AN ARGUS SPECIALIST PUBLICATION
AUGUST 1985 £1.00

COMPUTING Today

THE DOMESDAY PROJECT

William I's inventory revisited

SHORT STORY

The Case of the Moriarty Engine

UNIX

micros in the Bell Labs environment

DATA PROTECTION

is the new Bill old hat?

LANGUAGES

ADA on test

GRAPHICS

the nature of computer generated images

THE CRACKER

The spreadsheet designed for normal people who make mistakes. Instant error detection and easy correction. Yes, this is a special feature. It means that what you do is right, first time, most times. For Businessmen, Engineers, Scientists and most simpletons.
£100+£2pp+VAT, CP/M-Z80, CP/M-86, MP/M-86, CCP/M-86, MSDOS, PCDOS

DISASSEMBLERS, Z80, 8086

Powerful practical file based disassemblers. Produces error mesages, full listings and cross-reference tables. The 16-bit version suitable for whole 8086 family and 8087. This version can handle. .CMD, .COM and .EXE files also ROMS.
£80+VAT, CP/M-Z80, CP/M-86, MP/M-86, CCP/M-86, MSDOS, PCDOS

TRANSLATOR Z80 TO 8086

This is a single pass translator designed to allow you to get your Z80 source code into an 8086 form easily. It has no real size limit and works fast. Data areas handled intelligently. Output for popular assemblers. An easy way to learn 8086 assembly language.
£80+VAT CP/M-Z80, CP/M-86, MP/M-86, CCP/M-86, MSDOS, PCDOS

Software Technology Ltd
PO BOX 724, BIRMINGHAM B15 3HQ
TEL: 021-454 3330
TELEX: 337675 TELPES G

THE MALTRON KEYBOARD

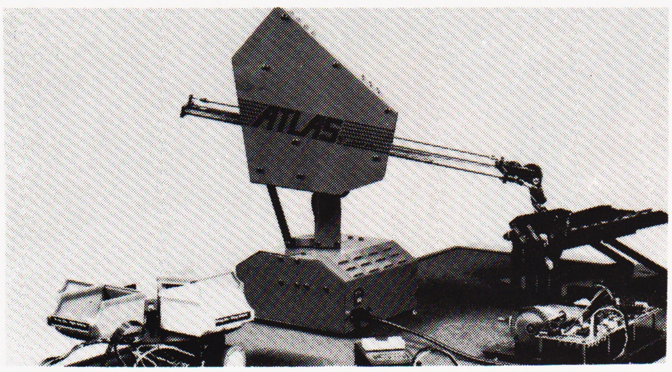
Why Hasn't Anyone Thought Of It Before?

If you spend a lot of time at your keyboard it is very likely that you get unpleasant aches and pains that can extend from your wrists to your neck and shoulders.
The cause is the shape and letter layout of the standard QWERTY keyboard, whose basic design goes back to the last century.
The MALTRON keyboard is easy to learn and tests have shown that great improvements in efficiency and comfort can be obtained.

Contact: **P.C.D. MALTRON LTD**
15 Orchard Lane,
East Molesey,
Surrey KT8 0BN
Tel: 01-398-3265

The copyright subsisting in this keyboard is the property of P.C.D. Maltron Ltd.

The LJ Robotic Work-cell system



Based around the popular LJ ATLAS Robot, this work-cell provides an automatic parts-selection system running under full microcomputer control.

For full details of this new work-cell and other LJ robotic systems send for our comprehensive data sheets.

L.J. Electronics Ltd.
Francis Way, Bowthorpe Industrial Estate, Norwich NR5 9JA.
Telephone: (0603) 748001 Telex: 975504.

AMSTRAD INTERFACES

THIS IS NOT JUST A MODEM, BUT A COMPLETE SYSTEM. NOTHING ELSE TO BUY

★★ MODEM ★★
★ £153.00 ★

Incorporating serial and parallel interfaces, to allow software control of all functions, each feature controlled from basic with the bar commands. Call from m/c or on entering bar modem all controls are menu driven for ease of use, bell/ccitt standards 300/300 600 1200 1200/75 75/1200 full and half duplex. Auto dial and auto answer contact bulletin boards, prestel compatible, software buletin on its own sideways Rom. Unique panel display, it displays what the modem is doing, mode of operation, and digits when auto dialing, standard B.T. plug connector. *Note this modem is not B.T. approved.*

★★ SIDEWAYS ROM ★★
★ £26.05 ★

The unit holds 4 Roms. Each can be 2, 4, 8 or 16K in size incorporating a device to allow slower Roms to be used less than Amstrad suggested 200, that means cheaper Roms, free utility Rom with every unit.

RS232

Communicate with your modem
Talk to other computers
Use serial printers
Split Baud rates
Standard 25 way 'D' connector
£39.96

PARALLEL PORT

Make that Robot move
Control electrical appliance
Twin 8 bit ports
Operates direct from basic
2 × 14 way speedblock connectors
£22.57

8 BIT PRINTER PORT

Make use of that 8 bit printer
Allows character codes
Above 127 (ie 0 to 255)
Plugs in between centronics
Port and printer cable
£17.35

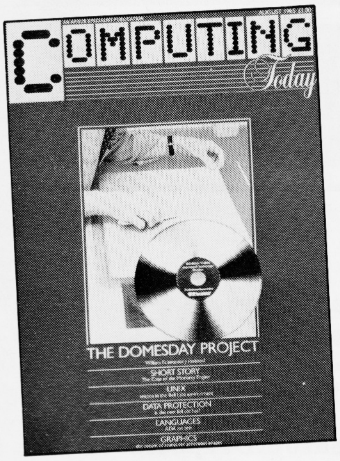
All units are cased and have through connectors
★ Please add VAT ★

COMPUTER HARDWARE & SOFTWARE

KDS ELECTRONICS

15 Hill Street, Hunstanton, Norfolk PE36 5BS
Tel: (04853) 2076

PD492



CONTENTS

VOL 7 NO 8 AUGUST 1985

REGULARS

SERIES

GENERAL FEATURES

SPECIAL FEATURES

COVER

As schools up and down the country gather material for the BBC's laser-disc Domesday Project, the original Domesday book (circa 1066) is enjoying a facelift by a team of restorers at the Public Records Office in Kew, London.
Story: Page 16.

Editor: Don Thomasson
Assistant Editor: Jamie Clary
Technical Illustrator: Jerry Fowler
Additional Illustration: Grant Robertson
Advertisement Manager: Anthony Shelton
Classified Sales Executive: Caroline Falkner
Advertisement Copy Control: Sue Couchman, Lynn Collis
Publishing Director: Peter Welham
Chairman: Jim Connell
Origination and Design: Design International
Cover Design: Argus Design

Member of the Audit Bureau of Circulation
ISSN 0142-7210

Computing Today is normally published on the second Friday in the month preceeding the cover date. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, Lodnon EC2A 4JS. 01-247 8233. Printed by: Alabaster Passmore & Sons Ltd, Maidstone, Kent.

Subscription notes; UK (£16.20) including postage. Airmail and other rates upon application to Computing Today Subscriptions Department, Infonet Ltd., Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB England (phone 0442 48432).

EDITORIAL & ADVERTISEMENT OFFICE: 1 Golden Square, London W1R 3AB.
Telephone: 01-437 0626. Telex: 8811896.

NEWS

SINTROM LAUNCH UNIX-BASED PORTABLE

The Instrument Systems Division of Sintrom Electronics , the Reading-based specialist computer distributor, has launched the first UNIX-based portable computer called the Integral Personal Computer from Hewlett Packard.

Providing full multi-tasking facilities with windows, the Integral Personal Computer is primarily designed for use in business, word processing and office automation applications. it incorporates the Motorola 68000 16/32 bit microprocessor running at a fast 8MHz and provides 512 Kbytes of internal RAM expandable to 1.5 Mbytes or 5.5 Mybes externally with the use of bus expansion modules. Half the total RAM can be allocated as RAM disk, enabling applications to function at higher speeds than with standard floppy disk drives.

Also featured is a special 16-bit graphics co-processor providing 32 Kbytes of dedicated graphics RAM.

Several programming options are available including Basic and 'C' which enables the Integral Personal Computer to be used as a powerful programming system. The Integral Personal Computer also provides software compatability with Series 80 computer software written in Basic.

Other features include a high-density double-sided 710 Kbyte, 3½" floppy disk drive, a built-in 150cps ink jet printer and a 9" solid state electroluminescent display screen.

The Integral Personal Computer also contains 256 Kbytes of ROM incorporating the Hewlett Packard version of the UNIX operating system, the Personal Applications Manager and all necessary 'windows' firmware. It also featues a fully-spaced detachable keyboard.

DEVPAC80 ASSEMBLER FOR CP/M MICROS

Hisoft has announced the release of Devpac80 — a suite of powerful flexible program development tools for Z80 CP/M computers.

HiSoft has been writing CP/M development software for many years and has always believed that the advent of the CP/M based home-computer was inevitable. Now, with machines like the AMSTRAD CPC464 and the Tatung Einstein there is a need for good programming tools running under the CP/M operating system.

HiSoft Devpac80 is *the new standard* (it says here — Ed) in CP/M development packages.

The package includes ED80, a fast, full-screen side-scrolling editor. Completely installable with disc-coded, pull-down help screens, cut and paste editing, wild-card find and substitute, GOTO line, recovery of deleted text and many other features.

GEN80 is a two-pass assembler handling over 4000 source lines a minute. Disk inclusion of library files, full textual macros, conditional assembly and complete operator-precedence arithmetic also feature.

MON80 is a high-specification, single-stepping monitor and debugger providing disassembly to disk (producing a file ready for ED80 or GEN80), multiple breakpoints, loop interpretation, pattern search (bytes text or mnemonics) and more.

David Link of HiSoft says: "We believe that Devpac80 is an extremely important product. It is a high quality package at an affordable price and backed up by our renowned support facilities. For the first time, a sophisticated development system is available to the home computer user."

Devpac80 is available from:

HiSoft
180 High Street North
DUNSTABLE
Beds. LU6 1AT
Tel. (0582) 696421

Price £39.95 inclusive for all CP/M 2.2 formats. □

BBC GEOSOFTWARE

Introducing Geography is a new title from BBC Publications designed to enhance children's understanding of the world's geography and help them acquire a geographer's eye.

The software package contains four separate programs, each constructed to test the user's knowledge of a particular geographical topic.

In **Flight,** a pilot and a navigator choose one of the several global air routes including refuelling stops, given by the computer. Using atlases, players fill in their flight log with names of countries, towns, rivers and mountains in order as they are passed on each leg of the flight.

With **Summit**, children learn the importance of supplying stores to each camp during an expedition to climb the southwest face of Everest. An understanding of the dangers in a mountain environment, including the weather conditions and the language used to describe them, are rapidly acquired as the team struggles to the top.

In **River**, a plane crashes on a mountain range. The survivor's map shows only the contour lines of the region. Finding and following the map ensures a successful escape. Children find out the landscape features at each stage of the river and interpret a cross section as well as the contours. Children can choose 'easy' and 'hard' options and there is a demonstration file for teachers. The simulation tests the children's vocabulary and encourages them to make informed decisions. In a sub-program the rate of flow in meanders is computed, showing how a raft on water responds to the slack and fast currents.

Nomad helps children learn the hardship of the Fulani, a race of nomadic pastoralists whose cattle graze in the Savanna of west Africa. The simulations contain a selection of the recorded 20 years rainfall figures for the area. These help to illustrate the struggle of keeping cattle alive as the annual rainfall decreases. Only the very lucky or clever survive by moving their cattle to the areas of highest rainfall, using the towns and avoiding disease, year by year.

Introducing Geography is availabe in good computer shops price £20.00. To order by mail, write to BBC Publications, PO Box 234, 144-152 Bermondsey Street, London SE1 3TH.

This three disk set is available only for the BBC micro. □

AMSTRAD GETS BRIDGE

The latest release from Kuma Computers is Bridge for the Amstrad CPC464.

Bridge from Kuma offers a challenging version of this traditional game using Blackwood and Stayman conventions.

Jon Day, Sales Manager of Kuma, says: "Not only do users want a challenging game, they also expect it to have good instructions and to be well presented. Bridge for the Amstrad fulfills all these requirements."

Bridge is available now at a cost of £8.95 incl VAT from all good computer retailers.

Kuma now have 14 titles for the Amstrad CPC464, including utilities, applications and entertainment packages. For full details on Bridge and the full range of Kuma programs contact:

Jon Day
Kuma Computers Ltd
12 Horseshoe Park
Pangbourne
RG8 7JW
Tel: 07357 4335 □

MINI MODEM

New from STC Electronic Services is the MM 100 Mini Modem.

Offering both business and personal users a sound and effective means of transferring computer data over the telephone, and at £86.25 +VAT, the MM 100 is believed to be among the most competitively-priced models on the UK market.

Mini Modem is compatible with most computer data bases and offers a choice of originate/answer modes with a transmit/receive rate of 300 Baud.

● For further details contact Stephen Smith, STC Electronic Services, Edinburgh Way, Harlow, Essex CM20 2DE. Tel: 0279 26777. □

ELECTROHOME INTRODUCE NEW PROJECTION SYSTEM

Eletrohome Limited have announced their new monochrome video data projector — the EDP 58.

The EDP 58 features a super-bright P53 phosphor CRT which has a maximum brightness rating of 500 lumens/300 foot lamberts — 40% brighter than CRTs used in previous projectors. The displayed image is in 'easy-on-the-eye' green with 1300 line resolution with less than 4% distortion.

The EDP 58 can be used with flat, curved or rear projection screens with a diagonal measurement of between 1.371 m (4.5 ft) and 3.048 m (10 ft), The EDP 58 has a horizontal scan rate of up to 33 kHz and a video bandwidth of 20 MHz allowing it to faithfully reproduce the high resolution graphics produced by practically all computer systems.

The EDP 58 is compact, portable and both shorter and lighter than previous designs. It has a wide range of optional accessories including ceiling mount, floor stand and desk stand. □

PASCAL COMPILER FOR THE SINCLAIR QL

Metacomco has announced the launch of QL Pascal — a powerful Pascal compiler for the Sinclair QL.

QL Pascal is a complete implementation of ISO 7185 — the international standard for Pascal.

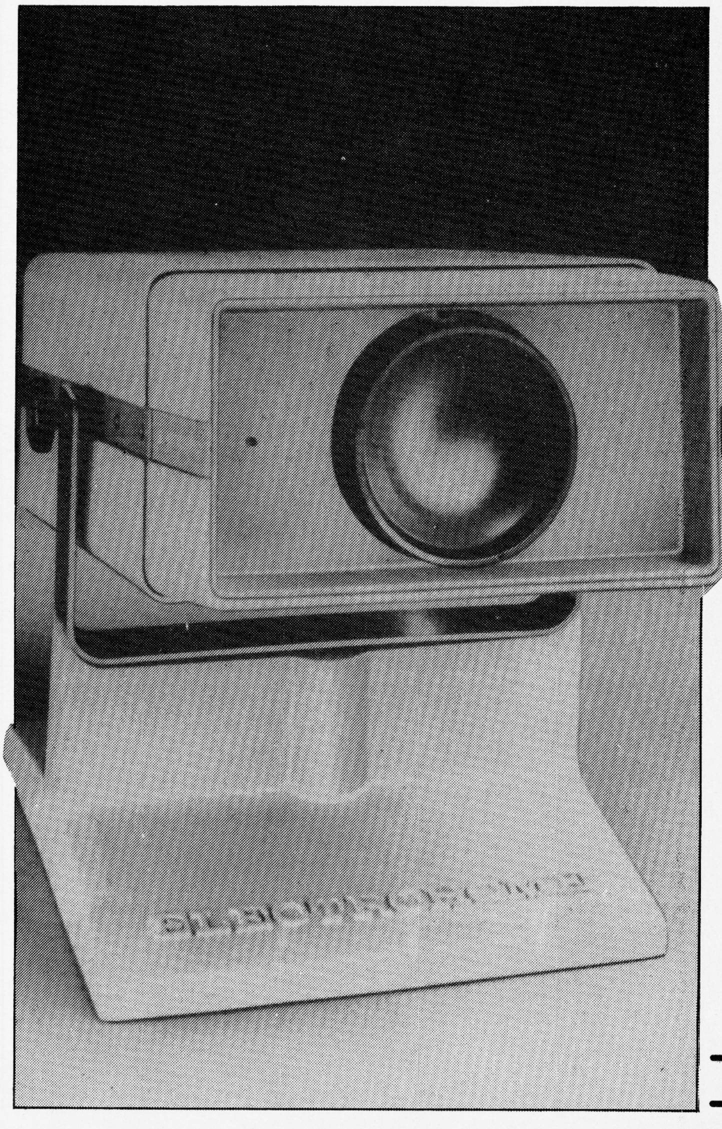
QL Pascal is a true compiler, which compiles quickly in a single pass, without intermediate stages. It generates native 68000 code, so programs execute fast taking full advantage of the QL's potential.

Features include: interfaces to QDOS, any length variable names, 32 bit integers, very large sets and arrays, and comprehensive error handling.

A spokesman for Metacomco said: "QL Pascal has been derived from ISO Pascal 68000, a powerful compiler which was recently validated by the British Standards Institution as meeting the exacting requirements of ISO 7185 — the international standard for Pascal. QL Pascal is therefore a high specification product — well suited to the requirements of professional developers. Newcomers to the language will find it equally attractive through its helpful error handling, standard dialect, and easy to use QDOS interfaces. and the fast, straightforward compilation will help programmers get the most out of their QL."

QL Pascal will be available in shops or by mail order at £89.95 (including VAT). It is the fourth language that Metacomco has launched for the QL; Assembler, BCPL and Lisp (reviewed in *CT* — Ed) have already won Metacomco considerable acclaim.

● For further information contact David Sykes on 0272 428781. □



SPEECH RECOGNITION SUBSYSTEM FROM STC MERCATOR

New from STC Mercator is the Speech Design SDR 600 — a powerful speech recognition subsystem.

Simple high-level I/O commands link the host computer to the SDR 600 and almost no workload is imposed on the user system as the Subsystem incorporates on-board micro-processors and memory.

The SDR 600, mounted on a single Euroboard, features an audio input via a microphone A/D conversion and a 128-word vocabulary in on-board 16Kbyte of static RAM which can be easily expanded externally to 80 blocks of 128 words each.

It accepts, for example, command words such as TRAINING or RECOGNITION as inputs while outputs include the number of the recognised word and status information. In TRAINING mode, vocabularies are stored in the reference memory and words may be assigned to sub-vocabularies to attain higher recognition rates thereby shortening the response time. For the RECOGNITION mode, current voice input is verified against stored data. The recognition time for the subsytem is 0.5s with a word length of between 0.2 and 2.0s.

The Subsystem offers a variety of features including ease of integration; a unique facility for data acquisition in a 'busy hands' commercial, industrial and medical environment.

● For further details contact Andrew Stubbins, STC Mercator, South Denes, Great Yarmouth, Norfolk NR30 3PX. Tel. 0493 844911. □

DEVELOPMENTS IN MICROLINK DATA ACQUISITION

The data acquisition and control interface, Microlink, has been enhanced and is now provided with better and more comprehensive hardware and software.

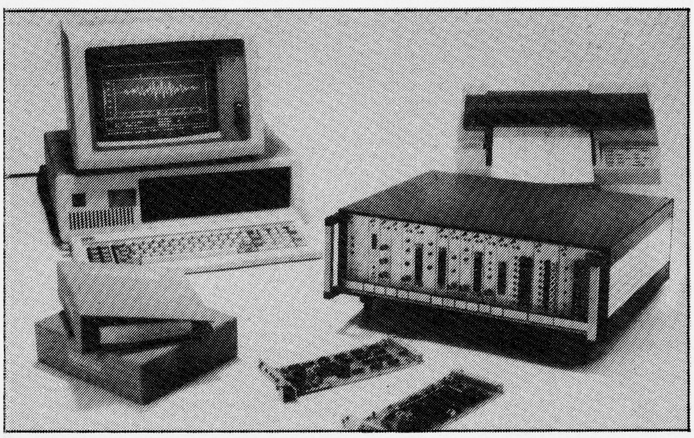
With the addition of 15 new modules, Microlink offers a choice of more than 40 modules including analogue and digital inputs/outputs, transducer inputs, timers and counters. The new modules include programmable gain amplifiers, transient capture devices, pH and R.T.D. inputs, frequency and event counters and alarms.

With the availability of new software for use with the HP150 and BBC micros, Microlink hardware can now be used with most popular micros, including the IBM PC, Apricot, HP 80 series, HP 9816, Sirius and Apple micros. Sample programs are provided in BASIC for each of these computers.

A new range of applications programs called Microlink Scientific Software has been launched. The first two packages are for data logging and for waveform capture, using the IBM PC or the Apricot.

Microlink hardware and software has been designed as an integrated package ensuring system compatibility and effective performance. The flexibility of the system means that most data capture requirements can be satisfied in fields as varied as electronics, engineering, chemistry and life sciences.

● For further details contact Biodata Limited, 10 Stocks Street, Manchester M8 8QG. Tel: 061-834 6688, Telex: 665608 Biodat G. □

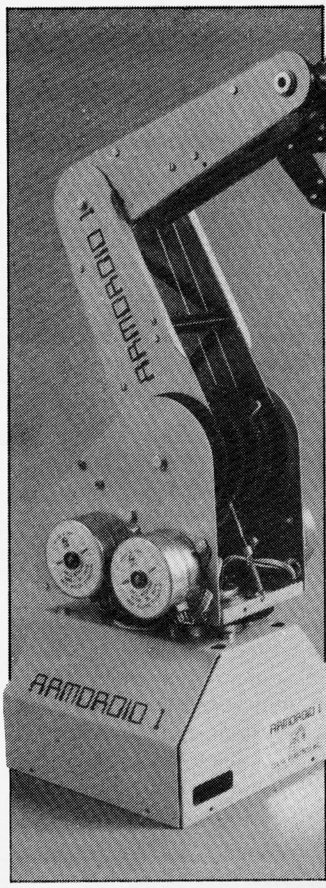


FORTH PACKAGE FOR ARMDROID I

A new software package called COLNEFORTH has just been released for the Colne Robotics ARMDROID I desktop robot.

Available on disk for BBC Model 'B' and Commodore 64 micro-computers, COLNEFORTH is highly interactive. It emphasizes communication between user and robot by means of a language structured in a way that approximates a natural language.

The new disk-based software for BBC and Commodore micros permits easy and versatile programming of ARMDROID I. Each joint of the stepper motor driven robot has a number of discrete positions and under COLNEFORTH control absolute commands can be used to achieve a specific position for each axis. Programming is highly structured so that complex move sequences can be built up from simple units and sub-sequences. Programs are convenient to edit, extend and save.



COLNEFORTH is well suited to experimenting with real world problems in robotics and, with several joints driven simultaneously, the robot can simulate the movement of industrial robots. Sub-sequences can be tested individually and used reliably to produce repetitive actions such as those of pick and place operations.

The new package is available for £75.00, comprising a disk and comprehensive manual detailing the use of COLNEFORTH with the ARMDROID I robot.

● For further details contact Elizabeth Newberry, Publicity Officer, Colne Robotics Co. Limited, Beaufort Road, off Richmond Road, East Twickenham, Middlesex TW1 2PQ. □

SPEECH SYNTHESISER SPEEDS RESISTOR TESTING

Welwyn Special Products is using a home computer speech synthesiser to speed the testing of high precision resistors.

A Currah Speech 64 synthesiser for the Commodore 64 has been adapted by Welwyn's engineers to give an audio signal to support a visual screen-based message.

A person carrying out the test has to examine resistors using a microscope, this means the operator has to lift his head to check results on a screen. Using the Speech 64 unit with special software held on disk, the operator no longer need lift his head, as test results are given audibly.

● More details from Carole Carr, Welwyn Electronics, Bedlington, Northumberland NE22 7AA. Tel. (089-426) 2652. □

INK JET COMES OF AGE?

Epson have recently announced two new ink-jet printers. Although ink jet printers have been around for some time — a year or more — and Epson are one of the world's leading producers of printers, these are their first products using ink-jet technology.

At the press launch, held in the plush surrounds of Maxim's de Paris restaurant, Epson were graceful in their dismissal of other manufacutrers: "We're not the first, there have been others that have experienced snags and pitfalls along the way . . . but these products that I'm introducing today are the result of a six year development programme by Epson to build-in that reliability that you have come to associate with the Epson name . . . ."

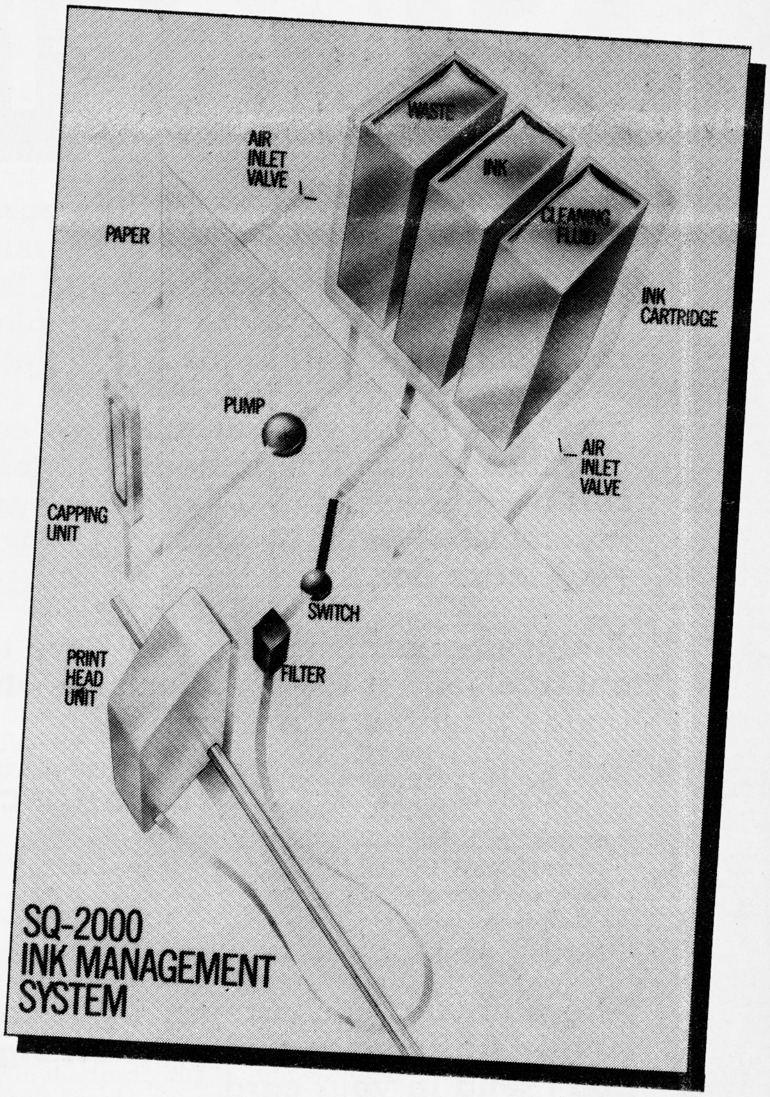
The two printers actually launched represent the top and bottom of the range that will eventually be available. They were the SQ2000, available immediately and priced at a cool £1825 + VAT, capable of 105 characters per second in a variety of typefaces, and the portable HS80 (available later this year) which could print at 160cps unidirectional (ie, when you take line-throws into account it gets much slower). The SQ2000 uses 24 nozzles while the HS80 has only nine. However, Epson claim a head life in excess of 100,000,000 characters for both.

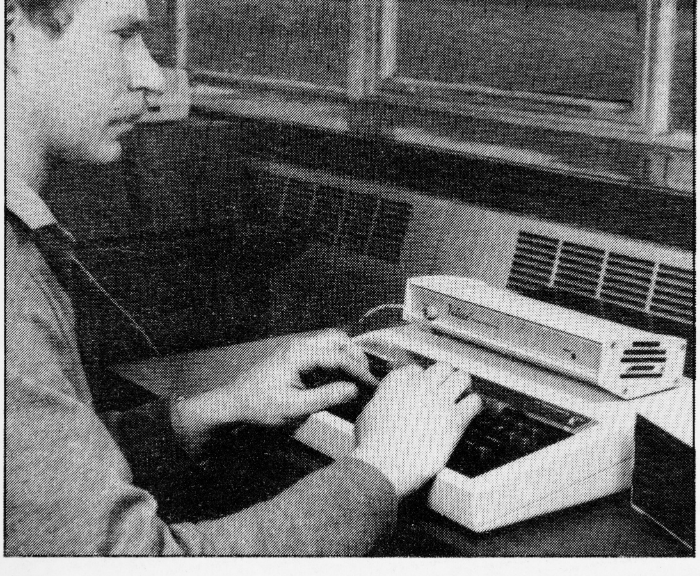
Both units have rather natty head cleaning and capping mechanisms built-in; on the 2000 it's all done under electronic control, while the 80 ingeniously uses a section attached to the on-off switch to do the cleaning manually, so you clean the head automatically when you operate the switch.

So why do Epson think that ink-jet technology is the one to go with? They reckon that it offers speed with quiet quality. With dot-matrix and daisy-wheel printers speed is limited by mechanical factors as well as heat generation in the printer head mechanism. Also, the faster they go, the more noisy they become — and Epson believe that a number of important employers (for example, local authorities) are becoming increasingly concerned about environmental noise in offices. So what is needed, in their opinion, is a change of technology.

Will the Epson printers succeed? I must say that I was disappointed with the quality of the print. They say that they regard the quality produced by the SQ2000 as 'letter quality' (admittedly much slower and noisier) daisy-wheel printer. The print quality on the HS80 is barely better than a reasonably good dot-matrix printer. So the crux of the question is just how important the noise factor is — and only time will tell there.

Dave Bradshaw □





NEW VOICE SYNTHESISER FROM CYBER

A new voice synthesiser from the USA is claimed as the most lifelike yet by its UK distributor, Cyber Robotics Limited, of Stone, Staffordshire.

The Votrax Personal Speech System offers 95 per cent correct pronunication, plus eight-octave music and sound effects. To operate, you type the words in at your computer keyboard. The synthesiser turns this into spoken words, complete with whatever speed and emphasis you choose.

In education, the Votrax can be a speaking textbook that also conducts oral tests, and for the blind verifies keyed-in data.

The Votrax synthesiser is Z80 based, has serial and parallel interfaces, and is claimed to work with virtually any micro. Priced at £375, it is available from: Cyber Robotics Ltd, Stone, Staffs ST15 0SA. Tel. 0785 812121. □

Subscriptions

Personally, we think you'll like our approach to microcomputing. Each month, we invite our readers to join us in an abundance of feature articles, projects, general topics, news and reviews — all to help committed micro users make more of their microcomputers at home or at work.

However, if you've ever missed a copy of Computing Today on the newstands, you'll not need us to tell you how valuable a subscription can be. Subscribe to CT and for a whole year you can sit back, assured that each issue, lovingly wrapped, will find its way through your letter box.

And it's not difficult! All you have to do is fill in the form below, cut it out and send it (or a photocopy) with your cheque or Postal Order (made payable to ASP Ltd) to:

COMPUTING TODAY Subscriptions,

Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

Alternatively, you can pay by Access or Barclaycard in which case, simply fill in your card number, sign the form and send it off. Please don't send in your card.

Looking for a magazine with a professional approach with material written by micro users for micro users? Why not do yourself a favour and make 1985 the year you subscribe to Computing Today and we'll give you a truly personal approach to microcomputing.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :
COMPUTING TODAY Subscriptions
INFONET LTD,
TIMES HOUSE,
179 THE MARLOWES,
HEMEL HEMPSTEAD,
HERTS HP1 1BB.

Please commence my subscription to Computing Today with the .... issue.

SUBSCRIPTION RATES (tick ☐ as appropriate)

£16.20 for 12 issues UK ☐
£18.70 for 12 issues Overseas Surface ☐
£51.20 for 12 issues Overseas Air Mail ☐

I am enclosing my (delete as necessary) cheque/ Postal Order/ International Money Order for £..........
(made payable to ASP Ltd)
or
Debit my Access/ Barclaycard*
(*delete as necessary)

BARCLAYCARD VISA

Please use BLOCK CAPITALS and include postcodes.

NAME (Mr/ Mrs/ Miss) ....
delete accordingly
ADDRESS ....
....
.... POSTCODE ....
Signature ....
Date ....

BACKNUMBERS

JANUARY 1984
TRS-80 programmer's aid, Apple music, Electron review, TRS-80 screen editor, calendar program.

FEBRUARY 1984
Using MX-80 graphics, Colour Genie monitor, non-random random numbers, ZX81 Forth, Program recovery on the Commodore 64.

MARCH 1984
Easycode part 1, BBC poker, Spectrum SCOPE review, Genie utilities, Spectrum Centronics interface.

APRIL 1984
MEMOTECH MTX500 review, Genie BASIC extensions, Brainstorm review, Disassembly techniques, Recursion.

MAY 1984
Debugging, Spectravideo SV318 review, Extending the Commodore 64's BASIC part 1, Z80 text compactor.

JUNE 1984
Adler Alphatronic review, Digithurst's Microsight review, Commodore search and replace, CP/M directory, Interrupts.

JULY 1984
Commodore BASIC extensions reviewed, The Art of Islam, a fast sort, Brother HR5 review, Random Thoughts, extended palette on the Dragon.

AUGUST 1984
Apricot xi review, BBC Mode 7 screen editor, Genie sprites, Microdrive-file line editor, TRS-80 screen scroller.

SEPTEMBER 1984
CUBE's Beebflex, Electron drawing utility, MTX real time clock, Commodore SX64 review, BBC disassembler, TRS-80 Fastsave.

OCTOBER 1984
AMSTRAD CPC464 review, Dragon sprites, Commodore 64 adventures, BBC Draughts, Nascom screen dump.

NOVEMBER 1984
Apple IIc review, Epson PX8 review, MTX utilities, Z80/TRS-80 memory move routine, 16-page Business supplement.

DECEMBER 1984
Acorn Bitstick package review, Art and the AMSTRAD, BBC Draw, Psion Organiser review, Koala Pad review.

JANUARY 1985
BBC Commodities, Tatung Einstein review, Fujitsu Micro 16 review, Commodore 64 prettyprint, MTX500 Life, Nascom string-save.

FEBRUARY 1985
The Intelligent Computer, Dragon interrupts, BBC Machine-code monitor, Tasword 464 review, Spectrum/BBC cassette volume meter, Sakata SCP800 printer/plotter review, Spectrum ON ERROR, TRS-80 mail list, BBC passwords; Deficiency, Abundance, Perfection.

If you've lost, lent or had stolen one of those precious back copies of Computing Today then now is your chance to fill the gap in your collection. The list of issues given here represents the few remaining copies that we have available to help complete your library of all that's good in features, programs and reviews.

If you want one of these issues, it's going to cost you £1.40 (including postage and packing) but we think that's a small price to pay for the satisfaction you'll get. Ordering could hardly be made simpler — just fill in the form, cut it out (or send a photocopy) together with your money to:

Backnumbers,
Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

If you wait until next month to do it, the chances are that we'll have run out of the very issue you wanted!

A COMPUTING Today Reader Service

BACKNUMBERS

Please send me the following Backnumbers

ISSUE

At £1.40 each. I enclose £ ....

NAME ....

ADDRESS ....

....

....

.... POSTCODE ....

Signature ....

I enclose a cheque/PO for £ ..... (Payable to ASP Ltd)
I wish to pay by credit card

Access ☐ Barclaycard ☐

Buy it with Access

BARCLAYCARD VISA

Insert Card No.

If you wish to pay by Access or Barclaycard, just fill in your card number and sign the form, do not send your card.

Please allow 21 days for delivery.

COURSES AND TRAINING

Education around the U.K.

MORE PLACES FOR SYSTEMS ENGINEERING

The University Grants Committee has awarded funds to provide for 20 additional places per year for the University of Kent's successful Computer Systems Engineering course in the first phase of a £43 million Government programme to strengthen the number of well-qualified science and engineering graduates.

It is one of only 20 universities to benefit in the first phase of this three-year programme. The selection was based on the relevance of course material to the needs of industry and the views of industrailists were sought by the University Grants Committee. A total of £3.2 million has been allocated so far to pay for 579 extra engineering and technology students from next October, and for the additional teachings post which will be required.

"We are very pleased that the UGC has recognised in this way the distinction of this University's teaching in Electronics and Computing and has chosen to allocate additional resources for our Laboratories" said the University's Vice-Chancellor, Dr. David Ingram.

Kent's BSc in Computer Systems Engineering course covers a broad spectrum of techniques and skills involved in the design and application of modern integrated computer systems, an area which has expanded rapidly in recent years and which shows every sign of continued future growth.

The course adopts an integrated approach to both the hardware and software aspects of the subject, but embraces also many related aspects of computer systems required to exploit current developments in microelectronics technology.

The funds allocated in this first phase of the programme will provide extra places for 475 undergraduates and 104 postgraduates in 20 universities. It is expected that some 4,000 additional places will eventually be provided when the programme is fully implemented. Allocations for the second phase of the programme will be made in the summer.

Further information about admission to the BSc programme in Computer Systems Engineering can be obtained from either the Admission Office, Registry or from the Admisions Officer for Computer Science Engineering, Electronics Laboratory, The University, Canterbury, Kent. Tel: (0227) 66822.

MICROPROCESSORS TECHNICAL FUNDAMENTAL COURSE ON VIDEOTAPE

Integrated Computer Systems has announced the availability of a comprehensive video course on microprocessor technology and applications. Course 660 Microprocessor Technical Fundamentals is designed to introduce the theory and practice of designing with microprocessors, to those who may have had little or no exposure to digital electronics.

In 8 three hour long sessions, the course introduces the fundamentals of digital electronics, microprocessor hardware, and the associated software and support circuitry. Interfacing, applications, and programming in machine language, assembly language and higher level languages are discussed in detail. The course is accompanied by a specially designed Microcomputer Training System, which is a self-contained microcomputer with interface circuitry, featuring Direct Memory Access, D/A and A/D conversion, programmable I/O ports, and monitor and demonstration programs. The student enjoys the benefit of hands-on experience with each of the concepts covered by the course, and makes extensive use of interfacing techniques and hardware.

The video sessions employ the latest in computer graphics and animation to illustrate technical concepts. The course assumes no prior exposure to microprocessors or integrated circuits, yet the final exercise is to design a microcomputer. It is currently being used in both academic and major industrial settings.

Free previews of the video may be obtained by contacting Lorna Warwick, ICS Publishing (UK) Co. Ltd., 3 Swan Court, Leatherhead, Surrey KT22 8AD, England. Telephone (0372) 379211; Telex 915133 ICSPUK G.

MICROPROCESSOR TROUBLESHOOTING COURSE

Two new courses complete the comprehensive video

training series on microprocessor troubleshooting for Integrated Computer Systems. These courses are based on their highly successful "live" troubleshooting workshops conducted for major companies such as IBM, Westinghouse, Kodak, and other clients throughout the U.S. and Europe over the past five years.

Microprocessor Troubleshooting with Diagnostic Software and In-Circuit Emulation provides a theoretical and hands-on background in troubleshooting with software diagnostic techniques and IN-Circuit Emulation instruments.

Microprocessor Troubleshooting with Signature Analysis is a comprehensive course in theory, hands-on troubleshooting, and design using the technique of Signature Analysis.

Computer graphics and animation are used extensively to increase the educational effectiveness of these colour videotapes by illustrating the techniques and keeping the pace lively. Included are tutorial materials which provide guidance and reinforcement. Each participant receives a tutorial Workbook, and a Video Course Guidebook, while the Administrator's Manual provides detailed training plans to assist with the organisation and administration of the training programme.

Also available in the hands-on videotape series are an introductory course on Microprocessor Hardware and Software and courses on troubleshooting microcomputers with a wide range of conventional and specialised instruments and test equipment.

Free previews of the video program may be obtained by contacting Lorna Warwick, ICS Publishing (UK) Co. Ltd., 3 Swan Court, Leatherhead, Surrey KT22 8AD, England. Telephone (0372) 379211; Telex 915133 ICSPUK G.

SPHINK OFFERS C TRAINING FOR MS-DOS PROGRAMMERS

Following its recent announcement of a new PC Division and an integrated catalogue of tools for MS-DOS 'C' programmers, Sphinx is now offering new training courses to complement its additional products.

Building on its 'C' expertise, Sphinx's first offerings will be introductory and advanced courses on the following topics:

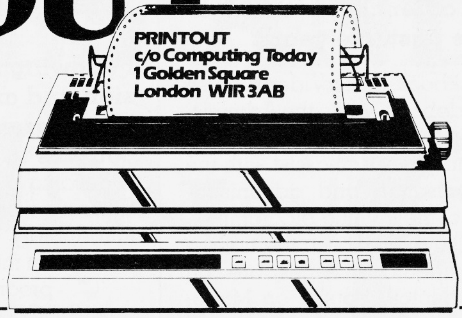
'C' programming in the MS-DOS environment.

Tools and utilities for 'C' programmers under MS-DOS.

Each course will range from two to five days in length. Courses based around tools will assume a knowledge of 'C' programming.

Courses are scheduled to start this summer. Full details of content, schedules and prices may be obtained from James Minter at Sphinx on 0628 75343.

PRINTOUT



Your opportunity to ask questions, put us straight, seek advice.

WILL AMSTRAD GET LISP?

Dear Sir,

I read with interst your article on LISP for the Sinclair QL (*Computing Today* June 1985). I own an Amstrad CPC464 and would like to know if there is a version for my machine.

Yours sincerely,
Ivalson Soreno,
West Germany.

Amsoft say that they themselves have no plans to produce a version of LISP for the CPC464, but they suggest that there may already be a version available under CP/M, and this should work with your CPC464. However, do try before you buy.

AMSTRAD LETTER WRITER

Dear Sir,

As a newcomer to the Amstrad CPC464 I was interested to find your word processing program, Amstrad Letter Writer, in your March '85 edition.

I had expected the program to generate a letter page format on the VDU upon which a letter could be typed. In practice, what resulted was a great disappointment — two bright spots on the screen; top margin Operation of any of the keyboard keys alone or in combination did not cause any movement of either spot although the arrow keys sometimes altered the symbol in the cursor.

ing duplicates that shown in the magazine (if necessary I can send you a tape to confirm this).

Have any other readers any experience of this program or can you, or the authors, offer any advice on this?

Meanwhile, I am back writing letters in the old fashioned way.

Missing lines from Amstrad letter writer

```
3190 GOTO 3000
3199 REM +- Handle control codes -+
3200 IF I$=CHR$(241) THEN GOSUB 3400:GOTO 3000:REM +- Down line -+
3220 IF I$=CHR$(240) THEN GOSUB 3500:GOTO 3000:REM +- Up line -+
3240 IF I$=CHR$(243) THEN GOTO 3120:REM +- Right one -+
3260 IF I$=CHR$(13) THEN GOSUB 3400:CX=LM+1:GOTO 3000
3280 IF I$<>CHR$(127) THEN 3320
3300 IF CX>LM+1 THEN CX=CX-1:T$(CX,CY+BL)=" ":GOTO 3000
3310 GOSUB 3500:IF FG=1 THEN CX=COLUMNS:T$(CX,CY+BL)=" ":GOTO 3000
3315 CX=LM+1:GOTO 3000
3320 IF I$<>CHR$(242) THEN 3360
3330 IF CX>LM+1 THEN CX=CX-1:GOTO 3000
3340 GOSUB 3500:IF FG=1 THEN CX=COLUMNS:GOTO 3000
3345 CX=LM+1:GOTO 3000
3360 IF I$=CHR$(22) THEN GOSUB 4000
3370 IF I$=CHR$(19) THEN GOSUB 4100:GOTO 3000
3380 IF I$=CHR$(12) THEN GOSUB 4200:GOTO 3000
3381 IF I$=CHR$(16) THEN GOSUB 4400:GOTO 3000
3382 IF I$=CHR$(17) THEN 9800
```

Yours sincerely,
R.S. Payne

And . . .

HELP FOR MR BELL

Dear Sir,

The following may help Mr Bell (Printout, *Computing Today* June 1985) to put his Word4Word cassette onto disc. It will not change the program in any way and it will still require any text files to use cassette, but it will enable the program to be up and running in about twenty seconds.

The procedure is as follows:

1. From DOS ready, type BASIC2 to get into LII BASIC
2.. Load the target program in the normal way but do not action it with the normal slash symbol
3. Press BREAK
4. type: PRINT PEEK(16607),PEEK(16608) and note down the two figures.
5. Press BREAK to reboot the DOS.
6. Enter the following BASIC program:

```
10FOR A=-32768 TO -32755:READ B:POKE A,B:NEXT A
20 DATA 33,14,128,17,233,66,1,23,61,237,176,195
25 DATA X,X
30 FOR C=-17115 TO -17102:READ D:POKE C,D:NEXT C
40DATA 33,233,66,17,14,128,1,23,61,237,176,195,0,0
```

7. RUN the program. This places two small machine code routines and a store area into high memory where they are safe from 'bombing' by either the target program or the DOS.
8. Press BREAK and RESET together to get back to LII BASIC
9. Answer "MEM SIZE?" and load the target program as usual but do not action it with the normal slash symbol
10. Type /48421 and ENTER. This actions the higher placed transfer routine and copies the target program byte-for-byte into the high mem store. It takes about half second, then the DOS reboots
11. Type: DUMP W/CMD (START=X'8000,END=X' BD24,TRA=X'402D). This places the lower of the two transfer routines and the whole of the stored code on disc, then returns to DOS ready.
12. Type: AUTO W and press ENTER

The program is now on disc. Switch the computer off.

To run the program:

1. Switch on again. This boots the DOS and automatically loads the W/CMD file which contains the wrongly located target program.
2. Type: BASIC2 and press ENTER
3. Press ENTER to answer "MEM SIZE?"
4. Type: SYSTEM and press ENTER. This actions the transfer routine. The target program is copied to the correct position and entered at the correct entry address

Since this routine adopts rather willy-nilly approach of saving the whole of the user-RAM area it can be used to save any machine code program which loads in this area irrespective of its length or location, provided it does not autostart. This letter was written with the Tandy cassette word processor, Scripsit, loaded down from disc. The load times are 3 minutes 5 seconds from cassette and twenty seconds from disc.

Mr Bell might like to know that Tandy are currently selling a lot of Model 1 software at rubbish prices. If he can locate a copy, he can get the true disc Scripsit for £9.95.

Yours faithfully,
Terry Fuller.

TALKING SHOP

Don Thomasson

When I first wrote a computer program, there was only one option available: machine code. The vocabulary of instructions was limited, with no more than twenty codes to be remembered, each instruction was a single word. Quite complicated routines were written in this way, with few problems. Since the main long-term storage medium was paper tape, and the only display device was a slow and noisy teleprinter, an 'in-store editor' was devised, this being the equivalent of the modern 'on-screen' editor.

A bright young programmer of today would probably view such conditions with horror. He would at least require an assembler, but we saw no need for that. Our 'mnemonics' were the actual instruction codes, and at the 'first pass' we could define the essential program structure, then make a 'second pass' to enter the addresses. That was usually done on paper, since printout was too slow for convenience, and was usually quite easy to patch in corrections on a temporary basis, to be sorted out later into a more formal version.

Nowadays, a programmer has many options. There are many compiling languages, word-pyramid languages like FORTH and LOGO, and — for those not in too much of a hurry — BASIC in a multitude of forms. In the final analysis, all are dependent on the execution of machine code routines, but that is concealed from the user, who might not even be aware that machine code exists. Now and then he may be brought down to earth, finding it necessary to examine machine code in order to discover the nature of a misunderstanding between himself and the language he is using, but that is usually done as a last desperate resort, like any venture into unknown territory.

BASIC, of one kind or another, has long held sway among beginners. This is deprecated by academics, who claim that BASIC breeds bad habits. Now that procedures have been introduced in some forms of BASIC, the academics are partially appeased, though the change has produced a distinction without much difference. However, by such means, BASIC continues to resist attacks from all sides, remaining supremely useful and convenient for many purposes.

A year or so back, there was great enthusiasm for FORTH, but it seems to have waned. One reason is that there have been some abysmal implementations on the market, some users find unacceptable. for those able to cope with it, FORTH has a certain fascination. It is fast, and for some kinds of work it is ideal, but it seems unlikely to challenge BASIC or the compiling languages. LOGO has also enjoyed some interest recently. Like FORTH, it can be seen as a 'word-pyramid' language, procedures defined by words being linked together by other word-identified procedures. One problem is that LOGO has been tied up in people's minds with 'turtle graphics', but as Garry Marshall has shown (*Computing Today* April 1985) turtles can equally well be linked with BASIC.

My own experience with LOGO is limited, but did not encourage me to see it as an ideal language for young children. In the first hour, I got more error messages than I have seen in many years of using BASIC, and the negative nature of the messages gave little guidance as to correction. This was due in part, I feel, to the rather scanty documentation available. Perhaps more practice would produce — if not perfection — at least an improvement.

☐ ☐ ☐

THE KEY test of any language is whether it can implement the kind of program that you want to write. I find LOGO frustrating in this respect. A reader recently took Bill Horne to task for suggesting that ARCSIN and ARCCOS were not essential, since they could be constructed from ARCTAN, but the LOGO I have lacks ARCTAN as well. How can you set up a special character shape? For that matter, how can you access anything external to LOGO itself? I will probably be told that the answers or obvious, or that the questions are not valid, but I believe they identify restrictive limitations.

Perhaps we could say that FORTH is too demanding, and LOGO too simple.

Among the compiling languages, FORTRAN still holds popular esteem, but seems rarely available on small machines. PASCAL is perhaps respected, but we hear little of it these days. CORAL and ADA are for the military scene. It is nominally possible to compile from BASIC, but experiments in that area have not produced impressive results, the compiled programs being too bulky for convenience. They run at relatively high speed, which is sometimes important, but apart from that they are little different from the source BASIC.

An encouraging sign is the increasing use of BASIC with machine code inserts, which is made especially convenient on the BBC Computer (and on the AMSTRAD with MAXAM). On other machines it is still possible but less easy to use, involving expedients like hiding machine code in REM statements. Those who use this approach will soon begin to expand their knowledge of machine code, and that is a good thing. Compiling languages are all very well, but they involve some loss of system efficiency. A manual programmer working in assembler can often eliminate store accesses by using registers to hold temporary values, but most compilers are unable to work out the logistics which this involves. If the processor in use has few free-use registers, this is unimportant, but the performance of some processors can be seriously degraded.

☐ ☐ ☐

THE ARGUMENT for using compilers is that they save a lot of thought and calculation, allowing the programmer to concentrate on more important things. In certain instances there has been evidence that the complex rules imposed on the user can offset this to a considerable degree. Prolonged arguments about the effect of quite a simple statement are not unknown, and the task of fine-tooth combing a suspect program is not easy.

The overall motto, then, is 'to each his own'. Some people are equipped to use compilers, but others may produce equally good — perhaps better — results with simpler means. Demonstrable popularity is a successful argument against academic prejudice.

LESSONS OF HISTORY

Bill Horne

Part five: Big is beautiful?

Towards the end of the nineteen-fifties, it became apparent that there was a degree of mis-match between the demand and supply of computer equipment. Enormous machines were available at astronomical prices, but there was a strong requirement for smaller machines that would prepare invoices and carry out similar limited tasks.

The principal computer manufacturers argued that no computer could be economically viable unless it was very big and very fast. This was based on the fact that the central processor in a small system might spend much of its time idling while data was printed out or fresh input data was being prepared. Adding more store and input/output channels would allow the processor to be used more efficiently, increasing the potential value of the machine. That potential, however, could only be realised if there was enough work to keep the machine busy.

Asked about the smaller applications, the manufacturers suggested that they could be dealt with by using off-line terminals, not directly connected to the computer, to prepare the relevant data, which would then be sent to a computer bureau to be processed. Some users adopted this approach, and found it acceptable, but others saw it as too roundabout and open to error, or objected to revealing their business statistics to outsiders.

In any case, the 'off-line approach' seemed clumsy. A good deal of keyboard work was needed to produce the data files which were sent to the bureau, and in many cases the calculations that had to be done were relatively simple. It would have made sense to provide computing equipment directly connected to the keyboards and do the job on the spot. This would have been called 'on-line' working.

DISADVANTAGE

The difficulty about 'on-line' working was that it created a need for 'software' support that was not readily available. Each customer wanted something a little different, and that meant writing different programs. Whereas the bigger off-line bureaux could afford to maintain adequate support staff, the smaller companies had to make do with the people they could find, and that put them at a disadvantage.

One essential to the effective use of a large machine was the technique of 'time sharing'. This required that a computer be loaded with a number of tasks, each given a priority number. Work began on the first task, but if and when that work was interrupted by the input or output of data, which proceeded much more slowly than the internal computer processes, that particular task was suspended, and another one was begun. The input or output of data meanwhile proceeded under separate control.

When the first task was again ready to run at full speed, the second task was suspended, and the first resumed. In this way, the central processor was always kept busy, and the machine seemed to be executing several tasks at once. In fact, they were being performed 'concurrently', rather than simultaneously.

One consequence of this technique was that complex control programs or 'operating systems' had to be created to cope with the sequencing of the various tasks. In particular, the concept of 'interrupt' had to be introduced. This - like the telephone bell which interrupts an important technical discussion - demanded program service. The central processor was automatically switched to provide that service, all essential data regarding the interrupted program being stored for reference when that task was resumed.

The operating system gradually became a barrier between the user and the actual computer hardware. He no longer had direct contact with the circuits that worked for him. It was necessary to communicate via the operating system. This led to the development of two types of programmer, the systems programmer who created and maintained the operating system and the user programmer who used the operating system.

STORAGE LIMITS

The operating system naturally increased the amount of storage space and processing time used, and it was suggested that a time might come when there would be no room for the 'application programs' which carried out the individual tasks. The reply was that it would always be possible to add more storage to solve that problem, which was true in theory, but not entirely true in practice. There was a practical limit to the amount of store which could be addressed.

The eventual solution to this came from the concept of 'virtual memory'. In simple terms, a given area of storage could hold several different sets of data. If data that was not already in store was needed, it could be read in from 'backing store', which was commonly magnetic tape or disc. There was some delay while the data was copied, but the store space appeared to grow enormously.

Stage by stage, these ingenious methods were used to make a computer look much bigger and more capable than it really was. Each new idea enhanced overall performance and called for a slightly larger operating system. Where the user had once known exactly what his program was making the computer do, he was now working through an intermediary which imposed strict disciplines on him.

Now, it must be made clear that such a line of development was essential to the concept of the 'big machine', but it seems a little surprising that more work was not done to develop small machines that could be used more simply. Perhaps the uncertainty regarding imminent developments in component technology played a part, but it is more likely that the economic climate had a more significant effect. Some firms made half-hearted attempts to open up the small computer market, but few carried this through to success. Confidence in such projects was limited, and there was a definite tendency to prefer outdated methods, both in hardware and in software.

Decimalisation of the coinage was relevant. We have seen how conversion from sterling to binary and back had troubled the LEO project. That was a greater stumbling block in small machines, and when the problem was removed it was expected that more progress would be made.

One impediment to small machine development was the lack of suitable peripheral equipment. Despite attempts to automate typewriters, favoured because of their more elegant typefaces, the teleprinter was still the most obvious input/output device. It was noisy, slow, and limited in reliability, but it served its purpose. Some faster devices were produced, notably in Italy, but they were not satisfactory. The actual computing system might only cost £1000, but the essential accessories trebled this price.

THE SYSTEM FOUR

In the middle nineteen-sixties, it looked as if progress might be resumed, when the English Electric System Four was announced. This was to be developed by a team assembled by combining the English Electric, LEO and Marconi computer departments. These were still geopgraphically separated, English Electric at Kidsgrove, LEO at Park Royal, and Marconi at Chelmsford, but the effect of the separation was to be minimised by developing one machine at each site.

The System Four design was not new. It was based on the RCA Spectra 70 series, which had a reasonable compatibility with IBM software. The central processor system was up to date, using integrated circuits and multi-layer printed circuit boards, but some of the peripheral devices still used transistors and other discrete components.

The peripherals were an important part of the overall concept, which was intended to supply component parts from which a wide range of computer configurations could be constructed. This would remove the need to rely on outside suppliers of such equipment, it was hoped.

The hope was not really fulfilled. At least the mechanisms of peripheral devices had to be purchased, and all that could be done in-house was to package them in a style compatible with the rest of the system and supply the necessary control circuitry. This led to continuous liaison work with mechanism suppliers and a great deal of travelling.

Much could be said about the way the development progressed, but at least the two bigger machines, the 4-50 and the 4-70, reached maturity. So did the 4-40, which was not on the official schedules, but which offered a particularly good performance. The key peripheral, the operating console, was completed just in time, after three weeks of night and day shift work, and other devices staggered along in its train.

Progress was not helped by a number of movements of key staff from one site to another, and there was some argument between sites, but it was not until 1968 that the background to this emerged. First, English Electric took over Elliot Brothers, then they themselves were taken over by General Electric, and out of this emerged a new alignment, the key feature of which was International Computers Ltd. This took over the 'commercial' computer work, while other groups handled specialist computers.

It is salutary to note that the heart of ICL was formed from International Computers and Tabulators, a company not previously mentioned herein. They had come to computing via punched card equipment, as their name suggests, and they had built up a satisfactory business on the commercial side, based on contacts made through their earlier work.

The LEO team continued as an element of ICL for a while, and were then shut down and merged with ICT at Stevenage. The Kidsgrove group remained independent longer, but were eventually absorbed. That left Elliott Brothers working as GEC Computers Ltd., and various other scattered units doing specialised work.

The fundamental idea of the mergers was that the British computer industry could only hope to survive if all its potential was combined into one common effort. The echo of the Big Computer concept in this idea may be pure coincidence, but much the same people were responsible for both.

In practice, the idea did not work too well. Some of the great names of the early days could be found in the ICL management structure, but they were surrounded by forceful people from very different backgrounds, and their contributions were thereby diluted. ICL remains in existence, but presents a rather different image these days, of which more anon...

THE SEVENTIES

By 1970, computer work had been going on in Britain for 25 years, but it could scarcely be said to be prospering. It was noted that even big groups which included a source of computers seemed reluctant to buy British. American products were said to be better designed and better supported. They were certainly supported more strongly on the sales side, and the Americans knew more about the process of fighting for a contract.

In software, on the other hand, Britain held a strong advantage, the value of exports helping to counterbalance the expediture on hardware imports. Yet in other fields connected with electronics, Britain was doing well. What had gone wrong?

It would be reasonable to suggest that the key lay in economic uncertainities during the nineteen-sixties. these deterred enterprise of even a slightly risky kind, and led to premature cancellation of a number of interesting projects. Software work required less forward investment, and was able to carry on with less interruption. Nor was it threatened by possible future developments in the technical field, especially where it was of a theoretical nature, applicable to a number of different hardware configurations.

There is a vital lesson here for all branches of industry. When something goes amiss, or for that matter when something goes well, the reason must often be sought in the previous decade. Young graduates come into industry with little experience, but they will tend towards firms which show the signs of success. Five years later, if those signs have been confirmed, the same graduates will still be there, learning to make themselves useful. Another five years, and they will be ready to take over the reins of control. The quality of management available then will depend on the image presented by the company and its branch of industry ten years earlier.

In 1960, the computer world looked attractive. By 1965, it was a little blown-upon, and by 1970 it looked to be in chaos. This applied mainly to the hardware side of the matter, however, and the software side had looked reasonably promising throughout. During the subsequent ten years, the effect of all this was increasingly apparent, but it was too late then to put matters right.

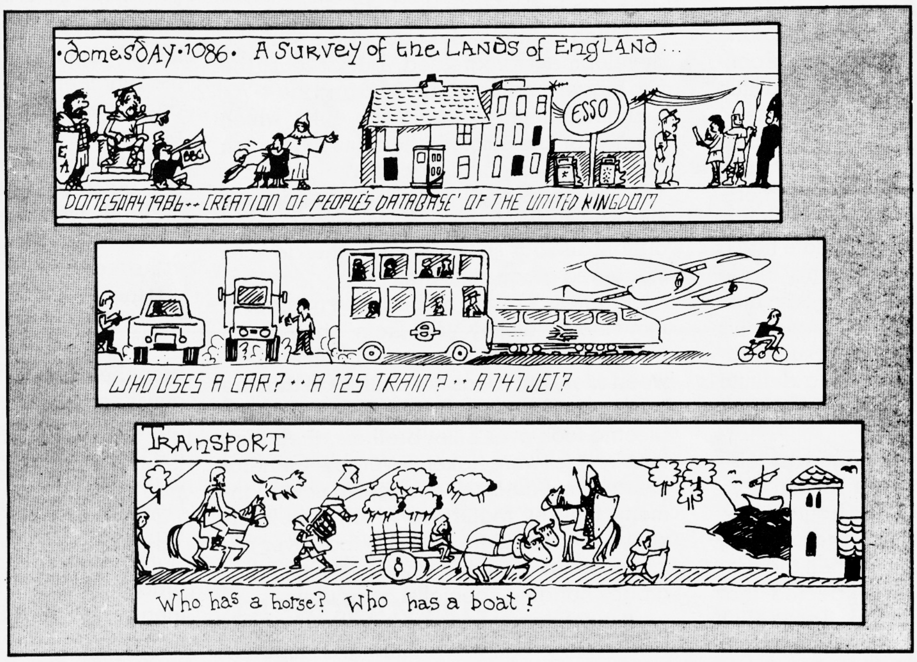
In any case, an era was ending. Hardware design was being taken over by integrated circuit manufacturers, who felt no need to respond too readily to any ideas put forward by their lesser customers. The engineer who produced a product on the basis of integrated circuits had to accept what he was offered and make the best of it. He was no longer in full control, and he needed to learn a completely new way of life.

Lessons of History continues in the next edition of Computing Today.

THE DOMESDAY PROJECT

Bill Horne

While essential restoration work is done to the original, 1086 edition of the Domesday book, children and teachers are busying themselves with a countrywide survey to celebrate the Domesday's anniversary.



Eight hundred and ninety-nine years ago, William the Norman saw a need to create a catalogue of the land which he had conquered, so that he would know the value of each piece of land, and be able to identifity its owner. Last November, another William - Bill Cotton, managing director of BBC Television - announced that a new Domesday book was to be created, and would become available in time for the ninth centenary of the original.

The word ' Domesday' is equivalent to Doom's Day, and originated from the fact that the contents of the book could have an influence on the fortunes of everyone mentioned in it. It is to be hoped that the new production will not be viewed in the same way in due course. The intention is that it should provide a source of information of a value to organisations and individuals, and it should not lead to anyone's doom unless the data is missed.

What will the new Domesday Book look like? It is intended that it should be made available on two videodiscs. The discs would contain, in addition to vast amounts of data, software which could be used to analyse and scan the data at will. The need for this can be seen when the amount of data involved is listed:

- 2,000,000 pages of text
- 20,000 maps
- 120,000 pictures

such figures make the mind boggle, Where will all the data come from? How can it be packed into two discs? How much will the equipment cost?

SOURCES

The data will initially be collected by schools and other local bodies, more than 12,000 schools being involved. The map of Britain has been divided into 3 x 4 km areas, and these have been 'farmed out' to the various collecting agencies.

One consequence of this is that the views expressed regarding a given area will be coloured by those who live in the area. This could be either a good thing or a bad thing. Local people know their area best (though that is not always true of Londoners!) but with so broad a base for data collection there would be many opportunities for political colouring to intrude. It is to be hoped that any such intrusions will be edited out.

The collection of information by schools is proceeding under the guidence of coordinators appointed by the local education authorities. Other sources include Scouts, Guides, Women's Institutes and the Nature Conservency Council.

EQUIPMENT

Existing videodisc equipment was unable to meet the required standards of packing density, so Phillips Electronics undertook the task of developing a new system. This alone is estimated to cost a million pounds. Should any limitation emerge in the course of development, the whole project might have to be re-thought, which illustrates the risks involved in such a complex matter.

In parallel with the videodisc development, Acorn Computers are developing the software needed to access and control the discs. Part of this work is involved with improved graphic techiques.

It will be necessary to use a computer in association with the videodisc, and here is an element of vagueness. It is first stated that the BBC Microcomputer will be ideal for the Domesday system, but it is then said that the Domeday system will be designed to be compatible with 32- bit processors. Perhaps these statements can be reconciled by remembering that a 16032 processor has been 'on the rocks' as a BBC Computer extension for some time past.

The original Domesday book was until recently stored in the Public Records Office in Chancery Lane, London. However, to celebrate its 900th anniversary the book has been transferred to the PRO in Kew, London, for rebinding and essential restoration work, and as a consequence it will not be on public display until April 1986.

THE DISCS

The first will be the 'Local' disc, covering the data collected on an area basis. Some 23,000 areas will be covered. There will be summaries of land usage, photographs and up to 20 pages of text for each area.

The National Disc, on the other hand, will be based on material selected under the guidance of an editorial board drawn from academic circles. It will deal with 'global' matters, such as natural recources, transport, population statistics, social activities, arts and crafts, education, and so on.

COST

While no selling is mentioned directly by the BBC, others have spoken of figures ranging from £1100 to £1500 for the discs and the equipment needed to view the data. In order to justify the original investment, it needed to view the data. In order to justify the original investment, it would be necessary to sell some tens of thousands of such systems. in view of the number of schools participating in collection, such a volume of sales should not be out of reach.

The original Domesday book was until recently stored in the Public Records Office in Chancery Lane, London. However, to celebrate its 900th anniversary the book has been transferred to the PRO in Kew, London for rebinding and essential restoration work, and as a consequence it will not be on public display until April 1986.

But that is not the end of the story. The hardware, and perhaps much of the software, will have other applications. Other countries might be interested to have their own Domesday compilations. A vast range of possibilities can be seen, but before they can be realised the project has to be brought to fruition.

CONCLUSION

It is always difficult to assess grandiose projects fairly. It is like watching an arrogant runner, and feeling that it would only be poetic justice if he stumbled. This is sheer jealousy. One would like to emulate the runner, or be a significant figure associated with the project.

In this case, one can only hope that all will go well. Quite apart from the direct benefits of the scheme, there would be an important spinoff in terms of familiarising people with the real capabilities of computers, and the possibility that the equipment would open up new capabiliteies. On the other hand, failure of the project in any marked degree would have very unfortunate consequences, since the credibility of big computer schemes would inevitably suffer.

As to the possibility that the information might be missed, those who might do so seem to manage quite well without a computerised Domesday Book to help them.

COMPUTING *Today*

NEXT ISSUE OUT FRIDAY AUGUST 9TH

SPEED - The speed and reliability of disk-based systems can be greatly - and all too innocently - diminished through bad programming practice. We offer practical advide on economic disk usage.

SUSPENSE STORY - lap timing, fuel control, and now suspension design. Computers have made great inroads in sport and particularly in motor racing. A recently-introduced system will now design you a complete suspension system tough enough for Formula One.

LANGUAGES - Metacomco Pascal for the Sinclair QL gets the *Computing Today* treatment next issue. We'll be carrying a complete, no-holds-barred review of Metacomco Pascal, so be sure of your copy of CT.

CODE MACHINE - Z80 assemblers for the Amstrad CPC464 are now appearing thick and fast. But Code Machine from the Picturesque stable is no *ordinary* assembler. Full review in the September issue.

BLOCKCODE - ever thought that your language needed tidying-up (your computer language, that is)? Contributor Dan Jonsson felt his language was missing something, so he proposed a new system of programming structures, BLOCKCODE, to make up for his system's deficiencies.

All this and more in the next edition of

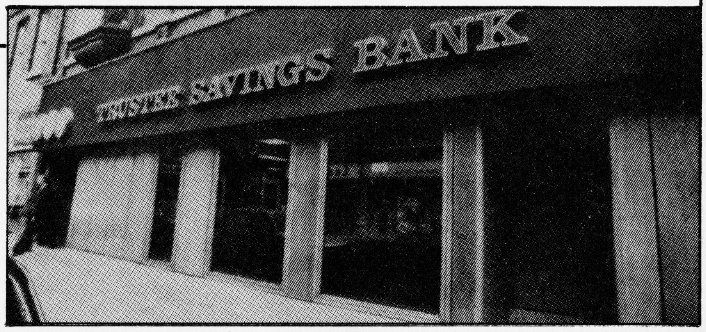
Computing Today

THE EVOLUTION CONTINUES..

MONEY MANAGER

Bill Horne

One possible use for a home micro is for recording financial transactions. However, any such program must be extremely user-friendly as the process of entering and analysing data can otherwise become extremely tedious. We examine here a low cost offering in this field for the Amstrad CPC464



Money Manager, produced by **Connect Systems,** is *almost* a Very good program. If that sounds ominous, bear in mind that we are looking for very high standards. We believe that we are sufficiently bright to be able to run any program successfully after reading the documentation through once. If we can't, we start to get critical. That is, perhaps, a grossly unfair attitude but it is likely to reflect the attitude of many buyers.

But let us go back to square one. Money Manager comes as a tape supported by a neat 12-page manual (A disc version is in the pipeline, but not yet available). The Manual lists fifteen options, each described in succinct terms, but it will be seen that a little more explanation might have been useful.

Loading the 11-block main program takes a while, which is where the disc version should score, and then you are instructed to load a data tape. at this stage, the only data tape available is the demo carried on the main tape, so you load that.

Unfortunately, we were using an early version, which had Hard read errors in both copies (*This has now been rectified - Ed).* The errors were in different blocks, so that little snag was overcome by the exercise of a little patience.

Having loaded the demo, and scanned the figures somewhat cynically - they reflect a life-style to which we are not very accustomed - we called on Option 15 to erase the data ready for a fresh start. It was a little disconcerting to find that the last balance of the demo was brought forward as a contribution to the figures which we entered.

The answer, we found - by making a telephone call - was that Option 12, which edits the account codes, also allows brought-forward figures to be adjusted. The manual says so, but we must have missed the relevant paragraph, not expecting to find the answer in that quarter...

ACTIVE MONTH

The system will store data for twelve months, only one month being 'active' at a time. The month in question is chosen by Option 9. In this option pressing return moves you forward one month at a time, while ']' selects the month displayed as active and exits the option. It is then possible to operate on the data for that month.

When all twelve months have been used, it is possible to re-identify the months to suit the following year.

Option 1 is for data entry. Each new line has a serial number, a date, an account number, an optional cheque number, a class code, a verbal description, a monetary amount and a mark code. This called for some hard thinking. The class codes for the demo tape were still set up, but they were not appropriate to the trial run which was planned, so a visit to Option 10 (Edit Class Codes) was necessary. Once you get the hang of this option it is quite easy to use, but you may have a bit of difficulty in deciding what codes to select. The answers become clearer after a little experimentation. a start can be made by deleting some of the demo codes, which survive total clearance. The same applies to the account codes edited by Option 12.

At this stage, you may be learning more about your own financial affairs than about the program, but there is no need to worry. If you change your mind about something there are provisions for editing almost everything. Option 2 edits a given entry, while option 3 will delete the complete entry.

SORTING

Should you make entries 'out-of-order' (it may be convenient to put income entries first, for example , followed by expenditure), Option 4 will sort the entries on a basis of date, class code, account code or mark, producing a neatly-ordered listing.

When the data has been entered and sorted, you can use option 5 to produce a statement for the active month or for all twelve months in turn. Alternatively, you can use option 6 to give an annual summary with rounded figures. You can create a bar chart or a pie chart based on selected items, but in the latter case beware of picking items which total zero, because the pie chart is then based on a very large number for the full diameter, and takes a very long time to draw.

TAPE STORAGE

Option 10 stores data on tape, with facilities for making a check on the result, but taped data can only be brought in as part of the initial load procedure, which is fair enough when you think about it. The only situation where this is slightly frustrating is when you have been working on the current data and want to check back on the data for the precious year, and have to reload the 11 blocks of the main program. To offset this, there is a 'find' option, which will locate a given string anywhere in the stored date.

COMMENT

After a certain amount of practice, a set of accounts covering one year was entered, and this showed that the program was entirely practicable, though some of the optional features were ignored. 'Mark', for example, was not used. It might have been useful for certain analytical processes, but it was not missed.

FACTSHEET: **Money Manager**

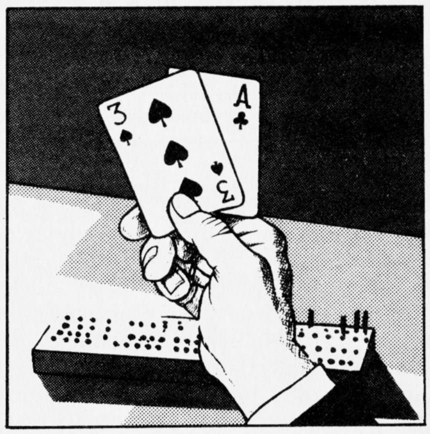
Price: £14.95

Available from: Connect Systems, 3 Flanchford Road, London W12 9ND.

CRIBBAGE PLAYER

Part Two: card dealing and display

Original Program: Bob Stafford
Adaptation by Computing Today staff



In the first part of this study, we reached a point where the title had been shown, initialisation had been performed, the ceremony of cutting for deal had been carried out, and the bare bones of the main display had been put up. We must now move on to the more complex functions.

The cards must now be dealt and displayed, the three routines involved being shown in Listings 5A and 5B.

We are going to be faced with the need to process three hands: the computer's hand, the player's hand and the crib hand. Some processes will apply to all three, so it will be convenient to introduce array HH(A,B,C). Subscript A defines the hand, being 1 for the player, 2 for the computer, and 3 for the crib hand. Subscript B indicates which card is involved, and has a maximum value of 6. Subscript C is 1 for the card denomination and 2 for the suit.

This array must be dimensioned for the BBC version, but the CPC464 will dimension it automatically, as none of the subscripts exceeds 10.

PROCF picks out the cards, giving six to the computer and six to the player. A thirteenth card is chosen for the turn-up, and this is registered in array TU, again an addition to the BBC dimensioning statements. As dealt, the cards may be in any order, but PROCG sorts them into value order. It will sort hand SK, containing J5 cards.

PROCI handles the display. The computer's cards and the turnup are shown face down, the player's cards are shown face up. Extensive use is made of the subroutines introduced for the cut procedure.

So far, everything has reasonably straightforward, but we now come to the actual play, beginning with the process of discard to the crib. The computer discards first, giving the player time to examine his cards and work out which he wants most.

CRIB DISCARD

Selecting cards to go into the crib is a tricky business. You want to keep cards that will give you a good score in the 'show' phase, but you need to be aware that a careless choice may make the crib valuable, while different cards may be useful in the 'lay down' phase. The computer's choice is simplified, to the extent that only the show phase scoring is considered. Since this phase produces the largest scores, the simplification is not unreasonable.

PROCJ handles the task, calling PROCL to calculate scores and PROCM to transfer cards to the crib hand.

The first step is to copy all six combinations of five cards from the six initially held into arrays HN,HS. Each combination is then scored by PROCL, the score being returned in HT. If this is greater than the previous highest score, shown in LS, LS=HT and SK=IJ, which identifies the card omitted in that combination. If none of the combinations score anything, SK=6 selects the last card. To assure the player that something is happening, the computer's cards in the display are 'flipped' in turn.

When all combinations have been examined, PROCM is called with F=1, to set up the first discard. The card removed is replaced by card 6.

The process is then repeated, taking all combinations of four cards from the remaining five. Card SK is passed to the second card position in the crib, and is overwritten by card 5 (If cards 5 or 6 are chosen, they may overwrite themselves, but that doesn't matter, because only cards 1 to 4 are used after the discard).

The two right-hand cards in the display are erased, and the message 'I'm choosing cards..." is erased.

SCORING IN SHOW

It is a little unfair that the computer can use PROCL to calculate scores, because the player has to do it in his head, and that is not too easy. The scoring basis in the 'show' phase is as follows;

- If all four cards in hand are of the same suit, the score is 4, but if the fifth card is of the same suit as well the score is 5. In the actual show phase, the fifth card is the turnup, but that does not apply during crib discard.

- If three or more cards form a run, the score is the number of cards in the run. However, once a run of given length has been scored, parts of the run are not scored again. Otherwise, a run of four, say, would also score two runs of three.

- For any group of cards with values totalling to fifteen, two points are scored.

- For any two cards of equal denomination, two points are scored.

There is an additional score, not relevant in discard, which is 'one for his Nob'. This applies if the Jack of the turn-up suit is held. It is scored by ZJ, which is zeroed in the discard routine before PROCL is called.

The routine first looks for a flush. The method may look a trifle odd, but it duly sets ZF, the flush score.

The cards are then sorted into denomination order, and if the difference between all adjacent pairs is 1, ZR = 5 to score a run of five. If all five cards total fifteen, ZQ=2.

All combinations of four cards are then checked for runs and fifteens. The run score is conditional on ZR being 0, but ZZ can accumulate more than one score of 4, which is passed to ZR when the combinations have all been checked.

The process is repeated for all combinations of three cards, runs and fifteens again being scored, and finally the pairs are examined, equal cards adding 2 to ZP, and totals of 15 adding 2 to ZQ.

In the original program, Bob Stafford used machine code routines to speed up the scoring process, but increased speed over the past five years makes the BASIC routines fast enough, especially as the player is likely to be spending the time mulling over his own discards.

PLAYER DISCARDS

The discards chosen by the player are processed by PROCK (listings 9A, 9B). First, the array PC$ is set up with six two-character strings representing the cards which the player holds. Then a two-iteration FOR-NEXT loop is entered to select the third and fourth crib cards.

Here we face a minor problem. The cards chosen for discard are indicated by input of two-character strings, such as TD for the ten of diamonds. If we use INPUT, Enter has to be pressed as well, but if we don't use input there is a risk of error, due to entry of the first character before the system is ready, which means that the second is interpreted as the first. For the BBC version, we have used the traditional INPUT command, but for the CPC464 we offer an alternative which is not fool proof but works most of the time. It relies on the player not fiddling with the keys while awaiting his turn, other than to put in a valid two-character string.

When the string has been input, PROCN checks to see whether it is duplicated in PC$, returning EF=0 if the entry is valid, and erasing the relevant entry from PC. (We left that out originally, and found that someone cheated by discarding the same card twice!)

If the card is not valid, a reproving message is put up and the routine loops for another try. PROCM otherwise transfers the card to the crib.

TURN UP

The discards being complete, the turn-up card can be shown, but this brings us into the realm of scoring, because if the card is a Jack the dealer scores 2. This has to be marked up on the scoreboard, which is no simple matter.

The question also arises as to whether it is sensible to go by the academic book and put

PROCEDURES

PROCF	2500	Deal
PROCG	2600	Sort hand
(PROCH)	2100	Character display
PROCI	2700	Display cards
PROCJ	2800	Computer discard
PROCK	3500	Player discard
PROCL	3100	Show scoring
PROCLL	5700	Jack scoring
PROCM	3400	Cards to crib
PROCN	3800	Validity check
(PROCO)	3700	Input
PROCP	3900	Player update
PROCQ	4100	Turnup score
PROCQA	4200	Turnup
PROCR	4300	Scoreboard
PROCRA	4400	Scoreboard
PROCRB	4500	Scoreboard
PROCRC	4600	Scoreboard
PROCRD	4700	Scoreboard
PROCRE	4800	Scoreboard

Listing 5a Dealing

```
2500 DD=53:FOR A=1 TO 52:CC(A)=A:NEXT
2510 FOR A=1 TO 2:FOR B=1 TO 6:GOSUB 1700
2520 HH(A,B,1)=NN:HH(A,B,2)=SS:NEXT:NEXT
2530 GOSUB 1700:TU(1)=NN:TU(2)=SS
2540 RETURN
2600 FOR A=1 TO J5-1:FOR B=A+1 TO J5
2610 IF HH(SK,A,1)<=HH(SK,B,1) THEN 2640
2620 F=HH(SK,A,1):HH(SK,A,1)=HH(SK,B,1):HH(SK,B,1)=F
2630 F=HH(SK,A,2):HH(SK,A,2)=HH(SK,B,2):HH(SK,B,2)=F
2640 NEXT:NEXT:RETURN
2700 FOR N=16 TO 21:GOSUB 1800:GOSUB 1900:NEXT
2710 FOR N=9 TO 14:GOSUB 1800
2720 SUIT=HH(1,N-8,2):DEN=HH(1,N-8,1):GOSUB 2100
2730 NEXT
2740 N=15:GOSUB 1800:GOSUB 1900
2750 RETURN
```

Listing 6a: Computer discards

```
2800 PRINT#1," I'm putting cards in the crib"
2810 LS=0:SK=0:ZJ=0:HP=2
2820 FOR IJ=1 TO 6:C=0:FOR K=1 TO 6
2830 IF K=IJ THEN 2850
2840 C=C+1:HN(C)=HH(2,K,1):HS(C)=HH(2,K,2)
2850 NEXT:GOSUB 3100
2860 IF HT>LS THEN LS=HT:SK=IJ
2870 N=IJ+15:GOSUB 2000:GOSUB 1900:GOSUB 1900
2880 NEXT
2890 IF SK=0 THEN SK=6
2900 F=1:GOSUB 3400:SK=0
2910 LS=0:FOR IJ=1 TO 5:C=0:FOR K=1 TO 5
2920 IF K=IJ THEN 2940
2930 C=C+1:HN(C)=HH(2,K,1):HS(C)=HH(2,K,2)
2940 NEXT:GOSUB 3100
2950 IF HT>LS THEN LS=HT:SK=IJ
2960 NEXT:IF SK=0 THEN SK=5
2970 F=2:GOSUB 3400
2980 N=21:GOSUB 2000:N=20:GOSUB 2000
2990 CLS #1
3000 RETURN
```

Listing 7a: Scoring

```
3100 ZF=0:RT=0:FOR A=1 TO 4:RT=RT+(HS(A)=HS(1)):NEXT
3110 IF RT=-4 THEN ZF=4-(HS(5)=HS(1))
3120 FOR A=1 TO 4:FOR B=A+1 TO 5:IF HN(A)<HN(B) THEN 3140
3130 F=HN(A):HN(A)=HN(B):HN(B)=F
3140 NEXT:NEXT
3150 ZP=0:ZQ=0:ZR=0
3160 IF (HN(2)-HN(1)=1)AND(HN(3)-HN(2)=1)AND(HN(4)-HN(3)=1)AND(HN(5)-HN(4)=
1)THEN ZR=5
3170 IF VN(HN(1))+VN(HN(2))+VN(HN(3))+VN(HN(4))+VN(HN(5))=15 THEN ZQ=2
3180 ZZ=0:FOR A=1 TO 2:FOR B=A+1 TO 3:FOR C=B+1 TO 4:FOR D=C+1 TO 5
3190 IF (HN(B)-HN(A)=1)AND(HN(C)-HN(B)=1)AND(HN(D)-HN(C)=1)AND(ZR=0) THEN ZZ
=ZZ+4
3200 IF VN(HN(A))+VN(HN(B))+VN(HN(C))+VN(HN(D))=15 THEN ZQ=ZQ+2
3210 NEXT:NEXT:NEXT:NEXT
3220 ZR=ZR+ZZ:ZZ=0
```

everything into procedures or subroutines.

There are arguments for and against this. There is something very impressive about a program in which a ten-line main routine calls five hundred lines in subroutines, but such a program can be quite difficult to read.

The turn up process is perhaps a marginal case, in that it is quite short, is called from one point only, and is not linked with other action. However, we have taken it out of its place in the original main program and turned it into PROCQ, with PROCQA as a subsidiary. PROCQ calls PROCQA to display the turn up card, and if TU(1) = 11, the turn up card is flipped four times to draw attention, and a score message is output. PROCR is then called with SP=DL (score to dealer) and SC=2 (score is two) to update the scoreboard.

PROCQ and PROCQA are given in listings 10A, 10B.

SCORING

We must here turn aside from the mainstream of action to consider the way to handle the scoreboard. This is covered by the PROCR family of routines, given in Listings 11A,11B.

As indicated above, the parameters SP (scorer) and SC (score) are passed to PROCR. SC is added to SC (SP). The numeric display of the score for SP is updated, and if the result exceeds 120 the WIN flag is set. Remember that this can happen whenever PROCR is called, but PROCR itself cannot conveniently respond to the flag being set.

SC is now used in a different role, which is a little naughty but perfectly practical in this case. If the total score for player SP exceeds 60, SC = SC(SP) - 60, since we are now on the second circuit of the 60 holes. The subsidiary PROCRA or PROCRB is then called, according to which player's score is being updated.

PROCRA has to keep track of the current score and the last score, updating both whenever it is called. Initially, B1 = 0, so F3 is zeroed, and the first line takes no further action. For subsequent calls, the line erases the trailing peg. The next line updates B1 from F1 and F1 from SC, while F3=SC. The new leading peg is then entered. The actual entry of the peg or hole is handled by PROCRE which in turn uses PROCRD. Note the need to allow for the grouping of the peg positions in sets of five.

For the other player, PROCRB, PROCRC, and PROCRD are used in the same way.

This is the routine which would become rather more difficult if an attempt was made to fit the scoreboard into a 32-column screen, with hole positions determined by graphics coordinates. However, you are welcome to try, if you want to...

SUMMARY

It may seem that progress is slow, but we have now defined a number of the key routines, and it is a convenient point at which to make a break.

```
3230 FOR A=1 TO 3:FOR B=A+1 TO 4:FOR C=B+1 TO 5
3240 IF (HN(B)-HN(A)=1)AND(HN(C)-HN(B)=1)AND ZR=0 THEN ZZ=ZZ+3
3250 IF VN(HN(A))+VN(HN(B))+VN(HN(C))=15 THEN Z0=Z0+2
3260 NEXT:NEXT:NEXT
3270 ZR=ZR+ZZ
3280 FOR A=1 TO 4:FOR B=A+1 TO 5
3290 IF VN(HN(A))+VN(HN(B))=15 THEN Z0=Z0+2
3300 IF HN(A)=HN(B) THEN ZP=ZP+2
3310 NEXT:NEXT
3320 HT=ZP+Z0+ZR+ZF+ZJ
3330 RETURN
```

Listing 8a: Cards to crib

```
3400 HH(3,F,1)=HH(2,SK,1)
3410 HH(3,F,2)=HH(2,SK,2)
3420 HH(2,SK,1)=HH(2,7-F,1)
3430 HH(2,SK,2)=HH(2,7-F,2)
3440 RETURN
```

Listing 9a: Player's discards

```
3500 FOR J=1 TO 6:PC$(J)=N$(HH(1,J,1))+S$(HH(1,J,2)):NEXT
3510 FOR OP=3 TO 4:IF OP=3 THEN O$="first" ELSE O$="second"
3520 PRINT #1," Your ";O$;" discard to the crib."
3530 GOSUB 3700:A$=UPPER$(A$):CLS #1
3540 GOSUB 3800:IF EF=0 THEN 3570
3550 PRINT#1,"You do not hold card ";A$
3560 M=150:GOSUB 1100:CLS #1:GOTO 3520
3570 HH(3,OP,1)=HH(1,K,1)
3580 HH(3,OP,2)=HH(1,K,2)
3590 GOSUB 3900:NEXT
3600 RETURN

3700 A$=INKEY$:IF A$=""THEN 3700
3710 B$=INKEY$:IF B$=""THEN 3710
3720 A$=A$+B$:RETURN

3800 EF=1:FOR J2=1 TO C1
3810 IF A$=PC$(J2) THEN K=J2:EF=0
3820 NEXT:RETURN

3900 AA=HH(1,K,1):BB=HH(1,K,2)
3910 N=C1+8:GOSUB 2000
3920 IF K=C1 THEN 3990
3930 FOR C=K TO C1-1
3940 HH(1,C,1)=HH(1,C+1,1):HH(1,C,2)=HH(1,C+1,2)
3950 N=C+8:GOSUB 1800
3960 SUIT=HH(1,C,2):DEN=HH(1,C,1):GOSUB 2100
3970 PC$(C)=PC$(C+1):NEXT
3980 HH(1,C,1)=AA:HH(1,C,2)=BB:PC$(C1)=""
3990 C1=C1-1:RETURN
```

Listing 10a: Turn up

```
4100 GOSUB 4200
4110 IF TU(1)<>11 THEN RETURN
4120 FOR IJ=1 TO 4:GOSUB 4200:M=100:GOSUB 1100:NEXT
4130 SP=DL:SC=2
4140 PRINT#1," Jack turned up. ";W$(DL);" score 2 as dealer."
4150 m=200:GOSUB 1100:CLS #1
4160 GOSUB 4300
4170 RETURN

4200 N=15:GOSUB 2000:GOSUB 1800
4210 SUIT=TU(2):DEN=TU(1):GOSUB 2100
4220 RETURN
```

Listing 11a: Scoreboard

```
4300 SC(SP)=SC(SP)+SC
4310 IF SP=1 THEN Y=9 ELSE Y=1
4320 LOCATE 15,Y:PRINT STR$(SC(SP))
4330 IF SC(SP)>120 THEN SC=61:WIN=1:GOTO 4350
4340 SC=SC(SP):IF SC>60 THEN SC=SC-60
4350 IF SP=1 THEN GOSUB 4400 ELSE GOSUB 4500
4360 RETURN

4400 F3=B2:IF F3>0 THEN M1$=CHR$(144):M2$=CHR$(231):GOSUB 4800:GOSUB 4700
4410 B2=F2:F2=SC:F3=SC:M1$=CHR$(231):M2$=CHR$(144)
4420 GOSUB 4800:GOSUB 4700:RETURN

4500 F3=B1:IF F3>0 THEN M1$=CHR$(144):M2$=CHR$(231):GOSUB 4600:GOSUB 4700
4510 B1=F1:F1=SC:F3=SC:M1$=CHR$(231):M2$=CHR$(144)
4520 GOSUB 4600:GOSUB 4700:RETURN

4600 IF F3<31 THEN Y=3:X=F3+2+(F3-1)\5 ELSE Y=4:X=68-F3-(F3-31)\5
4610 RETURN

4700 FOR K=1 TO 3:LOCATE X,Y:PRINT M2$
4710 M=30:GOSUB 1100
4720 LOCATE X,Y:PRINT M1$
4730 GOSUB 1100
4740 NEXT:RETURN

4800 IF F3<31 THEN Y=7:X=F3+2+(F3-1)\5 ELSE Y=6:X=68-F3-(F3-31)\5
4810 RETURN
```

BBC Listings

Listing 5b: Dealing

```
2600 DEFPROCF
2610 DD=53:FOR A=1 TO 52:CC(A):NEXT
2620 FOR A=1 TO 2:FOR B=1 TO 6:PROCA
2630 HN(A,B,1)=NN:HH(A,B,2)=SS:NEXT:NEXT
2640 PROCA:TU(1)=NN:TU(2)=SS
2650 ENDPROC


2700 DEFPROCG
2710 FOR A=1 TO J5-1:FOR B=A+1 TO J5
2720 IF HH(SK,A,1)<=HH(SK,B,1) THEN 2750
2730 F=HH(SK,A,1):HH(SK,A,1)=HH(SK,B,1):HH(SK,B,1)=F
2740 F=HH(SK,A,2):HH(SK,A,2)=HH(SK,B,2):HH(SK,B,2)=F
2750 NEXT:NEXT:ENDPROC

2800 DEFPROCI
2810 FOR G=16 TO 21:PROCCB(G):PROCCC(G):NEXT
2820 FOR G=9 TO 14:PROCCB(G)
2830 PRINT TAB(CP(G)MOD 40+2,CP(G) DIV 40+3);SG$(HH(1,G-8,2))
2840 PRINT TAB(CP(G)MOD 40+2,CP(G) DIV 40+1;N$(HH(1,G-8,1))
2850 NEXT
2860 PROCCB(15):PROCCC(15)
2870 ENDPROC
1230 DIM HH(3,6,2),TU(2),PC$(6)   (Additional arrays)
```

Listing 6b: Computer discards

```
2900 DEFPROCJ
2910 PRINT TAB(2,26);"I'm choosing cards to put in the crib."
2920 LS=0:SK=0:ZJ=0:HP=2
2930 FOR IJ=1 TO 6:C=0:FOR K=1 TO 6
2940 IF K=IJ THEN 2960
2950 C=C+1:HN(C)=HH(2,K,1):HS(C)=HH(2,K,2)
2960 NEXT K:PROCL
2970 IF HT>LS THEN LS=HT:SK=IJ
2980 Q=IJ+J5:PROCCD(Q):PROCCB(Q):PROCC(Q):PROCDEL(100):NEXT IJ
2990 IF SK=0 THEN SK=6
3000 F=1:PROCM:SK=0
3010 LS=0:FFR IJ=1 TO 5:C=0:FOR K=1 TO 5
3020 IF K=IJ THEN 3040
3030 C=C+1:HN(C)=HH(2,K,1):HS(C)=HH(2,K,2)
3040 NEXT K:PROCL
3050 IF HT>LS THEN LS=HT:SK=IJ
3060 NEXT IJ:IF SK=0 THEN SK=5
3070 F=2:PROCM
3080 PROCD(21):PROCCD(20)
3090 PRINT TAB(0,26);SPC(80)
3100 ENDPROC
```

Listing 7b: Scoring

```
3200 DEFPROCL
3210 ZF=0:RT=0:FOR A=1 TO 4:RT=RT+(HS(A)=HS(1)):NEXT
3220 IF RT=-4 THEN ZF=4-(HS(5)=HS(1))
3230 FOR A=1 TO 4:FOR B=A+1 TO 5:IF HN(A)<=HN(B) THEN 3250
3240 F=HN(A):HN(A)=HN(B):HN(B)=F
3250 NEXT:NEXT
3260 ZP=0:ZQ=0:ZR=0
3270 IF (HN(2)-HN(1)=1)AND(HN(3)-HN(2)=1)AND HN(4)-HN(4)-HN(3)=1)AND
HN(5)-HN(4)=1) THEN ZR=5
3280 IF VN(HN(1))+VN(HN(2))+VN(HN(3))+VN(HN(4))+VN(HN(5))=15 THEN ZQ=2
3290 ZZ=0:FOR A=1 TO 2:FOR B=A+1 TO 3:FOR C=B+1 TO 4:FOR D=C+1 TO 5
3300 IF (HN(B)-HN(A)=1) AND (HN(C)-HN(B)=1) AND (HN(D)=HN(C)=1) AND
ZR=0 THEN ZZ=ZZ+4
3310 IF VN(HN(A))+VN(HN(B))+VN(HN(C))+VN(HN(D))=15 THEN ZQ=ZQ+2
3320 NEXT:NEXT:NEXT:NEXT
3330 ZR=ZR+ZZ:ZZ=0
3340 FOR A=1 TO 3:FOR B=A+1 TO 4:FOR C=B+1 TO 5
3350 IF (HN(B)-HN(A)=1) AND (HN(C)-HN(B)=1) AND ZR=0 THEN ZZ=ZZ+3
3360 IF VN(HN(A))+VN(HN(B))+VN(HN(C))=15 THEN ZQ=ZQ+2
3370 NEXT:NEXT:NEXT
3380 ZR=ZR+ZZ
3390 FIR A= 1 TO 4:FOR B=A+1 TO 5
3400 IF VN(HN(A))+VN(HN(B))=15 THEN ZQ=ZQ+2
3410 IF HN(A)=HN(B) THEN ZP=ZP+2
3420 NEXT:NEXT
3430 HT=ZP+ZQ+ZR+ZF+ZJ
3430 ENDPROC
```

Listing 8b: Cards to crib

```
3500 DEFPROCM
3510 HH(3,F,1)=HH(2,SK,1)
```

```
3520 HH(3,F,2)=HH(2,SK,2)
2530 HH(2,SK,1)=HH(2.7-F,1)
3540 HH(2,SK,2=HH(2.7-F,2)
3550 ENDPROC
```

Listing 9b: Player discards

```
3600 DEFPROCK
3610 FOR J=1 TO 6:PC$(J)=N$(HH(1,J,1))+S$(HH(1,J,2)):NEXT
3620 FOR QP=3 TO 4:IF QP=3 THEN Q$="first" ELSE Q$="second"
3630 PRINT TAB(2,26);"What is the ";Q$;" card you wish to put"'"in the
crib?"
3640 INPUT A$:PRINT TAB(0,28);SPC(80)
3650 PROCN:IF EF=0 THEN 3680
3660 PRINT TAB(2,26);"You do not hold card ";A$
3670 PROCDEL(250):PRINT TAB(0,26);SPC(80):GOTO 3630
3680 HH(3,QP,1)=HH(1,K,1)
3690 HH(3,QP,2)=HH(1,K,2)
3700 PROCP:NEXT
3710 ENDPROC


3800 DEFPROCN
3810 EF=1:FOR J2=1 TO C1
3820 IS A$=PC$(J2) THEN K=J2:EF=0
3830 NEXT:ENDPROC


3900 DEFPROCP
3910 AA=HH(1,K,1=:BB=HH(1,K,2)
3920 PROCCD(C1+8)
3930 IF K=C1 THEN 3990
3940 FOR C=K TO C1-1:HH(1,C,1)=HH(1,C+1,1):HH(1,C,2)=HH(1,C+1,2)
3950 PRINT TAB(CP(C+8) MOD 40 +2,CP(C+8) DIV 40 +3);SG$(HH(1,C,2))
3960 PRINT TAB(CP(C+9) MOD 40 + 1,CP(C+8) DIV 40 +1);N$(HH(1,C,1))
3970 PC$(C)=PC$(C+1):NEXT
3980 HH(1,C1,1)=AA:HH(1,C1,2)=BB:PC$(C1)=""
3990 C1=C1-1:ENDPROC
```

Listing 10b: Turn up

```
4100 DEFPROCQ
4110 PROCQA
4120 IF TU(1) <> THEN 4190
4130 FOR IJ=1 TO 4:PROCQA:PROCDEL(100):NEXT
4140 SP=DL:SC=2
4150 PRINT TAB(2,26);"Jack turned up.";W$(DL);" score 2 as dealer."
4160 PROCDEL(250):PRINT TAB(0,26);SPC(80)
4180 PROCR:PROCDEL(250)
4190 ENDPROC
4200 DEFPROCQA
4210 PROCCD(15):PROCCB(15)
4220 PRINT TAB (CP(15) MOD 40+2 ,CP(15)DIV 40+3);SG$(TU(2))
4230 PRINT TAB(CP(15) MOD 40 +1,CP(15) DIV 40+1);N$(TU(1))
4240 ENDPROC
```

Listing 11b: Scoreboard

```
4300DEFPROCR
4310SC(SP)=SC(SP)+SC
4320IF S=2 THEN PP=1 ELSE PP=9
4330PRINT TAB(15,PP);STR$(SC(SP))
4340IF SC(SP)>120 THEN WIN=1:GOTO 4370
4350SC=SC(SP):IF SC>60 THEN SC=SC-60
4360IF SP=1 THEN PROCRA ELSE PROCRB
4370ENDPROC
4400DEFPROCRA
4410F3=B2:IF F3<>0 THEN M1$=CHR$(224):M2$=CHR$(225):PROCRE:PROCRD
4420B2=F2:F2=SC:F3=SC:M1$=CHR$(225):M2$=CHR$(224)
4430PROCRE:PROCRD:ENDPROC
4500DEFPROCRB
4510F3=B1:IF F3<>0 THEN M1$=CHR$(224):M2$=CHR$(225):PROCRC:PROCRD
4520B1=F1:F1=SC:F3+SC:M1$=CHR$(225):M2$=CHR$(224)
4530PROCRC:PROCRD:ENDPROC
4600DEFPROCRC
4610IF F3<31 THEN PP=122+F3+(F3-1)DIV 5 ELSE PP=228-F3-(F3-31)DIV 5
4620ENDPROC
4700DEFPROCRD
4710FOR K=1 TO 3:PRINT TAB(PP MOD 40,PP DIV 40);M2$
4720PROCDEL(20):PRINT TAB(PP MOD 40,PP DIV 40);M1$
4730PROCDEL(20):NEXT
4740ENDPROC
4800DEFPROCRE
4810IF F3<31 THEM PP=282+F3+(F3-1)DIV 5 ELSE PP=308-F3-(F3-31) DIV 5
4820ENDPROC
```

THE MALTRON KEYBOARD

Owen and Audrey Bishop

There aren't many alternatives to the QWERTY keyboard although the DEVORAK keyboard is now being installed into some government departments in the USA . However, the MALTRON keyboard is claimed to be safer and more efficient than either the QWERTY or the DEVORAK keybaords, so we put it to the test.

Most readers will already have seen photographs of the computer keyboard that looks as if it comes from the world of Salvador Dali. From the photographs, it might be thought that it has an unnerving jelly-like texture (world of Sinclair?), but this is not so. Its white plastic case is rigid and robust, well able to stand up to the rigours of word-processing and other high-intensity usage. The model supplied for review was the new IBM PC version. We tested it on a Zenith Z-150, an IBM PC compatible machine.

Versions for the IBM PC and BBC micros differ very slightly. Both versions are available ready to plug-in, but the model for the BBC requires special operating software supplied on ROM. It is ready for use immediately without any adaptatation of the machine or additional software. If you are buying a system from scratch, there is no need to buy the IBM (or IBM compatible). Both the Maltron keyboard *instead.*

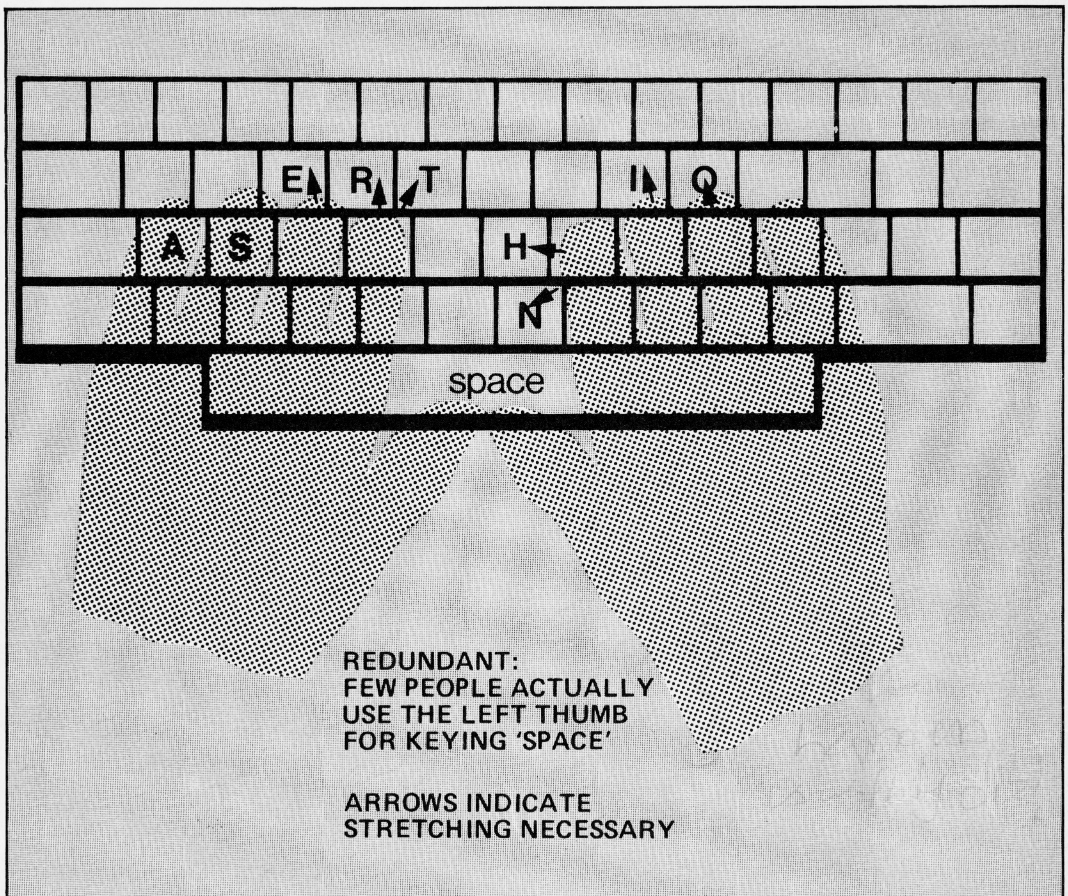


Figure 1: QWERTY keyboard.

ERGONOMICS

According to *The Concise Oxford Dictionary,* ergonomics is the study of the efficiency of persons in their working environment. A device which is described as being 'ergonomic' is thus one which has been designed for efficient use by *people.* The designer must take into account those features of the human body and mind which relate to the operation of the machine. Muscle action, eyesight, tension (mental and physical) and fatigue must all be considered in bringing the machine into harmony with its human operator. Only then can the maximum efficiency of the human/machine combination be realised.

The early typewriters were far from ergonomic. The designers were constrained by the fact that the keyboard had to be linked mechanically to the type heads. Staggered rows of keys were the only practicable solution. Worse still, the early mechanisms acted more slowly than the typist! The allocation of letters were scattered all over the keyboard to make them relatively inaccessible and slow to key (Fig 1). Nowadays, the keyboard is linked electronically to the computer (or electronic typewriter). No longer are there constraints on how the keys may be arranged, and we can be certain that the computer can easily keep pace with the speediest operator. Designers are free to produce the ergonomic keyboard at last.

The designers of the Maltron keyboard have aimed for greatest ease of use with the minimum of effort, reduction in errors, increased keyboarding speed and the elimination or reduction of keyboard-associated diseases. Diseases that can be caused by non-ergonomic keyboards include repetitive strain injury (in the form of Tenosynovitis and Writer's Cramp) and persistent, often painful, aches of the shoulders, arms and back.

The makers of the Maltron keyboard claim that theirs is an effective ergonomic design. It is not practicable to run an extensive field trial to support this claim, but evidence is mounting that the new keyboard does indeed do what is claimed. People are finding that the keyboard is easier to operate, that they make fewer keying mistakes and achieve higher speeds. It has been estimated that the keyboard can produce a 50% improvement in productivity because fewer keying mistakes are made, fatigue is reduced and higher keying speeds are reached. Many say it is easier to learn to type on the new keyboard than on the old QWERTY version. Touch-typists experienced on the QWERTY keyboard have had no problems in converting to the Maltron keyboard. Their fears that they might then lose their proficiency on the QWERTY keyboard have turned out to be entirely unfounded.

THE PRINCIPLES

These may be discussed under two main headings. First of all, let us consider the general shape of the keyboard. This has two main areas for alphanumeric and symbol keys. These are on the extreme left and right of the console. The hands may be comfortably placed over these separated areas. In the QWERTY keyboard, the alphanumeric keys are all in the centre of the keyboard; the hands must be placed close together, necessitating unnatural bending at the wrists. The key areas are concave, the concavity being shaped to take account of the relative lengths of the four fingers and to allow the fingers to sweep up and down the unstaggered columns of keys without having to move the hand itself. The thumbs each have their own key-pad, consisting of 8 keys, though only one key in each pad ('E' and SPACE) is in frequent use. With the arrangement described above, the operator is able to reach all keys with the minimum of stretching.

The Maltron keyboard console is deeper than that of the IBM and many other micros. The manufacturers suggest that, for maximal ergonomic effect, it is preferable for the work-surface to be 24-in from the floor instead of the standard 26-in of most micro benches. We tried it at both heights and found that the recommended height was marginally preferable. However, it would be expensive to alter or replace office furniture and the slight difference in operating ease would hardly make this worth while.

The second aspect of design is the allocation of characters and functions to the keys. The keyboard has two configurations, and the operator can select one or the other by pressing a single QWERTY/MALTRON key at the top of the keyboard. The QWERTY arrangement is retained for use by operators trained in QWERTY who do not wish to learn the new system. But to stay with QWERTY is to lose the substantial benefits of the Maltron layout. The keys are marked in white with the Maltron designations, the QWERTY designations being much smaller and red.

The most frequently used character (17% of key-presses) is the space. This is keyed by the right thumb (Fig 2). The most commonly occurring letters in English text are ETAONIRSH, in that order. As mentioned above, 'E' (11% of key-presses) is keyed by the left thumb. The other 8 letters come directly under the finger-tips. Thus, most of the letters occurring in English text may be keyed without moving the fingers or thumbs from their 'home' keys. By contrast, only 3 of our 10 digits rest on commonly-used keys on a QWERTY keyboard (space, 'A' and 'S', which rank 1, 4 and 9 in frequency of use).

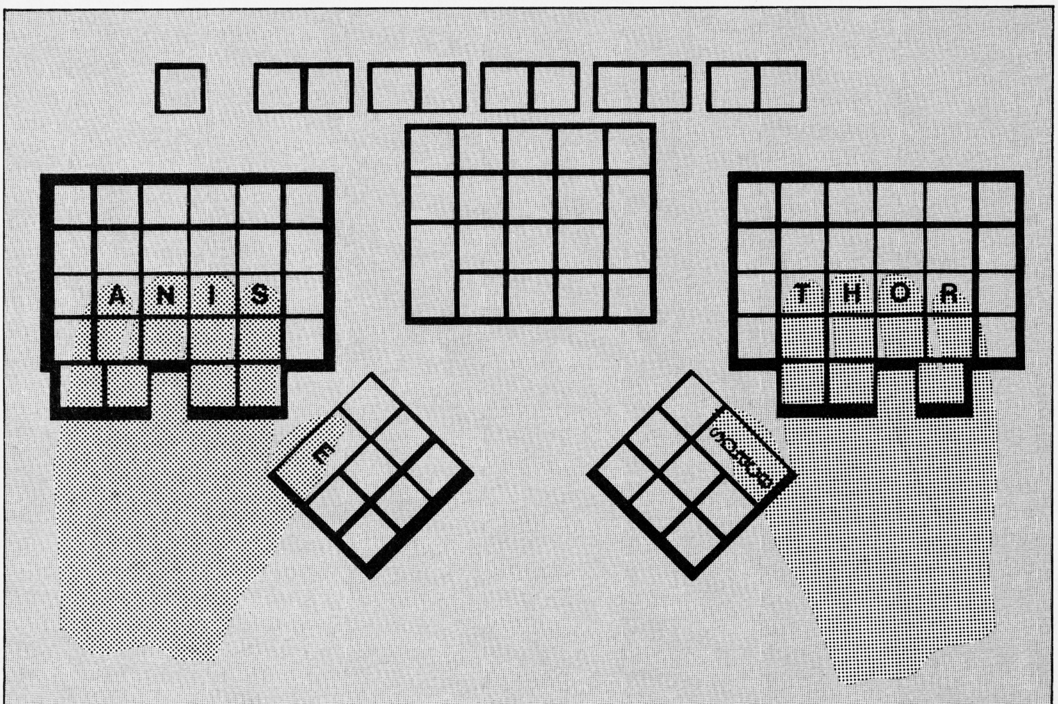


Figure 2: the MALTRON layout.

The placing of the control keys is sensible. All the 'lock' keys are at the left side of the keyboard, with an LED beside each. Other function keys, including the cursor keys, are in the pads controlled by the thumbs. the backspace key is very conveniently placed where it can be operated by the left thumb, allowing almost instantaneous correction of keying errors. The 10 function keys are arranged in a row across the top of the keyboard, so that the position of each relates clearly to the function legends displayed at the bottom of the screen. The arrangement of keys seems to be as perfect as possible and easy to learn.

Situated between the left and right groups of alphanumeric keys is the editing/numeric keypad, similar in layout to the keypad region at the right-hand end of the IBM keyboard. Placing this pad in the centre has the advantage that it may be used equally well by left-handed or right-handed operators.

TRAINING

The keyboard is accompanied by a training manual and a set of instructional audio tapes. Three courses are available: a short course and a longer course for executives, and a course for professional keyboard operators. We were supplied with the executives courses. The manuals include instructions on posture and fingering, and a graded sequence of keying exercises. These would take the executive to a speed of 25-35 wpm (short course) or to 30-50 wpm (long course), speeds which are fully adequate for most purposes. The audio tapes are intended to help the user progress step-by-step through the manual. We were surprised when we ran the first tape. After a short musical excerpt from the works of J.S. Bach, the voice of the instructor commenced in strong American accents. Maybe this tape was intended for the US market. We found the accent too foreign and the manner patronising. Maybe the content was educationally sound, but we became tired of hearing 'letters' referred to as 'ledders' and switched off long before he progressed from 'a' to 'zee'. We think an English-accent version should be made available for the British market.

HANDS-ON EXPERIENCES

We report on a necessarily brief acquaintance with the keyboard, but we have different backgrounds in typing and our comments may be of interest to a reasonably wide range of readers.

Owen tinkered with typewriters from an early age and never learned to type 'properly', let alone touch-type. But he can type articles and books fast enough to satisfy himself and his publishers. He found the keyboard extremely comfortable and quick to use. His accuracy, which is poor (thank goodness for word-processors!), was marginally better with the Maltron mode to the QWERTY. Since he tends to look at the keys (at least the less frequently-used ones) when typing, it was easier to look for the large white letters and numerals than for the small red ones. Learning the QWERTY keyboard since the location of the keys appeared to be easier than with the commonly-used letters are grouped together and 'E' is in a memorable place. He would have liked to have had the '0' (zero) in its logical position at the extreme left of the split row of numeric keys rather than in the conventional position at the extreme right. He thinks that learning ~~to use~~ the Malton really well would take very little time and his typing would be improved.

Audrey, with no previous typing experience, attended an introductory typing course and was taught to touch-type on a typewriter. Her first impression on using the Maltron keyboard (in its QWERTY mode) was that the keys were set closer together than on a standard typewriter keyboard. Measurement revealed that the Maltron keys are spaced on 19mm centres, exactly as on a conventional keyboard. The 'closer-together' feeling was an illusion originating from the fact that she did not have to move her fingers as far to reach the keys. An initial difficulty was that the absence of stepping between rows caused her to mis-key sometimes, pressing a key above or below of the one she intended. But, over a period of 15-20 minutes, this confusion disappeared and mis-keying was eliminated. When typing for long sessions on a standard QWERTY keyboard she has often experienced discomforting shoulder-ach. She found the Maltron keyboard much more comfortable and relaxing to use. Although the period of trial was necessarily limited, she is convinced that the Maltron keyboard would be easier and quicker to learn. She also tried touch-typing some of the training exercises, using the board in the Maltron mode. This presented no problems and she began to get the feel of it very quickly.

SUMMING UP

The Malton keyboard lives up to the claims of its designers. We consider that it is a considerable step forward in keyboard design, and in the consequent improvement of working efficiency.

We would like to thank Zenith Data Systems *and* Mr. Henry Budgett *for lending us the Zenith Z-150 which we used for reviewing the Maltron keyboard.*

THE NATURE OF COMPUTER GRAPHICS

F M Botto

We begin a short series on computer graphics with an introduction to display methods and mathematical conventions.

The widespread use of computer graphics today, can almost certainly be attributed to the recent and rapid development of microprocessor technology. Microprocessors forming the main constituent of microcomputers, are now capable of supporting sophisticated computer graphics software.

Prior to microprocessor technology, complex computer graphics necessitated the use of minicomputers or mainframe computers, both of which were extremely expensive in comparison to the micro-computers of today.

Two-dimensional computer graphics may be used, and are used in CAD stations dedicated to the transformation of circuit diagram into circuit art-work. If you are unfamiliar with electronics associated processes, figure 1 shows the function of such a CAD station.

Using CAD technology, three-dimensional computer graphics techniques allow the modelling and design of solid objects, such as mechanical components and structures. The CAD comcepts mentioned utilise computer graphics in its interactive form. Interactive indicates that the user may model the graphical output immediately following an appropriate implementation of user communication. Methods by which the user may communicate with the graphics output are numerous, and presently include:

- digitiser or graphics tablet (see figure 4),
- light pen (see figure 3), joystick (see figure 4)

and control dials. The most common forms of user communication at present are the light pen and digitiser. The light pen has been with us for some time, and like many other computer associated devices and concepts, originated from the United States. The light pen was a spin-off from the SAGE project, which was fundamentally concerned with the development of an early warning radar system, and implemented in the mid nineteen fifties. This breakthrough was a significant advancement in the development of interactive computer graphics (IG). As such this was an important milestone in the evolution of the CAD concepts mentioned.

GRAPHICS SYSTEM ORGANISATION

Any microcomputer based graphics system will include the following:

i) A method of display.
ii) A microcomputer with appropriate input/output ports to drive a VDU (visual display unit) and disk or cassette storage device.
iii) Disk or cassette storage device.

Figure 5 shows the essential elements of such a microcomputer graphics system. A current low cost home computer would undoubtably conform to this arrangement, and although limited will offer a degree of computer graphics capability.

TYPICAL METHODS

Using a cathode ray tube arrangement, there are basically three methods of producing an image, which are the storage tube, refresh graphics screen and raster scan displays. The refresh graphics screen was developed in conjunction with the light pen, as an integral part of the SAGE project.

A cathode ray tube (CRT) may be

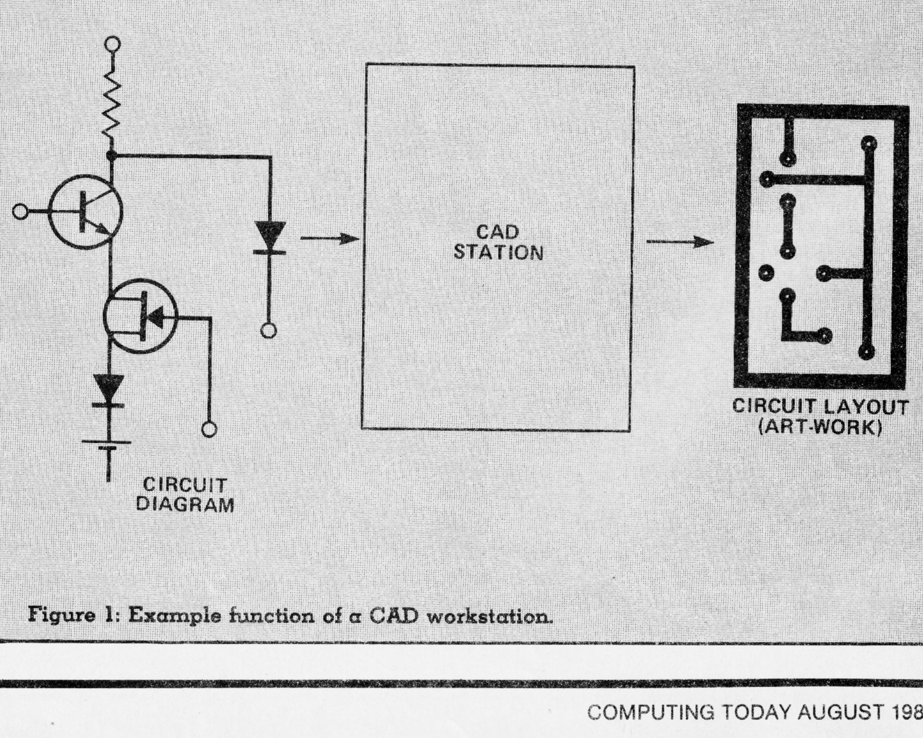


Figure 1: Example function of a CAD workstation.

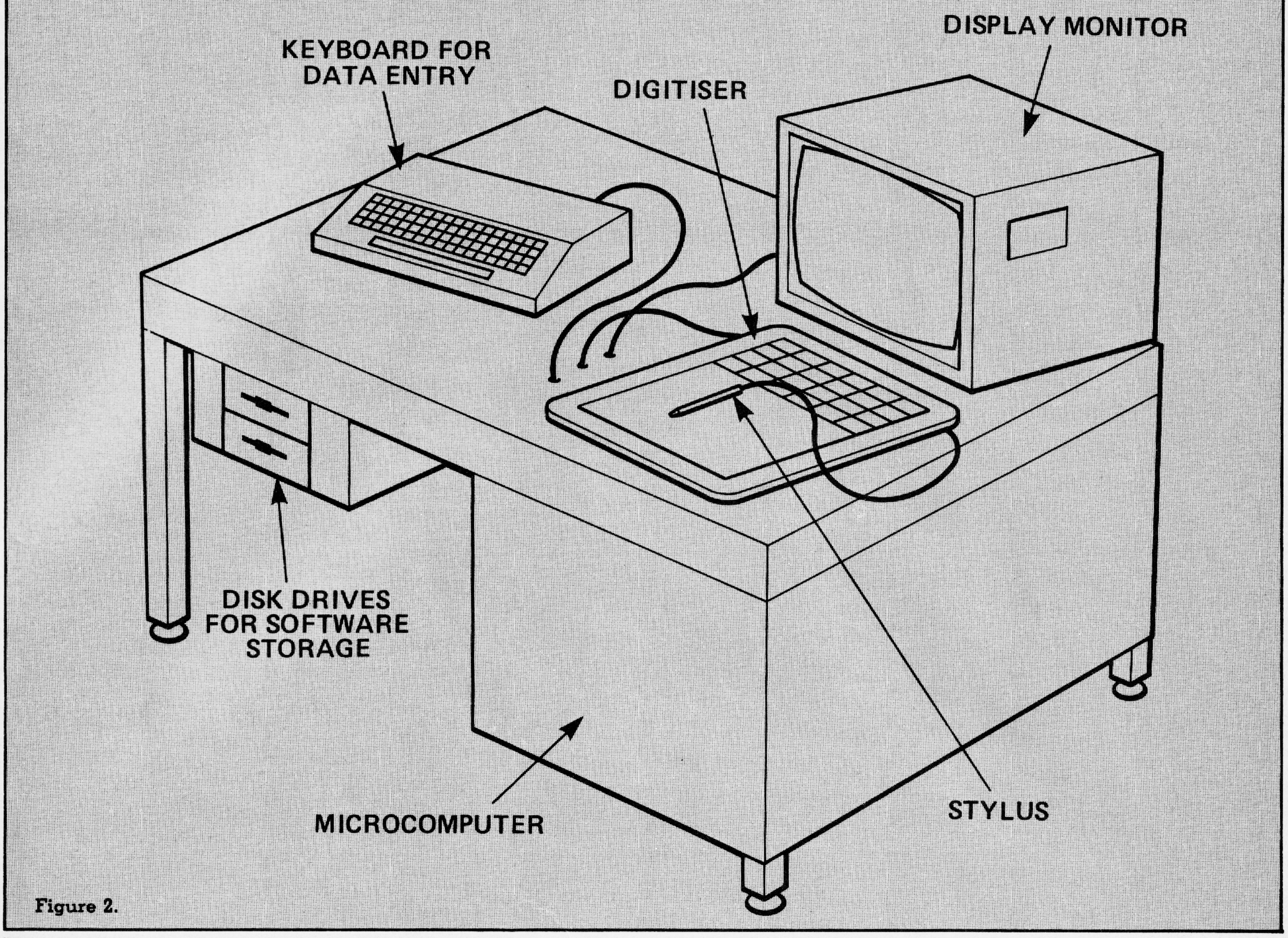


Figure 2.

Figure 4. below: simplified construction of an X/Y joystick.

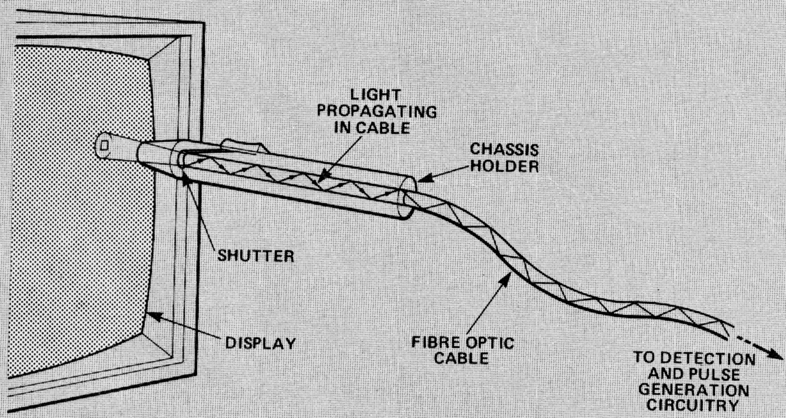
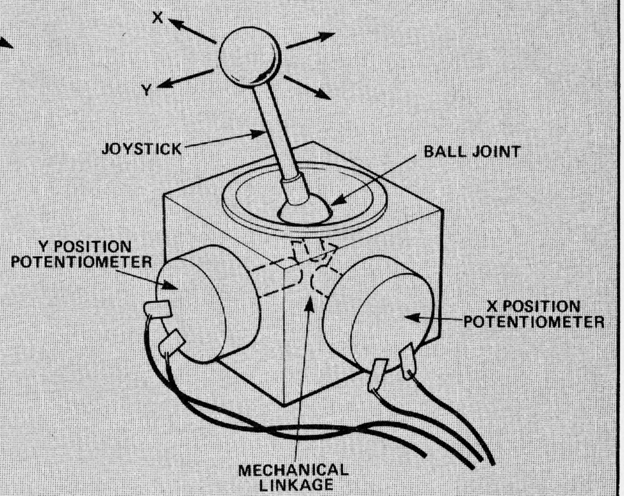


Figure 3: When the shutter is opened, the light produced by the illuminated section of the display, is appropriately launched into the optical fibre. The entry angle is such that propagation occurs. The basic operation involves a comparison between the instance of light detection via the light pen relative to the field scan period consumed. Hence, information is achieved from which the position of the light pen in relation to the display may be determined.



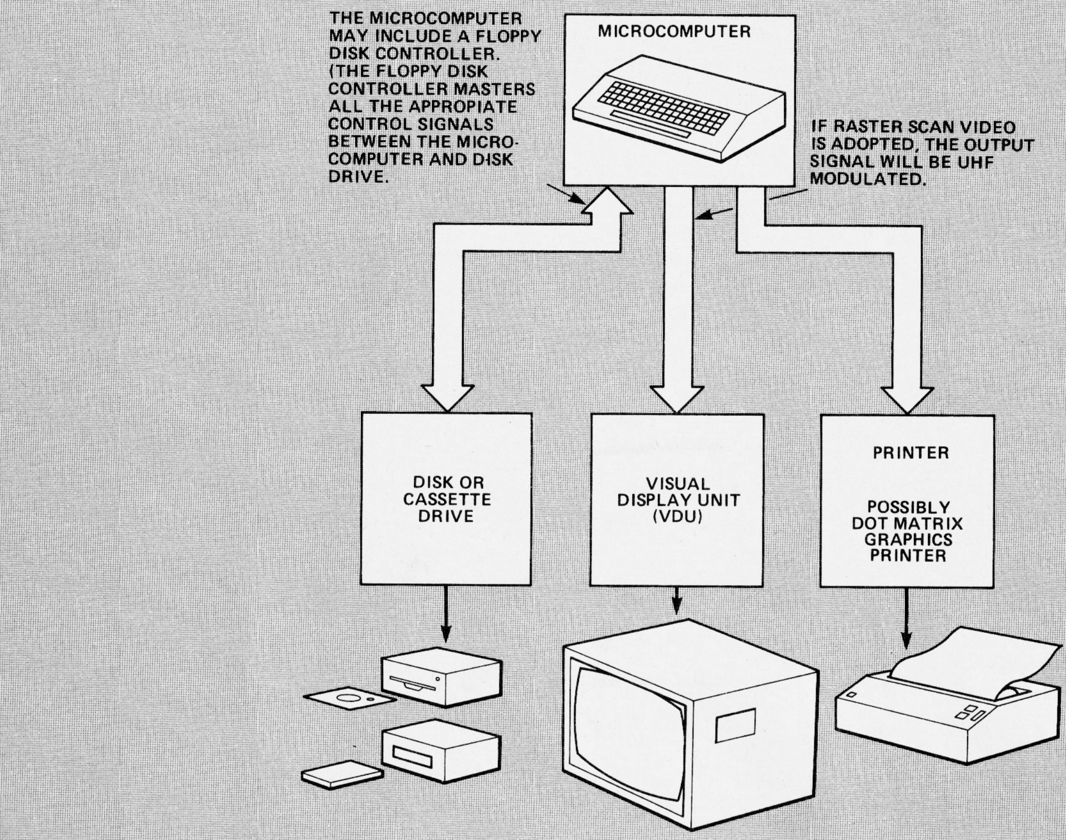
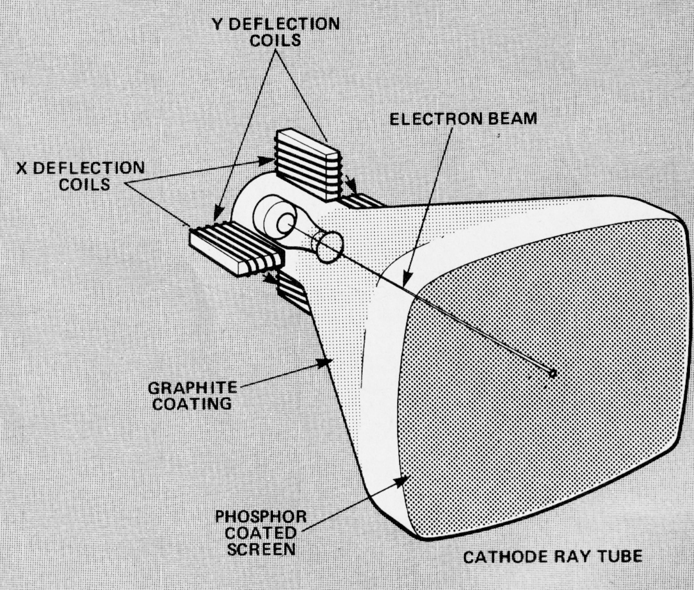
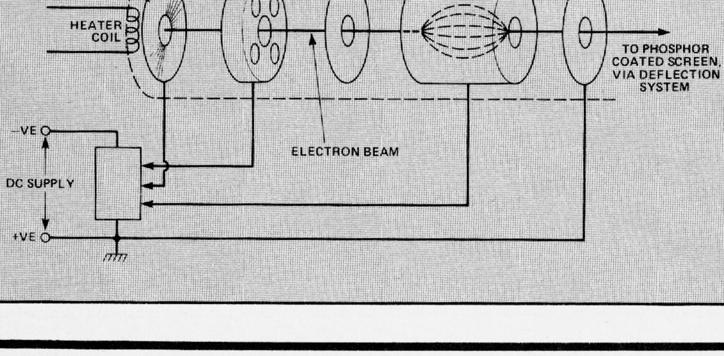


Figure 5: essential elements of a microcomputer graphics system.

Figure 6: Electrons are dissipated from the heated cathode due to thermionic emission. The released electrons are initially attracted to the negatively biased control grid. By varying the negative bias, the rate of electron emission may be controlled. The more negative the control grid, the greater the electron impact upon the screen phosphor, hence, a brighter illumination may be obtained. Anode B offers a means by which the electron beam may be focused, this is achieved by adjustment of the anode potential. Anode, A,B and C constitute an electron lens, and perform the final electron beam manipulation of the electron gun.





described as an optoelectronic device, and will essentially consist of an electron gun, a method of deflecting an electron beam, and an appropriate phosphor coated screen. The deflection system will be either electrostatic or electromagnetic. However, the large displays used in computer graphics systems will normally utilise electromagnietic deflection techniques, having both x and y deflection coils mounted outside the cathode ray tube. Figure 6 illustrates a basic CRT format.

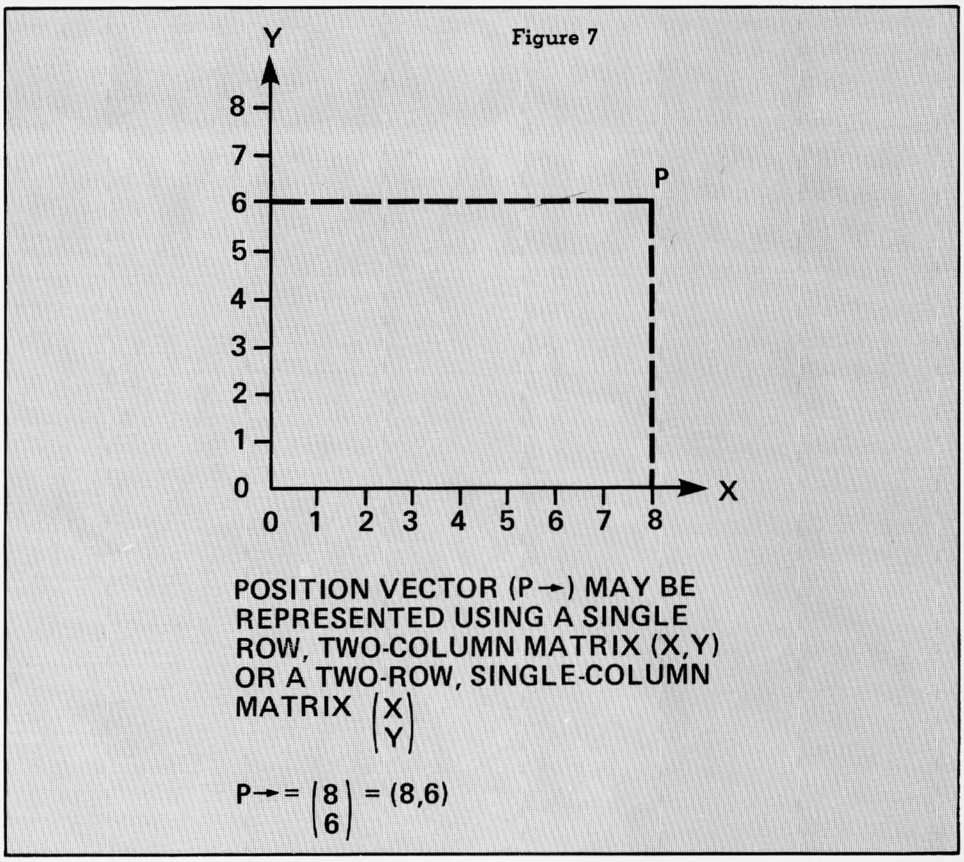


Figure 7

MATHEMATICAL CONVENTIONS

Points (coordinates) forming a graphical shape may be quantised in the form of a vector. A vector quantity may consist of several numerical values, unlike scalar quantities which comprise of a single numerical value, which will often represent a number of units. Graphical coordinates represented in the form of position vectors may be conveniently stored in a matrix. Position vectors forming the elements of a matrix, offer a means by which a graphical point or shape may be mathematically manipulated. A graphical coordinate may be represented as a two-dimensional position vector, comprising of an x dimension (horizontal distance) and a y dimension (vertical distance), relative to a common origin (0). Figure 7 illustrates the representation of such a two-dimensional coordinate.

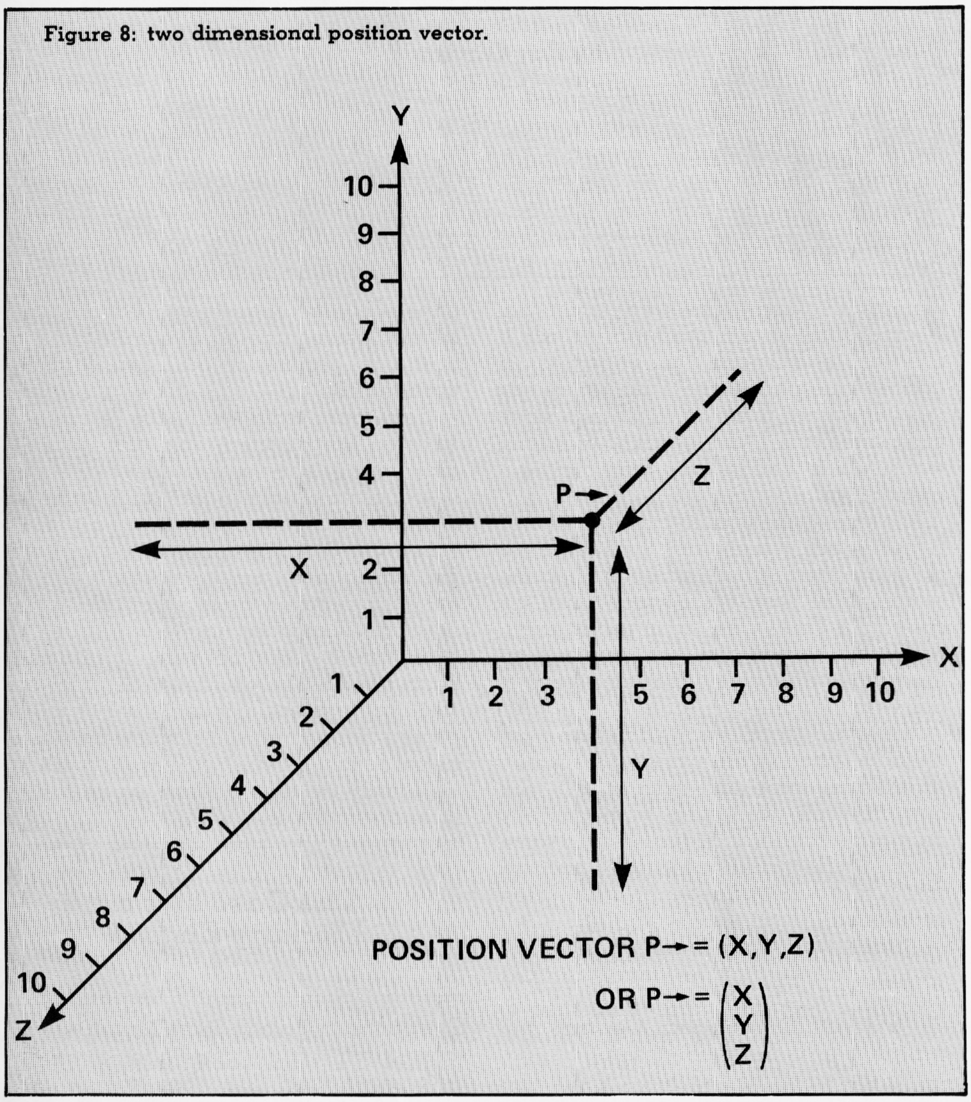


Figure 8: two dimensional position vector.

Simarly, a graphical coordinate may be represented using a three-dimensional position vector. The third component being a further dimension Z. Figure 8 shows the arrangement of such a vector.

The z component provides a means by which a solid object may be both graphically illustrated and manipulated (if required). When realising the shape of a solid object, one attempts to view it from a number of angles. Hence, to appreciate the physical appearance of an object, it must be observed in three dimensional space. The significance of the z compartment is therefore obvious.

When position vectors are arranged in an array format whether two dimensional,

$$\begin{bmatrix} x & y \\ x & y \\ x & y \end{bmatrix}$$

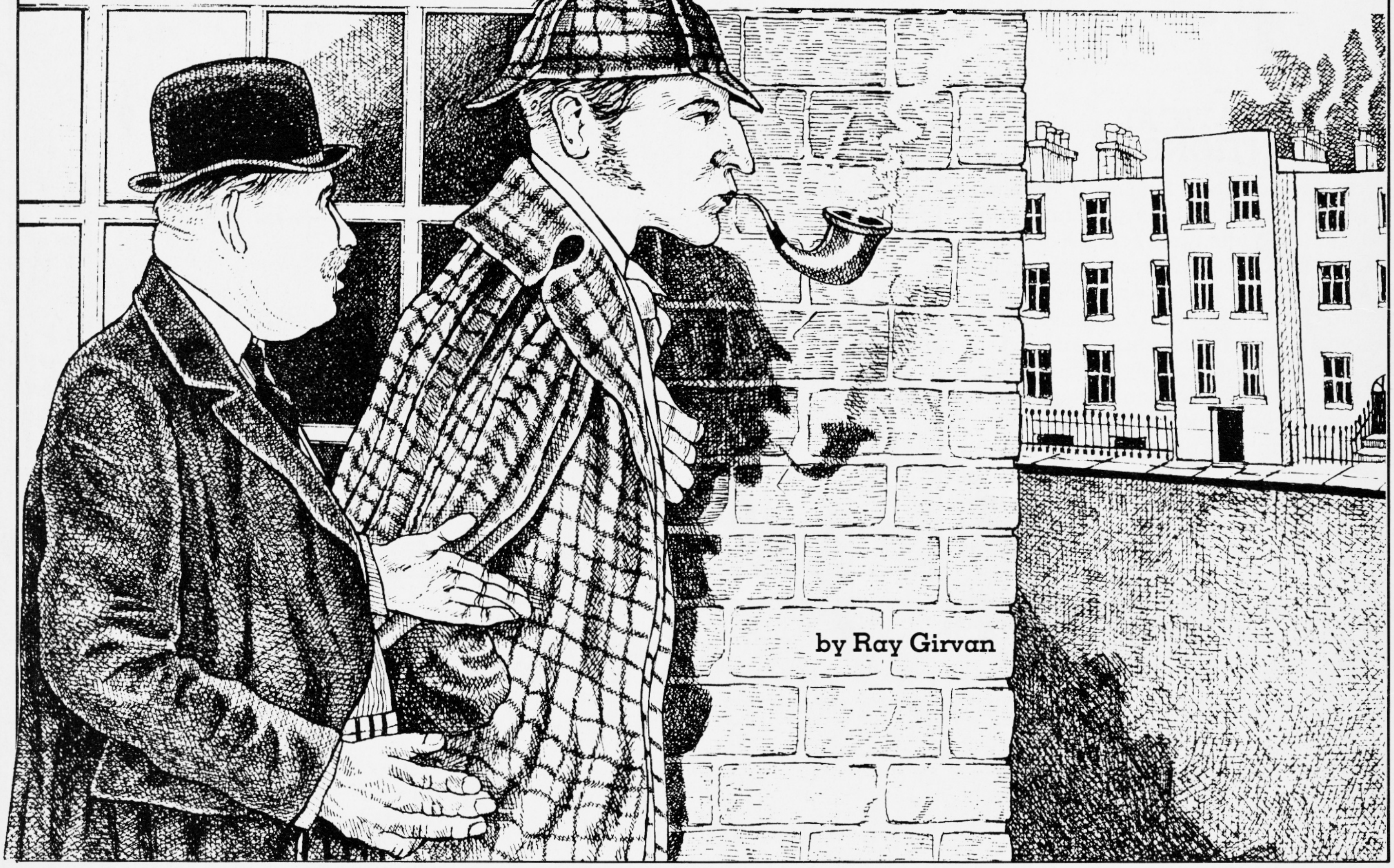
or three dimensional

$$\begin{bmatrix} x & y & z \\ x & y & z \\ x & y & z \\ x & y & z \end{bmatrix}$$

by multiplication of a suitable transformation matrix T, the position vector or vectors may be 'transformed'. The transformation may take the form of rotation, scaling, reflection, shearing or distortion. Some of these operations shall be explored later.

Part two of this series appears in the next edition of Computing Today.

THE CASE OF THE MORIARTY ENGINE



by Ray Girvan

"It is quite iniquitous, Holmes," I said. "My dear fellow," Holmes replied, preceding me up the seventeen steps to our Baker Street rooms. "I can neither agree nor disagree unless you tell me to what you are referring."

I gesticulated with the rolled newspaper. "Another factory fire; a Nottingham lace mill burned to the ground. According to *The Times,* Hartmann the anarchist has claimed resonsibility. The man must be mad!"

Holmes took out his key. "You may be right." said he. "But never presume madness merely because you do not understand the motive. It may be a diabolical one, I grant, but even then will have its logic. Take Professor Moriarty . . . "

— "Moriarty?" I scoffed. "Surely he is less of a mastermind than you believed?"

Not a month before, the said Professor had resigned his university post following investigations into his finances; not least, how one ostensibly earning £700 per annum could afford a Greuze painting valued at some six times that figure. I gathered he had set himself up as an army tutor here in London.

— "Pah!" Holmes expostulated, hanging up his hat. "You vastly underestimate him, Watson. No charges were bought; a man of his intellect could easily weather a few dark rumours. It is clear he is involved in some new villainy so time-consuming that he no longer chooses to maintain the imposture of academic respectability. A Mr. James has recently shown uncanny success in stock market dealings. I have as yet no proof, but feel sure it is the Professor acting under a pseudonym."

We entered the small sitting room, where I settled myself in a chair by the hearth; but Holmes paced restlessly. Recognising the symptoms of boredom, I handed him my copy of *The Times.* Casting himself full-length on the sofa, he read for several minutes, then tossed the paper on the crumpled heap of dailies in the corner.

— "I feel the black claws of stagnation in my brain!" he announced, springing to his feet. "Nothing but loose ends to tie up!" He took from its shelf one of his alphabetically arranged scrapbooks and crossed out some entry. "So, Henry Slater, Moriarty's accountant and book-keeper, has been imprisoned."

— "Surely that is excellent news, Holmes," said I, noting that he hardly looked pleased.

— "Indeed? Once again, we trap the acolyte, but the master eludes us."

— "Is there nothing to interest you in the papers?" I asked cautiously, changing the subject. "I thought perhaps you might wish to attend the lectures at the Royal Institute this afternoon."

— "I saw the advertisement in *The Telegraph,* Watson. The planned topic of vegetable alkaloids has been replaced by that of the new science of economics, which interests me not a jot."

I shrugged. Holmes claimed to read only the criminal news and the agony columns, but often betrayed the fact that little in the newspapers escaped his eye.

Lighting his briar-root pipe, he resumed pacing. I feared he would soon reach for the small bottle on the mantlepiece; but as he passed the window, something in the street below caught his eye.

— "I fancy we have a client," he said in a more optimistic tone.

The Case of the Moriarty Engine

Shortly the bell rang, and Mrs. Hudson showed in a young woman. She was short, dark-haired, and tidily though inexpensively dressed in brown. Her face was rounded and serious, and she clutched her purse tightly in both hands.

Holmes, at his most courteous, showed her to a seat.

She smiled nervously, removing her gloves. "My name is Lucy Scrope," she said. "I would not normally bother you, Mr. Holmes, but I fear for the life of my father, Edward."

Holmes puffed reflectively on his pipe. "You are not bothering me in the least, Miss Scrope. Tell me more about your father, and why you carry such an unaccustomed sum of money. You grip your purse as if your life depended on it."

— "Two men abducted him from his shop on the Edgware Road last week. I should have told someone sooner, but I feared going to the police would endanger him more."

— "Was anything stolen?" Holmes asked.

— "No. On the contrary, they left behind an envelope containing five hundred pounds. I could not afford your fee otherwise."

She opened the purse, but Holmes shook his head, frowning. "No doubt payment for your silence," he said. "But I wonder why. What is you father's trade?"

— "A watchmaker."

Holmes tapped out his pipe, eyebrows raised. For once I assumed I shared his thoughts. Two days before, the body of another watchmaker had been picked from the river at Woolwich. He had vanished six months previously; but the autopsy showed he had died of natural causes only hours before being found.

Holmes sat slowly, placing the tips of his fingers together. "I agree that your father may be in danger," he said. "But it may comfort you to know that I am already observing those responsible. Return home now — if all goes well, I will contact you tonight."

I showed her out. Returning, I took Holmes to task. "However can you claim to have the matter in hand?" I exclaimed. "You have not stirred from this room!"

He gave a short humourless laugh. "O ye of little faith, Watson! This is but the final confirmation that great matters are afoot. Unless I am very much mistaken, Professor Moriarty has built himself an Analytical Engine!"

— "What on earth is that, Holmes?"

— "A device capable of the most abstruse mathematical articulations. Babbage, its inventor, presented the idea to the government some twenty years before we were born . . . "

— "I have heard of Babbage. He designed the cow-catcher on American trains, did he not?"

— "Aye; and less commendably, skeleton keys. Regrettably, no engineering works then existing was capable of the precision required to build the Engine. What a challenge for the Professor!"

— "But how can you know of his plans?"

Holmes opened a hand, slapping the fist of the other into his palm.

— "Facts, Watson, facts. Each an insignificant premise, yet when juxtaposed forming an incontrovertible conclusion. Take the first — Slater's arrest. None of the usual bail or defence was forthcoming from the Professor's organisation — ergo, his services are no longer needed. Yet Moriarty can still adroitly manipulate the stock market! Thus, he has found some better way."

— "You think the calculating engine could do that?."

— "Second," Holmes continued, ignoring my interruption. "It was intended that the Babbage Engines read their instructions from punch cards."

— "As in a pianola?"

— "Precisely; though more to the point, the Jacquard loom system. I will wager that not Hartmann but Moriarty set the mill fire to hide the theft of equipment for his own machine. Finally, the disappearing watchmakers. Who better to use intricate work on cogs, gears, and bearings? It seems that Moriarty's first helper died on him, and he needed a replacement."

I considered admitting I was less than convinced, but thought better of it. Leaving Holmes deep in contemplation, I went to ask Mrs. Hudson to make tea.

The Case of the Moriarty Engine

Later that evening I was eating dinner alone — Holmes had gone out earlier — when I heard footsteps on the stair. A stooped old man in frock coat and pince-nez spectacles entered the room.

— "Hello, Holmes." I said, and continued my meal.

Somewhat disgruntled, he drew himself up to his full height, plucking off false mutton-chop whiskers. "I see your powers of observation are improving, Doctor," he said sourly. "I should fetch your old service revolver, if you feel up to an adventurous night's work!"

"Disguised as an elderly academic, I attended the economics lecture. As I expected, Moriarty was there, no doubt to improve his knowledge of the workings of commerce. It was a simple matter to follow his hansom afterward. Now we shall pierce to the centre of the spider's web. Come, Watson, the game is afoot!"

A cab journey in the fading light took us to a shabby riverside street in Wapping. Holmes led me down an alley between warehouses, halting at a rusty iron door. With the aid of Mr. Babbage's more questionable invention, we were soon inside one of the buildings.

I had expected musty darkness in the storeroom. Instead, the air was clean and warm, the ceiling strung with electric lamps. Drawing our revolvers, we picked our way between tall crates.

— "Behold!" Holmes whispered.

In an open area stood the Analytical Engine, if such it was. In shape and size it was not unlike a four-poster bed, but with the space between the pillars packed with an intricate mechanism of brass gears and vertical steel spindles. To the right, a concertina of perforated card hung from a smaller assembly bolted to the Engine. A well-stocked workbench stood nearby.

There was a shuffling and clink of metal from our left. I raised my revolver, expecting danger, but a mild-looking little man came towards us, dragging a length of chain shackled to his ankle.

— "I am Edward Scrope," he said weekly. "I implore you, whoever you are, free me before they come back!"

— "We shall do just that," said Holmes, kneeling to work on the padlock.

— "If we are quick," I said. "Moriarty can be arrested. Shall I call Lestrade?"

Holmes stared dreamily at the Engine. "I had hoped the Professor would be here, but . . . " He stood, suddenly resolute. "No. Watson, take a file from that bench and cut the chain."

I saw the padlock lying on the floor. "But you have already freed . . . " Too late; Holmes was already elsewhere, presumably examining the Engine. I was through the link in minutes, and led Scrope to the exit.

Holmes joined us shortly. "I decided to leave the Professor to his own devices," he said, a furtive twinkle in his eye. "I cannot begrudge him a little honest mathematical research. Let us be on our way — I am sure Miss Scrope will be delighted to see her father again."

The Case of the Moriarty Engine

"I do not understand," I admitted at breakfast a week later. "Why did you not have Moriarty arrested while you had the chance?"

Holmes sipped his tea. "Have you seen the stock exchange notes today?"

I riffled through the paper. "Oh, I see that the luck of 'Mr. James' has failed him."

Holmes chuckled and took a bite of toast. "Yes, Moriarty is financially ruined, which should temporarily curtail his activities."

— "I knew a machine could never match the judgement of the human brain."

— "Oh, it could," Holmes contradicted. "And did — while it worked correctly."

— "You sabotaged the Engine?" I laughed.

Holmes snorted. "Hardly! I am no engineer; and anything so subtle would have been noticed. Cryptography is more my line; by comparing the punched-card instructions with the handwritten notes on the bench, I was able to deduce the meaning of some of the patterns of holes. I took the opportunity to alter a few signs in the calculation." He glanced at the newspaper, smiling. "I admit the results exceeded my expectations."

— "So you made it appear that Scrope escaped by his own efforts, so that Moriarty would think nothing amiss, and use the Engine as planned?"

— "Yes, indeed. He may never trust it again; when a tool fails him, he discards it."

— "He will eventually deduce that you have caused him this trouble," I warned.

Holmes nodded grimly. "I fear so; there will be a final reckoning one day. But as for the Moriarty Engine, we have definitely seen the last of that."

THE END

DATA PROTECTION

Bill Horne

The 1984 Data Protection Act is designed to protect individuals whose records are held in any computer-oriented database. This means that each and everyone of us now enjoys legal protection from those who might misuse our records; there isn't a single British national whose name isn't held in a computer somewhere in the country.

No, we are not going to talk about how to deal with software pirates. This is a rather different matter. It concerns the Data Protection Act 1984.

Of no interest to you? Perhaps you should read on a little.

The praisworthy purpose of the act was to provide a measure of control over the massive data banks that have come into existence in recent years, holding personal information on all and sundry. There was an understandable fear that misuse of this information might pose a threat to individuals.

Quite apart from that concern, there was a danger to British international trading, in that failure to ratify the "Council of Europe Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data" might lead to restrictions on the transfer of personal data to or from other countries.

Now, there is no doubt that some law of this kind was needed. Speaking personally, I am not aware of having suffered in any way from the activities of data base users, but it is reassuring to know that there might be some means of redress should a problem arise. However, an act framed to catch killer whales can sometimes prove extremely inconvenient for tiddlers.

COMPUTER BUREAUX

Do you operate a Computer Bureau? Not such a silly question as you may think. If you provide other persons with services in respect of data, the answer is that you do operate a computer Bureau, and may be required to register as such under the Act. The same applies if you allow other people to use your equipment to process data. The only exemption seems to be a case where the data processed is entirely personal to the individual concerned.

Do you keep a file of club members' names and addresses? Quite a lot of people do, unaware that they may be operating a 'Computer Bureau'. I use a computer to set up articles. If those articles make personal reference to any individual, I could be required to register, I suppose.

A nasty snag is that responsibility for assessing the practical cover of the act rests with data users and computer bureaux, because the Register cannot examine every single case to see if it qualifies. It could be said that operation of the act relies on the conscience of individuals, and it has often been evident that people in the computer world are not always troubled by their consciences . . .

GUIDELINE No 1:
An Introduction and Guide.

Published by the Office of the Data Protection Register, Springfield House, Water Lane, Wilmslow, Cheshire SK9 5AX.

To be practical, it is extremely doubtful whether the register would ever be aware of everyone who runs an address file service, but failure to register is a punishable offence.

As with many of our laws these days, the odds are that the fly-by-night who makes a rude gesture at authority will get away with it, while the innocent get clobbered.

The answer might well be to exclude cases in which the stored data is held with the knowledge and agreement of the person to whom it refers. That would rule out the clever-sides who wants to make trouble and does so by claiming that the club records show that he is behind with his subsciription.

The worst snag concerns the inclusion of use of your equipment by other people. During the war, my car went missing from camp, and was found some distance away with stolen petrol in cans in the back. How could I prove that it had not been taken with my permission, and with knowledge of the intended purpose? It would be just as difficult to prove that you had no knowledge of the purpose for which your computer had been used.

OTHER CASES

There is probably no need to worry unduly about the possibilities outlined above. The Registrar is going to be busy enough catching up with the big data users. But it is an example of legislation affecting computer users that has been pushed through in a manner that betrays a vast ignorance of the subject. There was another, not long ago. An American running a mailbox service was held to be responsible for the content of the messages which his customers sent. It seems that he was expected to read and vet every message to make sure that there was nothing illegal about the text.

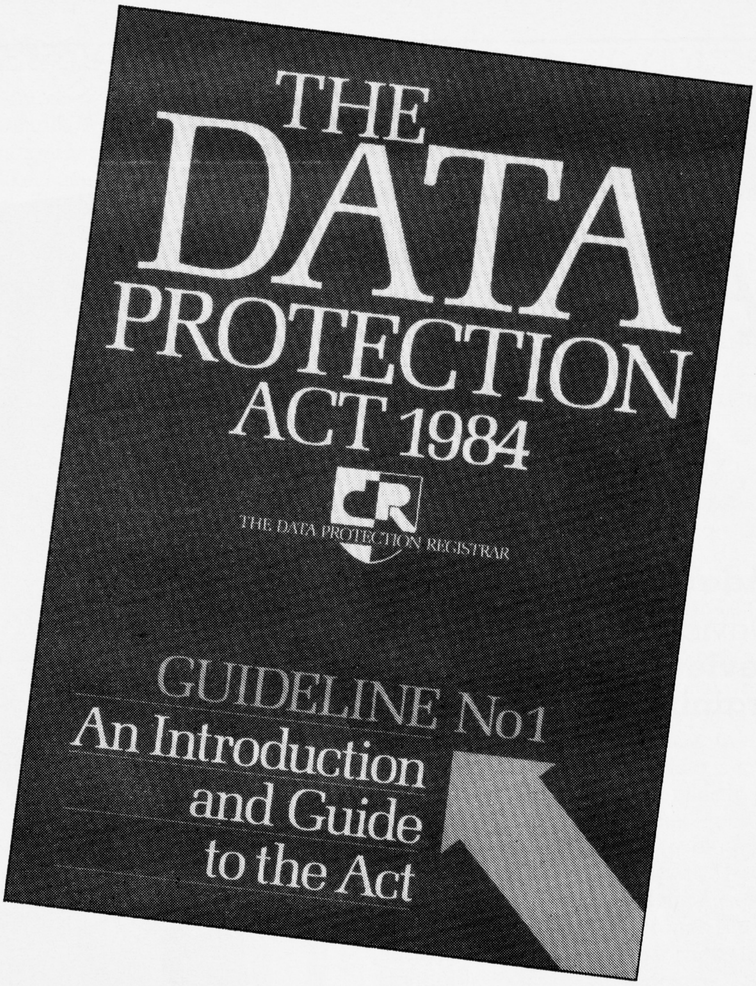
The trouble is that computers have created entirely new situations which are beyond the scope of existing laws, and — it seems — beyond the understanding of legislators. It is rash for anyone to become involved in any legal dispute involving computers, since the outcome will probably be dependent on a chance combination of legal and computer knowledge.

Let us, after all, look briefly at software piracy. Applied literally, the copyright law could prevent any new programs being written, since they are bound to contain segments of code that are identical to segments in some existing program. On the other hand hand, if you take the broader view, a small change might be said to produce a totally different program. Where do you draw the line? I remember one program that had a fatal bug in it. After a lot of probing, I found and removed the bug. I then had a workable program, but could not offer it for sale, despite the fact that there was no workable equivalent in existence.

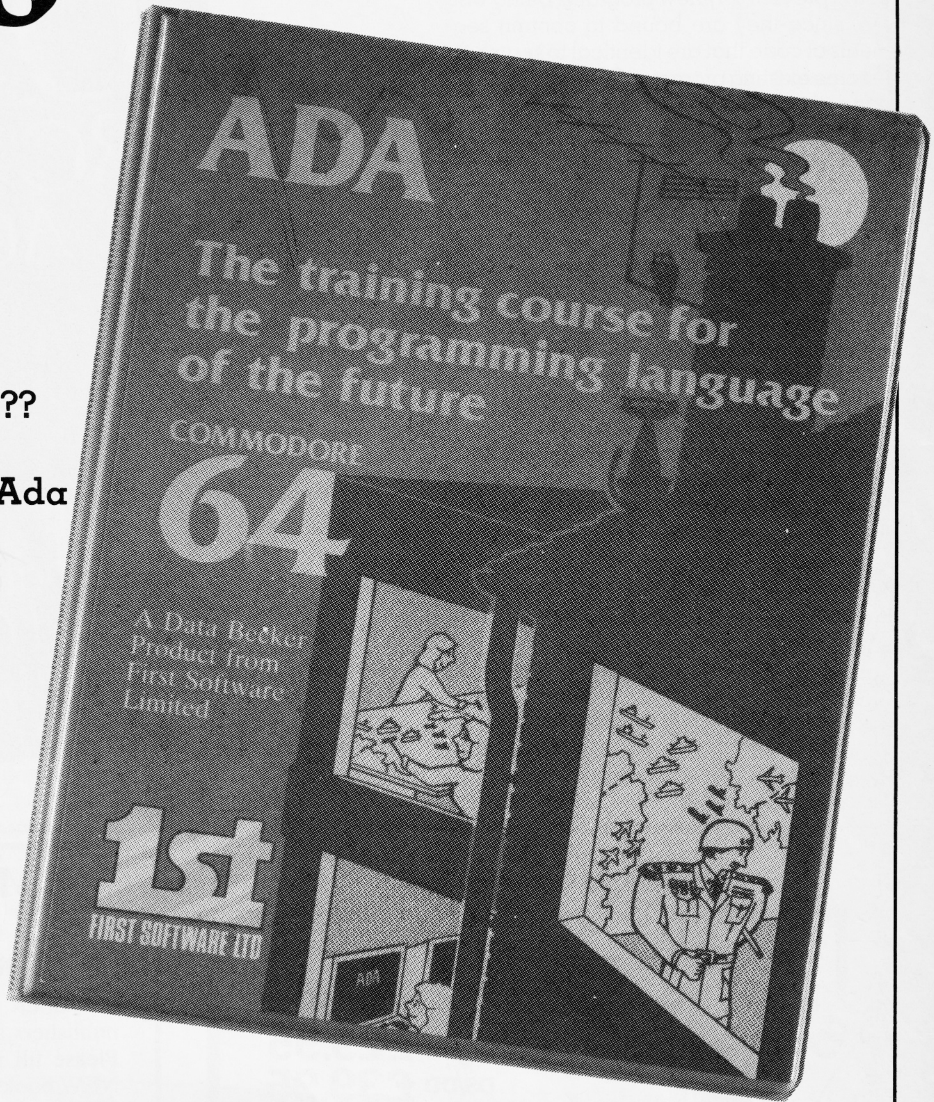
Ah, you will say, but it's all a matter of common sense. No, sir. It is well known that the law is an ass, and rarely takes a common sense view. Finding a safe course through the legal minefield is something we may have to live with, at least until — some time in the next century? — computer matters are more clearly understood and accepted.

GUIDELINES

In the case of the Data Protection Act, there is one ray of hope. The Data Protection Registrar is publishing a series of books providing Guidelines for those who may be affected by the Act. These are not definitive interpretations of the Act, but provide 'rules of thumb' to help you to assess your position.



FIRST'S ADA



Ada? On a Commodore 64??

David Barltett

reviews First Publishing's Ada training course.

This review of the First Software Ada training course must be split into two distinct sections. Firstly an explanation of what Ada is, and secondly, how the software package relates to this.

To begin with let's have a look at Ada, a language that most readers will not be familiar with due to its uncommoness on micro computer systems.

Ada is a *constructed* language. It was designed by a committee with international participation, but mainly in conjunction with the American department of defence. About ten years ago they looked at how much they were spending on computer systems and programming. They decided that a lot of money was being wasted due to the vast number of languages being used. Fortran, Cobol, and Pascal were all in common use in the defence department's big computers, along with odder languages such as Coral and Forth for run-time systems in missiles and other embedded computer system.

The fundamental problem with this was that none of the languages came particularly close to an ideal language for the defence environment. One single language was definitely desirable, so that different development systems, departments, and programmers could all get along together.

So the next few years plodded along and all the existing languages were considered, but none really stood out. A decision was made to invent one based around Pascal, with a lot of Pascal's good points included (because of all the languages that were considered, Pascal came the closest) and a lot of Pascal's bad points removed. Pascal was never really intended for implementation anyway, but was merely designed as an intellectual exercise.

To cut a long story short, the various committess that made up the designing body came upon the final design and named it Ada. This is after Lady August Ada Byron, Countess of Lovelace. Otherwise known as Lady Lovelace. She is thought to be the first real programmer, and worked closely with Babbage on his Analytical Engine and published several papers outlining the processes that we use today and call programming. She was also the daughter of Lord Byron.

Eventually, a language was created that was supposedly able to be used for writing stock control programs for the quartermasters stores, or even wordprocessors. Whether they succeeded or not is largely a matter of opinion. But one thing is for sure, and that is that a lot of money is being spent on ada, and it is looking like becoming a standard amongst the bigger computers, which means that in a few years it will be down to the super-micro, and business micro level;.

ADA SUPPORT ENVIRONMENT

The 'standardness' of Ada is ensured by the American department of defence holding the copyright to the name; to use it you have to assure them that the compiler you have written is up to the current Ada standard. What isn't so standard is the ASE or Ada Support Environment. This is a common collection of routines and functions that the Ada programmer can call upon whilst writing or operating his software. The last I heard about a standard ASE was a couple of years ago when the committe was still mulling over it, and the 100+ routines that had been suggested.

The routines are a big feature of Ada and are called 'packages'. There are 62 reserved keywords initially, any of these can be redefined by the user, in a similar manner to Forth, though redefining these is not advised due to other packages using them and expecting them to perform in a particular way. There is also an assumed number of packages already predefined, that define how the maths works and what ASCII is and so on. these don't actually exist, but it helps to think about them as if they did.

The structure and 'feel' behind Ada is that of Pascal, from which it was taken. CASE, BEGIN, and WHILE are all there along with a number of Pascal favourites (some in a slightly changed form). Programs are built up out of blocks — again similar to Forth, and data typing is very flexible.

Another function that was considered important by the US DoD was that programs written under Ada should be bug free when they went out into the wide world — after all you only know if an ICBM guidance program is really going to work when the thing is actually falling towards Moscow (or perhaps Washington). So, exception handling was introduced. This takes error trapping away from the operating system and compiler in a way that 'ON ERR GOTO' never did in BASIC. You can trap almost anything. And, with luck, produce some really good bug-free code.

FIRST'S ADA

Ada is going to be very important so watch out for it. It is already implemented in a limited form on MS-DOS and PC-DOS, and there is even a CPM-80 version. However, these cost around £800 to £1000. But what is really interesting is the £50 Ada training course from First Publishing for the Commodore 64.

The Ada training course is so called because of the previously mentioned restrictions on calling cut-down Ada compilers 'Ada'. Even so there is a fair Ada compiler included. The other software included in the package is an assembler, a disassembler, exercise and example programs for the instruction course, and the editor.

Unlike most languages for the C64, Ada uses its own, quite good editor that has a built in checker. So you can't enter in an illegal character where it isn't wanted. The editor is menu driven with three main menus. The menus are accessed by using the comfortably placed '@*' keys. I puzzled over the choice of keys as they seemed a bit odd at first. Most systems use the function keys or the numeric keys, but these are needed for other functions and the '@*' keys are very well placed.

THE MENUS

The Start menu allows you to alter the operating environment for the colours on the screen. This is very useful on colour TVs and Monitors alike. I use a green screen monitor, and what most manufacturers consider to be a good combination of colours doesn't actually apply to the green and blacks that they get translated to.

The Commands menu allows you to access the disk drive, to save, load, and scratch programs, and to transmit disk commands to the disk drive and display directories. It also controls the printing-out of the program and how much memory is left.

The Write/Edit menu is the most interesting though, as this is used when creating or changing an Ada program. When creating a program the function keys assume controls for starting new text, ending text, scrolling through the text, inserting text, and so on. The editor is a line editor with each line being given a number, though these line numbers are purely for reference and editing purposes and are not used by Ada, except for the trace function discussed later.

FACTSHEET:	Ada Training Courses
Machine:	Commodore 64
Price:	£49.99
Available from:	First Software Limited Unit 20B Horseshoe Park Pangbourne Berkshore. Tel. 07357 5244

COMPILATION

Once you have your suitably edited and created program it is ready for compilation. F2 from the command menu intialises the procedure. Firstly you are asked whether you want a trace option. This is an unusual, but by no means unwanted feature for a compiler. Compilers are very tricky things to write for as the sequence usually goes edit-compile-results, edit-compile-results, and so on. Interpreters are much easier as you do not have to wait for compilation to take place, and you can always pause the program to see what is happening at any one point. The trace function is a very worthy addition to this, package and shows some thought.

Now starts a lot of disk swapping between the data disk and the program disk as the compiler is in several discrete sections. The Lexical analysis comes first, which checks the syntax of all the reserved words and functions. Then the Suntactic check is made to see if the grammer of the program is correct. Then the Semantic analysis is made, which checks to see if the program basically makes sense. The semantic checker then creates an assembly source listing which can be modified or altered in the normal way by anybody who is competent at assembly language. The assembler source is then assembled by the included assembler.

This gives a very flexible system that enables the results of the compiler to be easily customisable and optimised.

THE TRAINING COURSE

The training course itself is very good and explains all of the various functions of Ada. Most of the standard packages are present, but on a more limited scale. As the author says about the TEXT_10 package which has to be different for all types of computer yet its effects have to be standard, the most gmportant bits are there without taking up all of the memory at the same time.

The first lesson deals with the use of this package and its associated functions and procedures. This is one of the more important packages as you can get very little on the screen without it.

Other lessons include screen control, data I/O, data typing, functions, decisions, loops, and many other necessary lessons towards understanding the language. At the end of each lesson there is a series of questions and there are suggested answers at the end of the book.

A good half of the course book is given over to explaining how a compiler works and the methods used in the C64 Ada compiler. At the beginning of the book the author explains that the idea behind it is to give a fuller understanding of a compiled high level language. There is more than enough information to do this both the experienced BASIC user and the real expert. The system is *not* designed for the first time user and is improved because of that. Much software of this type is ruined by trying to make it understandable to a complete novice. This isn't, but I think that somebody with a thorough enough understanding of computers will be quite happy as long as they took it at the right rate.

Another objective of the course is to teach low level processing so that a greater understanding of high level languages is achieved. To this end some sections in the book are given over to this, along with the associated disassembler and monitor packages.

CONCLUSION

On the whole this is an excellent package and well worth the money. The software is first class, though I felt the editor was a little strange but this was due to being used to screen editors. The documentation is to an extremely high standard, and the course progresses through easy stages.

The version of Ada in itself is cut down from the full implementation for obvious reasons, but is still good to use. To my knowledge, no other compiler of this type, or any type for the Commodore 64, produces source code that can be modified and assembled at will. It even means that by using a cross assembler for another computer (such as a BBC) you can use the compiled code elsewhere.

So to conclude, this is one of the best pieces of software of this type that I have seen. First Publishing and First Software have set an extremely high standard for their products and I hope that they can keep up the good work.

● *Thanks to Ian (Gnome) Graham of Marconi Avionics who helped with the preparation of this article.*

LEARN UNIX

Mark Woodley

Complementing our earlier series on 'C' programming, we begin a three-part look at the UNIX environment.

The development of the UNIX operating system could have been described as something of an accident. It began in 1969 with the American Telephone and Telegraph Company, in the Computing Science Research Centre based at Bell Laboratories, in Murray Hill, New Jersey. You may have noticed from my previous series that this was also the home of the C programming language. Dennis Ritchie began his work on C in the hope of improving the first versions of UNIX. Now in the latest System V release of UNIX, all but the most low-level code is written in C.

At Murray Hill, Ken Thompson had been working on the development of a file system, for which he was able to produce simulation programs on the Centre's GECOS multi-user system. Thompson and GECOS system. So when there became available a cast-off PDP 7, their overriding objective was to create a computing environment in which they could effectively and comfortably persue their programming research.

If the operating system on the PDP 7 had not been so crude, UNIX may not have come to fruition. Bell Labs, had recently abandoned work on the Multics operating system, so at the time, operating system development was not popular. But when Thompson and Ritchie wrote a space travel program for the GECOS system (as part of a project to simulate celestial mechanics), it performed so badly, that it was decided to port it to the PDP 7. The PDP7 was only provided with an assembler and a loader, so they were forced to rewrite the core of the operating system and provide a new assembler, cross-assembling from the GECOS system. Before long a command interpreter and the file system were added, to produce the first child in the UNIX family.

After this, the group obtained a PDP 11/20 in 1970, for a text preparation project, which gave birth to the second version of UNIX. In 1973, when C became available, the whole operating system was rewritten to run on all PDP 11 series computers and a better, fourth system was provided which could also run on the Interdata 8/32 mainframe in 1979. At about the same time the University of California at Berkeley developed a version 4.0 UNIX for DEC VAX machines, and so became a distributor of 'Berkeley' UNIX systems.

HIGHLY POPULAR

Unix has become so well liked by members of the computing profession that it is now peaking the popularity polls, and AT & T are now competing with IBM to establish a standard operating system. It seems that AT & T may win the battle, and that IBM will market its own versions of UNIX in competition, perhaps putting many of the smaller UNIX software houses out of business.

Most of the firms that have jumped on the UNIX bandwagon have produced their own versions of the system, not necessarily recognised by Bell Labs. One version that is recognised, is the popular XENIX. Its authors, Microsoft, have recently been assured by AT & T that they will have support in implementing the latest version of XENIX, which is expected to match System V. It appears that AT & T are doing this reluctantly, so as to stake a claim in some of the revenue for its best competitor.

DEC (who manufactured the PDP computers on which UNIX was first developed) produced various systems, mainly ULTRIX, VNX-VM and DEC/Shell. Dec had shown some reluctance to market these UNIX systems in the UK, and have recently announced that they are to pull out of the UNIX market, in preference to their own operating systems. At much the same time however, ICL have announced that they are to spend money in developing a new machine that will be able to support System V. And Hewlett-Packard are working on HP-UX which is hoped will run on all of their machines.

Proven systems for mainframes include CENTIX, that runs on the Burroughs Megaframe, VM/IX that runs on the IBM 370 series, and CONVEX UNIX that runs on the Convex C1 60, one of the most powerful computers in the World.

On a much smaller scale, there are several versions of UNIX for the IBM PC that include XENIX. These versions are primarily aimed at the small business end of the micro-market.

The production of UNIX systems by other manufacturers strive to prove compatibility with the original versions of UNIX but compatibility is not much of a problem, since most commercial versions are in fact better, since they include extra features that are not normally available in standard UNIX. One such important feature is file-locking which I will say more about nearer the end of this series.

THE SHELL

The user communicates with the UNIX system at a terminal using the Shell. The Shell is the name given to the UNIX command interpreter.

Looking at UNIX is like looking at a nut. The Shell is the outermost part of this nut that provides the interface between the user and the system, whereas at its core is the software that controls all the dedicated I/O and other essential low-level activities, called the kernel.

On a multi-user version of UNIX, you would get a Shell by switching on a terminal and hitting a key at the keyboard to 'wake it up'. In response, the UNIX system will display the name of the UNIX system you are using (they are all given names), followed by the login prompt:

```
Berkeley 4.1
Nehemiah VAX
Login:
```

Berkeley 4.1 is the version demonstrated here, and it has been named Nehemiah VAX by the system administrator. The prompt - login: is provided by a login program, running under the Shell.

The first thing that you will notice about UNIX is that although it was very well designed in most respects, its user - friendliness or lack of it, leaves much to be desired. Now the system will do nothing until you type in your allocated user-name, followed by a <return>, but if this doesn't work try typing 'guest' (the letters of the passwork won't be displayed at your terminal). You might have to experiment with login for a while if you have not been formally introduced to your system.

You will soon know if a mistake has been made because the login program will respond with,

login incorrect

and puts you right back where you started. If you are successful however, you will be greeted by the messge of the day, and a few arbitrary lines of daily announcements written by the system administrator. They may be asking users to purge their directories of unwanted files or to take note of some ensuing system maintenance. Whatever the message, the system will then start running a Shell, from which you will be able to issue commands.

Normally, the prompt given to a user of the Shell is a '$', but if you have been given the privilages of the system administrator it will be a . A privileged user can, for example, alter another user's file as if he was the owner of that file.

SHELL COMMANDS

The input to each program is treated as a constant stream of characters, terminated by the end-of-file character, which in ASCII is <Control>D. The Shell behaves just like any other program, so when you want to log off, simply type CONTROL D immediately after a prompt.

After the prompt, you can invoke a command by typing in its name. Let's start by looking at the date command.

date <return>

This produces today's date and the correct time at your terminal :

Mon Sep 30 09:00:00 GMT 1985

GMT, of course, stands for Greenwich Mean Time.

Each command is held as a program, or a 'binary' file in a directory called 'bin'. When you give the Shell a command, it searches 'bin' for a program with the same name and then runs it. As you will learn later, you can also set up your own commands in a directory for the Shell to use.

COMMAND ARGUMENTS

On the same line as a command, the user can introduce arguments as parameters to be passed to the program by the Shell when the command is brought into action. Arguments must be separated by spaces.

The 'cal' command is used to print out a calendar. It can produce the calendar for either a whole year or a single month. With only one argument, for example, '1985' in cal

cal 1985<return>

it prints out the whole calendar for 1985. If there were two arguments, i.e.

cal 10 1985 <return>

the cal program would produce the calendar for October 1985.

October 1985

S	M	TU	W	TU	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

The first argument is now being treated as the month (which is from 1 to 12) and the second argument is being treated as the year (which is from 1 to 9999). If these arguments were out of range, then it will be the cal program and not the Shell that complains.

Attention is drawn to the manual entry for this command which points out that the year must be given in full and cannot be abbreviated to two digits, as this would then refer to the early Christian Era!

ARGUMENT CONVENTIONS

There are also loosely defined sets of argument conventions that are followed in the design of commands. Some commands have options. For example, a complier may offer options to turn off warning messages, or switch on diagnostics. The compiler writer will think of letters to represent each of these options, like 'W' to refer to the warning option and 'D' to refer to diagnostics.

To switch something like warning messages off, this letter is included as an argument, preceded by a minus sign,

-W

and to switch something on, the letter is preceded by a plus sign,

+d

A complete command to compile a C program from the source code in a file called 'sce', using the C Compiler (cc), without warning messages, would be,

cc -w sce<return>

The options are always written as the first arguments in a command. There are no fixed letters for options, these will have to be decided by the writer of the command.

TERMINAL CHARACTERISTICS

One unfortunate feature of UNIX is that you will probably have to start by defining at least one of the keys at your terminal before you can use it properly.

The default key for delete is # and the delete button is used as a break key. You can change this state of affairs with the 'stty' command. The 'stty' command is very powerful and allows you to do three things; set up certain keys such as the delete key, set up a communications protocol and preserve or remove tabs. There is also a special 'tabs' command that lets you set up tab positions.

To show how stty works, you can set up the <rubout> key as your delete key with

stty erase <rubout><return>

The user can put commands such as this in a file called profile and the Shell will execute these commands each time you log on.

KEEPING TRACK OF EVENTS

We have already seen in the cal and date commands that the system is able to maintain the current system time and date. A further 'calendar' command allows you to take advantage of this. The calender command looks for a file called calendar in the user's directory. It then produces any lines from the file that contains either today's or tomorrow's date. This way a user can keep a calendar of events and have the computer periodically remind him of the immediately ensuing ones.

SENDING MESAGES

An interesting feature of the UNIX system, is that you can instruct the Shell to send a real-time message, to another user logged onto the system. Firstly, to get a list of users on the system type 'who'. Who will give the user-names of all the logged users, with the number of the terminal that they are using and the time and date that they started:

guest	tty01	Sep 30	10:20
steve	tty02	Sep 30	09:01
paul	tty08	Sep 30	10:15
nigel	tty11	Sep 30	13:10
teresa	tty14	Sep 30	11:32
wendy	tty19	Sep 30	08:51

If we now wanted to send a message to Wendy, we would use the write command to echo everything from our terminal, to the one that Wendy is logged onto:

Write wendy<return>

(If Wendy was greedy enough to be logged on to two or more terminals at the same time, then we could specify the device as a second argument.)

As we type in the message, it would appear at her terminal like this;

Message from guest tty01
See you later!

until you typed <control>D, the end-of-file character, to terminate the input to the write command (pressing <control>D again would log you off). Wendy could make her replies at any time using write in the same way.

An unfortunate consequence of the write command is that mesages can appear on your screen at any time, even to interrupt the output of a program. It is therefore necessary on occasions, to refuse to accept messages with the mesg command:

mesg n<return>

which antisocial user will put in their 'profile'

Messages can be accepted again if the user types,

mesg y<return>

SENDING MAIL

Each user has a mailbox file called 'mbox'. You can read your own mailbox, and send mail to other users with the powerful 'mail' command. Mail is menu-driven and allows you to purge your own mailbox and send files to either a single user or a group of users in one go. Thus allowing you to broadcast to as large a portion of the user population as you wish.

UNIX

UNIX PROGRAMMER'S MANUAL

To find out the details of a particular command and the options available, you can always refer to the UNIX Programmer's Manual, because it is available to each user of the system from the Shell. To get an entry in the manual, you use the man command. Any page describing a command (or page of commands) can be called up by putting the command names as arguments. For example, we could get details of the man command itself with.

man man <return>

which would produce something like that shown in figure 1.

```
MAN(1)                UNIX Programmer's Manual                MAN(1)

NAME
     man - print out the command.

SYNOPSIS
     man [ section ] title...

DESCRIPTION
     man  formats a specified set of manual pages.  If a  section
is  requested,  man  looks in that section of the manual for  the
given titles...
```

Figure 1: details of the 'man' command

ED AND VI

Files are created using one of the system editors. There are two standard editors available: ed and vi. A full description of these editors is well beyond the scope of this series, but here is a brief overview:

ed is a line-by-line text editor based on a famous editor called QED. It is normally brought into action with,

ed <filename>

ed stores text in a portion of the computer's memory called a buffer. If <filename> does not exist, then ed will prepare to create a new file with this name, and the buffer will start off empty. If you want to edit an existing file, then the contents of the file will be loaded into the buffer where they can be edited, before writing them back to the file. There is a current line-pointer within this buffer called Dot. When you start, it points to the first line in the buffer.

You can insert from Dot onwards using the insert command 'i',

i<return>
text
.**<return>**

The full-stop at the end is used to terminate the text. You can also append after Dot using the append command 'a'. in the same way,

a<return>
text
.**<return>**

You can move Dot forwards and backwards through the text with lines like,

+3<return>
−4<return>

to move forward three lines and back four.

You can also find the value of Dot at any time by typing,

.=<return>

You can also use mathematical expressions, like those above, to print portions of the buffer, for example,

1,.+10p<return>

prints from the first line, to ten lines after Dot, and

5,$p<return>

prints from line five to the end of the file.

The delete command can be used in the same way to delete lines at a time, e.g.

10,15d<return>

and text on the current line can be changed with the 'switch' command, 's', e.g.

s/Pascl/Pascal/<return>

replaces the first occurence of 'Pascl' with 'Pascal'.

Finally, the modified buffer can be written to the file by typing,

w<return>

and you can quit the editor with,

q<return>

The other editor 'vi' offers all the features of ed with on-screen editing. Any ed commands can be performed by preceding the command with a colon, e.g.

:w<return>

There are also special onscreen controls. 'x' is used to delete characters and 'i' is used to insert them. For example, to insert the word 'diskette' at the cursor position, you use the following keystrokes,

idiskette<escape>

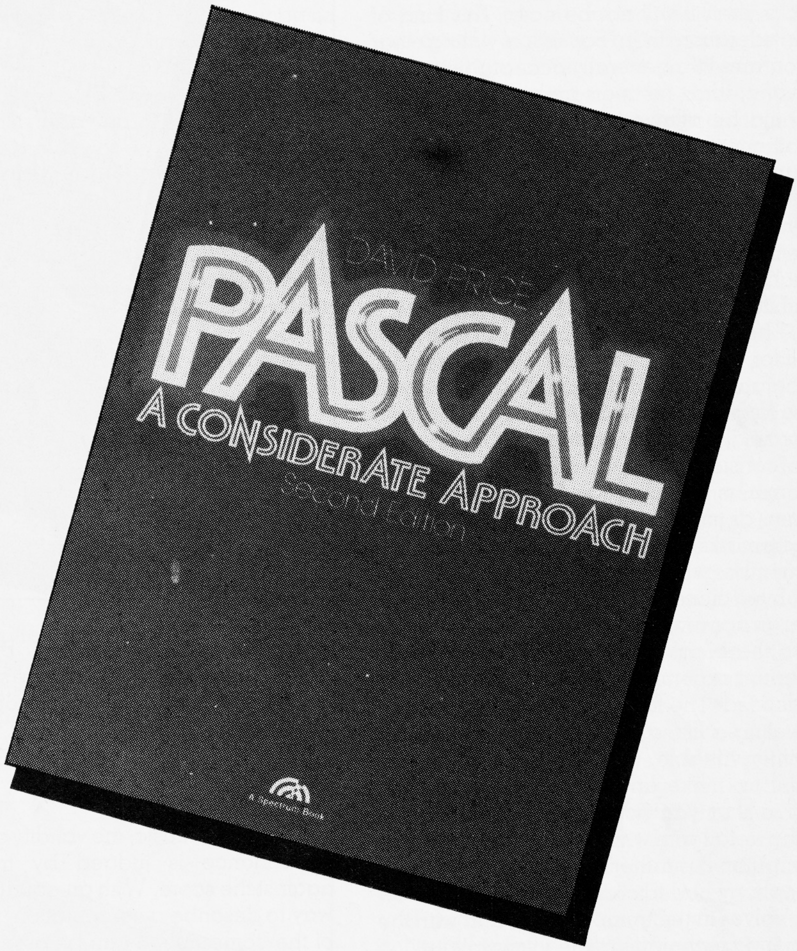
You can move the cursor about the screen with the cursor control keys and you can additionally scroll the screen up with <control> F and down with <control>B

Learn UNIX is continued in the next edition of *Computing Today.*

BOOK PAGE

Garry Marshall

Three varied books given the once-over by our reviewer.



I saw 'Amadeus' at the cinema recently - a tremendous film. Forgive me if you know, but the film is about Wolfgang Amadeus Mozart. It shows his life as seen by Salieri, a competent composer of the time, who is intensely jealous of Mozart but, at the same time, the only person to appreciate the quality of his work. The film catches the effortless ease with which Mozart must have composed, showing the music flowing, fully realised, from him.

Now, we are unlikely to be Mozarts: certainly not in musical terms and almost as certainly not in computer terms (whatever they might be). But we can imagine that we could be computer Mozarts, composing the computer equivalent of a symphony with effortless ease. And if we could, how would we express it? I just don't think that it would come out in BASIC. Could Pascal be the answer?

Still supposing: if we could come to computing for the first time with no preconceptions, no half-baked ideas about what is good and what is bad - with a completely open mind, in fact - where could we find an introduction to Pascal that would provide us with a notation to express our computational ideas as freely as the standard musical notation allowed Mozart to express his musical ideas? I don't know the answer of course, but there is a book that is at least a candidate. It is **Pascal: a considerate approach** by David Price.

This isn't just another book about how to program in Pascal, it is a book that can provide a sensible approach to Pascal in particular and to programming in general to the person coming to computing with a 'clean slate'.

The book introduces Pascal from adding two numbers together through to the more advanced matters such as file handling and pointer variables. The ways in which it claims to be different, and with justification, are by presenting the principles of Pascal programming with a minimum of theoretical discussion to make it suitable for readers learning to program for the first time, by conforming to the standard definition of Pascal, and by using short example programs to illustrate and provide insights into the programming process.

It succeeds in these aims by presenting what seems to me to be just enough information and no more. I am a great believer in making a book just as long as it needs to be and no longer: this one doesn't waste a word. The text is well written and the example programs are provocative. In conjunction with a pleasant typeface and good production, this makes the book a pleasure to read while it conveys what it has to say about its chosen subject.

By 'considerate programming', the author means three things. Programs should be written with consideration for their users so that the person using the program need never be in any doubt about what the program does and about what he has to do to make the program do it. This means making the program produce considerate output. No messages like

?

when "How many points?" would be better and "How many points do you have (1 - 50)?" would be better still. This also implies that the user of a program should never need to look at the listing of a program at all, which sounds obvious really, except that it might be the only way to find out what an inconsiderate programmer expected of you.

Consideration should also be shown for the people who may one day need to read your program. Other programmers might need to improve or extend your program, or they might want to borrow from it. If your program is written in a clear and straightforward manner, then this will all be easy. But if your program is not clear, reflecting your unclear thoughts in the first place perhaps, or uses tricks, then it will not be easy. This kind of consideration, then, consists of writing clear programs in a generally acceptable way. It means that no personal idiosyncracies should be allowed to creep into the programming. That special methods intended to increase efficiency should not be used, but that readability should always be the first concern. That programs should be laid out in the most helpful way, with one statement on each line and pretty printing used to highlight the organisation of the program.

The third target for coinsideration is yourself for, even if no-one else ever reads your program again, you probably will. And you can forget what it is about and how it works amazingly quickly. So you will benefit from the consideration you show to other programmers who may need to read your programs as much as they will, and if you add a few remarks to them it will be even easier to remember what they were all about.

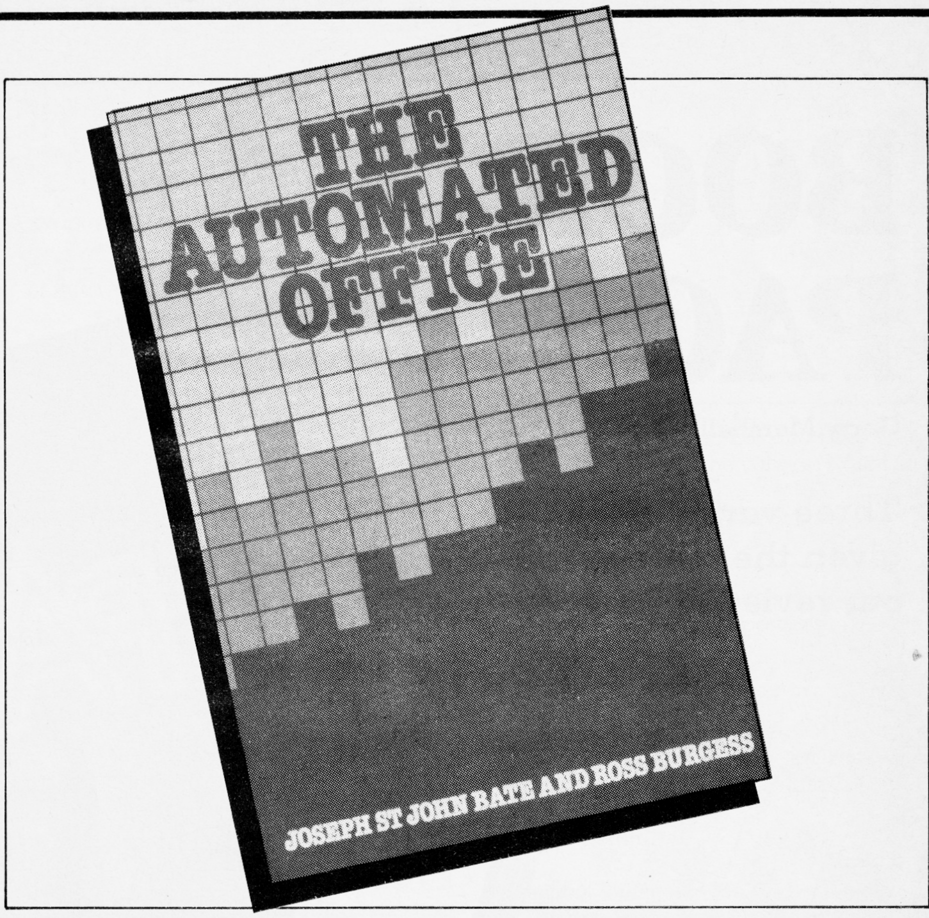
Even though a book like this one can give you an appreciation of how to communicate with the computer, it can only present an account of communication as it is currently practiced. The fact that when you are constructing a list, say, you have to declare a pointer variable, create a variable for it to point to, make an assignment and so on, shows that you have to be concerned in some detail with what is going on inside the computer. It is rather as if the composer of a symphony had to concern himself with how the valves in the trumpets work and with the fingering technique of the third violinist.

Pascal does not provide computing's equivalent of musical notation. The score for a piece of music totally describes the piece. While leaving scope for interpretation by the creators of the music, the score fixes the sound that is to be produced. The nearest equivalent to this at present lies with languages such as Prolog, with which a problem is described to the computer, thereby fixing it, so that the computer can then interpret for itself how to go about solving it.

Price illustrates the importance of testing software for its reliability, in the context of Pascal programs, with referece to the pretty hoary example of solving a quadratic equation by using the formulae for its roots. But he extracts some new milage from it. The testing of a program should include test cases of three kinds: normal cases, exceptional cases and invalid cases. He shows how all three cases can be designed for this problem.

When solving the equation $A*X^2+B*X+C=0$ using the formula X+(-B+sqrt(B*B-4*A*C)/(2*A) and its companion, the following values for A, B, and C provide examples of the three cases

(1) Normal: A=1, B=2, C=1



(2) Exceptional. A=5, B=4, C=3 (because it gives a negative square root) and A=0, B=5, C=1 (because it gives division by zero).
(3) Invalid. A=0, B=0, C=1 (because it gives 1=0).

Back with the music, the validity of a musical performance is judged by matching it against the score. With an equally powerful way to describe a computation, the validity of the computation can be assessed, in an analogous way. With a language such as Prolog, it is done, essentially, by ensuring that the description of the problem is complete and correct. This avoids the futility of trying to test all the possible eventualities in a conventional program when the number of their combinations is expanding enormously.

And back with the book, it provides a good introduction to Pascal and programming with Pascal that can be recommended to newcomers to computing and is worth reading for its insights even if you are not a newcomer.

THE AUTOMATED OFFICE by J Bate and R Burgess at last gives us a book that was made by taking full advantage of the technology that it describes. The authors say in their preface that the plan and rough draft for the book were produced using Caxton's 'Brainstorm' ideas processor. The draft was then converted to document form and polished using Word Star and the Xionics text processor, after which SpellStar was used to check the spelling. the computers used were the IBM Personal Computer and Compaq portable computer, and the Xionics and HiNet local area networks were also used.

Full marks for this. But not even the most modern technology can help if you write paragraphs such as this one from page 49 of the book:

> Originally, each microcomputer had its own specific operating system. Most of the home micros still have specific operating systems, although the Japanese MSX machines are a move towards standardisation. Apple also still have various operating systems specific to their various ranges of machine. But the dominant operating systems today are not specific to one machine.

Phew! Could it be that each of these sentences went into 'Brainstorm' as an individual idea, and that they were then merged to make a paragraph that was never edited or revised on the word processor?

Nor can the technology save you from howlers such as, from page 31: "The nineteenth century, indeed, was not really the smooth middle class Victorian world of the Forsyths." I didn't know Bruce was that old. What? Oh, *Forsyte.* (The spelling checker must surely have highlighted the word 'Forsyth', so the technology did its best to help.)

Actually, the book isn't so bad. It gives a coherent account of the way that an office works, expressing its function in terms of processing and communicating information, and goes on to explain how the new technology can deal with these activities and improve the efficiency and effectiveness of the office at the same time. The ways in which this can be done are illustrated by giving examples of the technology, including executive workstations, PABXs, local area networks and so on.

The book's treatment is not particularly deep. Its attempts to involve more generally applicable items of the technology, from home computers to expert systems, are unconvincing.

One positive benefit to the book from the use of the aids of the technology is its strong structure. The authors would probably have planned it carefully in any case, but their use of 'Brainstorm' must have encouraged, and even required, the top-down development of a structure for the book. This can be illustrated by laying out the titles of sections and sub-sections for a chapter, and I have chosen chapter three (although any other would have served) to illustrate the structure and, as it seems to me, the result of using 'Brainstorm'. The headings are:

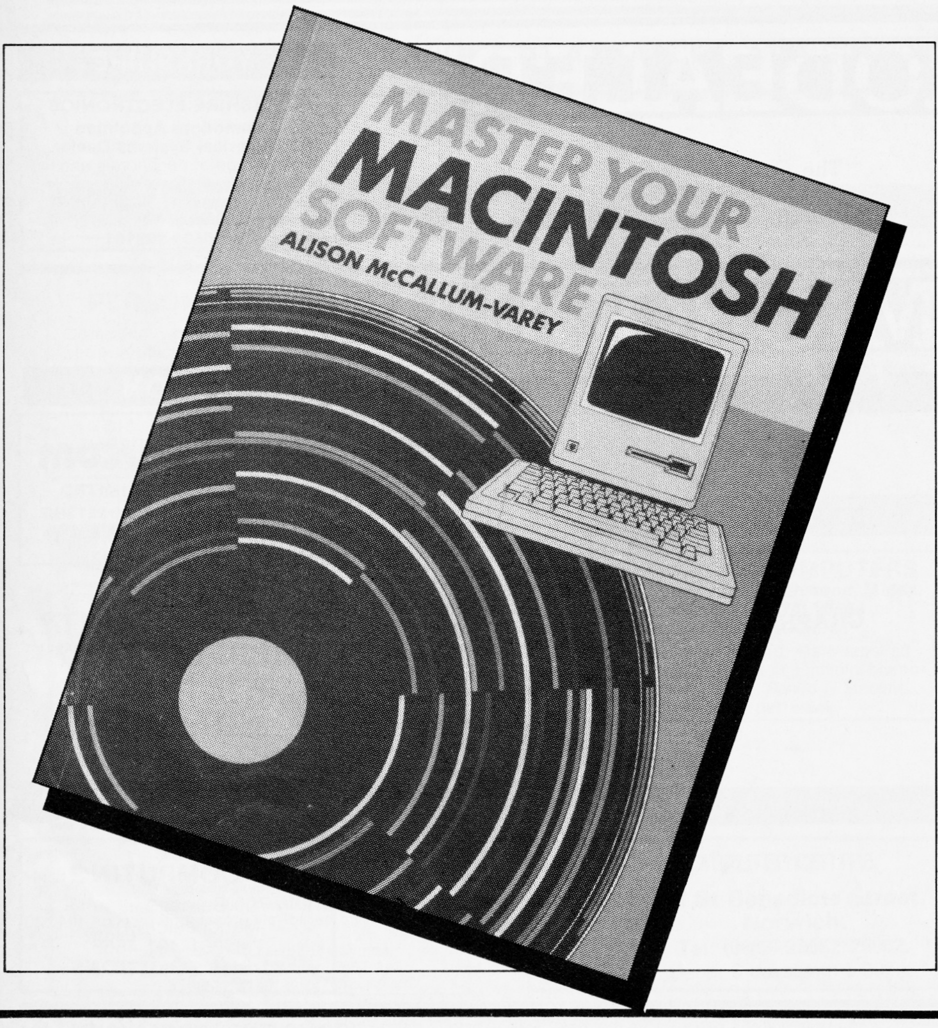
Concepts and goals of office automation
Examples from history
The effect within the office
The falling costs of equipment
Salary costs
Investment in staff and in equipment
Numbers of staff
Low productivity in the office
Paper by the mile
Measuring productivity
  Counting keystrokes
  Analysing information flow
  Assessing business effectiveness
Expanding business without expanding costs
The impact on profitability
  Cost reduction
  Saving time
Supporting the business environment
  Enhancing the quality of the work
  Widening the span of control
  Better tools for analysing results

These headings show exactly what the chapter is all about. And when you have read them you have read ninety per cent of what the chapter has to say about the concepts and goals of office automation. The small sections and sub-sections in the chapter really have very little to add to our basic general knowledge of, and to the ideas suggested by, their headings.

In case I seem to suggest it, let me say with emphasis that I am not against the use of the new technology as an aid to writers and as an aid to writing. In fact, I am all for it. Nor would I want to do anything that might discourage, even in the very smallest way, anyone from using these aids. But I do think that writers might make sure that they have mastered the technology to the extent that it does not degrade their output, and hopefully until it enhances it, before they impose the results on their readers.

MASTER YOUR MACINTOSH SOFTWARE By Alison McCallum-Varey explains how to get as much as you want from Finder, the Macintosh operating system, and from MacWrite and MacPaint, the word processor and sketching program that are supplied with it.

Starting from icons and the mouse, we move through clicking the mouse, dragging, pull-down menus and folders, to get a quick run-through of how to use the Macintosh and, at the same time, an appreciation of what it can do. Then we move on to the word processor and the sketching program. Their capabilities are explained and illustrated by lots of imaginative examples that are accompanied by illustration direct from the screen or from printed output.



The use of MacWrite to create letters, lists and tables with the standard word processing facilities is followed by more adventurous uses such as producing invitations and advertising material. The ways in which the capability to display text in different fonts can be used to good effect are explored, and the illustrations show very convincingly how the creative use of these facilities allows documents and notices to be created that not only convey their message, but at the same time convey precisely the image that the sender wishes to create. The medium reinforces the message, in fact.

Moving on to MacPaint, the sometimes-expressed idea that this is nothing more than a program for creating doodles is soon dispelled. It is shown creating detailed maps, flowcharts, plans of the arrangement of a warehouse, company logos, packaging symbols, holiday carts and more. The final example shows how a form can be designed, the example is of a job application form, and this leads quite naturally to the final chapter which is about the integration of MacWrite and MacPaint. Integration allows, for example, a form created with MacPaint to be passed to MacWrite to be filled in, and an illustration to be passed over for inclusion in a document.

The author claims that her book is intended to show business users of the Macintosh how to use it creatively. It succeeds admirably in this, and I think that it would show any Macintosh user how to get more from it. To go further, the main attraction of the book for me is that it shows very clearly and unpretentiously exactly what can be achieved with the best of the currently available hardware and software. Be warned that it gives the sort of demonstration that will make you reluctant to return to using anything inferior!

Odd things can come into your head when reading a computer book (well, into mine, anyway). At the risk of having others join my niece in calling me jokey, I record the following: David Price thanks his sister for her help with his Pascal book. Her name is lisa. So what's the Lisa Price? Answer: About ten grand at the moment. (With acknowledgements to "What's a Greek urn?" and apologies to Lisa Price) *(that's enough jokes - Ed)*

This month's books are:

Pascal: a considerate approach by David Price (Prentice-Hall), 181 pages, £12.55.
The Automated Office by J Bate and R Burgess (Collins), 182 pages, £7.95.
Master Your Macintosh Software by Alison McCallum-Varey (Sunshine), 116 pages, £8.95.

ACT MICRODEALER

xi APRICOT

CPU	8086
MEMORY	256K RAM
LANGUAGES	Microsoft BASIC, Personal BASIC
MASS STORAGE	No cassette drive Integral Sony 3½" 315K microfloppy disk drive Integral 5 or 10 Mb hard disk
OS	MS-DOS 2.11 with GSX bundled CP/M-86 (not yet available) Concurrent CP/M-86 (not yet available)
KEYBOARD	QWERTY, cursor, numeric pad, function keys
INTERFACES	RS-232C, Centronics, Microsoft mouse
DISPLAY	Monitor (supplied)
GRAPHICS	80 by 24 text with block graphics 800 by 400 high-res graphics under GSX
SOUND	No

Notes. The Apricot xi is a development of the award-winning Apricot, and replaces one of the latter's disk drives with an integral hard disk, providing vastly increased storage with faster access. Memory may be expanded in 128K increments to a maximum of 768K. The languages and operating systems mentioned above come bundled (except for Concurrent CP/M) and four software tools are also bundled, including an asynchronous package for use with the optional modem card.

CBM MICRODEALER

Notes: The Commodore 64 is a popular micro with a great deal of games software available. There is also some business software available.

The Commodore 715B is the top model in the 700 range of business machines.

NASCOM MICRODEALER

NASCOM 3

CPU	2 MHZ Z80	DISPLAY	40 or 80 column 25-line display
MEMORY	8K or 32K inbuilt RAM (expandable to 60K)		
LANGUAGE	Full Microsoft BASIC	GRAPHICS	High resolution graphics with 8 foreground and 8 background colours (400 x 256 pixels) Double density graphics with 2 colours (800 x 256 pixels)
MASS STORAGE	Single or twin 5.25" disc drives 350K capacity per drive		
OS	NAS-DOS or CP/M 2.2	SOUND	No
KEYBOARD	Full size QWERTY		
INTERFACES	RS232 and 16-bit parallel		

SHARP MICRODEALER

SHARP MZ-3541

CPU	Z80A (two), 80C49
MEMORY	128K RAM, 8K ROM
LANGUAGE	Sharp BASIC
MASS STORAGE	Twin integral 5¼" floppy disk drives, total capacity 1.28 Mb
KEYBOARD	QWERTY, cursor, numeric pad, function keys
INTERFACES	RS-232C, Centronics, interface for extra external floppy disks
DISPLAY	Monochrome monitor, colour optional
GRAPHICS	80 by 25 text, 640 by 400 high-resolution graphics
SOUND	Single channel

Notes: The Sharp MZ-3541 is aimed at the businessman. RAM is expandable to 256K, while two disk drives may be added externally to complement the integral pair. Colour is only possible with the optional graphics expansion RAM. One Z80 handles the main CPU activities while the other handles peripheral activities. The third processor handles the keyboard. The availability of CP/M means a ready supply of business software.

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

COMPUTING TODAY

Lineage: 40p per word.
Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
CAROLINE FAULKNER
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1.

HARDWARE

NEWBRAIN & SANYO
Professional Micro Computers for the prices of hobby machines.

NEWBRAIN ON SPECIAL OFFER
£99 PLUS VAT
Model AD with free beginners guide and tape, investment software and UK postage. As above plus about **£100** value of business/general software **£129** + VAT.
Limited offer-ring now
SANYO 550/555 COMPUTERS
Micropro Wordstar, Calcstar etc at no extra cost! **Printers:** Epson, Canon, Juki, Kaga, Daisystep 2000 etc.
Monitors & Recorders.
Call STEVENAGE (0438) 812439
anytime for hardware/software lists
Mail Order and **Access** facilities
ANGELA ENTERPRISES
4 Ninnings Lane, Rabley Heath, Welwyn, Herts AL6 9TD.

AMSTRAD accessories, video and power extension lead:- £9.95 stereo amplifier, connect two speakers for stereo:- £15.95 inclusive. Amware, 15 Heath End Road, Flackwell Heath, Bucks, HP10 9DT.

SOFTWARE

T199/4A BASIC
Seven games on two tapes one price. Bagbounder Food of the Gods, 3D Maze Hunt, Eat-Up, Laundry Man, Lower and Higher and Rebound Sector **£6.95.** 50% PROFITS FOR ETHIOPIA. P&P free. Send cheques or PO's to: **KABEE SOFTWARE, 73 THORNES PARK, RASTRICK, BRIGHOUSE, W. YORKS HD6 3DA.**

SOFTWARE

££ TAX MASTER! ££

END YOUR INCOME TAX WORRIES!
A user friendly tape/disc, password protected, 2nd processor, etc. (view change and print-out operations) **£9.75**
ANDERSON'S
56 Waterloo Rd., Freemantle, Southampton.

CONFUZION
BY INCENTIVE
£6.95

TURBO PASCAL
Extended Pascal for PC DOS, MS DOS, CP/M£86 and CP/M-80, includes full screen editor, floating point arithmetic, full string handling feature, random access data files, complies faster than IBM or MT + Pascal, requires less than 35K of disk space, 250 page manual and FREE spreadsheet program written in Turbo Pascal.
****ONLY £54.95****
All prices fully inclusive for prepaid orders.
CONGUIN SOFTWARE,
14 GOODWOOD CLOSE, MORDEN, SURREY SM4 5AW.
No callers please Phone 0524 381423

COLOUR GENIE owners quiz game £2.99 coming soon — Quiz Master £4.99, Fires of Mordor, 41 Pexwood Rd, Tormorden, Lancs.

DISKS

DISKS
3M—TDK—BASF

SS DD 40T	14.64
DS DD 40T	19.99
SS DD 80T	21.25
DS DD 80T	25.25

All Prices Box 10 And Include VAT & P+P.
Send cheque stating:
Qty, Brand and Type to:
CAROUSEL TAPES
"Disks", 3 Park Parade, Stonehouse, Glos GL10 2DB.
Tel: 0453-82-2151

REPAIRS

MICROSERV for expert repairs to BBC machines and disk drive. For details call or write to Unit 4. Denny Workspace, Denny, Scotland. FK6 6DW. Tel: Denny (0324) 823468.

HOME COMPUTER REPAIRS
Look at our fantastic prices on repairs!

BBC B	£27.50
VIC 20	£20.00
COMMODORE 64	£27.50
DRAGON	£30.00
ORIC/ATMOS	£25.00
ZX SPECTRUM	£17.25
ZX INTERFACE	£17.25
ZX MICRODRIVE	£17.25

PLUS OTHERS!
The above prices are inclusive of parts, labour, P&P. All repairs carry 6 months warranty on replaced parts. Extended warranties, peripheral repairs, upgrades etc. etc. All available. Ring for full details (0234) 213645.
ZEDEM COMPUTER LTD
2 Kimbolton Rd, Bedford.

ALARMS

BURGLAR ALARM Equipment. Please visit our 2,000 sq. ft. showrooms or write or phone for your free catalogue. CWAS Ltd., 100 Rooley Avenue, Bradford BD6 1DB. Telephone: (0274) 731532.

UTILITY SOFTWARE

*** UNLOCK YOUR AMSTRAD***
Amskey
EASY TO USE utility program. Removes protection and allows listing, studying and copying of your precious software.
YOU CHOOSE loading speed and protection of your back-up copies.
Only **£6.99 incl. p&p.** Overseas please add £1.00 postage.
Interlock Services Ltd. Dept C.T.
37B New Cavendish St, London W1M 8JR. Tel: 01-609 8301.

COURSES

LEARN TO USE your computer on a weekend/holiday course. Details from:- Jaysoft MICRO Developments, 2 Wester Row, Greenlaw, Berwickshire.

COMPUTING TODAY
CLASSIFIED ADVERTISEMENT — ORDER FORM

If you have something to sell now's your chance! Don't turn the page — turn to us!
Rates of charge: 40p per word per issue (minimum of 15 words)× 15% VAT. Please state classification and post to: **COMPUTING TODAY, CLASSIFIED DEPT., 1 GOLDEN SQUARE, LONDON W1.**

Please use BLOCK CAPITALS and include post codes.
Name (Mr/Mrs/Miss/Ms) ....
(delete accordingly)
Address ....
....
....
Signature .... **Date** ....
Daytime Tel. No. ....

Please place my advert in **COMPUTING TODAY** for .... issues commencing as soon as possible.

BLANK DISKS

5¼in blank discs (40 track)
Prices per box of ten

	single-sided/ double-density	double-sided/ double-density
Scotch 3M	£14.95	£19.95
Dysan	£16.95	£24.95
Memorex (unlabelled with free case)	£12.95	£16.95

S-J-B SUPER SAVERS!!!

50 Memorex SS/DD disks supplied in a perspex storage box.
only £59.95

50 Memorex DS/DD disks supplied in a perspex storage box
only £74.95

We also supply 3″ and 3½″ disks

FURTHER DISCOUNTS ON BULK ORDERS

CREDIT ACCOUNTS AVAILABLE

FREE FAST DELIVERY ON ALL ORDERS

ALL PRICES INCLUDE VAT

Please send cheques/PO's to:

S-J-B DISK SUPPLIES
DEPT CT 11 OUNDLE DRIVE, WOLLATON PARK, NOTTINGHAM NG8 1BN
TELEPHONE: 0602 782310

SYSTEM SCIENCE

C Compilers

DeSmet C	PC/MS-DOS	**£135.00**
Lattice C	PC/MS-DOS	**£450.00**
C86 & optimiser	PC/MS-DOS	**£365.00**
Microsoft C ver 3.0	PC/MS-DOS	**£475.00**
Aztec C86	PC/MS-DOS	**£225.00**
C80 requires LINK	PC/MS-DOS	**£50.00**
Aztec C][	App-DOS	**£175.00**
Aztec C II	CP/M-80	**£175.00**
C/80 Software Toolworks	CP/M-80	**£50.00**
C/80 Mathpak — floats and longs	CP/M-80	**£30.00**
BDS C	CP/M-80	**£125.00**
ECO-C for Z80 Code	CP/M-80	**£185.00**
VENIX-86 full Unix for IBM PC-XT and PC-AT		from **£850.00**

LISP Interpreters

LISP-80 Software Toolworks	CP/M-80 & MS/PC-DOS	**£45.00**
MuLISP/MuSTAR	CP/M & MS/PC-DOS	from **£175.00**
MuMATH	CP/M & PC/MS-DOS	from **£215.00**
IQ LISP	PC-DOS	**£160.00**

FORTH-83 from Laboratory Microsystems

Z80 Forth, PC-FORTH, 8086-FORTH	**£89.00**
Floating point extensions	**£89.00**

Editors

SEE Editor for IBM-PC and Apricot	**£50.00**
EC Editor (windows and DOS calls) PC-DOS	**£125.00**
PMATE IBM-PC and Apricot	**£225.00**
FirsTime C (with syntax checker)	**£295.00**
FirsTime Pascal (with syntax checker)	**£245.00**
Final Word for IBM-PC, Apricot and CP/M-80	**£295.00**
Tools — many C tools available	
dBase to C conversion package	**£995.00**
Crosstalk XVI communications	**£165.00**
Pascal Compilers incl Turbo 3.0	**£ call**
Fortran Compilers	**£ call**
Assemblers and cross-assemblers	**£ call**

HSC 16 bit Co-Processors for Z80 CP/M systems
— choice of 8086 or 68000 — 6MHz clock
— MS-DOS, CP/M-86 and CP/M-68K — 256Kb to 1.25Mb memory
— use as RAM DISK under CP/M — fit most Z80 systems
— simple to install, simple to use. Prices from **£625.00**

Prices are exclusive of VAT and postage.

6-7 West Smithfield, London EC1A 9JX Tel: 01-248 0962

NEW LOW PRICES

WHY PAY MORE

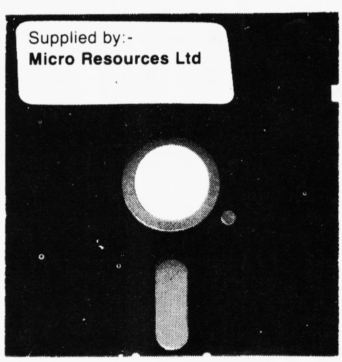
NEW LOW PRICES

QUALITY SS/DD AND DS/QD DISCS IN THEIR OWN LIBRARY BOX

CERTIFIED 100% ERROR FREE AND FULLY GUARANTEED

We are able to offer these prices due to our policy of bulk buying from a major manufacturer and feel that we can offer you the customer a better deal. Why not take advantage of these low prices.

48 HOUR DELIVERY SERVICE
24 Hr. ANSWERING SERVICE



PRICES INCLUDE DELIVERY PER BOX OF 10

PRICES EXCL. VAT	1—4	5—9	10+
SS/DD 48 TPI	12.50	12.00	11.50
DS/QD 96 TPI	17.50	17.00	16.50

SPECIAL OFFER BULK PACKED IN 25's WITH ENVELOPES — PRICES INCLUDE DELIVERY

PRICES EXCL. VAT	25	50	100
SS/DD 48 TPI	27.50	52.50	100.00
DS/QD 96 TPI	40.00	77.50	150.00

PLEASE SEND PAYMENT OR PHONE TO USE CREDIT CARD
LARGE QUANTITY AND EDUCATION DISCOUNTS GIVEN

Micro Resources Limited

Southfield House, 11 Liverpool Gardens, Worthing, Sussex BN11 1RY
Telephone: Worthing (0903) 213174

ATARI ST

520ST

POWER WITHOUT THE PRICE

THE NEW ATARI 520ST

Under the new leadership of Jack Tramiel (former boss and founder of Commodore Business Machines), Atari Corporation have marked their entry into the world of business/personal computers with a machine which leaves the competition standing. Tramiel's slogan 'Power Without the Price' has been implemented in the manufacture of the new 512K Atari 520ST colour computer which offers the user amazingly high performance at an incredibly low price. Launched as a work-station, this new system incorporates seven software packages as well as the 520ST computer with 512K RAM, mouse controller, high resolution monochrome monitor (640x400), 95 key keyboard (with 18 key numeric keypad), MIDI interface, GEM and a 500K 3½ inch disk drive, all for the package price of only £651.30 (+VAT = £749). Dubbed the 'Mac beater' and the 'Jackintosh' (after Atari's Chief, Jack Tramiel), Atari's new machine has been directly compared with the Apple Macintosh RRP £2595 (+VAT = £2985) which offers similar features and capabilities but at a much higher price. Favourably reviewed by the UK's highly critical specialist computer press, the 520ST is likely to make a great impact in this country as a sophisticated alternative to an IBM PC, APRICOT or APPLE MACINTOSH. Unlike its overpriced competitors, the Atari 520ST can be linked up to a colour monitor to unleash a choice of up to 512 colours. The addition of colour brings out the full potential of graphics packages such as GEM.

USER FRIENDLY GEM OPERATING SYSTEM

The power of the ST is harnessed and made user friendly by the new operating system 'GEM' from Digital Research. GEM stands for Graphics Environment Manager and allows a user friendly colour or B/W graphics interface which closely resembles that of the Macintosh. This similarity extends to the use of moveable resizeable windows, icons to represent objects such as disks and disk drives, and the use of pull down menus and a mouse. The advantage of all this is that the computer becomes extremely easy to use. GEM has now been implemented for the Acorn, ACT, Atari, IBM, ICL, and Olivetti. Software written for GEM on one computer should also run under GEM on another computer. This will enable the market to quickly produce a large library of standard interchangeable software.

FREE SOFTWARE AND FUTURE EXPANSION

The Atari 520ST comes supplied with seven free software packages as listed below: 1) TOS -Tramiel Operating System based on CPM 68K. 2) GEM Graphics Environment Manager by Digital Research (DR) giving a WIMP (Window, Icon, Mouse, Pull down menu) environment. 3) DR GEM Paint for creating graphics masterpieces. 4) DR GEM Write for word processing. 5) Logo learning language to enable you to write your own programs easily using turtle graphics. 6) DR Personal Basic a powerful user friendly version of the Basic programming language. 7) BOS operating system giving you access to dozens of business applications packages already available on the market. Designed with future expansion in mind, the ST also features a host of different interfaces to the outside world and an impressive list of accessories is planned. Atari will soon be releasing a 1000K (1MB) 3½ inch disk drive, and a 15MB hard disk storage system as well as a mass storage compact disk (CD) player capable of storing an entire 20 volume encyclopedia on one disk. A full range of inexpensive printers are planned including dot matrix, daisywheel and thermal colour printers. With its unbeatable graphics, speed and software at a price which is far below that of any comparable personal computer currently on the market, the ST is all set to do battle with the competition. To receive further details of the ST from Silica Shop, just fill in the coupon below with your name and address details and post it to us.

Silica Shop Price: £651.30 + £97.70 VAT = £749.00 This price includes:

- ★ **512K RAM**
- ★ **MOUSE**
- ★ **GEM**
- ★ **B/W MONITOR**
- ★ **500K 3.5" DISK DRIVE**
- ★ **KEYBOARD (95 KEYS)**

£749

ATARI 520ST SPECIFICATION

MEMORY
512K RAM (524,288 bytes)
16K ROM expandable to 320K
Port for addnl 128K plug-in ROM cartridges
200K TOS operating system

GRAPHICS
Individually addressable 32K bit-mapped screen with 3 screen graphics modes:
320x200 pixels in 16 colours (low resolution)
640x200 pixels in 4 colours (med resolution)
640x400 pixels in monochrome (high res)
16 shades of grey in low res mode
512 colours available in low/medium res
8 levels of each in red, green and blue

ARCHITECTURE
4 custom designed chips:
GLUE Chip - MMU Memory Mngmnt Unit
DMA Controller - Graphics Processing Unit
16/32 bit Motorola 68000 processor at 8MHz
eight 32 bit data registers
eight 32 bit address registers
16 bit data bus/24 bit address bus
7 levels of interrupts/56 instructions
14 addressing modes/5 data types

DATA STORAGE
High speed hard disk interface
Direct memory access 1.33 Mbytes per second
CD (Compact Disc) interface
Built in cartridge access
Dedicated floppy disk controller

DISK DRIVE
500K (unformatted) s/sided 3½" floppy d/drive
349K (formatted) storage capacity

SOUND AND MUSIC
Sound Generator
Frequency control from 30Hz to above audible
3 voices (channels) in wave shaping sound in addition to a noise generator
Separate frequency and volume controls
Dynamic envelope controls
ADSR (Attack, Decay, Sustain, Release)
Noise generator
MIDI interface for external music synthesizers

KEYBOARD
Separate keyboard microprocessor
Standard QWERTY typewriter styling
Ergonomic angle and height
95 keys including 10 function keys
Numeric keypad — 18 keys including ENTER
One touch cursor control keypad

MONITOR
12" screen - high res monochrome monitor
640x400 monochrome resolution

VIDEO PORTS
Display - Low Resolution - 40 columns
Med/High Res - 40/80 plus cols
Medium res RGB (Red/Green/Blue) output
High resolution monochrome (Black & White)

COMMUNICATIONS
Bidirectional centronics parallel interface for printers, or modems capable of input/output
RS232C serial modem/printer interface
VT52 Terminal Emulation Software
Maximum Baud Rate up to 19,200
High speed hard disk interface
Floppy disk controller (Western Digital)
2 joystick ports (one for 2 button mouse)
MIDI interface for external music synthesizers

GEM WIMP ENVIRONMENT
WIMP - Window Icon Mouse Pop-down menus
Two button mouse controller
Icons/Pull down menus/Windows
GEM VDI - Virtual Device Interface
GEM AES - Application Environment Services
GEM BBT - Bit Block Transfer
Real time clock & calendar

SOFTWARE
GEM environment
with user friendly Macintosh style operation
'TOS' - Tramiel Operating System
Atari's own system based on CP/M 68K with hierarchical directory & file structure plus a host of MS DOS & UNIX command structures
'BOS' - Business Operating System
to run any standard BOS business programs
GEM desktop
with GEM PAINT graphics mgmt system and GEM WRITE word processor
Personal BASIC and DR Logo
originally written by Digital Research (DR) Very much like those on other machines except for the extensive use of pull down menus, mouse control and windows

VARIOUS
Dimensions: 470mmx240mmx60mm
Replaceable external power supplies
Expansion: 3½" floppy disk drives 500K/1,000K
(two drives can be connected)
3½" 15Mb hard disk
CD (compact laser disc)
Dot matrix & d/wheel prntrs (black)
Thermal dot matrix (colour)
RGB & monochrome monitors

LANGUAGES
BASIC & LOGO supplied
Many others will soon be available, including:
Assembler, BCPL, C, Cobol, Compiled Basic
Lisp, Modular-2 and Pascal

Note: Some of the above specifications are pre-release and may therefore be subject to change

MACINTOSH v F1e v 520ST

"Imagine a Fat Mac - the 512K Apple Macintosh - but with a bigger screen, a far bigger keyboard with numeric keypad, cursor and function keys, and colour. That gives you some idea of what the Atari 520ST is like, except for two important things. First the Atari seems faster. Second the Atari system is about one third of the price."
June 1985 - Jack Schofield - **PRACTICAL COMPUTING**

	APPLE	APRICOT	ATARI
FEATURES OF BASIC SYSTEM	MACINTOSH	F1e	520ST
Price Includes B/W Monitor	YES	NO - extra £200	YES
Keyboard size mm (LxDxH)	330x147x50	450x167x28	470x240x60
Keyboard size ins (LxDxH)	13x5¾x2	17½x6½x1	18½x9½x2½
3½" D/Drive (Unformatted)	500K	500K	500K
3½" D/Drive (Formatted)	399K	315K	349K
WIMP (Window, Icon, Mouse ...)	Apple	ACT - Activity	GEM
Real-time Clock	YES	YES	YES
Polyphonic Sound Generator	YES	NO	YES
RS232 Serial Port	YES	YES	YES
Centronics Parallel Printer Port	NO	YES	YES
Dedicated Floppy Disk Controller	NO	YES	YES
Hard Disk DMA Interface	NO	YES	YES
Full stroke keyboard	YES	YES	YES
Number of keys on keyboard	59	92	95
Numeric Keypad	NO	YES (16 Keys)	YES (18 keys)
Cursor Control Keypad	NO	YES	YES
Function keys	NO	10	10
16-bit processor	68000	Intel 8086	68000
Processor running speed	8MHz	4.77MHz	8MHz
RAM size	512K	256K	512K
Number of graphics modes	1	4	3
Number of colours	Monochrome	16	512
Max Screen Resolution (pixels)	512 x 342	640 x 256	640 x 400
Mouse included	Single Button	NO - extra £95	Two Button
Replaceable External Power Pack	NO	NO	YES
Cartridge Socket	NO	NO	YES
Joystick Ports	NO	NO	YES (two)
MIDI Synthesiser Interface	NO	NO	YES
Monitor Size	9"	9" - extra £200	12"
RGB Video Output	NO	YES	YES

System Cost with: Mouse - Monochrome Monitor - 512K RAM - 500K Disk Drive

Price of basic system (exc VAT)	£2595+VAT	£595+VAT	£652+VAT
+ Mouse	Included	£95+VAT	Included
+ Monochrome Monitor	Included	£200+VAT	Included
+ Expansion to 512K RAM	Included	£295+VAT	Included
Price of complete system (exc VAT)	£2595+VAT	£1185+VAT	£652+VAT
PRICE rounded down including VAT	£2,984	£1,362	£749

PRESS COMMENT

"Atari's new corporate image as an aggressive low cost computer maker is likely to mirror that of Commodore where Mr. Tramiel established the maxim that 'Business is war'."
August 21st 1984 **FINANCIAL TIMES**

"This is the only personal computer I know of that comes with a Midi interface as standard."
Peter Bright March 1985 **PERSONAL COMPUTER WORLD**

"The (GEM) version running on the Atari 68000 machines will have the additional advantage of leaving the PC version standing." April 6th 1985 **PERSONAL COMPUTER NEWS**

"It would seem that GEM offers the ideal operating system."
March 7th 1985 **POPULAR COMPUTING WEEKLY**

"I found it (GEM) extremely easy to use and was very impressed with the way in which it disguises the unfriendly hardware and operating systems lurking under the surface."
Peter Bright Feb 1985 **PERSONAL COMPUTER WORLD**

"The electronics in the machine are a work of art ... The heart of the 520ST is a Motorola 68000, one of the most powerful 16-bit processors around and in many respects it is close to being a 32-bit chip ... when the machine appears in the shops, I'll be at the front end of the queue to buy one."
Peter Bright June 1985 **PERSONAL COMPUTER WORLD**

"This machine is significantly more powerful than an IBM PC ... If it's possible to design a sure-fire winning machine, this is it." May 11th 1985 **PERSONAL COMPUTER NEWS**

"... the use of GEM makes the new range of Atari computers so similar to the Macintosh (with the added attraction of colour), that they are already being called 'Jackintoshes'."
May 2nd 1985 **COMPUTING**

"The new Atari ST computers truly represent to the consumer what Jack Tramiel is saying — easy-to-use computing power without the price." March 1985 **ANALOG COMPUTING**

"It (the ST) uses the most modern technology that is affordable, in a package that gives a professional impression."
May 23rd 1985 **POPULAR COMPUTING WEEKLY**

"The Atari ST is one of the most elegant designs I have seen ... Atari has used an original and elegant method of memory management which should make the ST faster than any other PC on the market - in any price bracket ... The 64K dollar question is would I go out and spend money for one? To which the only answer is 'Try and stop me!'"
John Lambert July 1985 **ELECTRONICS & COMPUTING**

"The 520ST is technically excellent ... The 520ST hardware is the new standard by which others will be judged."
July 1985 **YOUR COMPUTER**

ATARI SILICA SHOP ATARI

WE ARE THE UK's No1 ATARI SPECIALISTS

At Silica we have been successfully dedicated to Atari ever since their products first appeared on the UK market. We can attribute our success largely to the Atari specialisation which we practice and to the user back-up we provide. Rest assured that when you buy a piece of Atari hardware at Silica you will be fully supported. Our mailings giving news of software releases and developments will keep you up to date with the Atari market and our technical support team and sales staff are at the end of the telephone line to deal with your problems and supply your every need. With our specialist bias, we aim to keep stocks of all the available Atari hardware, software, peripherals and accessories. We also stock a wide range of Atari dedicated books and through us, the owners on our list can subscribe to several American Atari dedicated magazines. We can provide a full service to all Atari owners and are now firmly established as the UK's NUMBER ONE Atari specialists. Here are just some of the things we can offer to our customers.

- ★ **FREE POST & PACKING ON MAIL ORDERS**
- ★ **FREE NEXT DAY SECURICOR DELIVERY**
- ★ **INFORMATION MAILING SERVICE**
- ★ **TECHNICAL SUPPORT TEAM**
- ★ **HIGHLY COMPETITIVE PRICES**
- ★ **AFTER SALES SUPPORT SERVICE**
- ★ **REPAIR SERVICE ON ATARI PRODUCTS**

If you would like to be registered on our mailing list as an Atari computer owner, or as a person interested in buying an Atari machine, let us know. We will be pleased to keep you up to date with new Atari developments free of charge. So, return the coupon today and begin experiencing a specialist Atari service that is second to none.

SILICA HOTLINE 01-309 1111

SILICA SHOP LTD, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX

SEND FOR FREE ATARI ST LITERATURE

To: Silica Shop Ltd, Dept CT 0885, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX

PLEASE SEND ME FREE LITERATURE

ON THE NEW ATARI 520ST COMPUTER

Mr/Mrs/Ms: .... Initials: .... Surname: ....

Address: ....

....

....

.... Postcode: ....

Do you already own a computer
If so, which one do you own? ....