstration without obligation.

We also have in stock the following top quality items.

High Speed Tape Cassette Interface. Comes complete with instructions showing how to interface to Nascom giving 'normal' and high speed operation. 300, 600, 1200, 2400 baud. At the highest speed this will load our 8K basic in about half a minute.
Price (Kit) £17.50 + VAT

Supercolour – at last top quality colour for your Apple II. Converts apply signals to separate red, green, blue, and sync. signals. No modulator required. 14" colour monitors available. Send for details now.

Also newly arrived – Buy brand new games for Nascom, Snarkhunt, Chase + Trail, Bulls + Cows, Life, Four in a Row. Runs on minimum Nascom – Any monitor. £10.00 inc. VAT p+p.

Printers – High speed. 112 Char/sec. Top quality. User plain paper tractor fed £575 + VAT.

PASCAL – Now available, the most powerful micro language PET. Features too numerous to fit this ad. Full details by return post.

Monitors – Hitachi, top quality B/W monitors now available. Plug straight into Apple. No modulator required.

Brand New Product. Chiptester. Converts Nascom to a super powerful I.C. tester. Plugs in to existing ports. Send now for full details. £19.50 Inc. VAT p+p

Callers welcome **STRATHAND** Callers welcome

44 ST. ANDREW'S SQ.
GLASGOW G1 5PL
041-552 6731

Tel. order welcome with Access and Barclaycard
Now on Telex 777268. 24 Hours Service

DISKS : LTT : MEMORY : DISKS : LTT : MEMORY : DISKS : LTT : MEMORY

GODBOUT Computer Products

High quality, fast (4 MHz), reliable static S-100 memory boards and other products e.g. –

	Kit	Ass.
2708 EPROM chip	£6.25	n/a
2716 EPROM chip	p.o.a.	n/a
Econorom 2708, 16K EPROM (No EPROMS)	£45	£65
Econorom 2716, 16K EPROM (No EPROMS) 8K RAM (No RAMs)	£55	£75
Econoram IIa, 8K	£80	£99
Interfacer 2 full RS232 serial I/O	£99	£125
Econoram IVa, 16K	£150	£169
Econoram VIIa, 24K	£230	£249
Econoram XIIIa, 32K	£315	£339

EXTRA LOW PRICES ON QUALITY DISKS

Brand name diskettes stocked for most micros.
Pack of ten 5¼" disks, £19 Ten packs (100 disks), £175
Pack of ten 8" disks, £23 Ten packs (100 disks), £210

When ordering please specify whether you require hard or soft sectored diskettes, and if hard sectored, the number of sectors.

Anadax DP8000 Printer. Ready to go! Includes RS232 cable, 1,000 sheets continuous stationery and Securicor delivery within UK. Only £525.

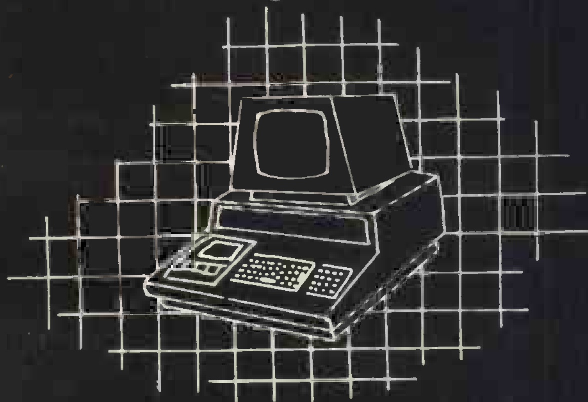
All prices given include postage and packing (overseas add £10) Just add VAT. Send 10p stamp for details.
Quantity discounts available on application. Credit terms (nett 30 days) given to large companies and government establishments.

Mail Order 'phone: 01-828 1785

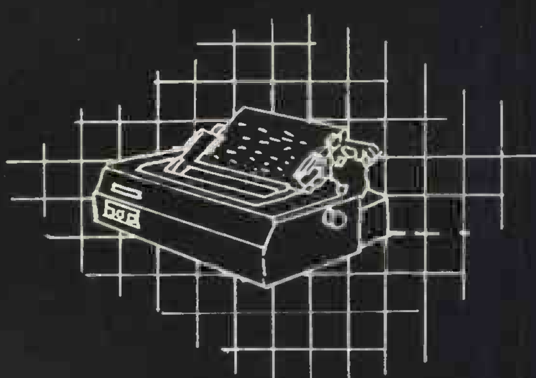
DISKS : LTT : MEMORY : DISKS : LTT : MEMORY : DISKS : LTT : MEMORY

Logic box

. . .Disks . . . calculators . . .
paper . . . printers . . . magazines . . .
8K,16K,32K PET's . . . Petsoft . . .
Petact . . . Compelec Series 1 . . .
and accessories like our . . .



. . .mains interference suppressor.
This plug in suppressor helps stop interference from the mains which can be a threat to data in small business systems; no wiring and handles 1500 Watts at 6 amps; reduces interference between 150 kHz and 100 mHz



If you're after a Hewlett Packard calculator, a sales accounting package, a Teletype 43 printer, a book on Basic or just a tape of 'Microchess' . . . drop in at 31 Palmer Street (by Caxton Hall) Westminster, London SW1. We are open Mon-Fri. and are right by St. James's Park tube near the Army and Navy Store.

31 Palmer St (by Caxton Hall) London SW1
(01) 222 1122
also at:
Planer Bldng. Windmill Rd., Sunbury Middx (09327) 86262

Katanna
Management Services Ltd

kms & COMPUT-A-CROP

WE ARE ON STAND 54
of the PCW SHOW
(1st - 3rd November)

0245-76127 01-677 3022

Come and discuss your requirements for Business Systems and Software:
Printers, Peripherals, and packages or individually written programs available:
For further information phone either of the above numbers or write to:-
22 Roughtons
Galleywood
Chelmsford
Essex

'Tandy' Equipment from S. J. Trott Ltd.,
(Franchise Dealer) — 01-969 6608

Printers... Printers... Printers...

- 16,20,32,40,64,80 and 132 characters per line.
- Impact, electrosensitive and thermal.
- O.E.M. mechanism/assemblies or complete packaged units.
- From only £149 for a printer mechanism with electronics.
- See them on stand A13 at the P.C.W. show.

Datac LIMITED
"The Printer People"
Tudor Rd, Altrincham, Cheshire, WA14 5TN
Tel: 061-941 2361/2

ADVERTISERS INDEX

TRS80 EXPANSION INTERFACE

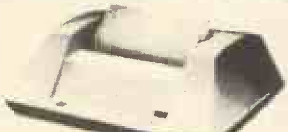
Upgrade your system as your needs increase. Contains sockets for additional 16K or 32K RAM and a disk controller for up to 4-mini-disks. Software selectable dual cassettes can be used. Features a Centronics parallel port, real time clock, and a connector for an RS-232C interface or whatever. Requires Level-II Basic. 220/240V AC.

Complete with 32 K RAM £299 plus VAT.

ANADEX DP8000

ONLY **£540** + VAT

PET Connector — £49



The DP 8000 prints the 96-character ASCII set in single or double width at 84 lines per minute. ● The unit operates bi-directionally to print a 9 x 7 matrix on multiple copy, pin-feed plain paper. ● This model accepts RS-232C or current loop serial data at baud rates switchable from 110 to 9600 and Parallel Bit data input at over 1000 characters per second. ● Standard storage capacity of 256 characters ● Other features include Out of Paper Detector, Top of Form Programming and Skip Over Perforation Control.

THE *NEW* ITT APPLE (2020)

★ Full colour — UHF output ★ Audio cassette tape interface ★ Up to 48K RAM on board ★ BASIC in ROM (graphics commands include COLOUR = VLIN, HLIN, PLOT and SCRN) ★ Built in loudspeaker ★ Buckets of software available ★ Disk System (110K byte per drive — includes controller) only £425 + VAT EX-STOCK

HITACHI PROFESSIONAL MONITORS

9" — £129 + VAT
12" — £199

● **Reliability** Solid state circuitry using an IC and silicon transistors ensures high reliability.
● **500 lines horizontal resolution** Horizontal resolution in excess of 500 lines is achieved at picture center.
● **Stable picture** Even played back pictures of VTR can be displayed without jittering.
● **Looping video input** Video input can be looped through with built-in termination switch.
● **External sync operation** (available as option for U and C types)
● **Compact construction** Two monitors are mountable side by side in a standard 19-inch rack.

video 100

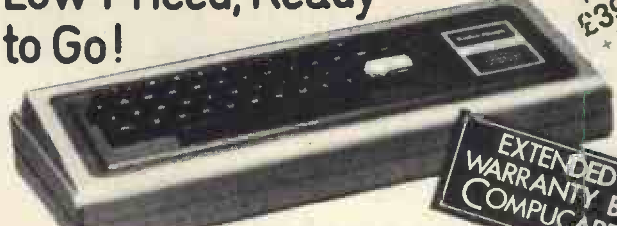
12" BLACK & WHITE LOW COST VIDEO MONITOR

- Ideal for home, personal and business computer systems
- 12" diagonal video monitor
- Composite video input
- Compatible with many computer systems
- Solid-state circuitry for a stable & sharp picture
- Video bandwidth - 12MHz + 3DB
- Input Impedance - 75 Ohms
- Resolution - 650 lines Minimum in Central 80% of CRT; 550 Lines Minimum beyond central 80%.

Only **£79** + VAT

THE TRS80 (Special Scoop) Low Priced, Ready to Go!

16K £399 + VAT



PLUGS INTO YOUR OWN TV
Use your own cassette

LEVEL II BASIC WITH 16K USER RAM provides you with possibly the most powerful micro around. All our TRS80s are fully converted to English Television Standard and include a U.K. Power Supply, Cassette Leads, Sample Tape, Level I & Level II programming manuals, and special lead that enables you to connect direct into your own television.

Special features of Level II Basic enable you to:-
— Set or reset any point on the screen — Test for the presence of a point on the screen (these features enable easy animation) — Save or load data from cassette under program control — File handling capabilities on cassette using named files. — Graphics blocks as standard — design your own pictures and many many more features for only £399 + VAT

PET COSTS LESS AT COMP and it's a pedigree

EXTENDED WARRANTY BY COMPUCARE

8K — Comes complete with integral cassette deck. Full manuals supplied. Powerful 8K Microsoft Basic in ROM. Masses of software available — £499 + VAT

16K — Same as above but with new improved keyboard and cassette supplied as extra. Machine code monitor on board so you can program in 6502 machine code — £590 + VAT

32K — for a little extra get 32K memory providing greater storage capacity for programs or data — £690 + VAT

External Cassette deck for 8K, 16K or 32K — £55 + VAT complete with cable and connector.

SORCERER SPEAKS YOUR LANGUAGE

EXTENDED WARRANTY BY COMPUCARE

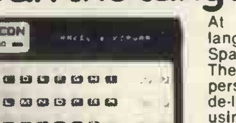
For personal or business use. The best value for money around. ★ 512 by 256 point screen resolution ★ 16K or 32K User RAM ★ Centronics Parallel Port ★ RS232C Serial Port ★ Composite IV peak to peak video output T.V. output supplied as extra. ★ 64 programmable graphics + 64 standard PET graphics ★ 79 key keyboard including 16 key numeric keypad. ★ Expansion bus for connection to S100 Expansion Box.

16K Sorcerer — £690.00 + VAT
32K Sorcerer — £790.00 + VAT
S100 Expansion Box — £210 + VAT

Word Processing Pac and Development Pac now available.

Word Processing Pac — £70 **Development Pac — £70**

Break the language barrier ONLY £138 + VAT



At a price equivalent to learning one language, LEXICON offers you, English, Spanish, French, German, Italian and Greek. The LK3000 comes to you with the person to person module which contains 6 languages, de-luxe carrying case and a charger adaptor using its own power source which will give you 4 - 5 hours continuous use, and can easily be re-charged from the mains supply, wherever you may be in the world. Every additional module carries a concise and understandable instruction book. Your deluxe carrying case has room for two additional modules.

COMPUCARE
(Part of the Compshop Ltd. Group)

Compucare is a company that has been set up to provide servicing and maintenance for the popular makes of micro-computers i.e. Sorcerer, Pet, Apple, TRS80, Nascom, Compukit.
Our charges are £7 per hour plus parts.
Because of the extensive range of spare parts stocked you can usually expect your micro to be repaired within 10 days for an average charge of £14 labour.
Emergency 24 hour repairs can be handled for a £10 surcharge where possible.
Compukits and Nascoms unsuccessfully constructed will be charged a standard £25.
Maintenance contracts for these machines are available, see your local dealer for a Compucare Maintenance Application Form.

Please add VAT to all prices — Delivery at cost, will be advised at time of purchase. Please make cheques and postal orders payable to COMP, or phone your order quoting BARCLAYCARD, ACCESS, DINERS CLUB or AMERICAN EXPRESS number. **OPEN — 10am to 7pm — Monday to Saturday** CREDIT FACILITIES ARRANGED

JUST COMPARE OUR CASH AND CARRY PRICES!

SEND LARGE S.A.E. FOR OUR FREE '79 CATALOGUE

AS SEEN IN
P.E. AUGUST, SEPTEMBER
OCTOBER 1979

EUROPES FASTEST SELLING ONE BOARD COMPUTER — JUST CHECK THE SPEC'S.

COMPUKIT UK101

SAMPLE TAPE
WITH EXTENDED
MACHINE CODE MONITOR
AND DISSASSEMBLER
INCLUDED FREE

LOW COST SUPERBOARD IN KIT FORM

The Compukit UK101 has everything a one board 'superboard' should have.

- ★ Uses ultra-powerful 6502 microprocessor.
- ★ 50Hz Frame refresh for steady clear picture (U.S.A. products with 60Hz frame refresh always results in jittery displays)
- ★ 48 chars by 16 lines — 1K memory mapped video system providing high speed access to screen display enabling animated games and graphs.
- ★ Extensive 256 character set which includes full upper and lower case alphanumerics. Greek symbols for mathematical constants and numerous graphic characters enabling you to form almost any shape you desire anywhere on the screen.
- ★ Video output and UHF Highgrade modulator (8Mz Bandwidth) which connects direct to the aerial socket of your T.V. Channel 36 UHF.
- ★ Fully stabilised 5V power supply including transformer on board.
- ★ Standard KANSAS city tape interface providing high reliability program storage — use on any standard domestic tape or cassette recorder.
- ★ 4K user RAM expandable to 8K on board £49 extra.
- ★ 40 line expansion interface socket on board for attachment of extender card containing 24K RAM and disk controller. (Ohio Scientific compatible).
- ★ 6502 machine code accessible through powerful 2K machine code monitor on board.
- ★ High quality thru plated P.C.B. with all I.C.'s mounted on sockets.
- ★ Professional 52 Key keyboard in 3 colours — software polled meaning that all debouncing and key decoding done in software.

COMMANDS
CONT LIST NEW NULL RUN

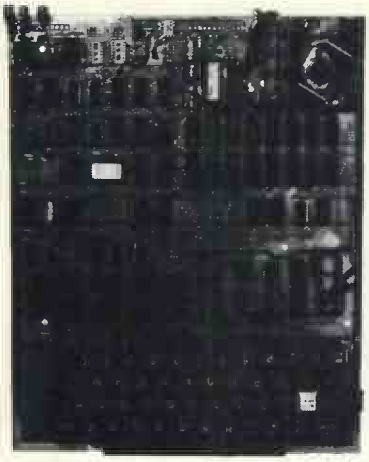
STATEMENTS

CLEAR DATA	DEF	DIM	END	FOR
GOTO GOSUB	IF..GOTO	IF..THEN	INPUT	LET
NEXT ON..GOTO	ON..GOSUB	POKE	PRINT	READ
REM RESTORE	RETURN	STOP		

EXPRESSIONS

OPERATORS
+,-,*,/,↑ NOT.AND.OR. >,<,<>,>=,<= RANGE 10^{-32} to 10^{+32}

VARIABLES
A.B.C. ....Z and two letter variables
The above can all be subscripted when used in an array. String variables use above names plus $.e.g.A$



*8K Microsoft Basic means conversion to and from Pet, Apple and Sorcerer easy. Many compatible programs already in print.

SPECIAL CHARACTERS
@ Erases line being typed, then provides carriage return, line feed.
← Erases last character typed.
CR Carriage Return — must be at the end of each line.
: Separates statements on a line.
CONTROL/C Execution or printing of a list is interrupted at the end of a line. "BREAK IN LINE XXXX" is printed, indicating line number of next statement to be executed or printed.
CONTROL/O No outputs occur until return made to command mode. If an Input statement is encountered, either another CONTROL/O is typed, or an error occurs.
? Equivalent to PRINT

Simple Soldering due to clear and consise instructions compiled by **Dr. A.A. Berk, BSc.PhD**

NO EXTRAS NEEDED JUST HIT 'RETURN' AND GO.

Build, understand, and program your own computer for only a small outlay.

KIT ONLY **£219** + VAT
including RF Modulator & Power supply.
Absolutely no extras.

Available ready assembled and tested, ready to go for
£269 + VAT

FUNCTIONS
ABS(X) ATN(X) COS(X) EXP(X)
LOG(X) PEEK(I) POS(I) RND(X)
SPC(I) SQR(X) TAB(I) TAN(X)
FRE(X) INT(X)
SGN(X) SIN(X)
USR(I)

STRING FUNCTIONS
ASC(X$) CHR$S(I) FRE(X$) LEFT$(X$.I)
RIGHT$(X$.I) STR$(X)
LEN(X$) MID$(X$.I.J)
VAL(X$)

EXTRAS AVAILABLE SOON

COLOUR ADD-ON enables you to choose your foreground and background colour anywhere on the screen. Flash any character on the screen at will. Full documentation and parts in kit form.

AD-A-RAM EXTENDER CARD provides up to 32K Dynamic RAM Expansion, 8 Eprom sockets for 2708's or 2716's. Parallel Port (centronics compatible) and an RS232C serial port.

WIN YOURSELF AN ANADEX DP8000 LINE PRINTER

There's never enough good software around. That's why **COMPUKIT LTD.** are sponsoring a software contest. There are 2 categories:
1) **Business and Education**
2) **Fun and Games**
One lineprinter will be awarded to the winner of each category.
Send or bring along to the address shown below the following:
1) The program on cassette in the format used by the **COMPUKIT UK101**
2) Any documentation that you have for the program (source listing not necessary)
3) This coupon signed by you accepting the rules and conditions of the competition.

RULES:
1) Entries, including documentation, must be printed by computer or typed double spaced, with your name on every page.
2) Send or bring your entries to the address shown below.
3) Entries must be received by midnight on 29/2/80, any received after this time are void.
Winners will be notified by post before 31/3/80.
4) You warrant by your signature that all programs and documentation material included is entirely your own creation, and that no rights to it have been given or sold to any other party, and you agree to allow **COMPUKIT LTD.** to use, publish, distribute, modify, and edit it as it sees fit.
5) All entries become the property of **COMPUKIT LTD.** No entries will be returned nor any questions answered regarding individual entries.
6) Judging will be by a selected panel chosen by, and including representatives of COMPUKIT LTD. Judges may assign programs to any of the categories as they see fit. Decision of the judges is final.
7) Employees of **COMPUKIT LTD**, its dealers, distributors, advertising agencies and media are not eligible to enter.

Name ______________________________

Address ______________________________

____

____

I agree to abide by the above mentioned rules.

Signature ______________________________

COMP COMPUTER COMPONENTS

14 STATION ROAD, NEW BARNET, HERTFORDSHIRE TEL: 01-441 2922 (Sales)
CLOSE TO NEW BARNET BR STATION — MOORGATE LINE 01-449 6596
OPEN — 10am to 7pm — Monday to Saturday TELEX: 298755

All Products Ex-Stock Please check availability

(Part of the Compshop Ltd. Group).

"My best Apple programs are on long-term deposit in the City... it pays rather well!"

We brought the first five Apples into the U.K. in November '77, with every penny we had. In November '79, we find several thousand throughout the country.
THANK YOU Apple owners.
Now we'd like to help you re-coup your investment by cataloguing and supporting the best Apple programs in the U.K. The Apple Software Bank is more like an old penny bank than a major clearing bank, but we know you'll help it grow. Telephone Stephen Derrick on 01-626-8121 to discuss your investment.
ATTENTION ALL Estate Agents, Employment Agencies, Yacht Brokers, Antique Dealers and Motor Traders.
Find out about *FINDER SOFTWARE!*

SOME BLUE CHIPS

TESKIM. This *ROM* will simulate the Tektronix 4010 family of graphics terminals. It's rather good!
UPPER/LOWER CASE ADAPTOR A chip for the chap considering word processing.

NEW ISSUES

We are continually trying to bring the latest add-ons for your Apples. Please phone for the latest product information and data sheets.

NEW PRODUCTS

8" SHUGART DISKS giving 1.2 Megabytes A twin drive (with room for a third.) disk system with controller and software, give tremendous commercial possibilities. £2350 Excl. V.A.T.
WORD PROCESSOR. Ask about our Apple II Plus word processor package. Complete System with Diablo 1650 Daisy-Wheel Printer. £4250 Excl. V.A.T.
PERSONAL COMPUTER PRINTERS. Sensational 40 & 80 Character printer (graphics options) from £243 Excl. V.A.T. Interfaces for Apple, Pet & TRS 80. High quality silent printers. It's your choice!
A/D BOARD At last we have either an 8 bit or 12 bit A/D card for Apple. Excellent spec from £125 Excl. V.A.T.
APPLE PASCAL £296

Personal Computers Limited

194-200 Bishopsgate, London EC2M 4NR.

Let us advise you about *COLOUR DISPLAY* on your Apple. Contact Technical Services.