

MSX

Moer

March 1985 95p

NEWS

MSX DATA

**PACKED WITH
PROGRAMS AND
PROGRAMMING
PROJECTS**

**DISK DRIVE
ROUNDUP**

**PERMANENT
DATA STORAGE**

**PROGRAMS
AND DATA
IN MEMORY**

**MSX DATA
STORAGE
SPECIAL**



**PANASONIC
CF-2700 MSX
REVIEWED**

MSX

"I'M A TOSHIBA HX10.
 I'VE GOT ALL THE
 BEST BITS FROM EVERY
 OTHER HOME COMPUTER.
 AND MORE. I HAVE A
 64K MEMORY, LIKE THE
 COMMODORE 64. A
 CASSETTE INTERFACE,
 LIKE THE BBC. TWO
 JOYSTICK PORTS, LIKE
 THE COMMODORE 64.
 A BUILT IN POWER
 SUPPLY, LIKE THE
 BBC. 16 USABLE
 COLOURS, LIKE THE
 ACORN ELECTRON. OVER
 70 FULL STROKE KEYS,
 LIKE THE BBC. A
 CARTRIDGE SLOT, LIKE
 THE COMMODORE 64.
 A PRINTER INTERFACE,
 LIKE THE ORIC ATMOS.
 SOUND OUTPUT THROUGH
 THE T.V., LIKE THE
 COMMODORE 64. AN
 AUDIO/VIDEO OUTPUT
 CONNECTION, LIKE
 THE COMMODORE 64.
 RF BUILT IN LIKE
 THE BBC. AND: A
 SEPARATE 16K VIDEO
 MEMORY, UNLIKE MOST
 NON-MSX COMPUTERS.
 32 SPRITES, MORE
 THAN MOST NON-MSX
 COMPUTERS. AND I USE
 MICROSOFT EXTENDED
 BASIC, LIKE EVERY
 OTHER MSX COMPUTER."

"WOW. WITH A
 SPECIFICATION LIST
 LIKE THAT.
 NO WONDER YOU'VE
 GOT A 64K MEMORY."

You'd expect one of the best-selling home computers in Japan to have a specification list as big as its memory.

But the Toshiba HX10 doesn't just limit itself to that.

It was developed along with other Japanese home computers to operate

on one language: MSX. You can swap programs, games, cassettes, even peripherals like disk drives, printers, and joysticks: they're all compatible with every other MSX computer.

All of which makes MSX the system of the future.

So if you want a computer that won't be obsolete in a few years, buy an MSX. If you want one of the best-selling MSX computers in Japan, buy a Toshiba HX10.

TOSHIBA **MSX**

Just when you thought it was safe to sit back and relax with your favourite MSX magazine, we have some changes to announce. Your previous editor Liz Coley and her assistant Sean Rothman having moved on to bigger and better things, a new team has stepped into the breach, namely myself as editor and Eric Doyle as editorial assistant, poor soul. Bear with us, and we will endeavour to entertain and inform you on every aspect of the MSX scene in the manner to which you were almost becoming accustomed.

The MSX scene provides an exciting challenge. In spite of gloomy talk of slow supplies, high prices and poor Christmas sales, we here at MSX User have faith in the future. A standardised computer needing no specialist knowledge and designed to become an integral part of a home entertainment system makes perfect sense, so much so that it need not rely on seasons or passing fads to grow and flourish. If MSX makers have any sense, prices will inevitably drop (and a standard *will* be maintained), sales will improve, and the supply of worthwhile software and interesting peripherals will follow suit.

However, there is no need to preach to the converted – if you are reading this, you have already bought, borrowed or been given an MSX, and quite right too. Many of you have written – most encouragingly – to tell us your views. Keep those letters coming, not only because we like to hear from you but because if your prose is printable you will also earn yourselves one of those snazzy little badges which Liz so cleverly devised (yes, they too will continue as before).

You will find your usual crop of games listings in this month's issue, along with a review of the new Panasonic CF-2700, all the latest gossip from Japan, an interview with Sony's whizz kid Mike Margolis, an insight into what the MSX people are saying, lots of easy-to-follow programming advice, and much more besides. So read on.

Nicole Segre

Editor
Nicole Segre

Editorial Assistant
Eric Doyle

Group Editor
Dave Bradshaw

Advertisement
Manager
Paul Stanyer

Divisional
Advertisement
Manager
Chris Northam

Advertisement Copy
Control
Lynn Collis

Chairman
Jim Connell

Typesetting and
Origination
Letterspace (London) Ltd

Design
MM Design

Editorial &
Advertisement Office
No 1 Golden Square
London W1R 3AB
Tel: 01-437 0626
Telex: 8811896

**MSX User is a monthly magazine
appearing on the second Friday
of each month**

Distribution
SM Distribution Ltd
16-18 Trinity Gardens
London SW9 8DX

Printing
Alabaster Passmore & Sons Ltd
Tovil, Maidstone, Kent

UK Subscription rates £15.50
per annum. Worldwide on
application to MSX User
Subscriptions Department,
Infonet Ltd, Times House,
179 The Marlowes, Hemel
Hempstead, Herts. HP1 1BB

What and where

NEXT MONTH

Interactive video - The shape of things to come

Music composer - We sound out Yamaha's cartridge

Communications - An exclusive look at Toshiba's RS232 interface

PLUS

News from all over, the latest reviews, more about programming
and a bumper crop of great games listings

MSX

User

REGULARS

- 3 Front** - Hello from the new editorial team
- 22 Profile** - Learn a thing or two from Sony's Mike Margolis
- 48 East** - Graham Knight with the latest from Japan
- 72 Contact** - Software houses and what they offer
- 83 First Aid** - Jeremy Vine has the answers to your queries
- 88 Feedback** - Your news and views

MSX REVIEWS

- 14 Panasonic CF2700** - Karl Dallas approves
- 86 First Editions** - Bookworm's corner

THERAPY

- Great DIY software for your MSX**
- 51 Blackjack** - Take a gamble
- 59 Shell Sort** - Order, please
- 62 Auto Racer** - Fast forward
- 64 Dodger** - Arcade action
- 66 Digital Clock** - Time keeper
- 68 Alien Planes** - Further arcade action
- 70 Address Book** - Easy filing
- 79 Quizmaster** - How clever are you?

PROGRAMMING

- 34 MSXercise** - Part 3 of our beginners' BASIC course
- 40 Conversion** - Tailoring programs to fit your machine
- 44 MSXplained** - More about interrupt routines in this month's instalment
- 74 Adventures** - Steve Lucas shows how to write them

FEATURES

- 17 Keep it Safe** - Everything you want to know about storage methods
- 26 Memory Matters** - Hidden secrets of the MSX

The contents of this publication including all articles, designs, plans, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. © 1985 Argus Specialist Publications Limited.

The new Mitsubishi

For those in the know

Anyone conversant with home computers will know precisely why MSX was worth waiting for.

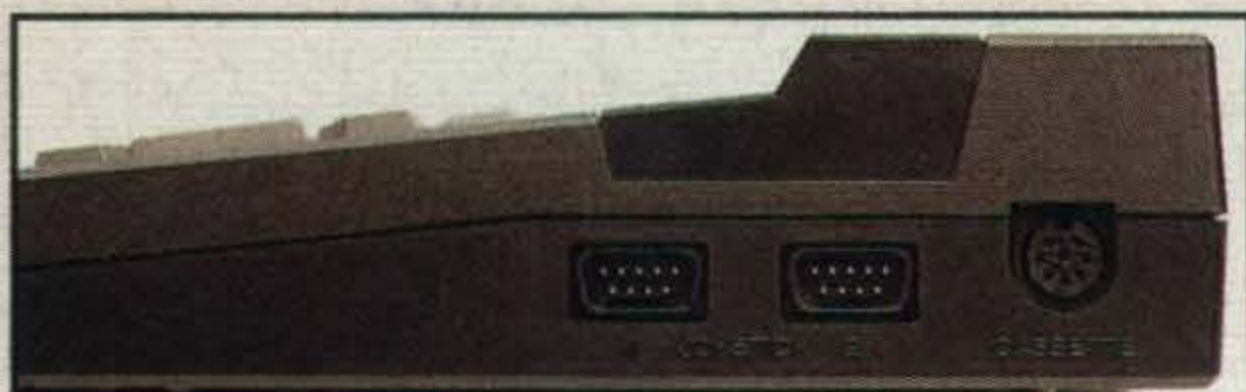
The sheer proliferation of computer and software systems flooding the market loudly underlined the need for a unified standard.

So the major companies jointly developed a single computer and software system. The result – MSX – the format that will be standard for all time.

And those in the know will not be surprised that Mitsubishi are in the vanguard of the MSX movement. For, with the F-series, Mitsubishi offers everything that MSX is and more.

GRAPHICS

Maximum resolution of 256 x 192 pixels with all 16 colours available on the screen at the same time. 32 sprites in two sizes and two magnifications allowing easy creation of '3D' graphics. 255 pre-defined characters all of which can be used as straight text or easily mixed with graphics.



JOYSTICK AND CASSETTE PORTS



EXPANSION BUS

PRINTER PORT

MONITOR OUTPUT

SOUND

Three independent channels which can be output through the TV loudspeakers at any volume, individually or simultaneously, at any of the available 8 octaves. All three channels can use the 'noise' generator for stunning sound effects.

KEYBOARD

73 moving keys, ergonomically designed for many hours of fatigue free use. Large cursor control keys which are excellent for both programme editing and game playing. 5 function keys giving 10 pre-defined functions which can easily be redefined from 'BASIC' using the 'KEY' command.

BASIC

MSX BASIC is possibly the most comprehensive version of the original language. There is a complete set of commands for creating graphics and sounds, manipulating text and moving sprites. In addition to this there are 'built-in' interrupt routines for detecting sprite collisions, function key selections and joy-stick fire buttons.

EXPANSION

The Mitsubishi 64k ML-F80 and 32k ML-F48 are both equipped with 2 cartridge ports, 2 joy-stick ports and a centronics compatible parallel interface. It is through these devices that the MSX system can be expanded for use with disc-drives, printers, serial interfaces, modems and other peripherals.

SOFTWARE ON CASSETTE

The MSX system can load and save data onto cassette at 1200 or 2400 baud and unlike certain other home computers, the Mitsubishi F-series can be used with a normal domestic tape recorder for this purpose.

When you put all of these features together, with the knowledge that Mitsubishi is the largest manufacturer of Mainframe computers in Japan, those in the know will immediately recognise the true potential of the Mitsubishi F-series.



Mitsubishi MSX Computers

For those who aren't

The Mitsubishi MSX family computer is everything you wanted to know about computers, but didn't know who to ask.

It's friendly, it's fun and so simple, a grown man can use it. Yet so versatile even his computer-versed children would be hard-stretched to over-tax it.

It operates with any colour TV set. Just plug it in, and the full power of the computer is instantly at your fingertips.

FOR FATHER

The Mitsubishi MSX can do many things, from keeping a simple check on the bank balance to running a complete business with customer account files, stock control programmes and word processing. It is just as much at home keeping control of your record or stamp collection or playing 'strategy' games such as chess, othello or contract bridge.

FOR MOTHER

There is the opportunity to store recipes and other household information or keeping record of the children's progress at school. Household accounts can also be recorded so that savings can be planned for holidays and other seasonal expenses.

FOR THE CHILDREN

There is education, particularly computer education. In a world where computer literacy is now of foremost importance, MSX offers a broad base of educational software. With simple programmes for the very young through to complex programmes for older students like language learning.

Also, the graphics system of the Mitsubishi computer ensures that the MSX versions of your favourite games are reproduced with incredible speed and accuracy.

Undoubtedly, MSX is the format for the future, and will become the byword for computer

education and entertainment.

And you can be secure in the knowledge that regardless of future developments, any investments made in MSX hardware, software and peripherals today will always be compatible with the Mitsubishi F-series.

So if you've waited until now to buy a computer, you couldn't have timed it more perfectly. Get to know one today.

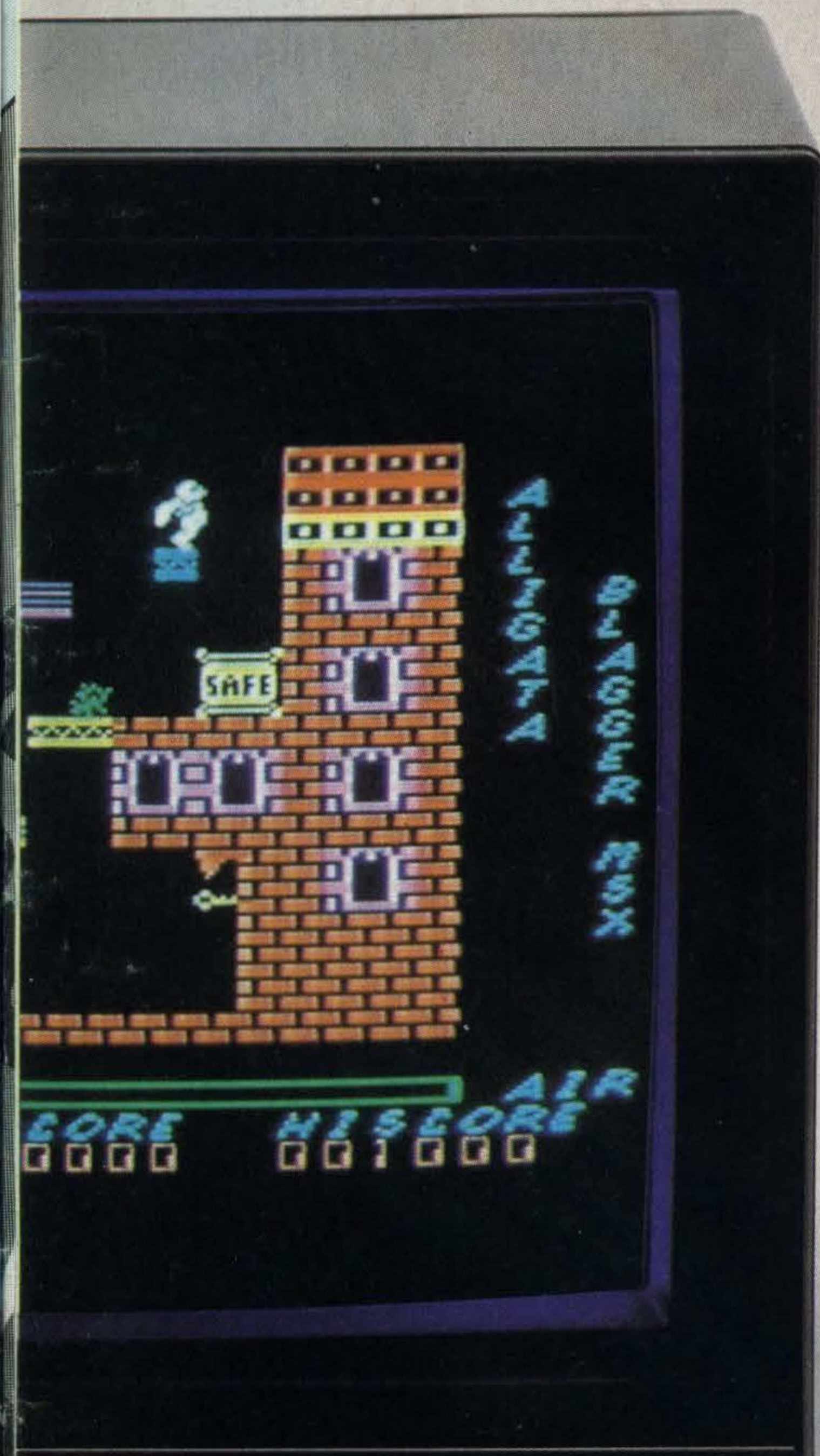


Mitsubishi Electric (UK) Ltd., Hertford Place, Denham Way, Rickmansworth, Herts WD3 2BJ. Tel: 0923 770000.

SPECIFICATIONS

CPU: Z80A (3.6 MHz)	Special keys for screen editing
Memory: ROM: 32 KB RAM: 64 KB (F80) RAM: 32 KB (F48) Video Ram: 16 KB	Sound: 8 octaves 3 channels for sound or 'noise' Output by TV sound or External Audio Amplifier
Screen Displays: *Text Mode: 40 columns x 24 lines *Graphics: 256 x 192 pixels Colours: 16 (15 + transparent) Sprites: 32 Output: RF, Composite Video	Cassette Interface: 1200-2400 baud Motor controlled by CPU Parallel Interface: Centronics Joy-Stick: 2 x 9 pin connectors Rom-Cartridge: 2 x 50 pin connector
Keyboard: 73 moving-key keyboard 5 function keys Cursor control keys	

*Subject to Scan of Monitor



ML-F80



ML-F48



MSX

MSX

Fighting Talk

So far the most striking thing about the launch of MSX has been the amount of words produced. Paul Walton listens to what the manufacturers are saying.

There are lies, damn lies . . . and there are marketing promotions! MSX might have been *conceived* by Americans and by Japanese but it was supposed to be *born* in home computer-crazy Britain this Christmas; and Georgie Gibbs can rightly claim that, for the past nine months, she was an important mid-wife, playing a crucial role in *delivering* the complete bundle of MSX goodies into the shops. But if MSX does not take off here then the pioneering work of her little group, "in getting people to market MSX as if the home computer were just another home appliance", will all have been in vain.

Because it doesn't matter how well the British market, the shop-keepers, the software houses and even the punters were prepared – if there aren't enough machines in the shops. A little bird tells us that the grand total of MSX computers in Britain is – wait for it – around 25,000. Which is peanuts: a quarter of the Spectrum sales in a bad month, let alone at Christmas.

Competitive

It must be said that Toshiba come up smelling of roses, having "gone for it this year, instead of waiting cautiously for 1985", according to rather sceptical marketing supremo Steve Skuce. "Our philosophy is to go for the market this year as it might not be there next year". Toshiba's HX-10 was in the shops in September, a good six weeks ahead of the pack, and is being shipped in "tens of thousands", as opposed to the measly

500-several hundreds from other MSX suppliers.

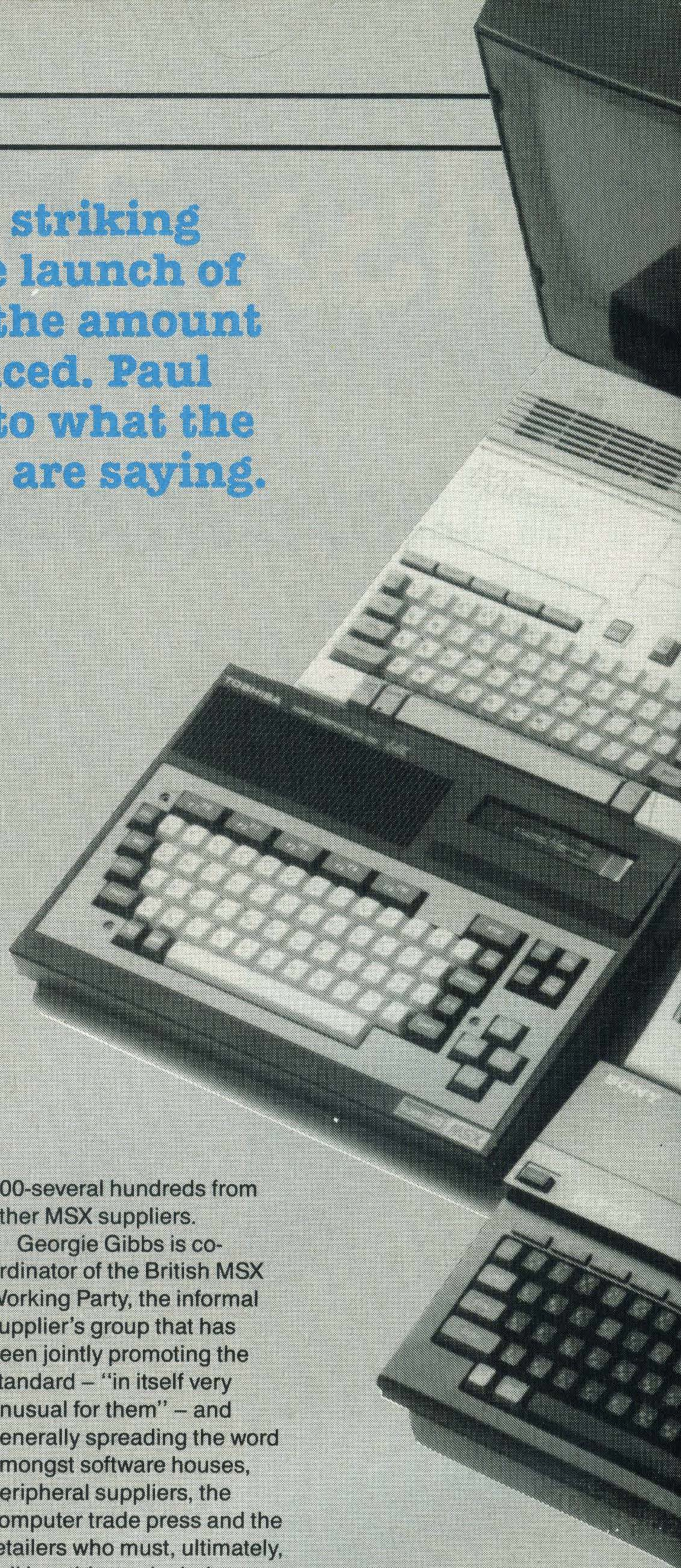
Georgie Gibbs is co-ordinator of the British MSX Working Party, the informal supplier's group that has been jointly promoting the standard – "in itself very unusual for them" – and generally spreading the word amongst software houses, peripheral suppliers, the computer trade press and the retailers who must, ultimately, sell into this particularly competitive section of the consumer electronics market. All of which, unfortunately, means nothing if there ain't no machines to purchase . . .

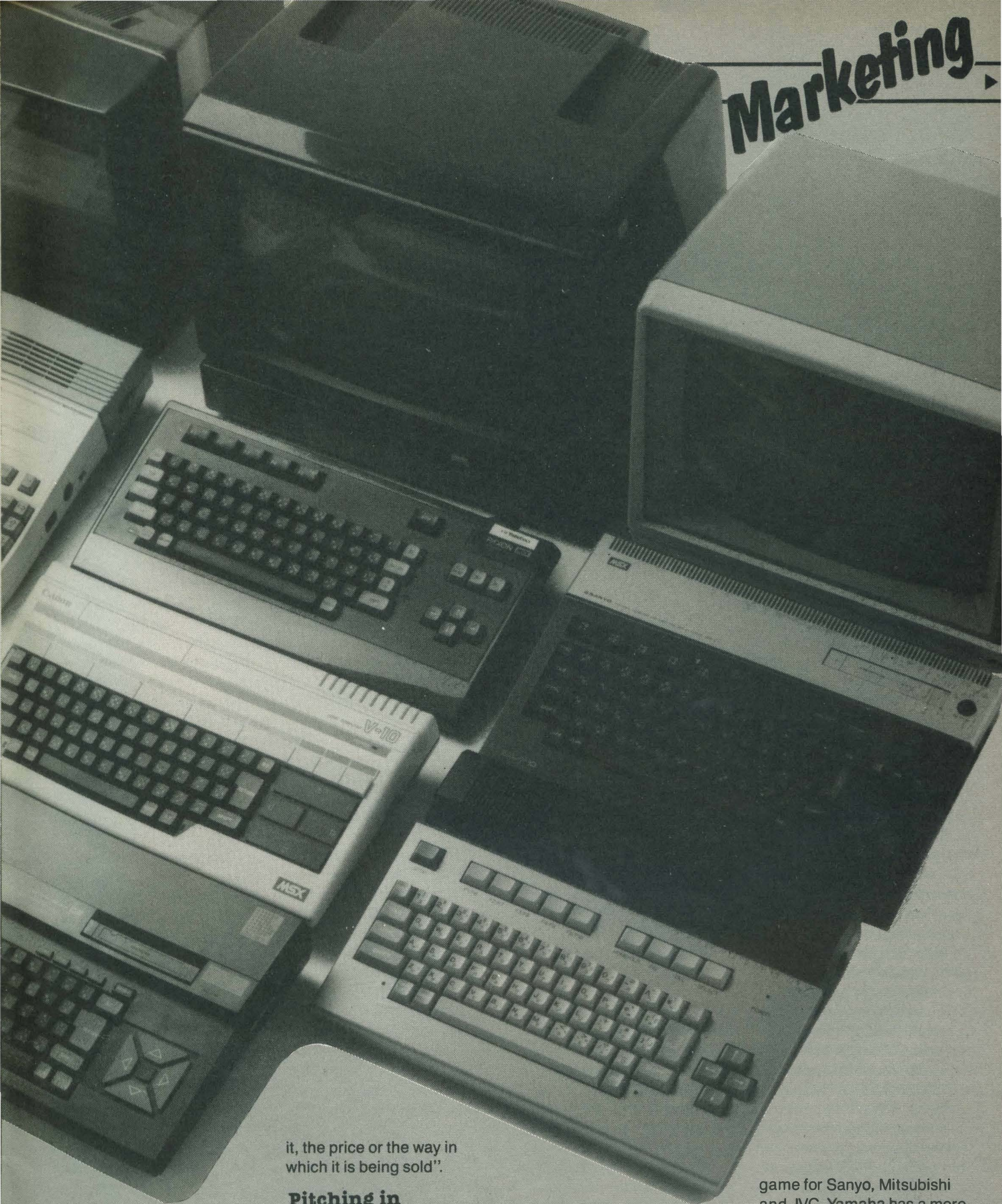
Informal group

Since the beginning of 1984 this informal group of suppliers, which includes Canon, Hitachi, JVC, Mitsubishi, Sanyo, Sony, Teleton and Toshiba, have been preparing the ground for successful sales 'behind-the-

scenes'. Other firms on the list of 23 licensing MSX (or the MicroSoft extended Basic operating system) were invited, and more may join next year, said Gibbs. But some decided to go it alone: however, as she pointed out, MSX is expressly designed for sharing.

"The whole idea of an MSX standard was that the home computer would become a home appliance, like the





it, the price or the way in which it is being sold”.

Pitching in

The way in which most MSX machines are being sold boils down to one word – words! A great deal has been said, by the Working Party and others, but few have put their money where their mouths have so patently been taking them.

Without wishing to cast

aspersions, **MSX User** picked on four members of that Working Party and one other supplier to find out how their machines are being sold for Christmas, and beyond...and, Toshiba aside, caution is very definitely the name of the

game for Sanyo, Mitsubishi and JVC. Yamaha has a more healthy attitude, but even it's not going to town with its unique musical speciality. Spot the difference in these sales pitches . . . (It's a trick question!) Some suppliers favour the dealers.

Ed Parker is in charge of promoting the Sanyo MPC-100 which, at £299, is a

record player, capable of allowing machines from different suppliers to run any piece of software”, said Gibbs. “And the differences only enter when you consider the unique features of a machine, the peripherals for

bit pricey and has nothing in particular going for it. "We've done a small amount of advertising, not a great deal, and otherwise a lot of PR".

Generous margin

Profits, it would seem, are the most important thing to the new High Street retailers handling – in MSX – their first home computer. "We've given the independent dealers a generous margin, to let them do their own local advertising". And his supplies, which were late? "Not many – it's been difficult to gauge the up-take this year".

Nick Dosanjh has the Mitsubishi MLF-48 (32K RAM), at £220, and the MLF-80 (64K RAM), for £275, on his hands: "We've spent a reasonable sum on advertising, largely in the trade press, and that will carry on to January".

Again the suppliers' own dealers, more used to video or TVs, are "getting to grips with the machine". How many machines supplied since November?: "A modest supply – it's a new product".

Steven Michaelis is in charge of pushing JVC's HC-7GB, for £279, and he described business as being "very low key on this side of Christmas – but then again we knew that the market was going to start small". JVC is concentrating on building MSX sales slowly through its Audio-Video dealers, with its promotional budget going into training them up for 1985.

The HC-7GB was late in to the UK as it was only being produced in September and is a new machine anyway, with a unique RGB output, and not the more usual European version of a Japanese best-seller.

Michaelis explained the general reluctance on the part of most MSX suppliers.

"It's a very long term project and, to be truthful, 1985 is really when it begins. The trouble is that we (the Working Party) were too successful in exciting public

interest and press speculation about our immediate prospects.

"The introduction of MSX is more analogous to the Japanese VHS video cassette recorder standard. Phillips were years ahead with their VCR, the old Betamax format, they had the dealers sewn up, the cassettes being produced and the growing market share.

"But they made the fundamental mistake in being just one firm selling that standard. Then we had the advent of umpteen VHS machines, which won over a larger share of the cassettes and, eventually, the dominant market share", he said. (And remember, Phillips had to concede victory and adopt VHS in 1984 . . .)

Toshiba, of course, simply can't wait for MSX to be adopted. So they "go for it" by addressing the consumer. Skuce said that was in part because the Japanese parent company was "becoming very much more aggressive" and trying to come out as the major competitor to Sony in consumer electronics. And that means fighting talk from Toshiba, for example. "No one's tipping Commodore and Sinclair over the brink just yet, but we want to be the firm that does".

Skuce described the detailed promotional campaign which Toshiba was beginning while the rest of The Working Party was chipping in for a measly all-in MSX ad in the nationals. "First of all it was aimed at the MSX purchaser, the father in other words. And that campaign is just coming to a close.

"Over the copy-line that said if you buy the wrong home computer now, you'll have to buy an MSX in two year's time we started to build the Toshiba identity", he explained. And if you've seen the 'Hello Tosh, Gotta Toshiba' ads then you'll know what identity can do . . .

"Next we advertised in the computer specialist press: our MSX needed a 6 foot tall

monitor to contain the specifications, while the ordinary home computer could fit them into a TV screen. In other words, showing that MSX had taken the best features from all home computers. The third campaign was much the same, but geared to the proliferating MSX press". (The 64K HX-10 is nothing wonderful, remember, just standard MSX).

"We've already spent £300,000 on advertising, and the bill for launching our MSX isn't likely to be anything less than a million pounds", said Skuce. Lotta tosh! "Our campaigns aren't 'angled' at any particular kind of buyer or seller, we believe in a balanced business where sales are allowed to come from anywhere – we don't dictate who sells what".

Just as Toshiba stands out from the bunch with its marketing, the only MSX supplier we spoke to that is really adding weird peripherals and specialising its machine is Yamaha.

Musical micro

"Anyone who handles the Yamaha MSX has first got to take our course and then pass an exam", according to Bruce Everiss – moonlighting as the marketing man for the launch of its £600 CX-5. Sold through the supplier's musical dealers, this 32K RAM machine adds a professional synthesiser chip, a full musical keyboard, a four-pack of musical ROM packages and additional 'voice-macros', or musical commands, to the MSX BASIC.

Because this musical micro is more difficult to sell Yamaha trains its dealers – and reckons that they don't get supplies until they've passed the exam. "It's a totally new concept, there is nothing like it", said Everiss – but still little spending on ads to proclaim this achievement, unfortunately, and still only "modest" supplies.

When JVC brings out

a VCR interface next year it will train dealers to use it; and the same goes for the rest. But by then the market share might have been largely soaked up . . .

Friends and foes

The MSX Working Party was founded in February on the common ground between suppliers, in recognition of the fact that certain pre-requisites – like software packages, improved system routines and peripherals – are common to all of them, with the stipulation that the group would just see to the basics and leave each firm to finalise its own marketing details.

Graylings was picked to administer the group, since it was Sony's bright idea and the firm did Sony's Public Relations. Gibbs, who had previously worked for Logica, was thrown in at the deep end. "I'm no buff", she said, "this was my first expedition into home computing and it was a hell of a lot of work. Since we are promoting the standard – not Sony – we have no particular axe to grind".

The Working Party was originally just going to prepare the basis for this Christmas marketing campaign but, since some suppliers will miss it and because there are now many more in the MSX market, it is likely to become a permanent and a useful feature of the European market "for the foreseeable future".

But is it a good thing, or is it just a polite club where the MSX suppliers will say much and do little? Toshiba, long thought to be a critic of The Working Party, said that it wanted it to keep generating software. Skuce said that the group would naturally handle less of the day-to-day business of MSX as the suppliers got around to the day-to-day business of stitching each other up.

This is already happening. Toshiba shocked the rest by dropping its price a whopping



£40 to £239.95, and has promised that as more savings come from mass production then these will be passed on to the consumer.

"We were fast into the market here and are now gearing up our supplies to meet demand", Skuce explained. "No-one else is going to risk a drop in price, with their supply so limited. As the other MSX machines start to disappear from the shops we'll still be carrying the banner for MSX".

No contest

With 10-weeks sales under its belt Toshiba is reckoned to be moving 70 per cent of all it can supply, or at least several thousand machines already sold. "It's got to be Number One in the MSX charts – no contest", according to Skuce. Sanyo has sold all it had. Mitsubishi's sales are "meeting expectations". JVC has sold 60 per cent of machines delivered – perhaps a couple of thousand – and expects to sell 90 per cent of stocks by Christmas.

The importance of the MSX Working Party should not be under-estimated; you

wouldn't get Sinclair, Acorn and Oric sitting round a table to discuss general improvement of the British market for home computers, let alone to do anything about it. But neither should it be seen as the MSX suppliers club – the gloves are off and it's every machine for itself.

Gibbs reckons that what is almost as illuminating about this fete of co-operation is that even those MSX suppliers who knew that they would be late and miss Christmas – notably Canon, Hitachi and Teleton – still "pulled together during the year". One wonders which direction they'll be pulling in 1985.

There are already well over a thousand retail outlets for MSX computers, according to estimates from the MSX Working Party, and several hundred applications and systems programs from a band of 60 software houses, 20-odd makers of peripherals, a dozen books and three

magazines – all in the High Street.

But the MSX machine which succeeds in grabbing sales will be that which is well-known, which is well-priced, and which is still on the shelves. And, as home computers have a life of between eighteen months and two years, that early lead is going to be an important asset as more peripherals, software and specialisations are introduced in 1985.

MSX machines which were in the shops by November – and confirmed at time of going to press – included those from Sony, Sanyo, Spectravideo, Toshiba, JVC, Mitsubishi, Yamaha (the Japanese) and Goldstar (Korean), with Philips also threatening to bring out the first European MSX.

Few machines

Prices are so similar that the naive might believe that MSX suppliers are operating some kind of high-tech, price-fixing cartel: the basic (32K)

MSX will cost between £199 and £299. But the truth is that the similarity between machines fostered by the hardware standard gives little room to manoeuvre on price, unless you either get into volume production quickly, or go for kami-kaze discounting to win market share.

The margins dealt to dealers and retailers are so slim that discounting, outside of a few pounds, is likely to be rare; differences are instead evident in the packaging and should also have been discernible in marketing and promotion.

Some MSX suppliers have chosen to launch many more of their machines at one time, in the expectation of huge demand. But with so few machines around this is only likely to further dilute the important brand-identity – more important in the UK than Japan – of the individual MSX supplier. It should also be noted that all of the Japanese

machines being sold in Britain have already been on sale in Japan since after Easter – although recent trade figures show that the demand was over-estimated, with only a half of the products being sold.

The soft touch

Again what will be common to all of these machines – although not necessarily all-in with the price! – is the ability to run a great deal of British software from Day One. By April the MSX Working Party had invited 70 software houses to its first demonstration of MSX machines; some 60 signed up soon after and agreed to write MSX packages.

But the problem, again, is that it is all the *same* software which is being bundled in with the first machines. (Partly forced by the late appearance of promised packages). The poor customer can't really make a choice on the basis of which program, or which peripheral, he gets with which MSX machine.

And Gibbs admitted that, for the first few months, getting hold of any MSX machines other than the Japanese-type NPSE as opposed to British-type (of television signal) PAL was, to say the least, difficult: "The Japanese were so busy selling them that they didn't get prepared to send machines to Europe". Here the MSX Working Party put pressure on their firms to ship-in the first sixty MSX machines for the software houses, service firms or, in some cases, for authors eager to complete books and manuals with some practical experience of the hardware.

The entire water-front of home computer software is now going into the shops, from basic business through to education and games, and much of it has been written originally for MSX. Kuma and Hi-Soft are emerging as leaders, being bundled in with most MSX machines, and with the majors such as

Thorn-EMI Software and Longman's also taking a close interest.

For instance, Knight's software house is the sole distributor for (Japanese-derived) Toshiba packages, and anyone buying a Toshiba machine will get five Knight's programs, including an introduction tape, graphics demonstration, junior maths and keyboard trainer.

And if you buy a Sanyo machine the Knight's package is also included, substituting the more studious graphics and maths for two games called Vicious Viper and Exploding Atoms.

The cassette-based software is being priced to undercut rivals for the popular Spectrum or Commodore 64, with the majority of packages being in the affordable £5 to £15 range, and the more specialised vertical market applications and system utilities not going over the £50-mark. But ROM cartridges might present a problem: at \$25 per thousand this dictates that the end-user price of cartridges is still going to be £40 to £50 – thought by retailers to be too steep for the home computer market.

Wait and see

There is every sign that MSX software will be as abundant as was CP/M, when it began, and that many suppliers are already following the same marketing gambit in bundling packages in with the machines. The major MSX software houses reads like a roll-call of the British games suppliers and also include Knights, PSS, Ivan Berg, Bubblebus, D K Tronics, Quicksilver, Megacycal, Alligata, Lotharian and Tasman.

A couple of software houses began solely to service MSX machines: MPL of Bexhill claims a list of 38 educational programs. Many packages are now being translated for MSX using utilities from Hi-Soft, for example. Software houses like Kuma are now also turning

to important peripherals, such as disc drives.

And then, more seriously, a very basic kind of (assembler) computer programming method was essential to pack complex programs into just 16K or 32K; this was developed by Kuma and Hi-Soft (called Jens) and distributed within the group. Gibbs said: "We tried to think of everything that might be required, but every time we thought we had, something else came up".

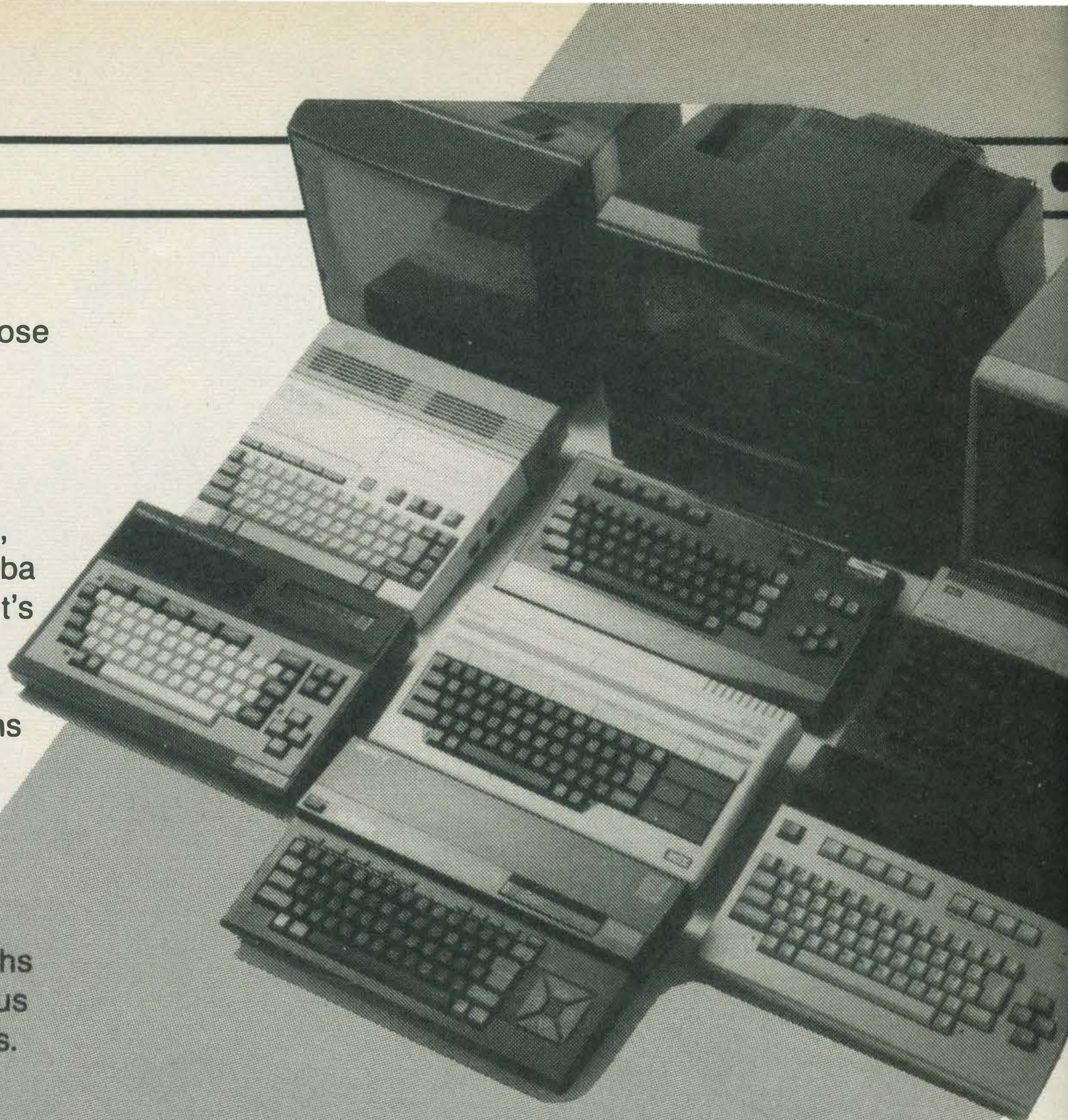
Nowhere was this better illustrated than in what she terms the "storm-in-a-teacup" that blew up over the correct, standard size and format discs, CP/M had made 5¼ in disc drives the standard; but MSX was technically capable of everything from the old 8 in to the mini-floppy 3 and 3½ in discs. Back came the word from Japan and from Microsoft. Offer all four size discs, but with a common format that will allow ease of inter-change between them! The software houses could get back to work.

The MSX Working Party is now using its experience to quicken the introduction of MSX in Europe, where it has just been launched in France, Germany and Holland, and in America, where a party of nine Japanese suppliers grouped together in January at the giant Consumer

Electronics show (CES) in Las Vegas. Britain was the first place for MSX to be sold, outside of Japan, and here the suppliers had to prove that they could persuade European retailers – many naive about how to sell computers – to handle MSX; it has been championed as the first *Computer Appliance*. But has it worked here? – "We'll have to wait and see", said Gibbs.

Still, it does appear that in almost every department The MSX Working Party was fighting an uphill battle. Perhaps Britain wasn't the testing ground for MSX exports which we were all lead to believe. Perhaps what might be born this year is not MSX, but a turkey: a tough old bird gobbling a lot about the product, but delivering little by way of real substance or flavour . . .

The complete list of MSX licenses issued by Microsoft is as follows: Canon, Casio, Fujitsu, General/Paxon, Goldstar/AVT, Hitachi, Daewoo, Mitsubishi, Panasonic, Pioneer, Samsung, Sanyo, Sharp, Sony, Teleton, Toshiba, Victor-JVC, Yamaha, Yashica, Kyocera, with the Dutch firm Philips, the French firm Thomson and the sole American supplier, Spectravideo.



Mirage

**SOFTWARE &
DISTRIBUTION**



9 Rossini Street
Seaforth
Liverpool
L21 4NS

Tel: 051-928 7443/4

PROGRAMMERS

WE ARE LOOKING FOR TOP QUALITY PROGRAMMES TO MARKET IN THE UK AND ABROAD. THE PROGRAMMES MUST BE OF TOP QUALITY AND IN MACHINE CODE WITH GOOD GRAPHICS AND SOUNDS AND ORIGINAL CONCEPT. WE ARE LOOKING FOR PROGRAMMERS TO CARRY OUT WORK ON NEW MACHINES SUCH AS THE C.B.M/16 AMSTRAD AND M.S.X. WE CAN SUPPLY THESE MACHINES TO CARRY OUT WORK FOR MIRAGE. WE ARE ALSO OFFERING £500 PRIZE FOR THE BEST PROGRAMME RECEIVED BY THE END OF FEBRUARY 1985 WHICH WE WILL DECIDE TO MARKET. WE OFFER AN OUT RIGHT PAYMENT OR 10% PLUS PAYMENT OR 20% ROYALTIES. FOR MORE INFORMATION CONTACT THE SOFTWARE MANAGER HE WILL BE GLAD TO LISTEN TO ANY QUESTIONS YOU MAY HAVE.

SOFTWARE HOUSES

WE ARE LOOKING FOR TOP SELLING PROGRAMMES TO DISTRIBUTE TO OUR WIDE RANGE OF OUTLETS THROUGHOUT THE UK AND GIVE YOU A VERY FAST SERVICE TO HELP WITH YOUR SALES AND TO MAKE THE MOST OF YOUR PRODUCTS. SEND A SAMPLE COPY OF YOUR PRODUCTS WITH RATES AND TERMS FOR A FAST ASSESMENT. FOR MORE INFORMATION CONTACT THE SALES MANAGER HE WILL BE GLAD TO HELP AND ASSIST YOU.

DEALERS AND DISTRIBUTERS

DEALERS WE HAVE THE BEST SELLERS FROM THE BEST PRODUCERS; WE ALSO OFFER VERY GOOD RATES AND TERMS TO HELP YOU AND US TO ACHIEVE THE BEST SALES AND SERVICE. FOR MORE INFORMATION CONTACT THE SALES MANAGER.

DISTRIBUTERS HAVE YOU GOT THE MIRAGE RANGE ON YOUR SHELVES, IF NOT THEN CONTACT THE SALES OFFICE FOR SAMPLES AND RATES.

**SALES MANAGER
T.C. SAPHIER**

**SOFTWARE MANAGER
C.A. SAPHIER**

MSX

Panasonic

With Hitachi still waiting to see what the other manufacturers report about the temperature of the water before dipping their toes in, launch of Panasonic's top-quality thoroughbred completes the first wave of MSX machines to hit the UK market.

Roughly in order of price, starting at the bottom end, the machines we've seen so far include the **GoldStar** from Microdeal, the **Spectravideo** SVI-728, the **JVC** HC-7, the **Canon** V-20, the **Toshiba** HX-10, the **Mitsubishi** MLF-48 and MLF-80, the **Sanyo** MPC100, the **Sony** HIT BIT HB75, and the **Yamaha** CX5M MSX music computer. (The scramble to reduce prices has now begun, affecting GoldStar, Toshiba, Sony and Sanyo initially, so the price order may change over the early months of the year.) We're still waiting for the **Philips**, which has been on

sale in Italy so far but nowhere else in Europe, as yet, not even its native Holland.

So far, we have had machines with only one cartridge slot (GoldStar, Sanyo, Spectravideo, Toshiba, Yamaha), with two cartridge slots (Canon, Mitsubishi, Sony), with numeric keypad (Spectravideo), with RGB video output as well as RF and composite video (JVC, Sony), with expansion bus (GoldStar, Mitsubishi, Sanyo, Spectravideo, Toshiba, Yamaha), with (RESET) button (JVC, Sanyo, Sony), and with specialist extra firmware (Sony).

The **Panasonic CF2700** is what you might describe as a "classic" machine, with two top-mounted cartridge slots, no expansion bus or (RESET) button, and otherwise no special bells and whistles, apart from the fact that it is a very attractive, professional-

feeling implementation of the MSX standard, supported by excellent documentation.

It comes in tasteful shades of grey: dark charcoal for the body of the machine and its large, diamond-shaped cursor control keys (outlined with green arrow heads), dark grey for control keys ((SHIFT), (SPACE), (GRAPH), (CODE), (CTRL), (TAB), and (ESC)), and light grey for the alphanumeric, function and (STOP), (HOME)/(CLS), (SELECT), (INS) and (DEL) keys – that's the order those latter keys are laid out across the top of the keyboard, incidentally.

The (RETURN) key is comfortably large (the depth of two alpha keys), and so are the two (SHIFT) keys, which are at least double-width (the one on the left is almost triple-width). Arrows are used to label them, a downward-and-left arrow for (RETURN), upward arrows for (SHIFT), plus a circled upward arrow



The new Panasonic CF2700 wins the Karl Dallas seal of approval.



for (CAPS LOCK) (which has a built-in LED rather than a separate indicator), and a leftward arrow for (BS).

None of the graphic characters are indicated on the key fronts, and the accents key displays only the grave and acute, not the circumflex and umlaut which can be accessed by pressing it at the same time as (CODE) and (SHIFT) + (CODE).

I really like this keyboard, which is easily the equal of my two other favourites, the Sony and the Canon. The keys have a nice springy feel and the keytops are slightly dished, in best Selectric manner.

The rear connectors are grouped rather close together instead of being strung out across the back as is normal, which makes for a rather tight tangle of wires if all are in use. Composite video and audio out sockets are both RCA phono-type, and suitable heavy-duty leads are supplied. The D-connector for printer and the two joystick ports at the right-hand side

come supplied with removable (and therefore losable) plastic covers.

Although there does not appear to be a micro-switch attached to either of the two cartridge covers, allowing them to function as a sort of cold reset, there is actually a switch inside each, which would be depressed by the insertion of a cartridge. However, you'd be ill-advised to try using either of these micro-switches as (RESET) buttons by poking around with your fingers – an insulated screwdriver might be OK, if you know what you're doing – and of course you should always switch off before inserting a cartridge. The micro-switch is there as a final fail-safe should you forget. (But this is one of those things you should *never* forget).

Inside, the layout confirms the workmanlike solidity of the exterior, with a sizeable powerpack contributing much to the half-a-kilo or so by which this machine is heavier than the competition – and

some of the extra heat, too, for it does run slightly warm, accounting for the fact that it draws 28 watts, compared with most other machines' ten or so. The keyboard is attached to the main PCB rather than coming away with the cover as is more usual, and the shortness of the ribbon cables forming this attachment might make servicing a little inconvenient.

However, the general appearance provides plenty of reassurance that servicing is unlikely to be necessary.

The monitor display is bright and crisp, with none of the weak colours I've noticed on some other machines.

The documentation is particularly good: a 130-page instruction manual with useful but simple programming examples, and a 196-page Basic Reference Guide, including 130 pages devoted to the commands and functions – mostly one to a page, including syntax and program examples – a chapter covering the main points of MSX Basic (rather

briefly), and ten pages of tables.

The only aspects inadequately covered are the VDP and SOUND command – but they probably need more than two or three pages each to make their functions clear.

Verdict

In general, I like this machine, though it has no special facilities apart from its second cartridge slot. I like the big cursor control keys, the fact that both cart slots are protected by hinged doors, the way its 3½-kilo weight prevents it from slipping around on the desk, and the professional feel of the keyboard.

I have reservations about the grouping of the rear connectors, and I'd be worried about buying a machine that didn't have an expansion bus. But then neither do my other two favourites. And it seems likely that the cartridge slot will be used for expansions.

For the serious user, this has got to be a contender.

Panasonic Review

The price tag. It's something rather special. When you take a look at the new Goldstar MSX you'll find an asking price of around £240. Quite a bit less than most of the others. And since you're choosing a micro that's designed and built to one standard, that'll leave you a whole lot more for the super new MSX games or business software.

What is standard, of course, is the superb MSX specification.

There's one thing about this MSX that isn't quite standard.



£199.99!

State-of-the-art feature like 64K of 'user' RAM and 32K of ROM and 16K Video RAM place the Goldstar right at the top of the MSX league.

Sixteen vivid colours and eight octaves of sound make it a great games player's micro.

And there's a powerful Z-80A processor to take on a world of home office tasks.

Goldstar MSX. The brightest new star among micros that's unbeatable value for money.

You'll find it at above standard computer dealers now.



GoldStar

Distributed by:-

MICRO DEALER UK Ltd

29 Burrowfield, Welwyn Garden City Herts AL7 4SS

Tel: (07073) 28181 Telex: 23674 MDUKG

Francs for the memory - Karl Dallas looks at the ways and means of storage.

Your MSX has a truly wonderful memory. Tell it something, and it'll be remembered for years - as long as you don't switch it off.

But if you do turn it off - or, more likely these days, if there's a power cut, or someone kicks out the plug - then *zilch!*

It'll just sit there blinking at you, telling how many thousand bytes of memory, it's got available for you, but not remembering a thing.

If you had a similarly forgetful friend, you could write a list of all the things you wanted it to remember, stick it in his or her pocket, and every time forgetfulness took over, there it'd be: the list! Instant recall.

Computers can't read bits of paper (well, not yet, anyway, though there's a gadget for the IBM PC that can read typewritten pages, and always remember that the IBM's MS-DOS operating

system is the big brother of MSX-DOS) but they *love* reading magnetic media.

Options

What that means, in real terms at this moment, is one of four options:

- Cassette tape
- Floppy diskette
- Data cartridge
- Tape streamer.

The first three of these are available right now, with the fourth on the horizon (meaning negotiations between the British importers and Mitsumi, the Japanese manufacturers, are still proceeding).

Everyone's familiar with tape cassettes, if only because that's what you use to tape all your mates' LPs on. The tapes used by computers are exactly the same, except that it makes sense to avoid using the pricey metal tapes, since they don't actually record computer programs

very well, and they are very expensive. You ought to steer clear of very long tapes (the C90 or C120 variety), too, because the tape is very thin and likely to stretch in use, and also it's a pain winding forward from one end of the tape to the other looking for exactly the program you want.

Disks are less of a pain, **LOADING** in seconds or less what could take minutes from tape. Of course, they're more expensive, both for the storage medium and for the hardware - the disk drives - required to use them.

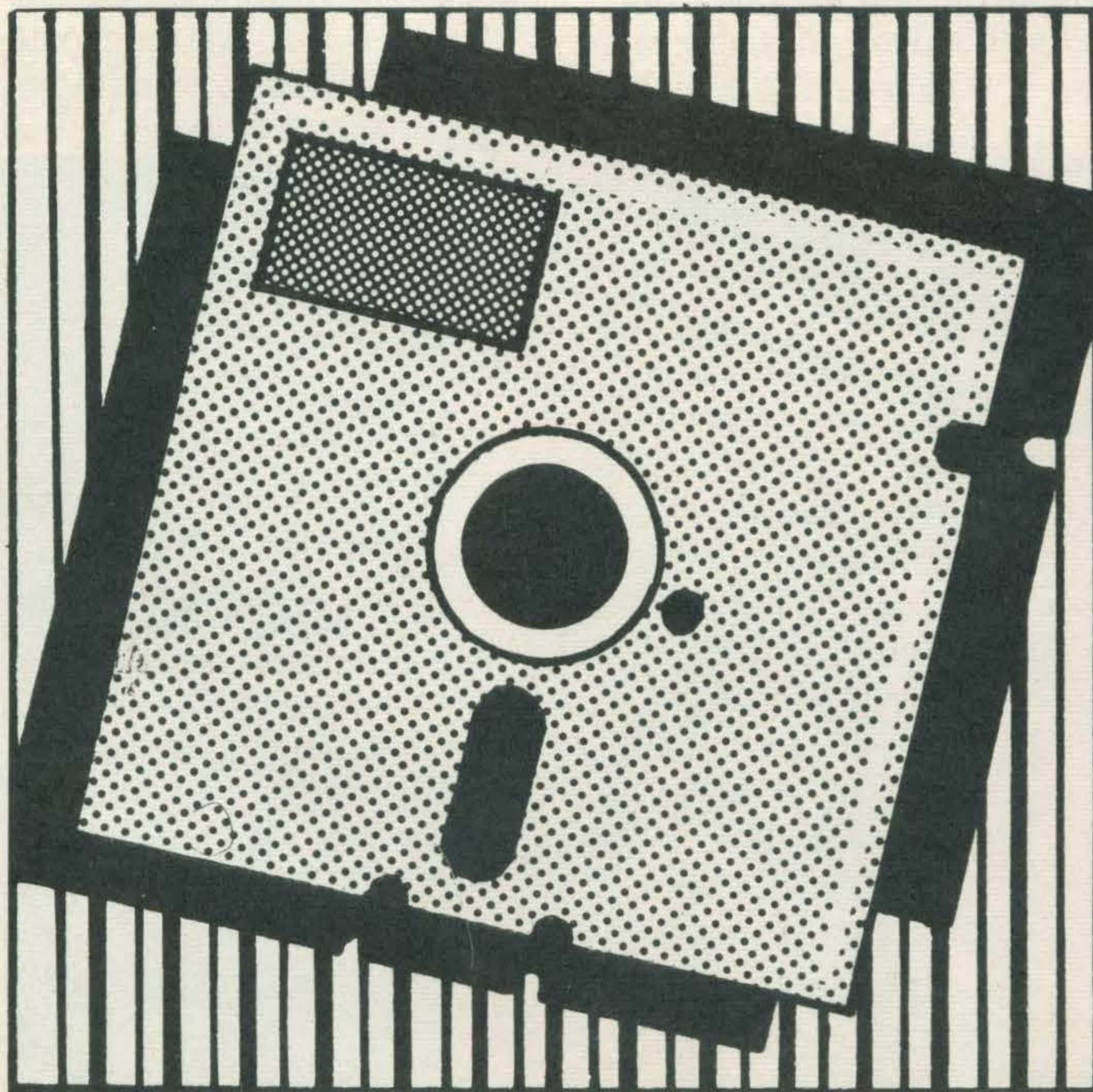
Floppies are strangely named, because they're pretty rigid, coming inside a strong protective metal or plastic casing. Compared with the hard disks that were used before they were invented, they're less rigid, of course. But floppy, strictly speaking, they ain't.

Disks are one area where the famed MSX compatibility doesn't exactly rule supreme, because while any MSX disk drive will fit into any MSX computer, any disk from one MSX drive won't necessarily fit into every other MSX disk drive, because they can be either 3in, 3½in, 5¼in, or even 8in wide.

The good news is that Hitachi, who produce the 3in format, haven't even launched a disk drive in Japa' (though I have a pre-production model, and it works fine) and with the exception of Spectravideo and possibly JVC, no one's producing 5¼in MSX drives, and there are no 8in MSX drives as yet.

That leaves Sony's 3½in drive as the only game in town, and believe me, it's sweet. Sony created this format, which is also used by the hugely popular Apricot and Macintosh machines as well as (rumour hath it) the soon-to-be-announced IBM lap-held computer. They were smart in making it available at the same time the Sony Hit-Bit MSX computer got into the shops before Christmas,

Keep it safe



which gave them something of a headstart, even though it cost almost as much as the computer.

Now others are becoming available, notably from JVC, who haven't yet decided whether they'll also be distributing a 5¼ in drive.

Data cartridge

The data cartridge also owes a lot to Sony, whose invention it is, although to the user it's a lot like the solid-state ROM cartridges that some of the more expensive games come on. The letters ROM stand for Read-Only Memory (I bet you already knew that) and a ROM cartridge is basically a program, on a chip, which can't be rewritten or erased (though if you try plugging or unplugging it without turning off the machine, you'll probably blow it).

The data cartridge is like an ultra-fast disk. It plugs into the cartridge port and hasn't got a lot of memory capacity, but thanks to a built-in battery pack, it holds its memory for four or five years.

When Sony announced it, the only machine you could use it with was their Hit-Bit, but it was only a matter of time before some clever sod worked out how to use it with other machines. Modesty forbids me from naming who that CS may be, but just watch this space in future issues!

The tape streamer is a sort of faster-than-usual tape cassette, using a non-standard cartridge with an endless loop of tape inside. It's not as fast as disk, but it's much faster than cassette, which is presumably why it's going to be marketed under the Quick Disk name.

Confusing, though, because it's not really a disk.

The main difference between cassette, disk, data cartridge, and tape streamer, is that the first and last are **sequential**, meaning that they store stuff in a long stream, one item after the other, so to get to item

number 100, you have to go through items 1 to 99 first, and disk and data cartridge are **random access**, meaning that you can go straight to any individual item.

The disk does this by holding a special directory of where everything is, like the label on an LP record. And then, like swinging the pick-up arm across the record until you find the track you like, the read/write head of the disk drive slides across the surface of the disk until it's reached exactly the sector it wants.

If you've ever compared picking an LP track with the time it takes to wind through a cassette of the same music to find it, you'll realise why disk is much faster than tape, and why no serious application can manage with tape.

The way I've told it, the disk has got to be a certain winner, but it's not as simple as that. For a start, disks are expensive. Also, certain markets seem to prefer one system or the other, so that it's impossible (or nearly so) to get anything different.

So Japan's mainly a cartridge market, and America's mainly disks. But here, in cost-conscious Britain, the people go for

tapes all the time, and so do the software companies. So let's have a look at tapes.

Tapes

The best kind of tape to use is something not longer than C30, and one which requires "normal" bias (which may be described on the cassette with something like "120µs EQ"). Avoid cheap tapes emanating from the more obscure parts of Asia, because while they may work 99 times out of 100, it's a racing certainty that the time they let you down is just after you've keyed in a zillion-line program, and you'll have to do it all again. If you haven't previously LISTED it to a printer, that means going back to square one.

Alternatively, you could go out and buy something to shoot yourself with.

When the computer is connected to the cassette recorder using the correct three-core cable, with the red plug into "mic", the white plug into "ear", and the black plug entering the 'remote' socket on the recorder, control of the motor is normally transferred to the computer (after first pressing the appropriate PLAY or RECORD buttons on the recorder). However, if it is necessary to rewind or fast-

forward through a tape to find the place from which to LOAD or to which to SAVE, you'll need to use the MOTOR command:

MOTOR OFF

returns control to the recorder, and

MOTOR ON

returns control to the computer. If the recorder motor is turning, the MOTOR OFF command will turn it off, MOTOR ON will start it again. Alternatively, the single command MOTOR can be used to act as a 'toggle', stopping the motor or transferring control to the recorder if the motor is turning, if the computer has control, or starting the motor or transferring control back to the computer if the motor is not turning or if the recorder has control.

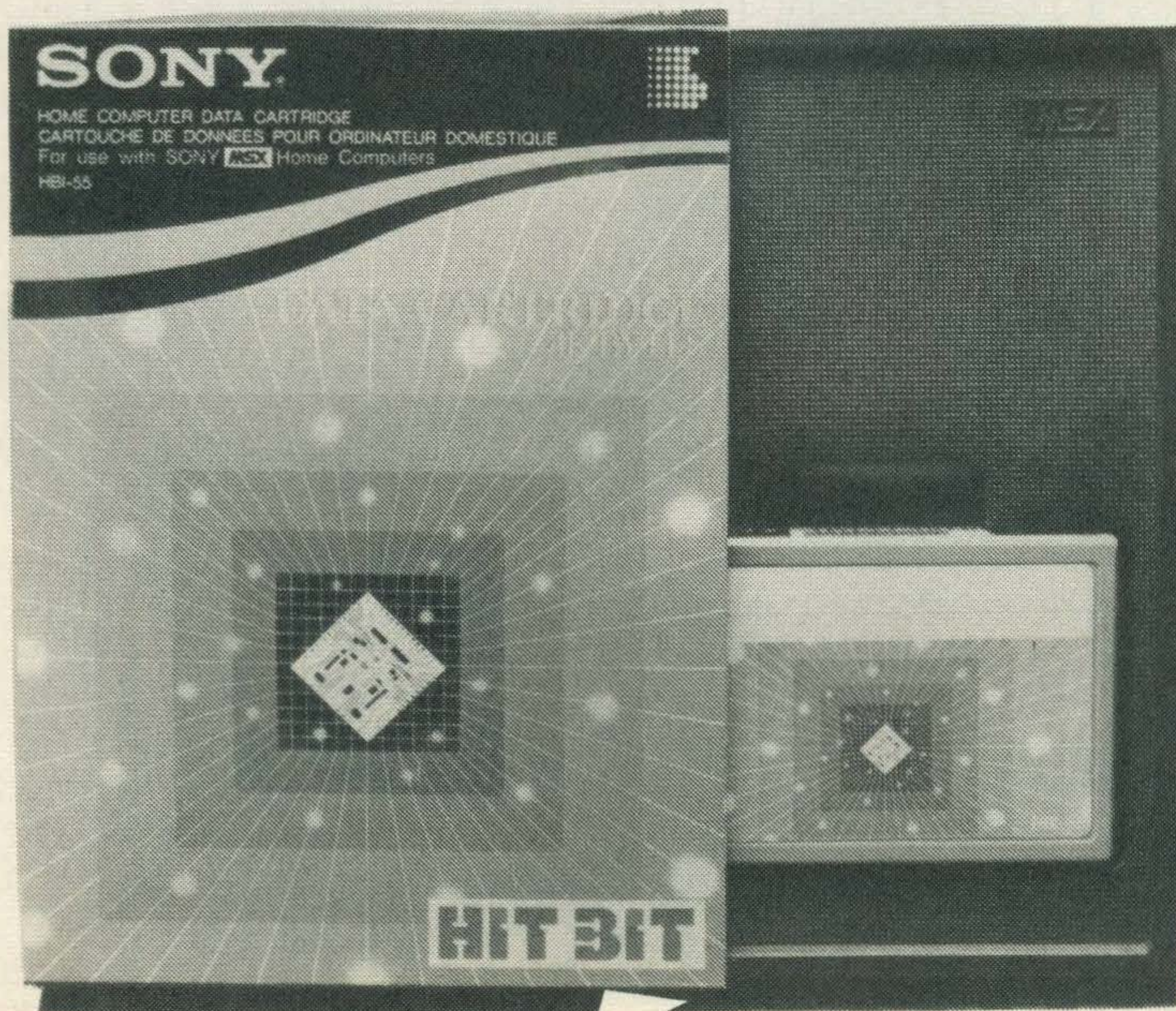
Before you start writing and SAVEing programs (I've capitalised the word SAVE because that's the actual command), you'll probably want to LOAD something from a commercial cassette.

The first problem you'll probably encounter is the playback volume level setting on your recorder, because if it's too high (loud) it will be distorted, and the computer won't understand it, and if it's too low (quiet), it won't be able to hear it. You'll need to do quite a bit of experimentation to get it exactly right, and the level setting may need to be varied from tape to tape.

Sorry, but that's the way it is.

If you play it back without plugging it into the computer, you'll be able to hear the sound of the program – and a dreadful caterwauling it is! If it sounds distorted, then you'll need to turn the volume down a bit.

This way, you can also hear the way the program is constructed. At first, there's a high "header" tone, which tells the computer that a program is about to begin. In this section, also, is a brief bit that tells the computer what baud rate was used to record it (don't worry, I'll explain



about Emile Baudot and his baud rate in a moment). Then the tone drops roughly an octave, and the program proper begins.

At the end, there's another tone to tell the computer that's all folks.

Talking Machine

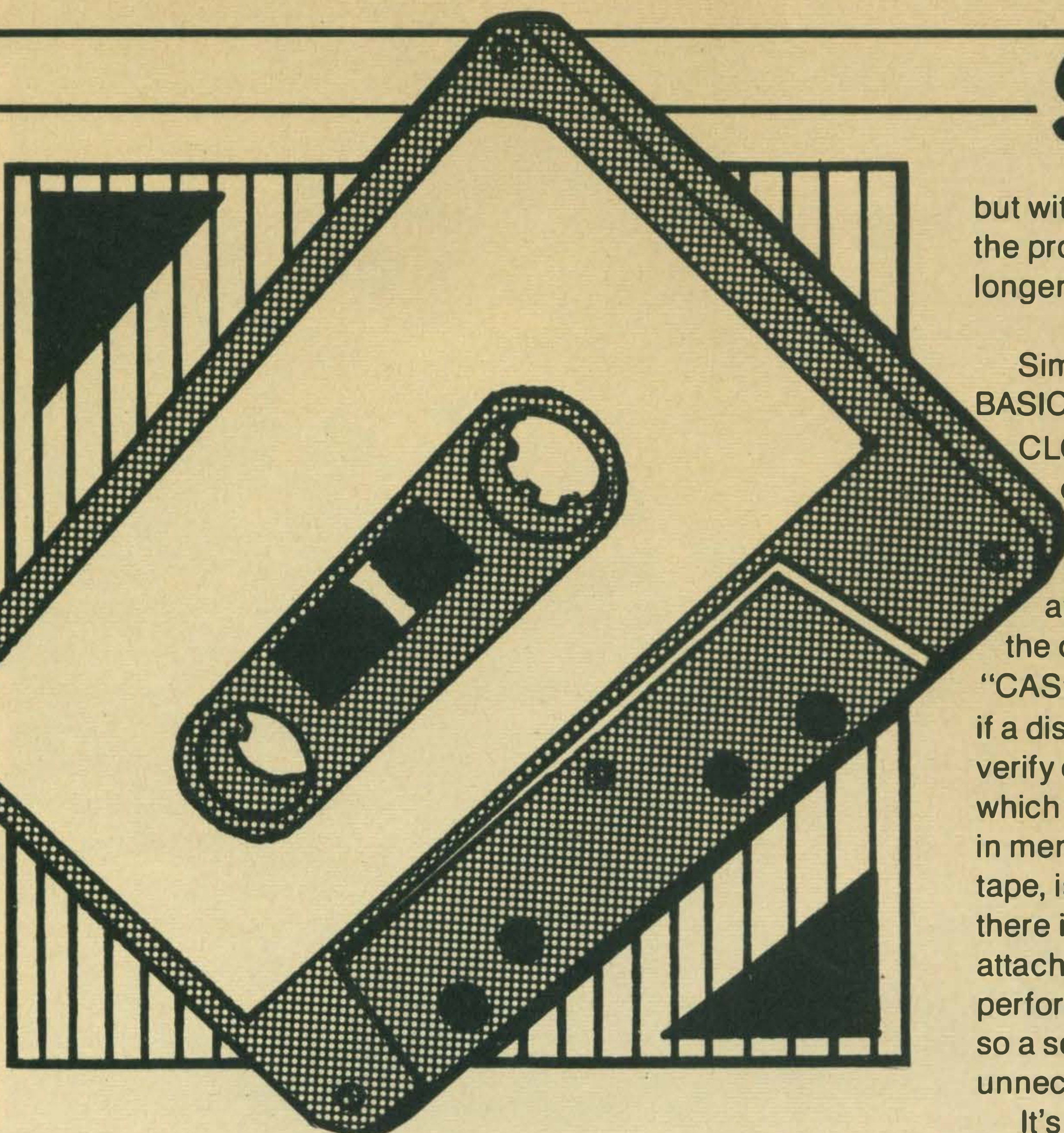
Although most programs you'll write yourself will probably be in the BASIC language, any commercial software will probably be in machine language or binary code. This is because games need to run fast, and if they're written in BASIC, every instruction has to be translated – interpreted is the technical term – into machine language, which is the only thing the stupid machine can really understand.

The command to load a machine language program is BLOAD (Binary LOAD). BLOAD will work on its own, if cassette is all you've got connected, but because it's possible to SAVE programs on other devices than cassette, and you may get a disk drive in due course, it's a good idea to get into the habit of telling the computer what device you're using, by adding "CAS:" after the command. For your own programs, you may need to add the program name (plus the area of memory it needs to be BLOADed into) but with commercial software, this is rarely necessary. If you add "R" after the command, it will RUN automatically, as soon as it's finished BLOADing. So the total command to BLOAD and RUN the first machine language program on a tape would be:

```
BLOAD "CAS:" ,R
```

If you specify a program name, and it's not the first on the tape, then it will display the name of each program it finds, and skip over it, until it finds the one you want.

Pretty soon, however, you'll start writing your own programs and will want to SAVE them, and the chances



are they'll certainly be in BASIC.

If you don't have a disk drive fitted, there are two options to SAVE to cassette: `CSAVE "programname"` which SAVES a program in compressed form
`SAVE "CAS: programname"` which SAVES a program in ASCII text form

The main difference between the two, as far as the user is concerned, is that because the program is compressed, `CSAVE` is faster. Also there is a choice of baud rates, and if a good recorder is used (essential), the higher baud rate will `CSAVE` it even faster.

What is a baud rate? I hear you cry.

Basically, it's the speed at which the program is going to be SAVEd, and gets its name from Emile Baudot, who invented a way of encoding telegraph signals into binary code back in 1874 (no, I'm not making this up, *honestly*) and MSX offers you two speeds: 1200 baud (which you choose by selecting the figure "1" after the `CSAVE` command) or 2400 baud (add a "2").

As in all things in life, of course, you gain something to lose something. The higher

baud rate may be faster, but it may also be harder to LOAD back, especially if your recorder is a bit manky. That's why certain software companies (like Electric, for instance), SAVE their programs on one side of the cassette at the faster, 2400 baud rate, but use the other for the slower, but safer, 1200 baud rate. This is a tip worth remembering, unless you're certain your recorder can actually handle the 2400 baud rate.

Normally, the slower 1200 baud rate is the default if the higher 2400 rate is not specified, but the higher rate can be made the default with the `SCREEN` command.

If a disk drive has not been attached, `SAVE "programname"` (without specifying the device) will SAVE to cassette. `CSAVE` cannot be used if a disk drive is attached and will produce an 'Illegal Function Call' error message if it is used. In that case, `SAVE` with no device specified assumes disk drive, so to SAVE to cassette the `SAVE "CAS:"` option is required.

The program name may be up to 11 characters long, including the device descriptor prefix and colon, eg `SAVE "CAS: 1234567"`

but with `BSAVE` or `CSAVE`, the program name must be no longer than six characters, eg `BSAVE "123456"`

Similar rules apply to BASIC program `LOAD` and `CLOAD`. `LOAD` with no disk drive will apply to the cassette recorder, and to the disk drive if one is attached, and in that case the command to `LOAD "CAS:"`. `CLOAD` is not usable if a disk drive is attached. The verify command, `CLOAD?`, which compares the program in memory with what is on the tape, is also usable only if there is no disk drive attached. `Disk SAVE` performs an automatic verify, so a separate command is unnecessary.

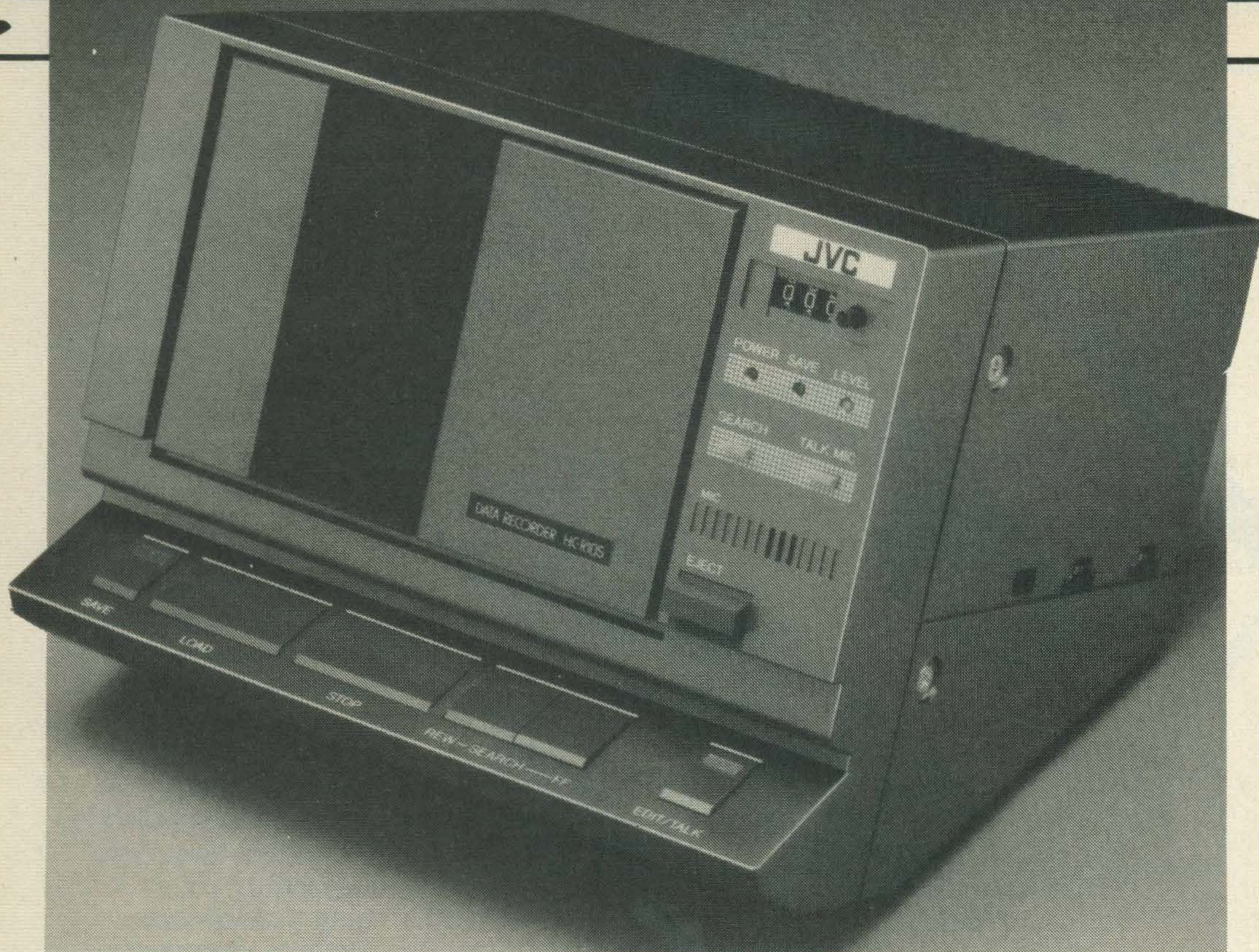
It's possible to use `CLOAD?` to get a list of the contents of a program tape to the screen. Some machines (eg the Amstrad), have a special `CATALOGUE` function for doing this, but you have to fool the MSX machine to get it to do it!

We've already met `BLOAD`. The command `BSAVE` is also used with binary (machine language) programs, and the syntax is similar: no device need be specified if only cassette is connected, but if disk is connected, it is necessary to specify `BLOAD "CAS:"` to direct the command to the appropriate device. In the case of `BSAVE`, it is also necessary to specify the start and end addresses of the program (and a run address if needed; otherwise the start address is taken as the run address default).

Immediate Runs

As we have already seen, if the suffix "R" is added to a `LOAD`, `BLOAD` or `CLOAD` command, the program will be RUN immediately after it is LOADed. Otherwise, you have to type in the three-letter instruction `RUN` (followed by `RETURN`, of course) to execute the program.

With `LOAD`, `BLOAD` or `CLOAD`, if no program name



is specified, the next appropriate file on tape is LOADED into memory.

Every time a program is RUN, everything in the computer's memory (except the program, of course) is wiped out, so if the previous time you ran it, you were asked to type in something, that will have been forgotten, since RUN sets all variables to zero (if they're numeric) or null (" ", if they're strings of characters, like words). If you know the number of the first line-number in the program, you can start it up again without losing this memory with the GOTO command. (If the program has stopped because something went wrong, you can sometimes start things up again where you left off by typing CONT, for CONTInue).

Better, you can store such information on a cassette data file – but I'll get into that in a moment.

RUN" " can be used to LOAD and RUN the next program on a tape; in this, it is identical to the LOAD "CAS:" ,R command.

RUN can carry a line number parameter, so that it is possible to by-pass the opening lines with a

command such as

```
RUN 100
```

but note that all variables will still be set to zero or null in this case. To preserve variables (eg during program debugging), it is better to use the

```
GOTO 100
```

command, as we've seen.

It is possible to use the MERGE command to amalgamate a program from tape with one in memory. The program on tape must be in ASCII text format (which means you didn't use CSAVE), and care must be taken to see that program lines do not overlap. If a line from the tape program has the same line number as one in memory, then the new line will replace the old. If the new program line numbers cover the same range as those in memory, they will become mixed up, probably garbling the entire program. So if the program in memory has lines 10 and 20, and the program to be MERGED has lines 15 and 25, then the amalgamated program lines will run 10, 15, 20, 25 – with unexpected and probably disastrous effects!

When MERGEing programs, it is often necessary either to DELETE

or RENUMBER lines in the program in memory – or both! – to avoid this sort of confusion.

Some BASICS have a CHAIN command to perform a similar function during program execution. MSX BASIC uses RUN "filename" for this purpose. If you've been using any data files (it won't be long before we get to them now) it's possible to keep them open during chaining, but all variables are set back to zero or null.

Unfortunately, it is not possible to combine GOTO, filename and line number to avoid this.

OK, so what *is* a data file?

You probably know that in an office a file is a sort of folder, usually kept in a filing cabinet, in which you can store things you may want to look at from time to time. Well, that's a pretty good description of a computer data file.

Before you get something out of a file, you've got to put it in first. And since it's possible to have more than one file open at a time (up to 15, as a matter of fact), you have to give it a number.

When you open a file, you have to tell the computer

where it can find it (in this case, on cassette), its name, whether you want to take something out of computer memory and output it into the file (print) or take something out and input it into the computer memory, and the file number, like this:

```
OPEN "CAS: filename"
FOR input/output AS 1-15
```

So, to write information into a file called ADDRESSES, you'd type in:

```
OPEN "CAS:
ADDRESSES" FOR
OUTPUT AS 1
```

Since you can have more than one file open at a time, you have to attach the file number to each PRINT or INPUT command, PRINT #1 in this case.

After you've finished outputting to the file, you must CLOSE it. In this case, you can either CLOSE one file, by specifying its number, several files, with their numbers separated by commas, or all files, by not specifying any numbers:

```
CLOSE 1
CLOSE 1, 3, 4
CLOSE
```

You can read the contents of your file in a similar way:

```
OPEN "CAS:
ADDRESSES" FOR
INPUT AS 1
```

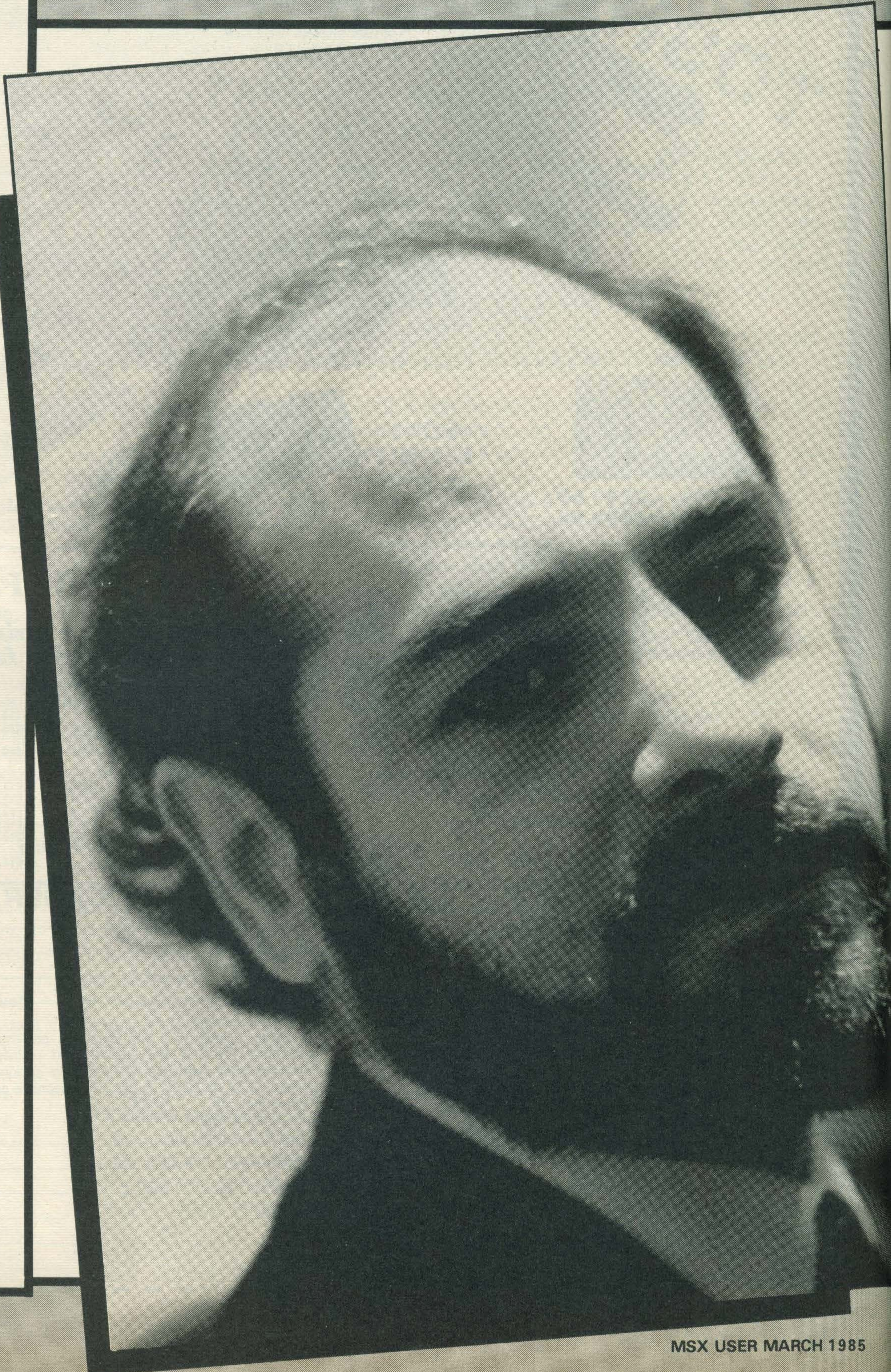
using INPUT #1 to look at the contents, and similar CLOSE commands.

Remember how I explained that tape was a *sequential* medium, meaning that it stored things in a long stream, one item after the other, and you had to read right through them all to get to the last one? This is also true of tape data files. But if you store your data files on disk, they can be *random access* files, meaning that you can pick out one record from right in the middle of the file.

Obviously, I haven't got space to go into all the ins and outs of how to manage data files, but next month, I'm going to look at how the disk drive works, and how these commands have to be modified to work with disk.

All right then, who wants to be an entrepreneur? Karl Dallas interviewed Mike Margolis to see how it's done.

Whizz kid



It seems like a double bluff to me. You know the sort of thing: I don't want you to know what I'm thinking, but instead of telling you something else, I tell you what I'm *really* thinking, because you're gonna be sure that if that was what I was thinking, that'd be the last thing I'd say.

If that seems just a shade convoluted, let me apply the principle to Mike Margolis.

Mike Margolis is described by Sony as their New Business Manager, which has a real whizz-kid sound to it, you'll agree.

In fact, Mike Margolis is such an obvious whizz-kid, with his neat beard and ready way with words and New York accent, that all my instincts incline me to write it all off as a bluff.

Most of the true whizz-kids I've ever known have been quiet, self-effacing Johnnies, and you only find out how brilliant they are by accident.

But having known Mike Margolis for a few years now, I'd say the old double-bluff is definitely working.

This man is a genuine whizz-kid, even if he looks and sounds like one.

Before I'd ever heard of MSX, before it was even a twinkle in the eyes of ASCII Microsoft, the Japanese-American company who first devised the MSX standard, I'd noticed that Mike Margolis tended to be around when new, untried, risky and – ultimately – successful Sony projects left the launching pad.

Take the little matter of the tape recorder that wouldn't record, which everyone said would be a failure.

Ostensibly, it was a tape

recorder system, but research showed that it couldn't possibly catch on. For a start, it couldn't record. It had no speakers, and the stuffy British public was being invited to invite ridicule by walking down the high street wearing headphones.

It was also too expensive.

"As soon as I put the headphones on, I knew it had to be a winner", he recalls today, and of course he was right.

The Walkman was a winner.

Then there was the TV system that needed at least three different boxes to house it and cost closer to £1000 than £400. It had a monitor, with a built-in stereo amplifier (who needs a stereo amp for TV? people scoffed), and a separate tuner with sockets for two video recorders.

At that price, of course, the Profeel component TV system has hardly had the runaway success of the Walkman, but it's definitely become a status symbol. Enough for a number of other TV companies to start copying the concept.

And now Mike Margolis is launching yet another consumer product that has divided the loyalties of the computer business: MSX. Would you be surprised to learn that once again he thinks he's on to a winner?

One thing about Margolis, you don't feel you'll get anywhere pussyfooting around, so I thought I owed it to him, to myself and, especially, to you the readers, to give the questions to him straight, the way they come from you to me, if you're still wondering whether to go MSX:

Why are the computers still so thin on the ground in the shops?

Isn't the MSX standard out of date?

Aren't all MSX computers the same under the skin?

And the 64 thousand dollar question:

When are prices going to start coming down?

To be fair to Mike Margolis, he took it all very well. But then, don't let the whizz-kid exterior fool you, he's really into computers. He may look as if he'd sell anything to anybody, if the profit margin was right, but when he talks about the future of MSX you've got to remember that he speaks with the authority of a man who had one of the first working micros in this country, because he built it himself.

"A friend gave me one of the first samples of the 8088 chip, and I built my own computer", he told me. "It had 256 bytes of memory and sometimes it even worked. I couldn't do much with it except learn about computers.

"I taught myself assembly language, learnt about computer architecture and – most important – learnt about the difference between the theoretical capabilities of the hardware, and a final product that could do something useful. I knew what I wanted to do with it, but the product just didn't exist, as yet".

Margolis bought himself an Apple II when they first became available, and he wrote a sales-forecasting program for it that the company still uses – though it's been modified considerably and adapted to run on

the IBM PCs that the Sony sales department has today.

When he graduated in electronics engineering from Syracuse University he set out to travel round the world, heading for Japan – but he only got as far as Britain when his money ran out. He ended up working for Sony UK, for a short time in Britain, then in Switzerland, and finally back in Britain, where he's been ever since.

His first impression of the micro revolution was that built-in obsolescence was a bit of a drag.

"It was a complete and utter pain in the ass", he recalls today, and there are plenty of users of machines like the Dragon, the Ace and the Newbrain who'd agree with him there. "No sooner had you got used to using a computer for a particular application than it became obsolete. That didn't make sense to me as a consumer – and it certainly didn't make any sense to me as a manufacturer.

"There was a very clear need for a system that could grow with a person's needs and sophistication. What were needed were computers suitable for today's needs that could still be around in ten years' time".

The answer to that quotation, as we all know, was MSX.

Meanwhile, Sony's corporate thinking was moving from considering the various items they sold – TV sets, hi fi's and the rest – as separate units, but more as part of an integrated home entertainment centre, in which the micro would play a significant role.

"There were a number of projects we were involved in:

"Component TV – the Profeel system – with its potential to connect up all sorts of devices that need a TV screen;

"Digital recording media, later to emerge as the compact disc;

and

"Communications systems like Viewdata.

"We had already begun thinking how intelligence could be added to the home entertainment system. When we were developing the compact disc, we built in features to prevent obsolescence – one of which is the ability to store text on the disc alongside the music. So it would be possible to display on the TV screen the selections playing, or even put up programme notes.

"The compact disc has a capacity of many, many megabytes. It's an ultra-reliable, ultra-fast method of storing digital data. There is no theoretical reason why it should not contain all text, incidentally, producing a software medium that should be piracy-proof.

"Sony were also one of the first two suppliers in the launch of Prestel, the British version of Viewdata. That was tremendously successful in industrial and commercial applications, though it still hasn't captured the imagination of enough of the home market, probably because the possibility of instant communication of information down a telephone line and hard-copy print-out was more applicable to business, and the range of services available were not

related to the needs of the consumer.

"It also became obvious very soon that its eventual success would depend on the intelligence of home computers providing a more useful front end".

OK, so much for the corporate philosophy. What about my cotton-pickin' questions? Like the one about MSX being old-fashioned, fairly slow computing technology, when the sharp end of micro-processing development is wrapped up in ultra-fast 16-bit hardware? In reply, Mike Margolis points to the example of TV.

"TV is not state of the art" he reminded me. "It's an obsolete technology, if you like. I could take you into Sony's development lab to see the high-definition TV we have. The real question is not the maximum level which we can achieve, but what relates to the customer in the home".

But if having a common standard means uniformity, does that mean all MSX machines are basically the same?

"You ain't seen nothin' yet", exclaimed Mike, when I put this point to him. "What we are seeing at the moment are the first generation of MSX machines built to the MSX standard. It's not been possible to fully exploit all the capabilities and variations possible behind the thinking of MSX.

"MSX is merely a platform upon which individual manufacturers can build their individual personalities into the machine. I draw an analogy with the compact cassette. It has to be a very specific size, have a very



specific tape width, even the EQ is exactly specified. But that doesn't mean that all compact cassette recorders are the same. Nor does it mean that all compact cassettes will work equally well in all machines".

The epitome of this individualistic philosophy is of course Sony itself, which has already produced the most individualistic of the first MSX machines, the Hit-Bit, with its 16k extra ROM and built-in "databank" software. But doesn't that run counter to the whole idea of MSX compatibility, since Sony data cartridges can't be used with non-Sony machines?

"The data cartridge does not carry an MSX logo", Margolis pointed out. "But you can still plug any other MSX peripheral into the Hit-Bit, and it will still work.

"Of course, you can also use data cartridges with other machines, as long as you have written the necessary software to handle them. There's already a word processing application in the pipeline which will use data cartridges as a storage medium, and that will run on any MSX machine.

"I envisage there will be more and more specialised MSX machines, dedicated to video applications, for instance, while still offering all the benefits of software compatibility. They'll be MSX machines, plus.

"People are going to do much cleverer things than just changing the RAM size and the styling. There will be some which will offer a certain level of convenience instead of facilities like the data cartridge.

"I expect there will be strong moves to develop machines to help the TV set to be used as a communications link, or to be an adjunct of the video recorder, for instance to overlay titles and other material material on to video tapes".

What about price, though? Is someone going to break through the ring and produce a basic machine at a basic price?

"I envisage the same thing will happen to MSX as happened to the Walkman. There will be machines with less features at a lower price, and others with more at a higher price. We have a Walkman at £30, and we have one at £200. But in each price category, Sony always try to introduce products in the upper end of that category".

What about supplies? They're still very thin on the ground.

"At this moment, "he agreed", our biggest problem is deliveries. Product availability is relatively small compared with what will be available in 1985. We hope to see a big improvement in the supply situation during the coming year".

FUTURE MUSIC

10 BADDOW ROAD, CHELMSFORD, ESSEX. TEL (0245) 352490
202 NEW KINGS ROAD, FULHAM, LONDON SW6. TEL. (01) 731 5993

104/106 ELM GROVE, SOUTHSEA, PORTSMOUTH, HANTS. TEL. (0705) 820595
85 ST. MARY'S STREET, SOUTHAMPTON. TEL. (0703) 26798

NEW!

YAMAHA CX5 MUSIC COMPUTER

If you're looking for a personal computer to make music, look no further! Yamaha's amazing CX5 offers the same incredible FM sound synthesizer quality as its famous DX synthesizer series! Plus all the features of the innovated MSX computer system!

CX5M Music Computer

The CX5M is an extremely versatile computer specifically designed for a wide range of music generation, programming and editing tasks, and for interfacing with other Yamaha digital instruments and components. The CX5M is a MIDI compatible computer, allowing it to serve as a control centre for playback and automatic sequencing of the Yamaha DX series synthesizers, RX drum machines and other MIDI compatible equipment.

The CX5M also has a Yamaha digital FM voice generator built-in — the same type of voice generator that has put our DX series Digital Programmable Algorithm Synthesizers at the forefront of the digital keyboard field. That means it is capable of producing rich, realistic sounds that are almost indistinguishable from acoustic instruments. In fact, 46 fine voices are provided pre-programmed. But you can also program your own to create virtually any voice you like. And you can save your original voices on a standard cassette tape.

A wide range of applications programs, interface units and accessories expand its music making potential enormously. **Sophisticated Music Software YRM101 FM Music Composer**

This optional program cartridge is a must for all composers and arrangers. It is simply the most sophisticated, versatile, easy-to-use music composition and arrangement package in its class. You get an on-screen music staff onto which you "write" notes by inputting them either from the computer keyboard or directly from the music keyboard. The ability to input notes from a piano-type keyboard is a real bonus for musicians. Up to eight parts can be entered, and each part can be assigned a different "instrument". You can use the superb pre-programmed FM voices in the CX5M, or voices you have programmed yourself. You also have full control over time signature, key signature, tempo, dynamics and phrasing. What's more, any parameter can be changed at any time during the piece. Of course, your compositions can be saved on a standard cassette recorder and reloaded whenever needed.

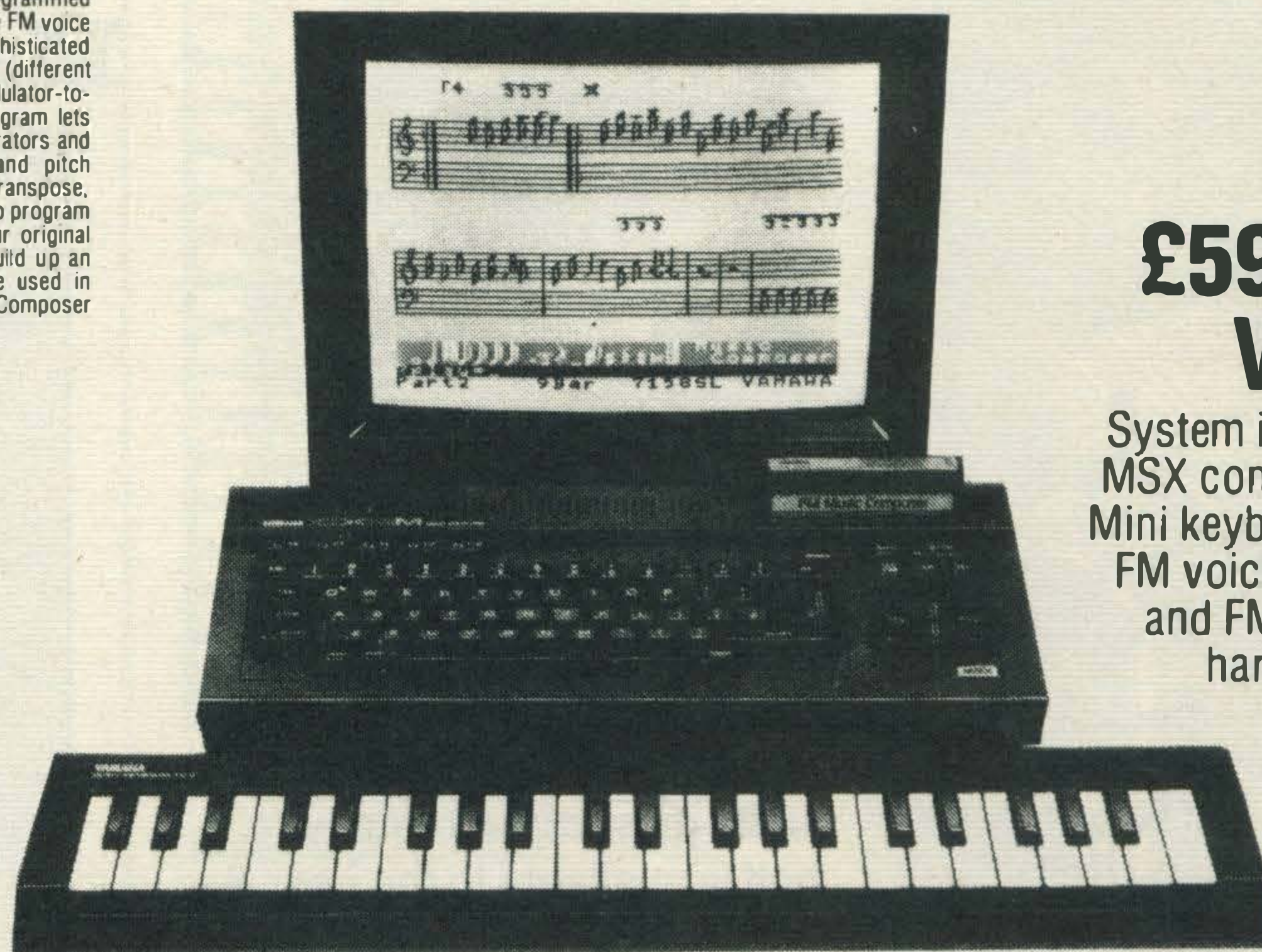
Furthermore, compositions written on the FM Music Composer can be used to "sequence" MIDI synthesizers, drum machines, etc via the CX5M's MIDI interface. The possibilities of this software package are virtually endless. **YRM102 FM Voicing Program**

This program gives you precise control over the CX5M digital

FM voice generator to edit and alter the pre-programmed voices or create totally new voices of your own. The FM voice generator employs 4 operators, each with a sophisticated envelope generator, and a choice of 8 algorithms (different configurations of operators with different modulator-to-carrier relationships). The YRM102 FM Voice Program lets you precisely set all parameters relating to the operators and algorithms, as well as extras like amplitude and pitch modulation, LFO waveform, keyboard scaling, transpose, etc. With a little practice you should be easily able to program just about any voice you can imagine. Save your original voices on any standard cassette recorder, and build up an original voice library. Voices you create can be used in arrangements created with the FM Music Composer program.

YRM103 DX7 Voicing Program
DX7 owner's, here is the key to easy DX7 voice programming. This program displays all DX7 voice parameters right on the video monitor, and lets you program from the CX5M computer keyboard. The data is transferred to the DX7 via the built-in MIDI interface. Voice parameters are displayed in easy-to-understand graph form. For example, when programming envelope generator parameters you can actually see what the programmed envelope curve looks like, rather than having to think entirely in terms of numbers. The DX7 voicing program makes programming the DX7 so easy, that even if you're not interested in the CX5M's other capabilities, it's worth having one just to program your DX7.

YRM104 Music Macro
The Music Macro is for people who want to incorporate top-quality musical voices into their BASIC computer programs. The Music Macro adds a special set of commands to the CX5M MSX BASIC language, permitting control of the digital FM voice generator from within BASIC programs. This makes it possible to program games or audio/visual type programs incorporating music or sound effects using FM voices for real quality and impact.



£599 inc VAT

System includes CX5 MSX computer, YK01 Mini keyboard, YRM12 FM voicing software and FM cartridge hardware!

NEW!

ROLAND DG DXY880 INTELLIGENT XY PLOTTERS

Incredible sophistication and unbeatable price of £699 inc VAT

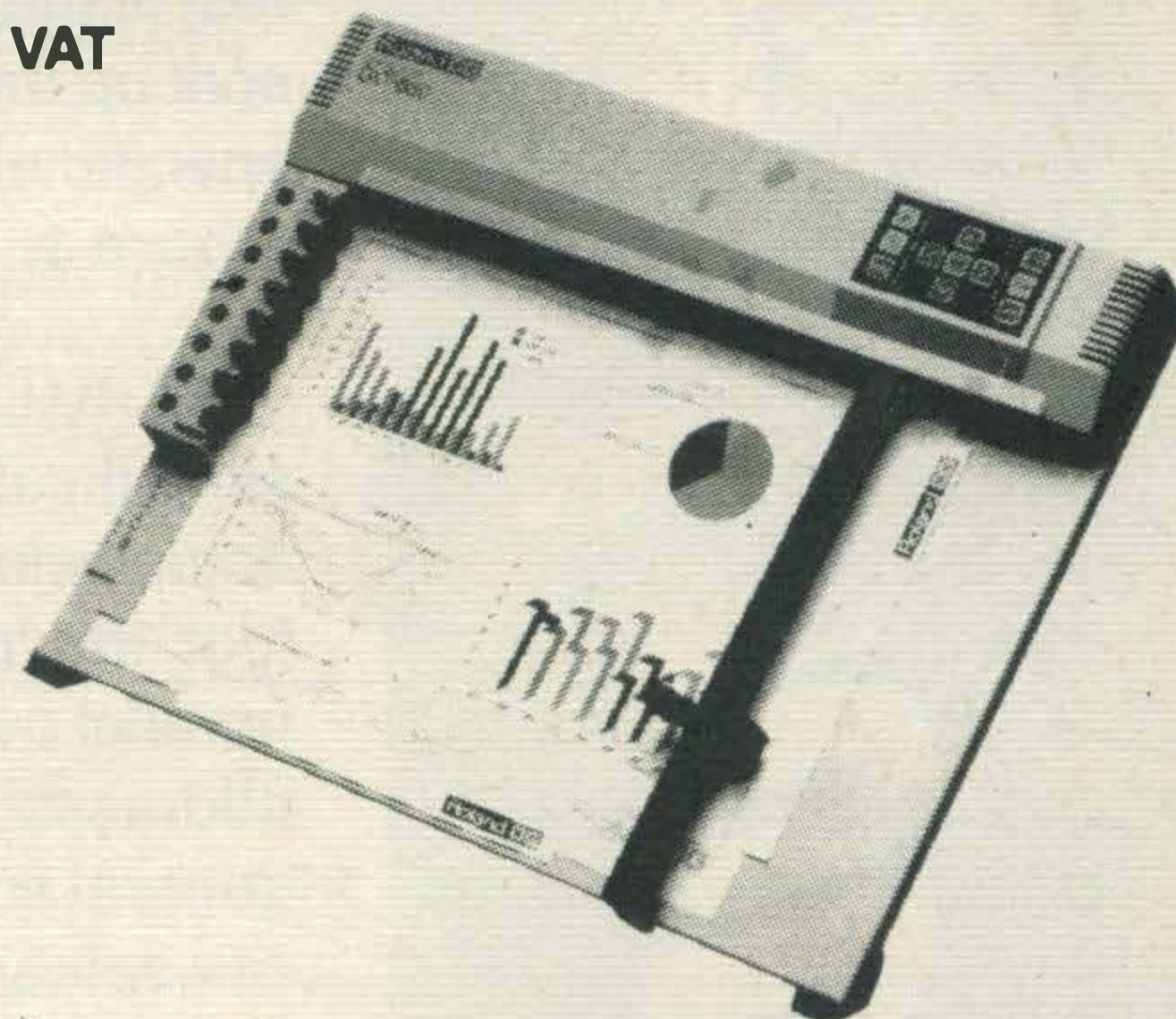
Quick and Accurate. The DXY-880's maximum plotting speed is 200 millimetres per second in all plotting directions. Thus, the DXY-880 ensures stable line quality at all times. The DXY-880 plots with a resolution of 0.05 millimetres per step. The graphics produced by the DXY-880 are excellent, even when projected on a screen by an overhead projector.

Compact. The DXY-880 has an effective plotting area of 380x270 millimetres (ANSI B or DIN A3 size). But the complete unit is only 553 millimetres wide and 435 millimetres deep. Furthermore, the DXY-880 can operate on a 60-degree incline to minimise installation space using a stand. The stand can be folded up inside the body when not in use. And the DXY-880 weighs only 4.3 kilograms so you can easily carry it anywhere.

Compatibility. The DXY-880 has both Centronics parallel and RS-232C serial interfaces. It is compatible with the IBM PC, Apple and other micros. The DXY-880 can also operate on most programs written for the H-P 7470, H-P 7475, and our DXY-800. The Lotus 1-2-3, pfs Graph, and BPS are only a few examples. New software will be introduced to the market continually. For the user who wants to write a program by himself, DXY and RD-GL commands are provided.

And Much More. Roland DG also supplies a variety of options, including an overhead transparency kit, several kinds of pens, and connection cables. Use the Roland DG's optional accessories to create beautiful graphics.

DXY-880 SPECIFICATIONS. Plotting area: X-axis 380mm, Y-axis 270mm. Plotting speed: 200m/sec in all directions. Recording paper size: 420x297mm (ISO A3), 17x11in (ANSI B). Recording paper setting: Paper holder and magnet strip. Resolution: 0.05mm/step. Distance accuracy: ±0.5% or less of travelling distance. Repeatability: ±0.3mm or less. Switches: Pen up/down, Home, Pause, P1, P2, Enter, Position (<, >). Fast, Power. DIP switches: SW-1, SW-2. LEDs: Power/Error, Pen up. Data buffer: 3K bytes (expandable to 10K bytes). International character font sets: English, German/French, Scandinavian, Spanish/Latin, Japanese, Special Symbols. Number of pens: 8 (black, red, blue, green, purple, brown, orange, pink). Power supply: AC adapter (DC 9V, 28V). Power consumption: 35W. Operation temperature: 0°C to 40°C (32°F to 104°F). Operation relative humidity: 20% to 80% (no dew forming). Dimensions: 533(W) x 90(H) x 430(D) mm (21.0" x 3.5" x 16.9" inches). Weight: 4.3kg (9.5lb) without AC adapter. Accessories: XY-4SPB-WN pen set x 1, XY-4SPC-WN pen set x 1, Magnet strip for holding paper x 2, Positioning seal x 1, XY-4PH pen holder set x 1, AC adapter x 1, Vinyl cover x 1. **DXY-880 INTERFACE SPECIFICATIONS.** Centronics Parallel Interface. Input signal: STROBE (1 bit), DATA (8 bits). Output signal: BUSY (1 bit), ACK (1 bit). I/O signal level: TTL level. Transfer system: Asynchronous. **RS-232C Serial Interface.** Transfer system: Asynchronous, Half-duplex data communication. Baud rate: 50, 70, 110, 134.5, 200, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200 or 9600 baud. Stop bit: 1 or 2 bits. Parity check: Odd, Even, None. Data bits: 7 or 8 bits. Connector: DB-25S



REAR PANEL

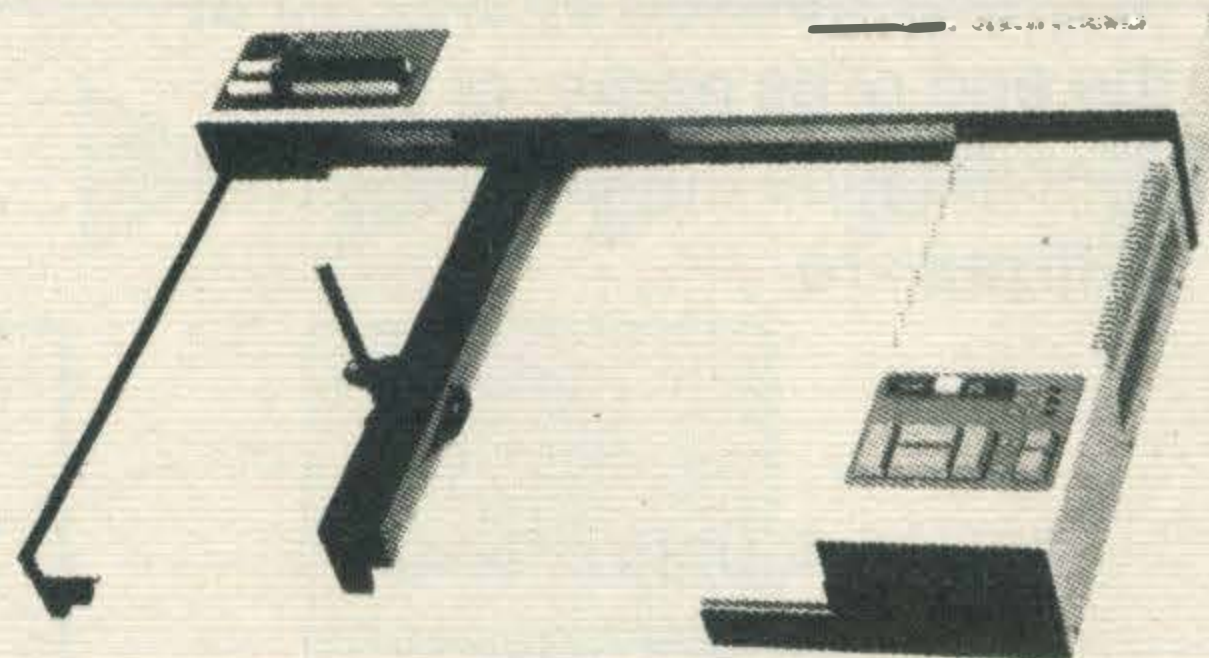


SPECIAL OFFER!

ROLAND PLOTTER PRICE MADNESS!! DXY 100R PLOTTERS

Ideal for BBC, Apple, Spectrum, etc ~~£699~~ now only £299 inc VAT

Bulk purchases make these Roland DG DXY100 A3 Plotters available at under HALF PRICE!! With free extension ROM 30 software and one year warranty! Future Music is a main agent for Roland DG computer peripherals. Call us for further information, demonstrations or advice!!



● Plotting speed of 70mm/s. Plots up to 360 x 260mm (14.2" x 10.2")
The DXY-100R, the first of its kind, is offered at an extremely reasonable price. This revolutionary plotter is characterised by full high-performance plotting and tabulating capacity. Quiet operation is an absolute requirement for a practical plotter. The DXY-100R successfully reduces mechanical noise to a minimum. It is suitable for use either at home or in the small-scale office. Effective plotting and tabulating size is up to 360 x 260mm. Since each step equals 0.1mm, calculation during programming is simple.
● Multiple intelligent functions
Fourteen control commands are included in the DXY-100R. The DXY-100R also provides eight vector commands for plotting and tabulating as well as drafting continuous lines, dotted lines and coordinates; five character commands to select English capital or small letters, numerals, various other symbols, and to set their size and slant. There is also a

built-in mode command. Since each command is very simple, programs are easy to master, even with BASIC. In addition, the DXY-100R includes a self-test function to quickly check performance and operation.
● Optional ROM for expanded intelligent functions
The DXY-100R has an added intelligent function for graphing, including circular arcs, curves, hatchings, etc. with the optional ROM #1 (XY-OR1). This allows simple programming to generate more sophisticated tabulating. Moreover, with the DXY-100R, it is possible to tabulate original characters such as trade marks and symbols simply by writing them into the PROM (2716).
● Compatible with virtually any personal computer
The DXY-100R can be connected to any computer with Centronics specification printer compatibility. Since computer output connectors differ, the DXY-100R is not supplied with connecting cords. Use appropriate connecting cords available separately.

PLEASE SEND ME THE FOLLOWING POST FREE
MSXU 3/85
 ROLAND DXY880 DXY-100R
NAME _____
ADDRESS _____
DEBIT MY ACCESS/BC



To be a good programmer, you need to understand how data is stored in memory, right? Bob Maunder sets the scene.

M SX machines have 32K of ROM and at least 8K of RAM. On a typical 64K micro, memory addresses are numbered from 0 to 65535, or in hexadecimal H0000 to HFFFF. The first half of this is taken up by MSX BASIC, while most of the remaining half is available to the user. A portion at the very top of memory (HF380 to HFFFF) is reserved for system variables – a scratch pad for the MSX system – while the rest is the user area. This is split up into several areas with specific roles, as shown in Figure 1.

In reality, this is an over-simplification as there are actually four 'slots', each of four 16K 'pages' which act in parallel over the 64K of addressable space – but in general such esoteric aspects can safely be ignored!

Two general types of data can be stored in MSX BASIC – **numbers** and **text**. Numbers can consist of several formats – **integer**, **single precision real** and **double precision real**. Each type is normally defined by a suffix at the end of its name, or by a specific type declaration. The suffixes are

- # ... double precision
- ! ... single precision
- % ... integer
- \$... text string

No suffix implies a double precision number by default.

Types may be specifically declared by the instructions DEFDBL, DEFINT, DEFSNG and DEFSTR. All of these specify that data-names beginning with certain characters should be of a given type, eg:

```
DEFINT I-N
DEFSTR S, T, Z
```

Hence any variables whose names begin with I, J, K, L, M or N will be integers, while those beginning with S, T or Z will be string variables.

Let's look at how data is stored in these different types of variables. The 'variable area' of memory (see Figure 1) holds data-names and their contents. Integers take up two bytes as we have seen, and some examples are shown below:

	byte 1	byte 2	
33 decimal stored as	21	00	hexadecimal
66 decimal stored as	42	00	hexadecimal
1026 decimal stored as	02	04	hexadecimal
- 33 decimal stored as	DF	FF	hexadecimal

The value is given by byte 2 × 256 + byte 1. In other words, the high order byte comes second, rather than first, as we might have expected. Negative integers are stored in two's complement form, ie. 2¹⁶ minus the number.

Single precision reals are stored in four bytes, but by a different convention to the pure binary of integers. The **exponent** is held in the first byte, and the **mantissa** in the remaining four bytes. Each digit of the mantissa is

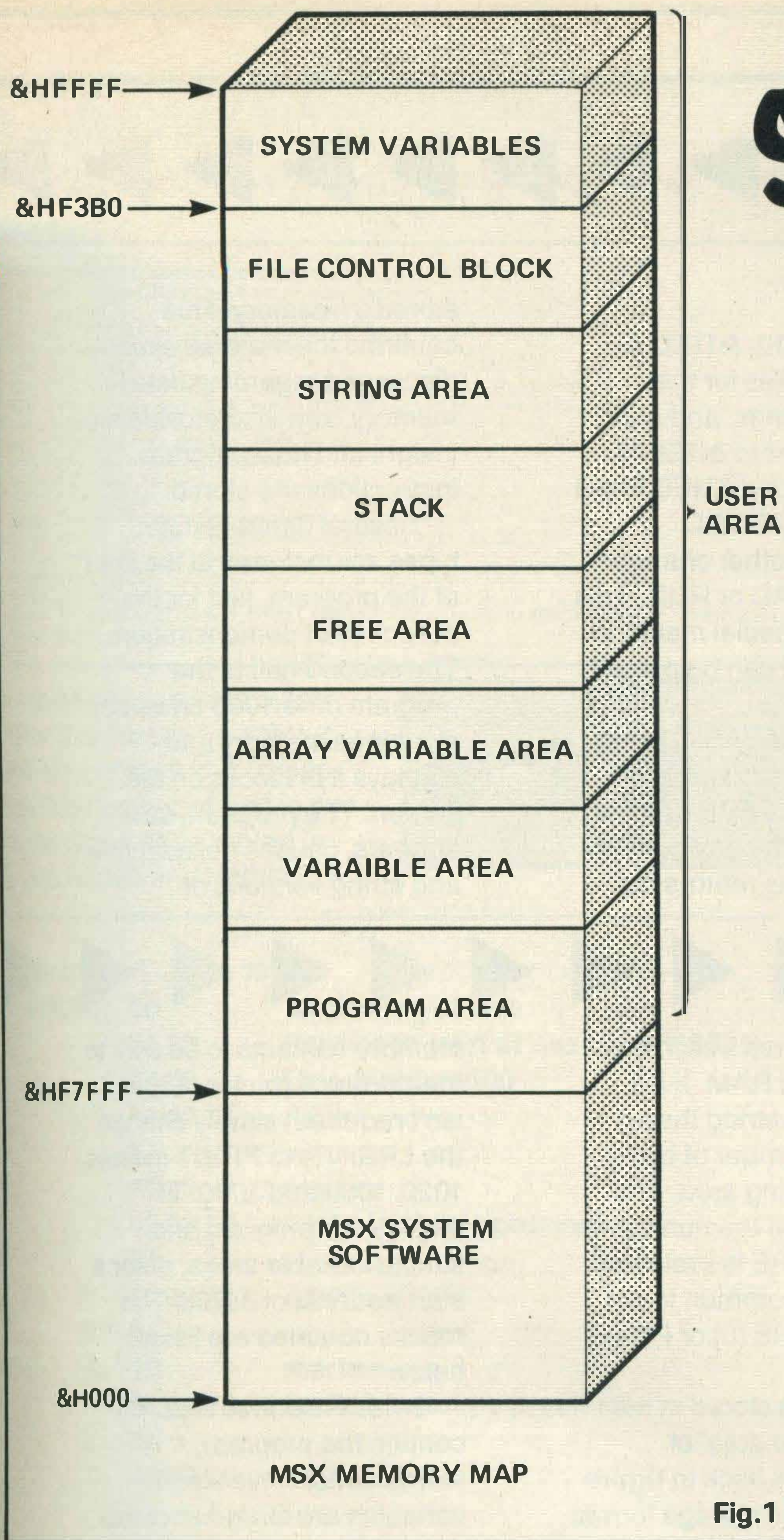
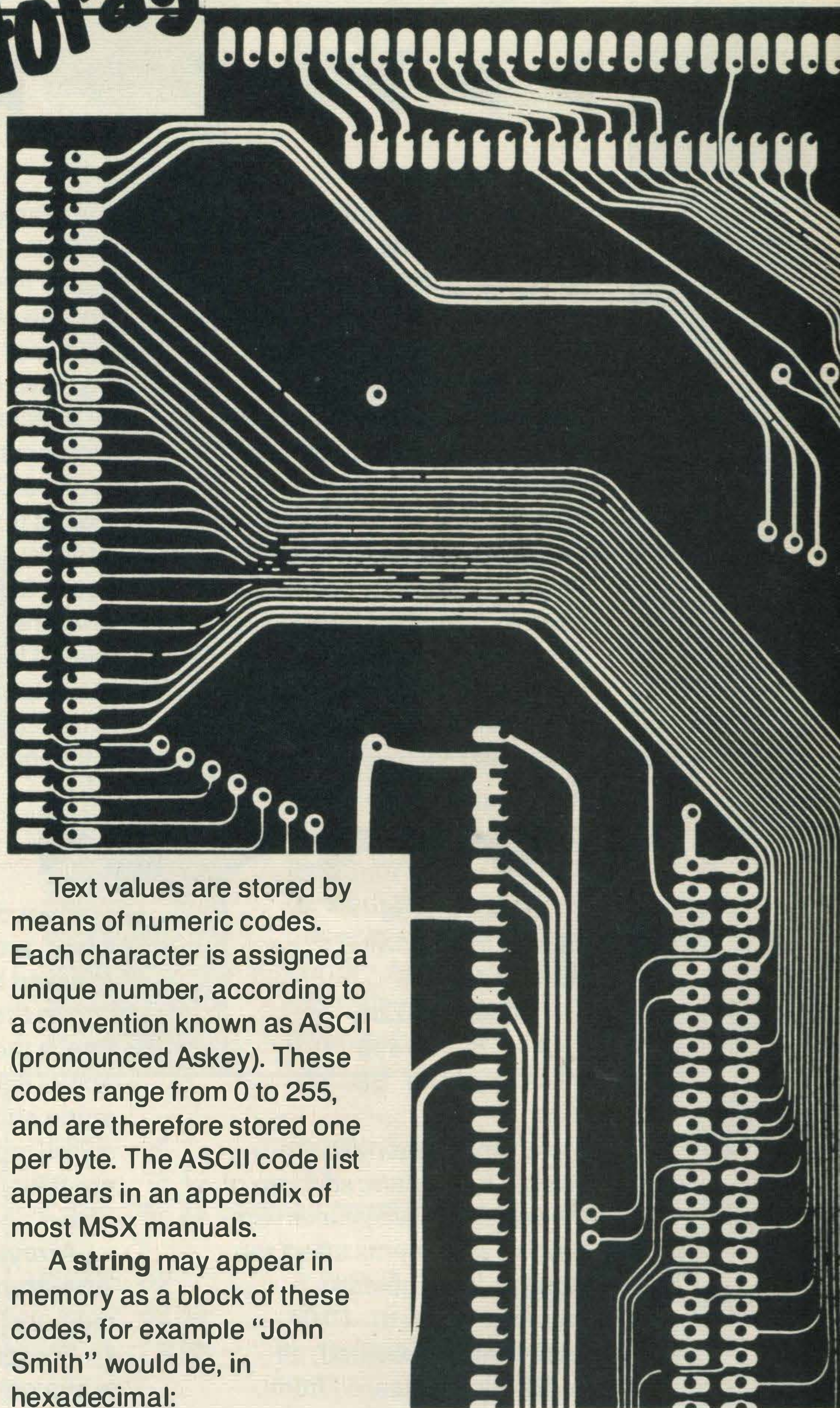


Fig.1

Storage



Text values are stored by means of numeric codes. Each character is assigned a unique number, according to a convention known as ASCII (pronounced Askey). These codes range from 0 to 255, and are therefore stored one per byte. The ASCII code list appears in an appendix of most MSX manuals.

A **string** may appear in memory as a block of these codes, for example "John Smith" would be, in hexadecimal:

4A	6F	68	6E	20	53	6D	69	74	68
J	o	h	n	space	S	m	i	t	h

represented as its binary value, using half a byte per digit. This is known as **Binary-Coded Decimal (BCD)**. The exponent is in binary but in 'excess 64' format – in other words it has 64 added to it, so that for example, an exponent of -30 would be stored as 34. A negative mantissa is shown by changing the highest bit of the exponent to a one, or effectively adding 128 to it. Examples of single precision values appear below.

Double precision data is stored in exactly the same manner as single precision except that seven bytes are available for the mantissa, instead of just the three in single precision. For example:

1.234D20	decimal stored as	55	12	34	00	00	00	00	hex
-35.125	decimal stored as	C2	35	12	50	00	00	00	hex

35.125	decimal stored as	42	35	12	50	hex
-35.125	decimal stored as	C2	35	12	50	hex
0.00035125	decimal stored as	3D	35	12	50	hex
1.23E-30	decimal stored as	23	12	30	00	hex
9.8765E-52	decimal stored as	0D	98	76	50	hex

There can be numeric strings as well as alphabetic ones, but by now it should be clear that "34.56" is stored in a very different way to 34.56. In fact, it's not possible to do arithmetic on strings, without first converting them back to proper numeric form, by means of the **VAL** function.

Returning to characters in general, character codes are used to store the first two characters of a data-name immediately in front of the data itself. For example, the statement

```
AB%= 23456
```

causes the following to be stored:

NAME	VALUE
A B	23456
41 42	A0 5B

As more data names are used in a program, so more of the **variable** area is taken up. Several statements affect the storage of variables in memory. When the **RUN** command is executed, all variables are cleared from memory. This can be a disadvantage in some cases, since no previously stored data can be re-used. However

as an alternative, **GOTO** can be employed to start a program without losing the contents of variables – **GOTO** should be followed by the first line number to be executed.

The **CLEAR** statement also clears the variable area. It doesn't affect program instructions, however. **CLEAR** may optionally have two parameters – the first defines how many bytes will be allocated to the 'String' area of memory (see Figure 1), and the second specifies the upper limit of RAM available to a user's BASIC program.

For example:

CLEAR 500, &HE000 allows 500 bytes for the storage of strings, and sets the top of RAM to **&HE000**. No bytes beyond **&HE000** will be affected by BASIC programs, or other changes such as **CLOAD** or **RUN**. This means that special machine code routines can be placed above **&HE000**.

A final MSX BASIC facility related to the storage of variables is the **FRE** function. **FRE** followed by any numeric within brackets returns the

number of bytes free in the 'User' area of RAM. If it's followed by a string then it shows the number of bytes free in the string area. The actual value of the number or string after **FRE** is irrelevant, so the most common forms are **PRINT FRE (0)** or **PRINT FRE (" ")**.

Arrays are stored in the 'array variable area' of memory (refer back to figure 1). For lists the storage format is straightforward – items are stored one after another in subscript sequence. There is always a zero subscript element, so it is a little

wasteful to start at 1. The amount of storage for each array element is decided by the data type of the array – **integers** get two bytes, **single precision reals** four bytes and **double precision reals** eight bytes. This is the same as for single values. **Strings** take up one byte per character.

To conclude this description of types and structures of data, program 1 performs an exploratory operation on its own innards, by displaying how its *own* data and instructions are

stored in memory. This confirms the material already discussed regarding data in memory, and also provide an insight into how program instructions are stored.

Several different data types are included at the start of the program, just for the purposes of demonstration. The second half of the program (**line 1000 onwards**) examines memory and displays it in blocks on the screen. The program, as it appears, causes hexadecimal and string versions of

memory contents to be sent to the printer: if printer output isn't required, simply change the **LPRINTs** to **PRINT** in lines 1020, 1050 and 1090. To display the program and simple variable areas, enter a start address of 32768. The results obtained are listed below.

Bytes **HB000** to **H823C** contain the program instructions, while simple variables are stored after this point, in the order in which they appear in the program. **Arrays** appear immediately after the simple **variables**.

First let's analyse how the

instructions are stored. Every instruction line starts with a **byte** containing **zero**. The next **two bytes** then contain a **pointer** to where in memory the following program line starts. Thus, bytes **H8001** and **H8002** point to **H8021**, where **line 110** starts; bytes **H8021** and **H8022** point to **H8045** where **line 120** starts; bytes **H8046** and **H8047** point to **H806A** where **line 130** starts, and so on. The last line, numbered **1110**, starts at **H8232**, and here the pointer refers to **H823A** which contains a program

Storage

```

100 AB!=34.56:AB£=64.789:AB%=1234
110 C(1)=123:C(2)=124:C(3)=125:C(4)=126
120 CD!=-9.87E-03:CD£=6.54E+30:CD%=-14
130 D$="ABCDEFGHIJKLMNPOQRSTUVWXYZ"
140 E$(1)="AAAAA":E$(2)="BBB":E$(3)="CCCCC"
150 DIM F(2,4)
160 F(1,1)=500:F(1,2)=65.8:F(1,3)=99.99:F(1,4)=5E-03
170 F(2,1)=666.6:F(2,2)=8.5:F(2,3)=111:F(2,4)=12345.678£
1000 INPUT "Start address=";S
1010 FOR A=S TO 65535: STEP 8
1020 LPRINT HEX$(A)";";S$=""
1030 FOR B=A TO A+7
1040 P%=PEEK(B)
1050 LPRINT RIGHT$("0"+HEX$(P%),2);" ";
1060 IF P%>31 AND P%<128 THEN C$=CHR$(P%) ELSE C$=""
1070 S$=S$+C$
1080 NEXT B
1090 LPRINT S$
1100 IF A=80*INT(A/80) THEN INPUT "More (y/n)";X$:IF X$="n" THEN STOP
1110 NEXT A

```

The data-names and their values are stored immediately after the program. All the single values come first, in the variable area, followed by the arrays, in the array variable area. The first item, AB%, occupies bytes H823D to H8243 inclusive, in the following way:

terminator – two zero bytes. After this next-line pointer at the start of each line comes the line number, stored in two bytes. Thus 100 is stored in bytes H8003 and H8004.

H823D: 04 length of value=4
 41 42 name AB
 42 34 56 00 value 34.56

Example 1 – line 100:

H8000: 00 Beginning of line
 21 80 Pointer to start of next line (H8021)
 64 00 Line number 100
 41 42 21 AB!
 EF =
 1D marker
 42 34 56 00 34.56 in single precision
 3A :
 41 42 23 AB#
 EF =
 1D marker
 42 64 78 90 64.789 in single precision
 3A :
 41 42 25 AB%
 EF =
 1C marker
 D2 04 1234 in integer form

In fact each of the variables is stored in this way, with a length byte first, name next (without suffix – it is unnecessary), and lastly the current value. Two more examples, CD# and CD%, are given below.

H825B: 08 length 8
 43 44 CD
 5F 65 40 00 00 00 00 00 double precision 6.54E+30

H8266: 02 length 2
 43 44 CD
 F2 FF -14 (two's complement form)

Example 2 – line 150

H80BA: 00 Beginning of new line
 C8 80 Pointer to start of next line (H80C8)
 96 00 Line number 150
 86 DIM
 20 space
 46 28 F(
 13 first dimension=2
 2C ,
 15 second dimension=4
 29)

String data-names also appear in this area, but their contents are not included. Instead for each there is a length byte and a pointer to where the string is stored. For example, string D\$:

H826B: 03 length 3
 44 00 D
 1A length of string=26
 72 80 pointer to H8972 where string is found

The instructions themselves are stored as mixtures of ASCII codes and keyword codes. Every keyword and symbol in MSX BASIC is assigned its own code, to economise on storage space. Let us look at some instructions in detail:

The numeric constants in line 100 conform to the standards for integer and single precision real that we have already discussed. Other examples are easily found by comparing the memory dump with the program listing.

Not all string pointers refer back to the program area – if the string contents have been changed substantially its contents will appear in the 'string area' (see memory map – Figure 1). This is the case for S\$, as readers may determine for themselves.

Next we consider the array variable area. This commences with array C at byte **H82A3**. Arrays start off looking like single variables, with a length byte and a name in a pair of bytes. After this however comes an array header. This header has data on the total number of bytes in the array, the number of dimensions, and the size of each dimension. The actual array elements follow the header. If the array is two-dimensional, columns are stored in sequence, namely column zero, then column one, then column two and so on. We consider each of the three arrays used in the program:

Array F

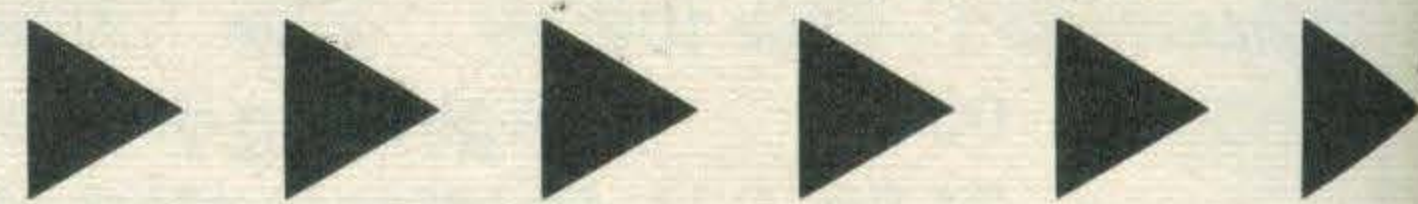
H832C: 08	length byte=8
46 00	name F
7D 00 02 05 00 03 00	HEADER: 125 bytes,
	2 dimensions
	of size 5 and 3
00 00 00 00 00 00 00 00	F(0, 0) = 0
00 00 00 00 00 00 00 00	F(1, 0) = 0
00 00 00 00 00 00 00 00	F(2, 0) = 0
00 00 00 00 00 00 00 00	F(0, 1) = 0
43 50 00 00 00 00 00 00	F(1, 1) = 500
43 66 66 00 00 00 00 00	F(2, 1) = 666.6
00 00 00 00 00 00 00 00	F(0, 2) = 0
42 65 80 00 00 00 00 00	F(1, 2) = 65.8
41 85 00 00 00 00 00 00	F(2, 2) = 8.5
00 00 00 00 00 00 00 00	F(0, 3) = 0
42 99 99 00 00 00 00 00	F(1, 3) = 99.99
43 11 10 00 00 00 00 00	F(2, 3) = 111
00 00 00 00 00 00 00 00	F(0, 4) = 0
3E50 00 00 00 00 00 00	F(1, 4) = 0.005
45 12 34 56 78 00 00 00	F(2, 4) = 12345.678

Array C

H82A3: 08	length byte=8
43 00	name C
5B 00 01 0B 00	HEADER: 91 bytes,
	1 dimension
	of size 11
00 00 00 00 00 00 00 00	C(0) = 0
43 12 30 00 00 00 00 00	C(1) = 123
43 12 40 00 00 00 00 00	C(2) = 124
43 12 50 00 00 00 00 00	C(3) = 125
43 12 60 00 00 00 00 00	C(4) = 126
00 00 00 00 00 00 00 00	C(5) = 0
00 00 00 00 00 00 00 00	C(6) = 0
00 00 00 00 00 00 00 00	C(7) = 0
00 00 00 00 00 00 00 00	C(8) = 0
00 00 00 00 00 00 00 00	C(9) = 0
00 00 00 00 00 00 00 00	C(10) = 0

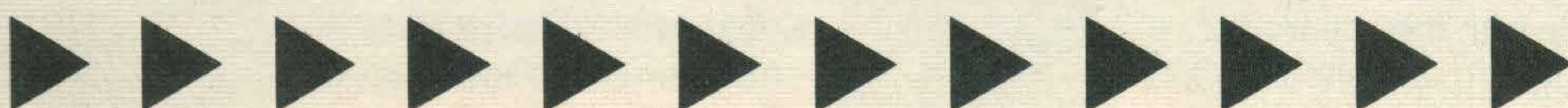
Array E\$

H8303: 03	length byte=3
45 00	name E
24 00 01 0B 00	HEADER: 36 bytes,
	1 dimension
	of length 11
00 00 00	E\$(1)=null
05 99 80	E\$(2)=5 bytes stored at H8099
03 A7 80	E\$(3)=3 bytes stored at H80A7
06 B3 80	E\$(4)=6 bytes stored at H80B3
00 00 00	E\$(5)=null
00 00 00	E\$(6)=null
00 00 00	E\$(7)=null
00 00 00	E\$(8)=null
00 00 00	E\$(9)=null
00 00 00	E\$(10)=null



There are several noteworthy points that arise from this analysis. First we see that array headers are not fixed in size, but expand according to the number of dimensions. We also note that string arrays are stored in a similar way to single strings – the strings themselves are not stored explicitly, but instead there is a length byte followed by a pair of bytes pointing to another location in memory where the string is to be found. The amount of wasted space is made obvious too – both from not using zero subscripts, and also from not declaring small arrays, since they default to eleven elements.

The array area ends at **H83AE**. In the dump listed here there are just blocks of 00s and FFs after it for some time. However, if you had been using the MSX machine before you typed in the examining routine, you may well find remnants of your previous program and its data in this area, even though you typed NEW. This is because NEW does not wipe memory clean, it simply adjusts the line pointers. Hence, with a little care, it is quite possible to recover a program that was lost by the NEW command.



ITING WE'RE SOLICITING WE'RE SOLICITING WE'RE SOLICITING WE'RE
 'RE SOLICITING **WE'RE SOLICITING** WE'RE SOLICITING WE'RE SOLICITING
 NG WE'RE SOLICITING WE'RE SOLICITING WE'RE SOLICITING WE'RE SOLICITING

Write MSX programs for us to print in our THERAPY feature and you'll not only become incredibly famous, popular and big-headed but we'll pay you as well! Create your therapeutic masterpieces, send them to us and we'll give them the once-over. If they're good, we'll list them in the mag. If not, we can give you advice on how to make them better.

Supply - Cassette of prog

List of variables used

Brief explanation of program flow

Printer listing of prog (if possible)

Explanation of how prog works/instructions

Check out the THERAPY feature in this issue for guidance.

Send to: MSX User

Argus Specialist Publications

No 1 Golden Square

London W1R 3AB

E'RE SOLICITING WE'RE SOLICITING WE'RE SOLICITING WE'RE SOLICITING
 TING WE'RE SOLICITING WE'RE SOLICITING WE'RE SOLICITING WE'RE SOLICITING

THERE'S ALWAYS SOMETHING EXCITING IN THE ALLIGATAS JUNGLE ...LIKE THESE AMAZING TITLES FOR



MSX **SON OF BLUGGER**

Platform programming at its best. Follow the exploits of Roger the Dodger, master burglar on seemingly endless looting of banks, shops and houses - robbing safe after safe, avoiding obstructions and alarm systems and dodging the nasty nightwatchmen. 30 screens.

TAPE
£7.95



BRIDGE

The ideal opportunity to improve your technique in a sophisticated program where the computer bids and plays the other three hands. Full rubber scoring plus advanced play routine.

TAPE
£9.95

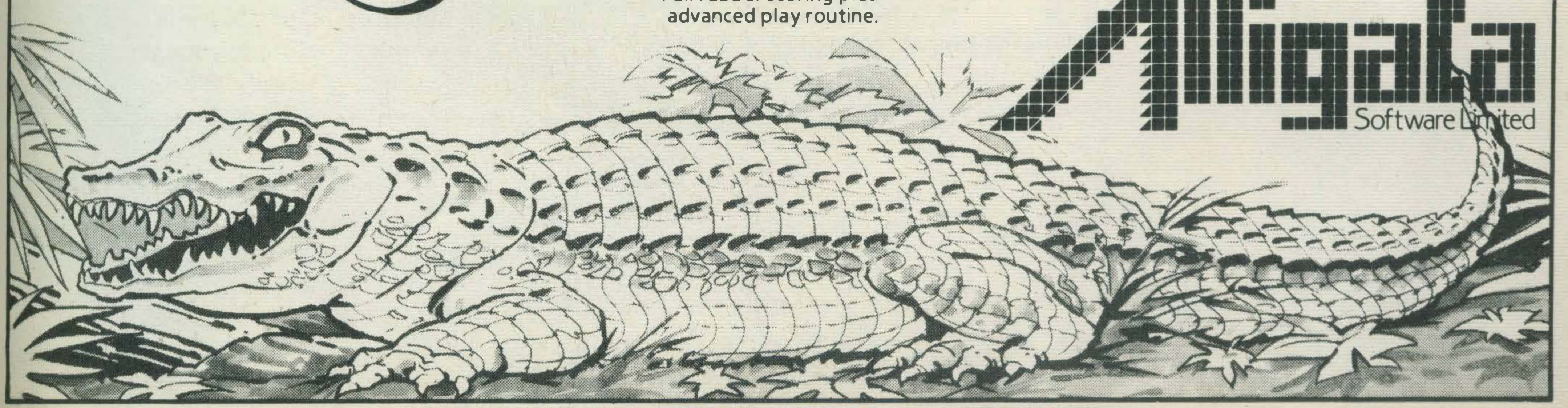
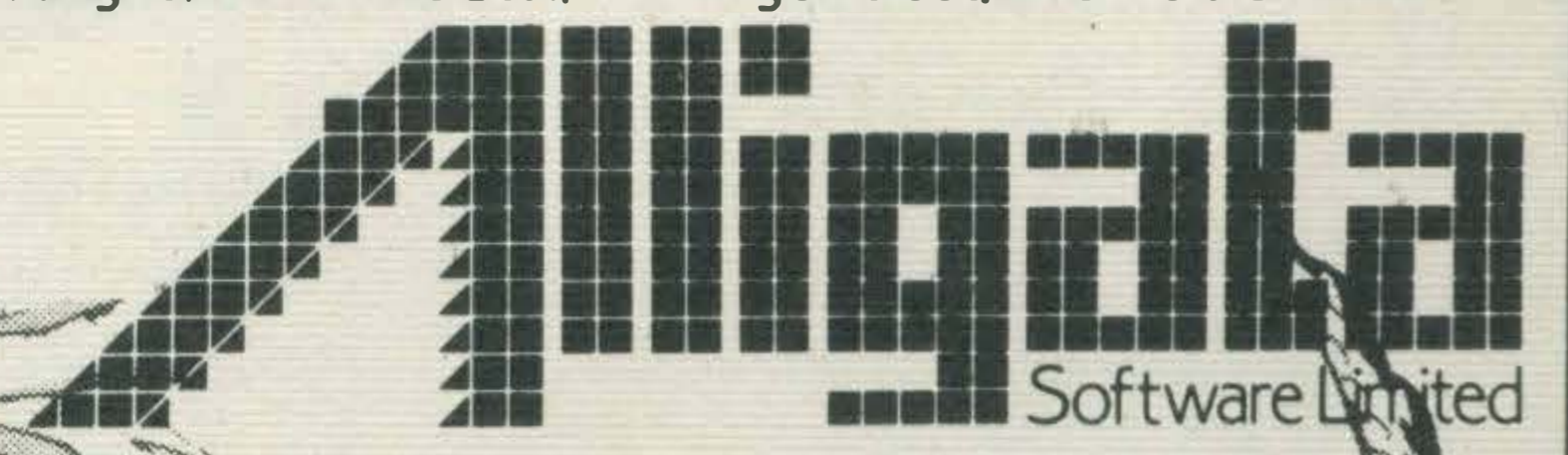


Disc WARRIOR

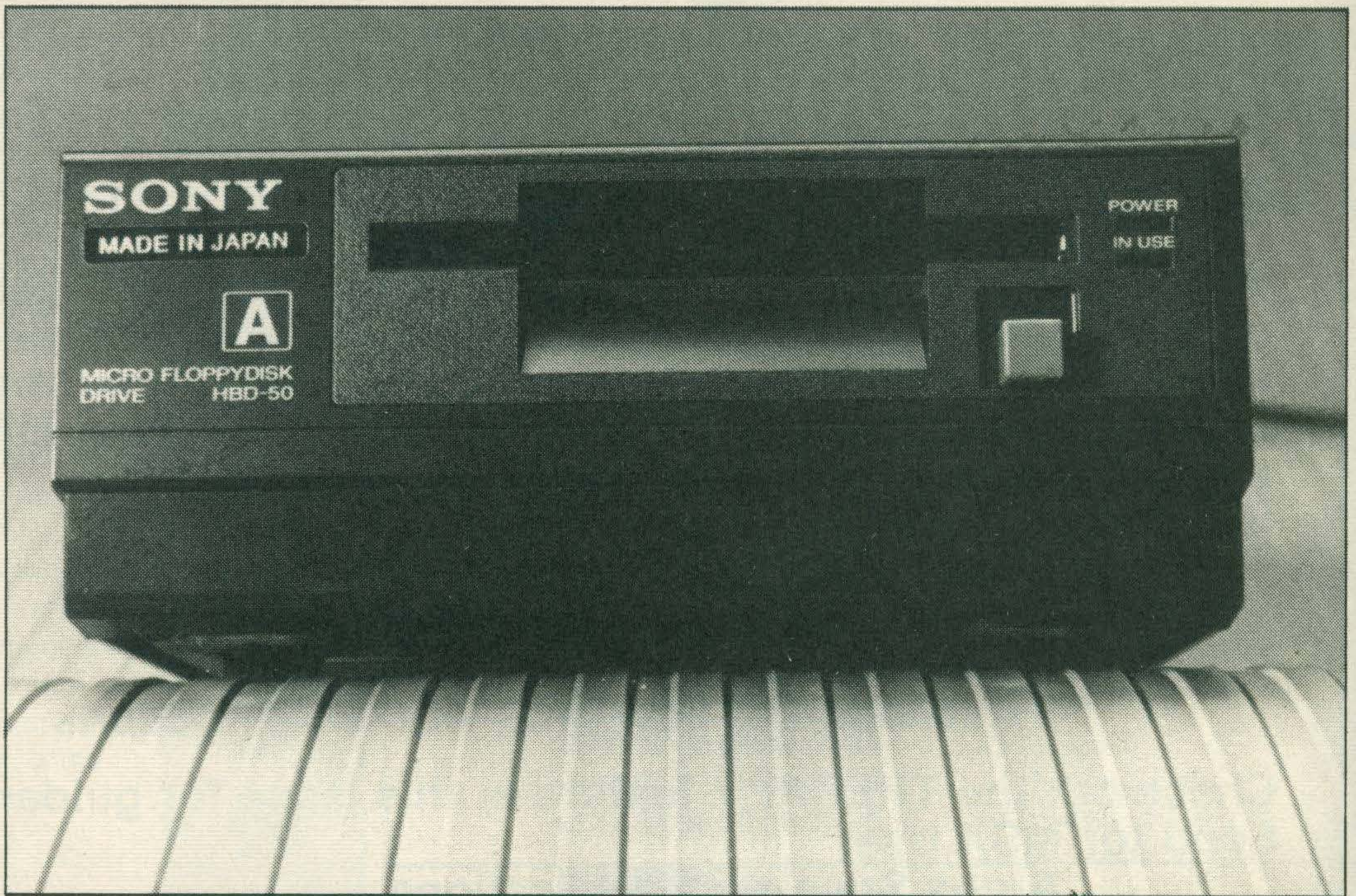
Travel the 3D computer complex alone armed only with your electrified "bouncing" discs. Blast holes through the electrified fences, defend yourself from the alien guards and robot canines as you search for the master key pieces and superbomb to blow the Master CPU to destruction. Keep an eye on your body clock when time runs out you'll roam on more.

TAPE
£7.95

Send for full colour product leaflet (enclose a stamp)
 Alligata Software Ltd., 1 Orange Street, Sheffield S1 4DW



A Spin With Sony



Goodbye Mickey Mouse – Hello Karl Dallas

Anyone who thinks that it's possible to manage a computer without a disk drive is really living in the steam age. Without a disk drive, an MSX computer still seems something of a Mickey Mouse affair – fine for playing games on, but strictly limited if you have wider ambitions.

Quite apart from the time taken to LOAD and SAVE programs, random access files are virtually impossible

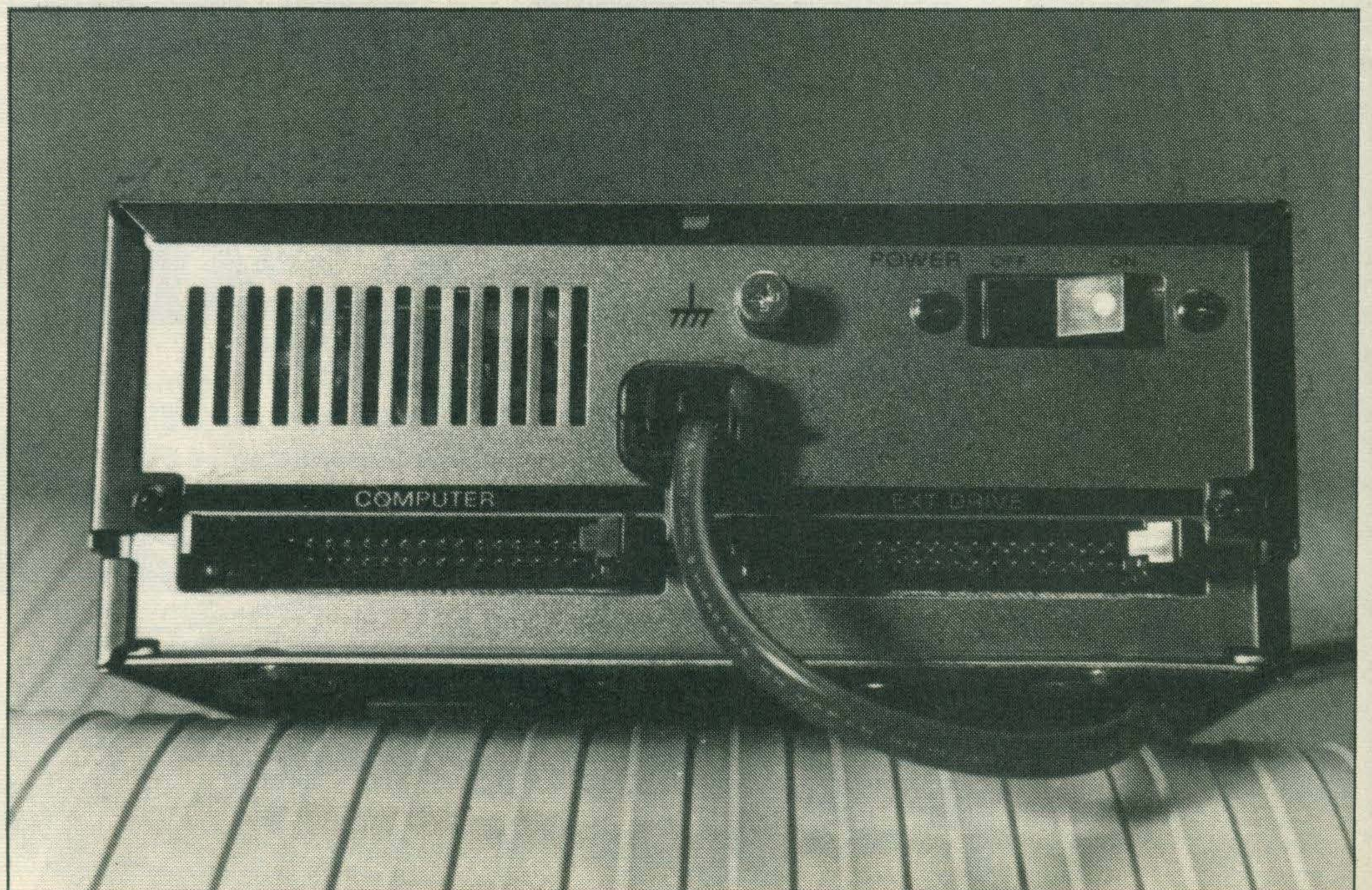
with tape, and even a games player will start demanding disk-based software, once the advantage of finishing a LOAD in seconds rather than minutes has sunk home.

Users of many other computers had to wait months before disks became available, but thanks to the genius of the people at Sony, MSX disk drives came on the scene from day one, and while other drives are said to

be in the pipeline, they'll have to be really something to challenge the supremacy of Sony's neat-looking, super-efficient peripheral.

Also, getting it up and running is simplicity itself: plug the interface into the cartridge socket, plug into the power supply, switch it on, and you're ready to go.

As soon as you've done this, a whole extra vocabulary of MSX Disk BASIC commands is available, plus 320K of virtually instant access for program and data files.



The new Disk BASIC commands and functions include:

COPY, CVI, CVS, CVD

DSKF

EOF

FIELD, FILES, FORMAT

GET

KILL

LOC, LOF, LSET

MKI\$, MKS\$, MKD\$

NAME

PUT

RSET

SYSTEM

VARPTR

Some of the Disk BASIC commands will be familiar to the user from cassette-based BASIC. SAVE and LOAD, BSAVE and BLOAD, INPUT # and PRINT # work much the same way as with cassette files, except that instead of the prefix "CAS:" to designate tape, you specify disk by the prefix "A:", if you only have one drive, or "A:" or

"B:" if there are two (it is possible to "daisychain" two drives together).

If you're still using tape, for instance to copy your previous programs from cassette to disk, then you'll need to distinguish the tape files with the "CAS:" prefix.

COPYing can be done with the COPY command, thus:
COPY "CAS: PROGNAME"

TO "A PROGNAME"

More than one file can be copied at a time, using the wild-card symbols "?" and "*".

Thus,

COPY "CAS:*" TO A:

will copy all the programs on a tape to the disk. (This may not work with commercial programs, which are probably protected. Companies may be willing to advise you on how to do such legitimate copying – of course, you should never copy a commercial program for anyone else!)

The disk itself is a sturdy little item, a square blue plastic casing surrounding the actual recording disk, which is protected by a sliding

cover. Accidental erasure or over-writing of precious programs or data can be avoided by sliding a small red "write-protect" tab a few millimetres. If you decide you want to reuse the disk and get rid of the old stuff, sliding the protect tab back is the work of a second.

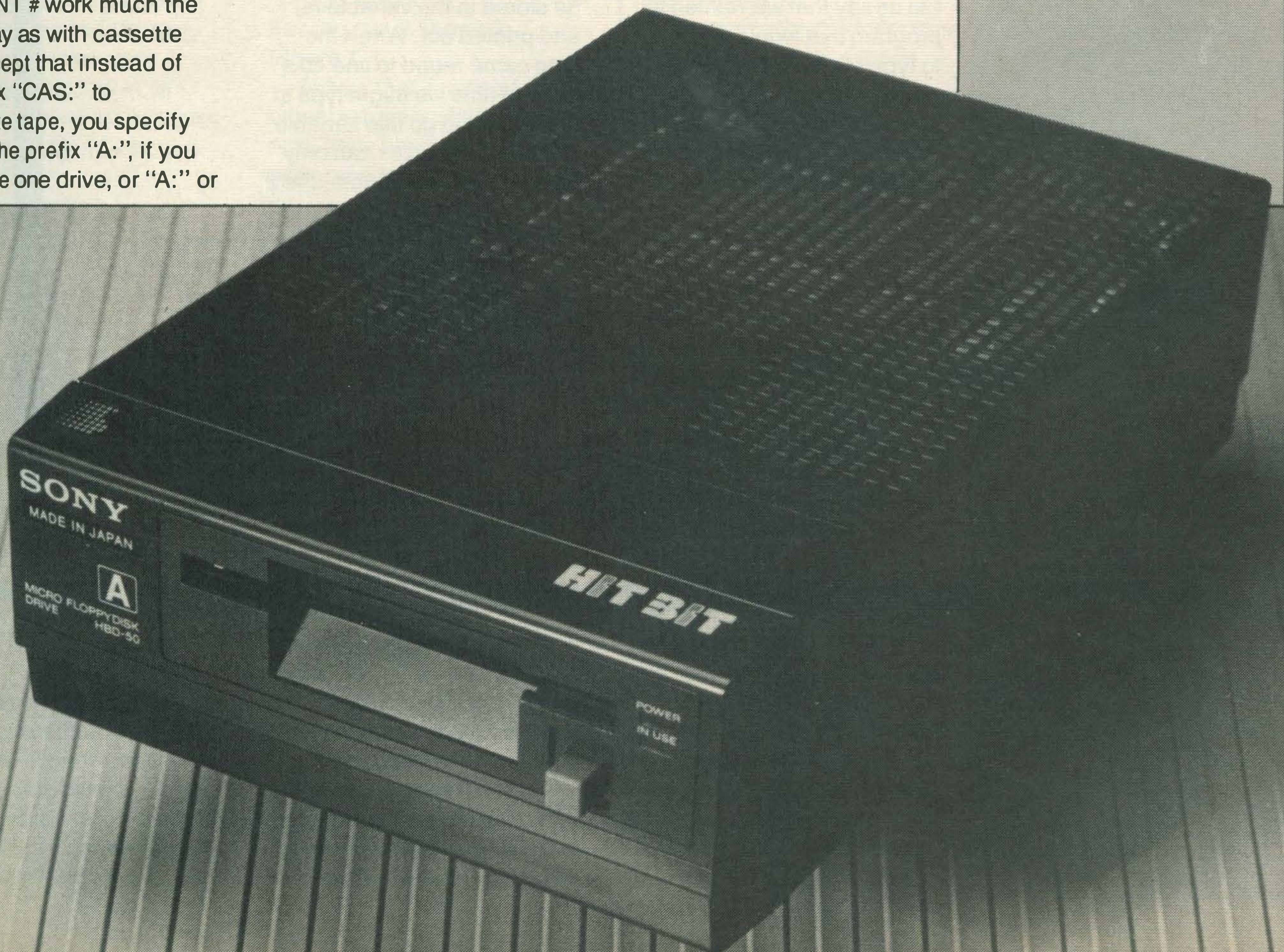
Like all MSX peripherals, of course, the Sony disk drive can be used with any other MSX computer, and is not limited to Sony's.

What's the catch?

You lose a small amount of computer memory, which goes down about 5½K, from 28815 to 23430 bytes.

It's not cheap, and the disks themselves are a bit expensive, too, though you'll probably find you can manage quite happily on a couple for quite a long time, unless you're doing a lot of word processing or something equally professional, in which case you can always set it against tax, can't you?

The Sony HBD50 3½in disk drive has a recommended retail price of £349.95.



Storing stuff is easy – Steve Colwill shows you how.

Your MSX computer is capable of much more than being just a games machine – it can help you to organise household accounts, draw up alphabetic lists of names and a number of other tasks that are boring and repetitive to do by hand. When jobs like those I've just mentioned are done by computer they are normally called 'data processing' applications. Data processing jobs have one particular thing in common: they all accept data, be it numbers or words, and reorganise it. This reorganisation may include making actual changes to the data that was input, or simply swapping the order of the data items. Later in this month's article we shall be looking at ways to handle letters and words as data, but let us start on slightly more familiar ground with number processing.

Let's start by thinking about a trivial job to be done. Let us say that we wanted a program that would allow us to type in a series of numbers and print them out on the screen. Our first thought might be to come up with a program like this one:

```
10 REM **** PRINT OUT 1 ****
20 INPUT "NUMBER OF ITEMS TO PRINT";N
30 FOR I=1 TO N
40 INPUT A
50 PRINT A
60 NEXT I
```

There is nothing in this program that we haven't met before. INPUT allows us to type something in at the keyboard (and press Return). The program asks the person using it to enter how many numbers they want to type in, and puts the user's answer in the variable called N. The value typed in sets the number of times that the FOR . . . NEXT loop between lines 30 and 60 will repeat. The loop goes on asking for numbers (if no message accompanies the INPUT command then only a question mark will appear on the screen) and immediately prints them out to the screen.

A variable called A is used temporarily to hold the number typed in by the user. As the same variable is used each time through the loop then only one number is ever held in the computer at any one time. If the first number we typed in when we ran the program was 2, then 2 would be stored in the variable A, and printed out. When the loop came round to line 40 a second time we might type in 6. This would go into variable A and, as variables can only hold one value at a time, the 2

that was previously stored there would be lost; or in the jargon, 'overwritten'.

This is fine if all we want to do is to print out the numbers in the order in which they were input, but what if we

wanted to reorganise the numbers in some way, say reverse the order in which they are printed out? Well, the obvious answer is to use different variable names so that we can hold all the

numbers that are typed in. This program will work but is rather limited.

```
10 REM **** REVERSE ORDER ****
20 INPUT A
30 INPUT B
40 INPUT C
50 PRINT C
60 PRINT B
70 PRINT A
```

Before we can use arrays in our programs we have to tell the computer how many members we want the array to have; this is called 'dimensioning' the array. The statement:

```
DIM A(50)
```

means set up a list with 50 members, calling them A(1), A(2), A(3) . . . A(49) and A(50), and set the value of each list

The obvious limitation of this program is that only three numbers can be typed in and reversed. If more numbers are required then additions have to be made to the program. What we really want is a different way of naming variables, so that we can combine the first program's ability to work for different amounts of numbers with the second's ability to reorganise the numbers. Fortunately, there is a special way of naming variables that allows us to do just that (you knew there would be!) These new variables are called 'arrays'. That sounds like a very mysterious title, but in its simplest form an array is just a list that we can store numbers in.

Hip Hip Array

Arrays work like this: instead of using different variable names for each number we want to keep, we use the same variable name but we follow it with a number in brackets afterwards. The array A() is made up of a number of different variables, A(1), A(2), A(3) . . . and so on. We can think of these as being a series of little boxes into which we can put numbers (just like other variables), so this command:

```
LET A(5) = 27
```

has the result shown in **Figure 1**.

Notice that if none of the other members of the list, say A(2), have been set (using an assignment like the one used

member to be zero. I think you'll agree it's easier to say DIM A(50)! Note that A(0) is also created by the DIM statement but is often ignored.

```
10 REM **** PRINT OUT 2 ****
20 INPUT "NUMBER OF ITEMS TO PRINT"; N
25 DIM A(N)
30 FOR I=1 TO N
40 INPUT A(I)
50 NEXT I
60 :
70 FOR I=1 TO N
80 PRINT A(I)
90 NEXT I
```

Let us go back to our first problem and see how we could do it using an array. The result is shown in **Program 2**. **Line 20** asks the user to say

Fig. 1

A(0)	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	etc
0	0	0	0	0	27	0	0	0	⋮

above) then they will each have the value zero placed in them. The main advantage of using arrays is that we can use a variable to tell the computer which member of the list we mean. This pair of commands has exactly the same effect as the one above:

```
LET X = 5
```

```
LET A(X) = 27
```

except that the member of the list that 27 is to be put into is selected by the variable X.

how many numbers are going to be entered. **Line 25** tells the computer to set up a list with 'N' members ready to take the numbers. **Lines 30-50** use a FOR . . . NEXT loop to input the numbers into the list as they are typed in; the first number going in A(1), the second in A(2) and so on. Notice how the loop counter I is used to specify which member of the list the number input is to be stored in. **Lines**

70-90 use another FOR . . . NEXT loop, this time to print out the values. Notice that this task can now be left until all the numbers have been typed in, as each number is kept separately in the list.

Now that we have seen how a series of numbers can be easily stored we can manipulate them. To print them in reverse order is easy, simply change line 70 of the above program to:

```
70 FOR I=N TO 1 STEP -1
```

Changing line 80 as follows makes the computer only print out the even numbers. If I tell you that the INT statement takes the whole number part only (so INT(3.3) or INT(3.9) both equal 3), you should be able to figure out how the computer does it!

```
80 IF A(I)/2=INT(A(I)/2) THEN PRINT A(I)
```

Fig.2

A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)
2	5	1	4	3	5	7	3

Fig.3

A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)
2	1	4	3	5	5	3	7

Swapping and Sorting

Your MSX computer can easily be made to sort things into order. Later we'll look at alphabetic sorting of words, but let's start with numbers. Before we can understand how a sorting program works, we must first understand how a computer can swap over two variables. Let us say we had two numbers, one stored in variable A and the other stored in variable B and we wanted to swap them so that the number in A ends up in B and vice-versa. No problem, you think to yourselves. Just do this:

```
LET A = B
LET B = A
```

No problem. But think again, Whittington. Let's say that A held 2, and B held 3. Let A=B means 'copy the number from B and put it in A'. After this command A would hold 3, and

so would B. Let B=A means 'copy the number from A and put in B'. A now holds 3, so it would copy this across to B. At the end of this second assignment A and B would both hold 3; the 2 originally held in A has been overwritten, rather than being swapped into B. The answer is to use another temporary variable to hold the 2, so that it is not overwritten by the 3 coming into A from B. If we call this temporary variable T, then the following sequence will swap the contents of A and B:

```
LET T = A: REM COPY A INTO TEMP STORE
LET A = B: REM COPY B INTO A
LET B = T: REM COPY ORIG CONTS OF A FROM T TO B
```

The reason we have spent time thinking this through is that swapping variable values forms a vital part of any sorting routine. Let's start by

MSXercise

assuming that the numbers we want to sort are held in an array A(). An example is shown in **Figure 2**. If we want to sort into ascending order (the smaller numbers earlier in the array) the program we write must carry out the following steps:

1. Compare each pair of neighbour members in the array in turn.
2. If they are out of order swap them, otherwise leave them as they are.
3. Repeat 1 and 2 until no more swaps can be made.

In our example, the program would start by looking at A(1) and A(2). These two values are already in the correct order so no swapping is required. The program would then move on to compare A(2) and A(3). These are definitely in the wrong order so A(2) and A(3) must be swapped. The process repeats, scanning the whole array until each pair has been considered. At the end of the first scan the array will look like **Figure 3**.

Obviously, the array is not yet fully sorted but some movement has occurred: most noticeably, the 5 that started off in A(2) has been relegated through the array to A(5). We need to scan the entire array again and again, making swaps, until all the values are in their correct positions and no more swaps can be made. If the whole array is scanned and no swaps can be made then this indicates that the array is fully sorted and the program can move onto the next task. Now all this activity seems fairly complicated, but see how easily it can be done in MSX BASIC, consider **Program 3**.

Lines 20-60 are concerned with entering the numbers into the array as before. Lines 90-130 sort the array. A FOR . . . NEXT loop scans through the array comparing each member of the array with its neighbour.

```
10 REM **** NUMBER SORT PROGRAM ****
20 INPUT "NUMBER OF ITEMS TO SORT":N
30 DIM A(N)
40 FOR I=1 TO N
50 INPUT A(I)
60 NEXT I
70 :
80 REM ** SORT ROUTINE **
90 F=0
100 FOR P=1 TO N-1
110 IF A(P)>A(P+1) THEN GOSUB 1000:REM SWAP
120 NEXT P
130 IF F=1 THEN 90:REM REPEAT
140 :
150 REM ** PRINT OUT IN ORDER **
160 FOR I=1 TO N
170 PRINT A(I)
180 NEXT I
190 END
200 :
1000 REM **** SWAP SUBROUTINE ****
1010 T=A(P)
1020 A(P)=A(P+1)
1030 A(P+1)=T
1040 F=1
1050 RETURN

READY.
```

Line 110 tests to see if the member is larger than its neighbour: if it is then the program is told to GO to a SUBroutine, starting at **line 1000**. This piece of program swaps over the two neighbours being tested and sets the variable F to 1, before RETURNing to continue from where it left off on **line 110**. **Line 130** tests the value of F, after one scan of the array is complete. If F has been set to

one, then this means that a swap took place during the last scan of the array. This in turn means that the array must be scanned again, as it is not yet fully sorted. Notice that F is reset to 0 each time at **line 90** before a new scan starts.

This program will sort many numbers very quickly. Try timing yourself to do the same job manually, and see which is faster, you or your MSX.

Strings

Most of our work in MSX BASIC has been concerned with numbers and number variables, but a different type of variable exists in MSX BASIC to help us handle letters and words in a similar way to numbers. These variable names always end in a dollar sign (\$), for example A\$, pronounced 'A-string'. We met this kind of variable very briefly in last month's issue, when we wanted to bring in keyboard control to a games program, but string variables are capable of much more than storing a single letter pressed on the keyboard. They can be used to hold words or sentences, and these can be chopped and changed around using special string-handling commands. The word 'string' is short for 'string of characters' or 'character string'.

We can use assignment statements in the same way as for numbers:

```
LET A$ = "PETER"  
LET B$ = "PAUL"
```

Strings can be compared using IF . . . THEN statements. For example:

```
IF A$(B$ THEN PRINT A$
```

Here 'less than' really means earlier in the alphabetic order. Equal strings are identical to each other. Strings can be linked together using the '+' sign. For example:

```
LET A$ = "JOHNNY"  
LET B$ = "AND"  
LET C$ = "MARY"  
LET D$ = A$ + B$ + C$
```

In this example, D\$ would be assigned the string of characters 'JOHNNY AND MARY'. Notice that the spaces between the words are only there because B\$ was defined as 'SPACE-A-N-D-SPACE'. If, instead, we LET B\$ = "NOT", then D\$ would be 'JOHNNYNOTMARY'.

As I've just mentioned, there are special commands in MSX BASIC that allow us to chop strings up. The two simplest are LEFT\$ and RIGHT\$. If B\$ is, say, 'HAROLD', then the assignments:

```
LET F$ = LEFT$(B$, 2)  
LET G$ = RIGHT$(B$, 3)
```

would make F\$ = 'HA' and G\$ = 'OLD'. We can think of these two commands as telling the computer to start at the left (or right) of the string specified before the comma in the bracket, and take the number of characters specified by the number after the comma.

As well as being able to isolate groups of characters (letters) from either end of a string we can also isolate a group of characters from the middle of a string using the command MID\$.

```
LET B$ = MID$(A$, 5, 4)
```

means start at the fifth character of A\$ and take 4 characters. If A\$ = 'EXISTENTIAL' then B\$ would be 'TENT'.

In addition it is often useful to know how long a string is. This can be found using the LEN command. If A\$ = "HELLO" then:

```
LET L = LEN(A$)
```

assigns L the value 5, the number of characters in A\$.

One final command that is useful is the INSTR command which searches for the occurrence of one string within another.

```
LET A$ = "AWARD FOR BRAVERY"  
LET B$ = "RAVER"  
LET L = INSTR(A$, B$)
```

After the previous instructions have been carried out, L will have the value 12, signifying that a match was found for B\$ starting at the twelfth character in A\$.

Sorting Strings

Following this whistle-stop tour of string-handling facilities let's look at a program that uses them: **Program 4**. Much of this program should be fairly familiar to you. It's a sorting routine for words instead of numbers, rearranging a given group of words into alphabetical order. Notice that we can use arrays for string variables in exactly the same way as for number variables. The main difference between this program and the 'NUMBER SORT' program given earlier is the routine between **lines 500 and 570**. The program asks the user to type in the names to be sorted, typing in the first name than a space, then the second name. Why is this? Well, most alphabetical lists of names, such as the telephone directory, create the alphabetical order by

considering the second name first (eg all the BROWNS are grouped together) and then considering the first name afterwards. If the names were sorted with first and second names in the usual order then they would be sorted on the

first name rather than the second name. The section of the program starting at **line 500** takes each name in the array and turns it around so that the second name comes first, followed by a space, followed by the first name. When the array members are later compared, the second name will be considered first.

The routine to turn the first and second names around works as follows. **Line 520** searches for the occurrence of a space and assigns PS with the position of the space within the array member being turned. **Line 530** uses the value of PS to isolate the first name and put it in CN\$. **Line 540** uses the value of PS and LS, the length of the name, to isolate the second name and put it in SN\$. **Line 550** reassembles the array member being turned, putting the second name first.

The program allows the user to type in a series of names. They are sorted and printed on the screen in alphabetic order. No attempt has been made to turn the first and second names back round after sorting and so the names will appear as second name followed by first name.

MSXercise

```
10 REM **** NAME SORT PROGRAM ****
20 INPUT "NUMBER OF NAMES TO SORT";N
30 DIM A$(N)
35 PRINT "ENTER NAMES IN THIS WAY"
37 PRINT "FIRST NAME (SPACE) LAST NAME"
38 PRINT "FOR EXAMPLE: HENRY GIBSON"
40 FOR I=1 TO N
50 INPUT A$(I)
60 NEXT I
65 :
67 GOSUB 500:REM REVERSE NAMES
70 :
80 REM ** SORT ROUTINE **
90 F=0
100 FOR P=1 TO N-1
110 IF A$(P)>A$(P+1) THEN GOSUB 1000:REM SWAP
120 NEXT P
130 IF F=1 THEN 90:REM REPEAT
140 :
150 REM ** PRINT OUT IN ORDER **
160 FOR I=1 TO N
170 PRINT A$(I)
180 NEXT I
190 END
200 :
500 REM ** TURN NAME AROUND SUBROUTINE **
505 FOR I=1 TO N
510 LS=LEN(A$(I))
520 PS=INSTR(A$(I)," ")
530 CN$=LEFT$(A$(I),S-1)
540 SN$=RIGHT$(A$(I),LS-PS)
550 A$(I)=SN$+" "+CN$
560 NEXT I
570 RETURN
580 :
1000 REM **** SWAP SUBROUTINE ****
1010 T$=A$(P)
1020 A$(P)=A$(P+1)
1030 A$(P+1)=T$
1040 F=1
1050 RETURN

READY.
```

Making Converts

Whenever a new micro is released, it takes at least a few months before good software for it hits the shops. The MSX system is no exception and if you have just got your hands on a machine for the first time, there is nothing more frustrating than having no programs to run on it. After you have busily typed in all the programs in **MSX User** and spent your precious pocket money on the few programs available now, what next?

Looking through any general computer magazine, you will find dozens of programs for Spectrums, BBCs, Commodores etc, and in this series I shall try to explain how to convert these listings so that they will run on an MSX machine. Of course these programs will never be as good as one written specially for MSX, but beggars can't always be choosers. Before you get too excited, don't expect that in a few weeks you will be able to convert such masterpieces as the *Hobbit*, *Elite* and *Ghostbusters* to run on MSX: but you should be able to get many listings in magazines to work.

Before starting conversion, it is worth spending a little time examining the listing to decide if it is going to be easy, difficult, very difficult or impossible! As a general guide, I would suggest that unless you have 'A' levels in ancient Greek, I'd forget trying to convert a Commodore game because the Commodore 64 and VIC 20 use a version of BASIC which is a throwback to the dark ages of the 1970's where POKES ruled OK. Machines which use Microsoft BASIC, such as the Dragon or TRS-80 will, usually, be the easiest to convert programs from, but conversion of games and utilities from Spectrum or BBC BASIC shouldn't prove too difficult unless they are machine-specific programs.

The BASIC used in MSX is

If the sort of software you like is thin on the MSX ground, why not raid some of the micros? Here's part 1 in a guide to program conversion.

an extended version of Microsoft BASIC and so the commands in Section 1 below will, generally, require no changes to be made in order to run.

Section 1: The Easy Ones!

These commands can usually be typed directly into MSX machines without any changes.

RUN, CLEAR, DATA, DIM, DEFINT, DEFSNG, DEFDBL, DEFSTR, DEF FN, FOR . . . NEXT . . . STEP, GOSUB, GOTO, IF . . . THEN . . . ELSE, INPUT, LINE INPUT, LET, LPRINT, LPRINT USING, MID\$, ON ERROR GOTO, ON . . . GOTO, ON . . . GOSUB, PRINT, PRINT USING, READ, RESTORE, RESUME, RETURN, STOP, ABS(X), ASC(X), ATN(X), BIN\$(X), CDBL(X), CHR\$(X), CINT(X), COS(X), CSNG(X), ERR, ERL, EXP(X), INT(X), FRE(O), FRE(" "), INKEY\$ (not quite the same on BBC/Electron), INSTR, LEFT\$, LOG, MID\$, RIGHT\$, SGN(X), SIN(X), SPC(X), SQR(X), STR\$, STRING\$, TAB(X), TAN(X), VAL(X), CLS.

Not all computers use the functions in the list above and the Spectrum, in particular,

uses different string handling commands.

Spectrum string handling

A couple of examples will, hopefully, make things clear. On the Spectrum, L\$=S\$ (1 TO 3) means the same as L\$=LEFT\$(S\$, 3) on an MSX machine. Again, L\$=S\$ (3 TO 7) on the Spectrum means L\$=MID\$(S\$, 3, 5) on an MSX.

The final two numbers in MID\$ are equal to the starting position and the number of letters. If the letters are at the end of the word, this may be interpreted as L\$=RIGHT\$(S\$, 5)

Arrays on the Spectrum

Arrays are handled in exactly the same way on most machines. The only difference in Spectrum arrays occurs when DIMensioning string arrays. Unlike other machines, a second number is required in one dimensional arrays to tell the computer the maximum number of letters which each element can hold, eg:

DIM S\$(40, 25)
is not a two-dimensional array in Spectrum BASIC and

should just be typed into MSX as DIM S\$(40).

Screen Positions

LOCATE x, y in MSX is used to position the cursor at position x in row y ready for printing text. I have listed some equivalents below.

MSX version	= LOCATE 2, 3: PRINT "Hello"
BBC version	= PRINT TAB (2, 3); "Hello"
ORIC version	= PLOT 2, 3, "Hello"
Spectrum version	= PRINT AT 2, 3; "Hello"
TRS-80 version	= PRINT @129, "Hello"
Amstrad version	= LOCATE 2, 3: PRINT "Hello"

Random numbers

The RND function used in MSX BASIC is fairly standard. There are a few machines, notably the Beeb and Tandy machines, which have a slightly different random number generator. For instance, F=RND(100) would return a whole number between 1 and 100. To implement this on the MSX, you should replace it with F=INT(RND(1)*100+1). If you have tried writing a program to generate random numbers on an MSX machine, you will notice that they are always chosen in the same sequence. Some machines have a RANDOMIZE statement which will force the computer to start the random number generator at a random point. The simplest way to do this on an MSX machine is to seed the random number generator with a negative number such as T=RND(-TIME), although you could also try inserting a line such as

```
101 FOR X=1 TO TIME STEP 10: P=RND(1): NEXT
```

to make the computer generate a series of random numbers or by forcing the computer to generate numbers at random while waiting for a keyboard input:

```
100 PRINT "Press the (Space Bar) to continue"
110 A$=INKEY$
120 P=RND(1)
130 IF A$(" " THEN 110
```

Procedures

A quick glance at any listing for a Beeb or Electron will reveal the rather unusual words DEFPROC and

ENDPROC, which are not available in other versions of BASIC. At first sight, you might be forgiven for closing the magazine and giving up, but in fact these commands are very similar to GOSUB and RETURN, except you don't specify a line number but instead give a name to it. Converting these is, in fact, comparatively easy. For example, the following BBC program:

```
10 PROCdraw circle
"
"
"
"
"
1000 DEFPROCdrawcircle
1010 ,, ,, ,,
1020 ,, ,,
1030 ,, ,,
1040 ENDPROC
```

could equally well be replaced with the MSX program:

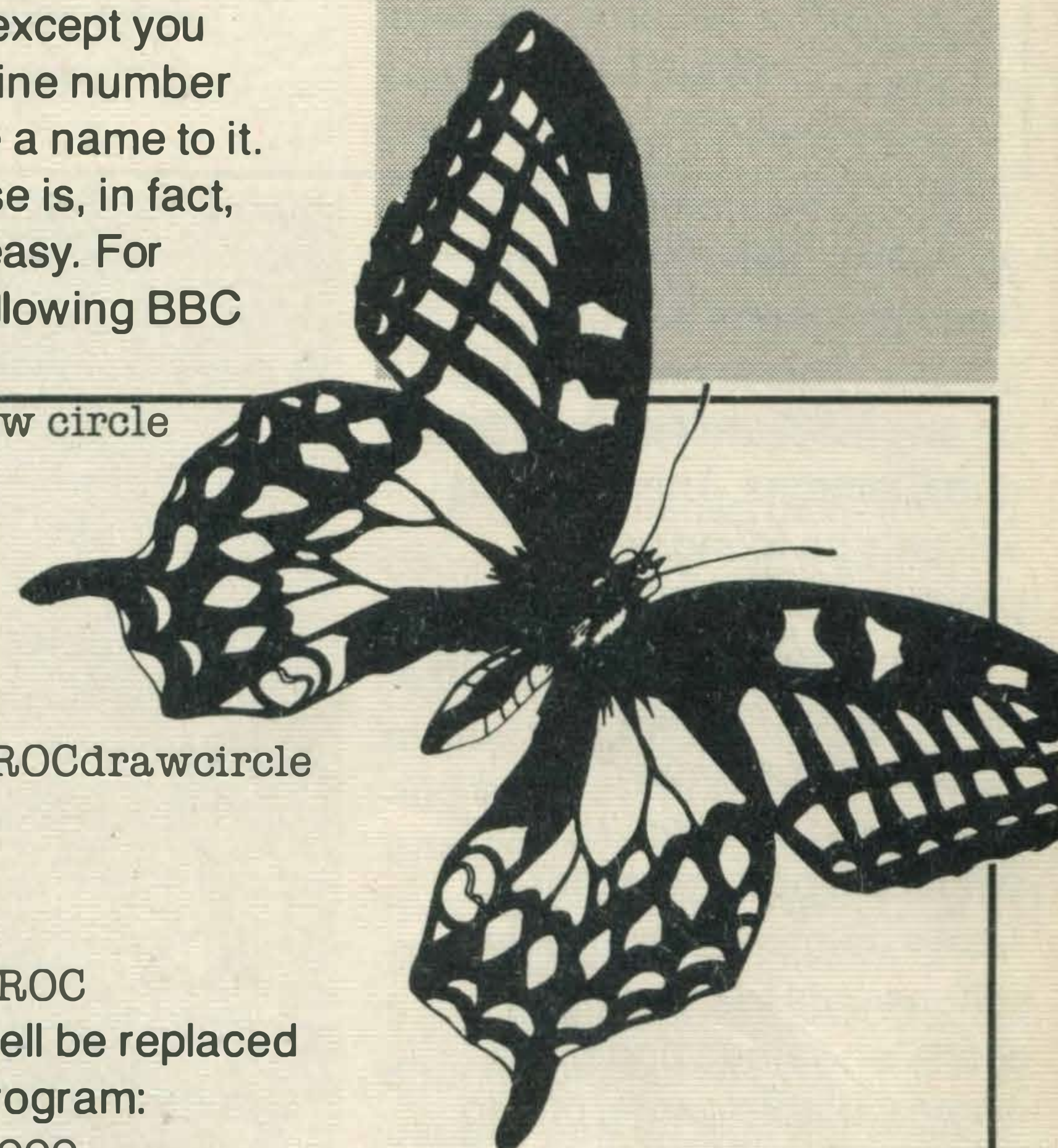
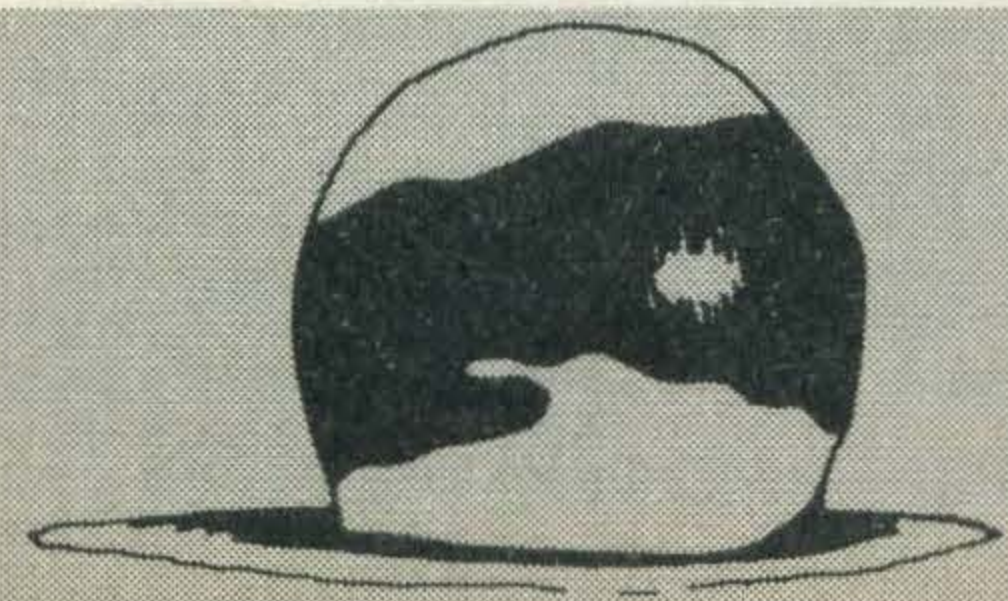
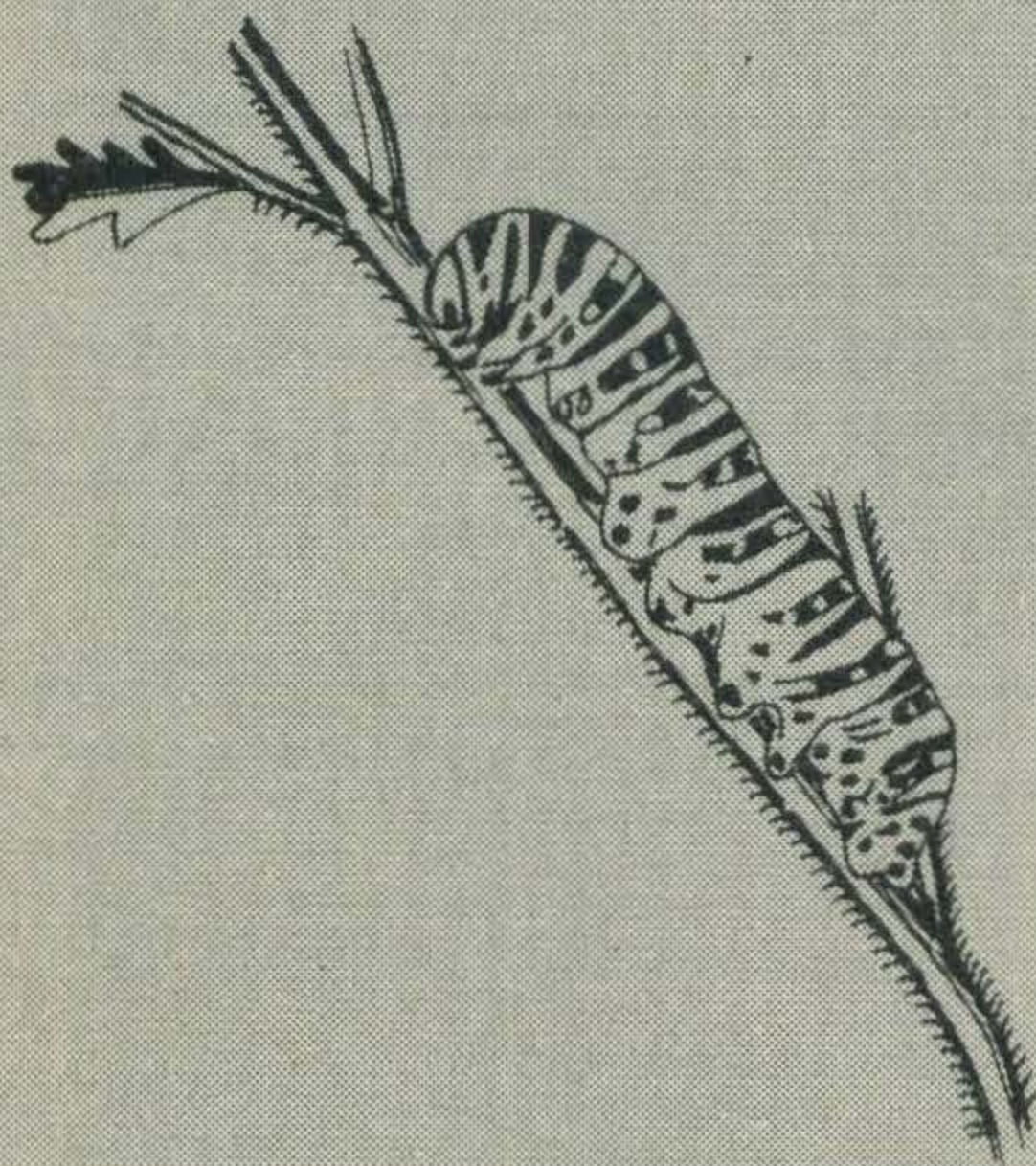
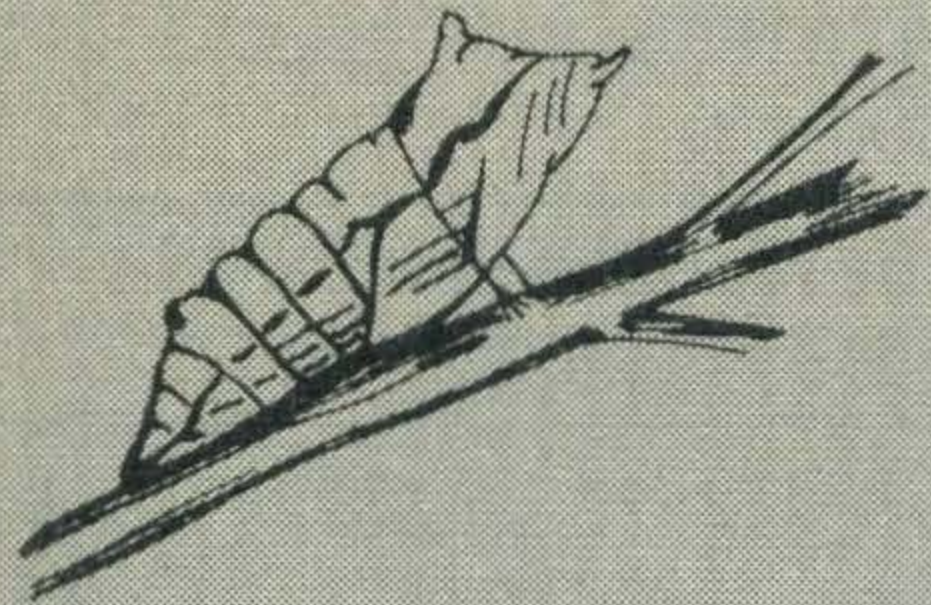
```
10 GOSUB 1000
"
"
"
"
"
1000 REM ** this line is not needed in MSX versions
1010 ,, ,,
1020 ,, ,,
1030 ,, ,,
1040 RETURN
```

Notice that you GOSUB the line number following the DEFPROC statement and NOT to the same line.

In some listings, you will come across rather more complex PROCEDURE calls such as:

```
200 PROCdrawcircle(200, 300, 3, 4)
"
"
"
"
2000 DEFPROCcircle(x, y, r, c)
"
"
"
2040 RETURN
```

The numbers in the brackets are used to send the values 200, 300, 3, 4 to the variables x, y, r, c in the procedure. This would probably be to draw a circle with its centre at coordinates 200, 300 with a radius of 3



and in colour 4. The MSX equivalent of this would be:

```
200 x=200: y=300: r=3: c=4: GOSUB 2010
''
''
''
2000 REM ** this line is not needed in MSX version **
2010
2020
2030
2040 RETURN
```

One other command which you will sometimes find in Beeb programs is the word LOCAL. This is not available on any machine except BBCs and Electrons. This keyword will only be found within procedures and allows the programmer to use a variable name which is already employed for some other purpose to be used 'Locally', within the procedure, for a totally different purpose. In other words the computer will store two values under the same variable name, one to use inside the procedure and one to use outside.

Converting this to MSX standard will require you to think up a new variable name. For example:

```
200 DEFPROC lower
210 LOCAL a, b: a=9: b=7
220 ,,
230 ,,
240 ,,
250 ENDPROC
```

This listing indicates that you have already used the variable names a and b and don't want the computer to forget their values. The value of a and b within the procedure will be different. The MSX version would look something like:

```
200 REM ** MSX version **
210 aa=9: ab=7
220 ,,
230 ,,
240 ,,
250 RETURN
```

Loops

In addition to FOR . NEXT . STEP and GOTO loops, some machines make use of REPEAT . . . UNTIL or WHILE . . . WEND loops. Beeb and Electron listings will usually contain REPEAT UNTIL loops, while AMSTRAD listings often

contain WHILE WEND loops. Although these are not directly available in MSX BASIC, it is possible to replace them with an appropriate GOTO. At first sight, a REPEAT UNTIL loop seems to behave in exactly the same way as WHILE WEND loops, but in fact there is a subtle difference in the way in which the loop is terminated. First, the REPEAT UNTIL loop in BBC BASIC:

```
10 REPEAT
20 ,,
30 ,,
40 ,,
50 PRINT "What shall I do now";
60 INPUT F$
70 UNTIL F$ = "quit"
80 END
```

can be replaced in MSX BASIC by the following:

```
10 REM ** This line is not needed in MSX version
20 ,,
30 ,,
40 ,,
50 PRINT "What shall I do now";
60 INPUT F$
70 IF F$() "quit" THEN GOTO 20
80 END
```

You will notice this this type of loop checks its condition at the end of the loop, whereas a WHILE . . . WEND loop checks the condition at the beginning; for example:

```
10 WHILE F$() "quit"
20 ,,
30 ,,
40 ,,
50 PRINT "What shall I do now";
60 INPUT F$
70 WEND
80 END
```

On the MSX this becomes:

```
10 IF F$ = "quit" THEN GOTO 80
20 ,,
30 ,,
40 ,,
50 PRINT "What shall I do now";
60 INPUT F$
70 GOTO 10
80 END
```

Occasionally, you will come across a rather strange loop in Beeb and Oric listings, the REPEAT UNTIL FALSE loop. Despite its fancy name, this really means REPEAT the loop forever and can simply

Programming

be replaced by a GOTO. Here is a BBC example:

```
10 REM ** BBC version **
20 REPEAT
30 PROCinstructions
40 PROCsetup
50 PROCyourguess
60 PROCmyguess
70 UNTIL FALSE
80 END
90 REM ** procedures **
```

This listing will, no doubt, look very strange to an MSX user, but is simply replaced with a listing such as:

```
10 REM ** MSX version **
20 REM ** This line is not necessary in MSX version **
30 GOSUB 1000
40 GOSUB 600
50 GOSUB 790
60 GOSUB 2000
70 GOTO 30
80 END
90 REM ** subroutines to replace procedures **
```

Screen width

So far, we have not mentioned one of the most important aspects of program conversion, screen width. The maximum screen width available for text on MSX machines is 40 characters, but the characters at the ends of the lines tend to disappear off the end of the screen when displayed on a TV set. If you intend to convert a program written with a screen width greater than 40 characters, you will probably need to adjust all the numbers in TAB and LOCATE commands and adjust the number of spaces in string variables. There are no hard and fast rules about this and it will probably be a matter of trial and error. The following list contains useful information about the screen widths of different MODES from other computers.

BBC/Electron

40 column modes: MODE 1, 4, 6, 7
20 column modes: MODE 2, 5
80 column modes: MODE 0, 3

Commodore 64

Always uses a 40 column mode.

Oric 1/Atmos

Always uses a 40 column mode.

Spectrum

Always uses a 32 column mode.

Amstrad

20 column mode: MODE 0

40 column mode: MODE 1

80 column mode: MODE 2

Obviously a list of this sort is very exclusive and I apologise for any machines missed out, but I've tried to include information on the most popular computers.

Colours

Different machines use different methods of introducing colour to a text display. The most common are PEN, INK, COLOUR, BORDER, PLOT etc.

My own method of introducing colour to the MSX version is to completely ignore the colour commands of the host computer and to introduce colour into the MSX version only after I've managed to get the program working in monochrome. If you try to convert the colour commands from the original computer to MSX, it is worth bearing in mind that the COLOR command contains three numbers. The first

number is the foreground colour, the second colour is the background colour and the final number is the border colour – the actual colours corresponding to the numbers should be found in your user manual. The foreground colour is the equivalent of the INK command (or PEN on the Amstrad) and the background colour is usually referred to as the PAPER colour on other machines.

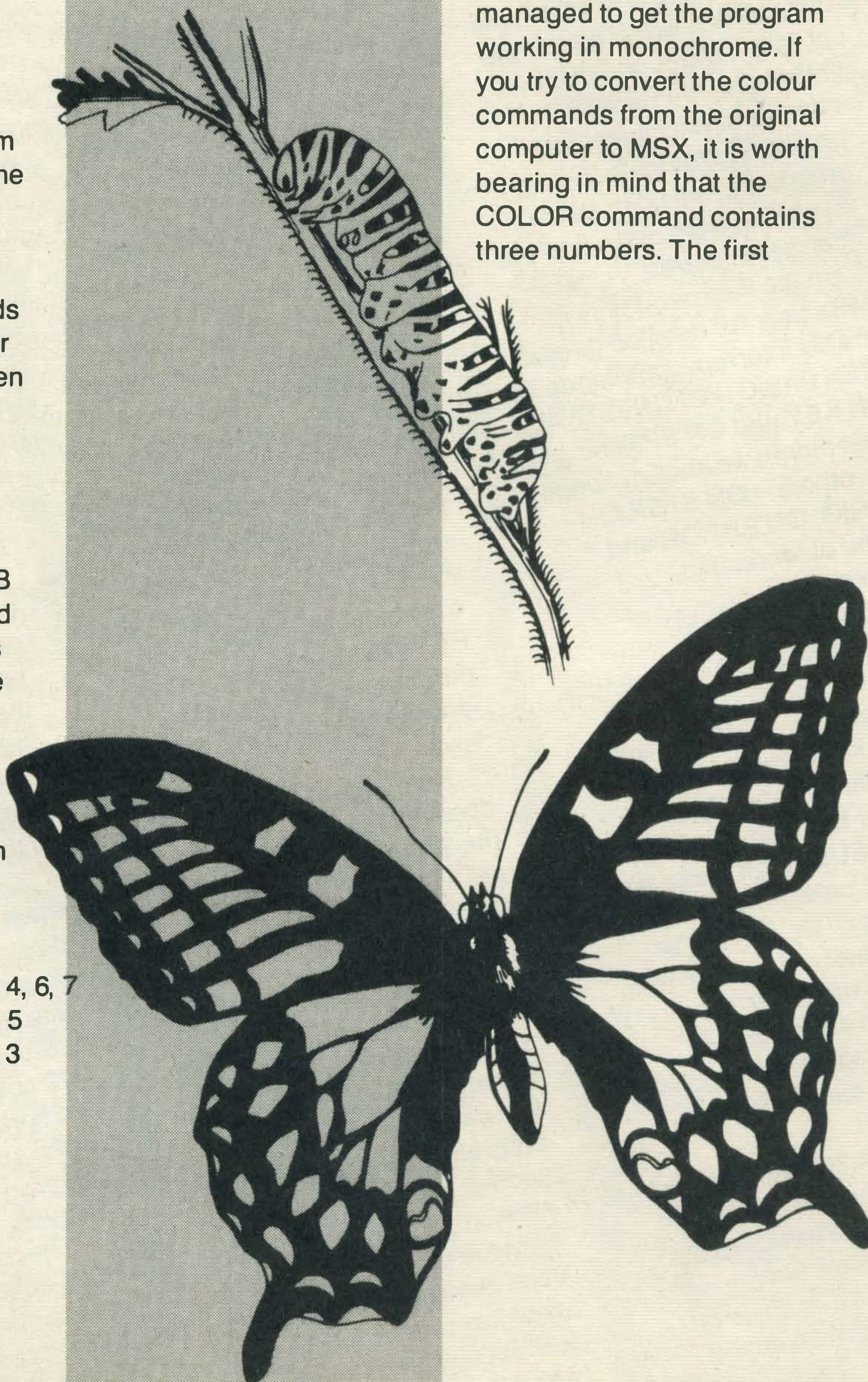
Sound Commands

MSX machines have a very comprehensive set of sound and play commands available to them, which are not available on other machines and whenever I convert a program from another machine, I always rewrite the SOUND effects from scratch. If you attempt to convert the sound commands, you will usually end up with a pathetic sound effect which fails to realise the potential of the MSX system.

Conclusions

In this article I have concentrated on aspects of programming which are not available in MSX machines and in most cases, it is extremely easy to convert such a program to run on an MSX computer. I hope that you don't go away with the impression that the MSX system is in any way inferior to other machines, because the BASIC specification contains so many other features which are not available on any other machine in the UK and so programs for other machines will still be mediocre even after conversion to the MSX standard! At the moment, however, there are so few programs written for MSX machines that we must make some sort of compromise.

In the next article, I shall look at how to convert user-defined graphics to run on a machine which uses sprites, converting graphics to run on MSX and how to convert functions which are machine-specific.



5X

MSX explained

More about interrupt routines in Graham Knight and Stuart Pirie's plain person's MSX User Guide.

Interrupts are special features of MSX BASIC which allow the user to specify certain conditions under which the program will branch to another part of the program. After one of these interrupts, the RETURN command will cause the program to return to the main program at the point which was being executed when the interrupt occurred.

In the February issue of MSX user we detailed three of the six interrupt commands in MSX BASIC – ON INTERVAL, ON KEY and ON SPRITE. This month we shall describe the others – ON STOP, ON STRIG, ON ERROR and some other related commands.

4 – ON STOP

The ON STOP command is used for trapping when the user presses the CTRL and STOP keys together. The program can specify a line number to which the program will GOSUB when these two keys are pressed. Normally, pressing CTRL and STOP would break out of a program but using the ON STOP command it is possible to write programs that cannot be broken out of after typing RUN.

Program 23 demonstrates the use of the ON STOP command. The basics of the program are very simple – it counts from one in steps of one until you switch off the

computer. If you want to keep the program, make sure you save it to tape before RUNNING as there is absolutely no way of breaking out of the program after it has started to count. Lines 100 and 110 set up the interrupt, line 100 specifies to GOSUB 1000 when the CTRL and STOP keys are pressed, and line 110 turns on the interrupt.

Lines 120-150 are very simple and do the counting. Line 1000 prints the word STOP on the screen. Line 1010 returns to the main program. This is to let you see that the computer does do something when you press the CTRL and STOP keys. Try changing line 1000 to RETURN and notice that the program seemingly totally ignores the CTRL and STOP keys.

Note that this command does not turn off the use of the STOP key on its own – this will still pause and restart a program.

5 – ON STRIG

The ON STRIG is used for trapping when any of the four joystick buttons or the space bar are pressed. A different line can be specified for the program to GOSUB to for each of these five buttons. After specifying where you want the program to GOSUB, each individual button being used has to be turned on with a STRIG (x) ON command where x is the number of the

button to be turned

Program 24 demonstrates the use of the ON STRIG command. This is a game in which a blue sprite of the shape of a face moves back and forth across the screen. There is a purple button in the middle of the screen

Press the space bar when the sprite is completely enclosed by the box. If the face was inside the box, it will turn white for half a second. If it was not inside, it will turn red. While the sprite is a different colour, the space bar will have no effect. The program will keep going until you press the CTRL and STOP keys together.

Line 100 sets up the interrupt and line 110 turns it on. Line 120 sets up the screen and sprite modes. The Hi-resolution screen mode 2 is used to draw the box. Sprite mode zero is used for 8 x 8 unmagnified sprites. Line 130 draws the box on the screen. Line 140 contains the data for the shape of the sprite and this is read in by lines 150-190. Line 150 sets the initial position, direction and colour of the sprite.

Lines 200-230 form the main program loop. Line 200 displays the sprite at its current position and colour. Line 210 updates the value of the position variable S% and line 220 changes the direction of the sprite if it is near the edge of the screen. Line 230 loops the program back round to line 200.

The routine from lines 1000-1060 is called when the space bar is pressed. First the sprite is assumed to be outside the box and the colour variable C% is set to red. Then the rest of line 10

Programming

PROGRAM 23

```
100 ON STOP GOSUB 1000
110 STOP ON
120 A=1
130 PRINT A
140 A=A+1
150 GOTO 130
1000 PRINT "STOP !"
1010 RETURN
```

PROGRAM 24

```
100 ON STRIG GOSUB 1000
110 STRIG(0)ON
120 COLOR 15,1,1:SCREEN 2,0:BEEP
130 LINE (111,91)-(124,104),13,B
140 DATA 3C,7E,DB,FF,DB,E7,7E,3C
150 S$="":FOR SZ=1 TO 8:READ M$
160 S$=S$+CHR$(VAL("&H"+M$))
170 NEXT SZ:SPRITE$(0)=S$
180 SZ=95:MZ=2:CZ=5
200 PUTSPRITE 0,(SZ,93),CZ
210 SZ=SZ+MZ
220 IF SZ<1 OR SZ>250 THEN MZ=-MZ
230 GOTO 200
1000 CZ=6:IF SZ>=113 AND SZ<=119 THEN CZ=15
1010 IF CZ=6 THEN PLAY "V1558M2002L100"
1020 IF CZ=15 THEN PLAY "V1559M200002L30CDEDEF"
1030 ON INTERVAL=25 GOSUB 1100
1040 INTERVAL ON
1050 STRIG(0) OFF
1060 RETURN
1100 INTERVAL OFF
1110 STRIG(0) ON
1120 CZ=5
1130 RETURN
```

tests whether the sprite was inside the box. If it was, the colour variable is changed to white.

Lines 1010-1020 play a buzz if the button was pressed at the wrong time and a friendly tune if it was pressed at the right time.

We want the sprite to stay this colour for half a second and during that time for the space bar to have no effect.

PROGRAM 25

```
100 ON ERROR GOTO 1000
110 PRINT
120 INPUT "FIRST NUMBER ";A
130 INPUT "SECOND NUMBER ";B
140 C=A/B
150 PRINT:PRINT "A/B IS";C
160 GOTO 110
1000 PRINT
1010 IF ERR=11 THEN PRINT "CANNOT DIVIDE BY ZERO":RESUME 110
1020 IF ERR=6 THEN PRINT "OVERFLOW ERROR":RESUME 110
1030 PRINT "ERROR";ERR;"IN LINE";ERL
1040 RESUME 110
```

We can do this by setting up a second interrupt, which is called after half a second. This is set up in lines 1010-1020. Then the space bar interrupt is turned off and the program returns.

The routine at line 1100 turns itself off so that it is not called again. Line 1110 re-enables the space bar interrupt. Line 1120 re-sets the colour to blue and line 1140 returns.

This program shows how interrupts can set up and call other interrupts and even turn themselves off and on!

6 - ON ERROR

This command is used to trap the 'Error . . . in line . . .'

type of errors. The program will then jump to the line specified in the ON ERROR command. When the program

recognised, the error number and the line at which it occurred is displayed on the screen.

routine. Lines 110-130 ask the user for the day and the number is held in variable D. Line 140 checks to see if it is

reaches there, a suitable message can be printed and then the program can continue. Without the use of ON ERROR GOTO, a program would stop executing completely when an error occurs. It is therefore good practice to incorporate error-trapping routines in your programs.

Each error that occurs in MSX BASIC has a corresponding error number (these will be detailed in your MSX BASIC manual). When the program jumps to your error routine the error number is stored in variable ERR and the number of the line in which it occurred is stored in variable ERL. The program can then look at these variables to find out which error has occurred. As ERR and ERL are specifically to store the error number and line of error, no values can be assigned to them.

Program 25 shows the use of the ON ERROR command. It is a simple program which divides the two numbers entered by the user. Line 100 sets up the program so that if any errors occur, the program jumps to line 1000.

This routine checks the number of the error with two that it knows – dividing by zero and overflow error. If it is either of these, a suitable message is printed. The program then returns to line 110 using the RESUME 110 command. This command has to be used when leaving an error trap routine, otherwise no more errors will be trapped. If there is no line number specified, the program will return to the statement after the one that generated the error. In our program, if the error is not

Program 25 illustrates a simple use of the ON ERROR command, trapping errors and returning to a main part of the program. There is a command in MSX BASIC which allows any error you want to 'appear' to have occurred. For example the error number for 'Out of memory' error is 7 (as explained earlier). To generate this error, type ERROR 7 and press RETURN in direct mode, ie with OK before the cursor. The computer should respond with 'Out of memory'.

This command on its own is not of much use but when combined with the ON ERROR GOTO command, it becomes very useful. Only about 30 of the possible error numbers 1-255 are used by the BASIC – the rest can be used by your own program.

Program 26 will allow you to display all the error messages on the screen. Function key F1 is defined in line as GOTO 130 (with a carriage return). Press the F1 key to display the first error message – indicating a syntax error. As the programming has no error trapping routine you are immediately returned to command mode. Keep pressing the F1 function key till all the error messages have been displayed on the screen.

Program 27 shows how the ERROR command can be used to generate very specific error messages. When you run the program, you are asked to enter the number of the day and the month today. The program will check these for errors and display any that occur.

Line 100 sets up the error

in range and if it is not, then error 210 is generated. This will cause the computer to jump to the error routine at line 1000 with error number 210.

If there has been no error, the program will continue and the user will be asked for the month. This is then checked for range 1-12 and if it is outside this range error 211 is generated by line 180. Lines 190 and 200 hold data for the number of days in every month. Line 210 reads in the correct number for the month entered by the user. Line 220 checks whether the day given is greater than the number of days in the chosen month. If it is invalid, error 212 is generated. If no errors have occurred, the program prints CORRECT and stops.

The error routine at 1000 checks through errors 210-212, displays appropriate messages and returns to line 110. If any other errors had occurred, line 1040 displays the values of ERR and ERL.

Precautions

Care must be taken with what is done within subroutines of this kind – you must be careful that the subroutines do not upset the rest of the program. This is possible because they can occur any time. To demonstrate this, type in and RUN **programs 28** and **29** separately. They are both simple programs – **28** displays MSX MICROS FROM JAPAN flashing in the middle of the screen and **29** displays one random character near the top of the screen using interrupts.

The problem occurs when you try to run the two

PROGRAM 26

```
100 KEY1,"GOTO 130"+CHR$(13)
110 ER=1
120 ERROR ER
130 ER=ER+1
140 GOTO 120
```

PROGRAM 28

```
200 CLS
210 LOCATE 8,12
220 PRINT "MSX MICROS FROM JAPAN"
230 FOR F=1 TO 10:NEXT F
240 LOCATE 8,12
250 PRINT SPC(25)
260 FOR F=1 TO 10:NEXT F
270 GOTO 210
```

programs together - MERGE them together as explained in earlier issues and then RUN the combined program. You will see that after a few moments, the MSX message appears under the random character.

This is because the interrupt routine has occurred between the LOCATE and PRINT commands for the MSX message. The interrupt

PROGRAM 29

```
100 CLS
110 ON INTERVAL=10 GOSUB 1000
120 INTERVAL ON
999 GOTO 999
1000 LOCATE 15,4
1010 PRINT CHR$(RND(1)*100+34)
1020 RETURN
```

PROGRAM 27

```
100 ON ERROR GOTO 1000
110 PRINT
120 PRINT "ENTER THE DAY TODAY"
130 INPUT "(1-31) ";D
140 IF D<1 OR D>31 THEN ERROR 210
150 PRINT
160 PRINT "ENTER THE MONTH TODAY"
170 INPUT "(1-12) ";M
180 IF M<1 OR M>12 THEN ERROR 211
190 DATA 31,28,31,30,31,30
200 DATA 31,31,30,31,30,31
210 FOR F=1 TO M:READ DM:NEXT F
220 IF D>DM THEN ERROR 212
230 PRINT
240 PRINT "CORRECT."
250 END
```

```
1000 PRINT
1010 IF ERR=210 THEN PRINT "THAT IS AN INVALID DAY":RESUME 110
1020 IF ERR=211 THEN PRINT "THAT IS AN INVALID MONTH":RESUME 110
1030 IF ERR=212 THEN PRINT "INCORRECT DAY FOR THAT MONTH":RESUME 110
1040 PRINT "ERROR";ERR;"IN LINE";ERL
1050 RESUME 110
```

routine leaves the cursor on the line below the random character so when it returns to the main program, the message is printed at that location.

There is an easy way to solve this problem: insert a line 1015 LOCATE 8, 12 which will make sure that the program returns with the cursor at the right place.

For more complicated subroutines, a lot of programming might be required to make sure the rest of the program still works properly.

Unique MSX features

MSX BASIC is unique in having the interrupt driven routines featured in programs

20-29. No other home micro has ON INTERVAL GOSUB, ON KEY GOSUB, ON SPRITE GOSUB, ON STOP GOSUB, ON STRIG GOSUB, and ON ERROR GOTO commands incorporated into its BASIC.

These interrupt commands greatly simplify all kinds of programs. These commands can be simulated on other computers but this would require longer and more complex programming routines which would inevitably be much slower in operation. In fact some of these interrupt driven routines would require programmers to revert to machine code programming to get exactly the same results as a single line of MSX BASIC.

Graham Knight has gathered the latest hot tips from Japan.

East

The giant Casio company make millions of calculators each year and have a very successful computer division making large business systems. While many of the established MSX manufacturers expected cheap MSX micros to come on the market most of them thought they would be produced by companies based in Taiwan, Macau, or Korea. Casio have surprised many by introducing an MSX micro which is being sold in Akihabara stores for 27,000 yen (about £89).

The Casio computer is smaller both in size and in memory than any other MSX micro. It sticks rigidly to the MSX specification and has the usual Z80A processor running at 3.6 Mhz, a single cartridge slot, 32K BASIC in ROM, the full 16K of video RAM, cassette and joystick sockets, and so on.

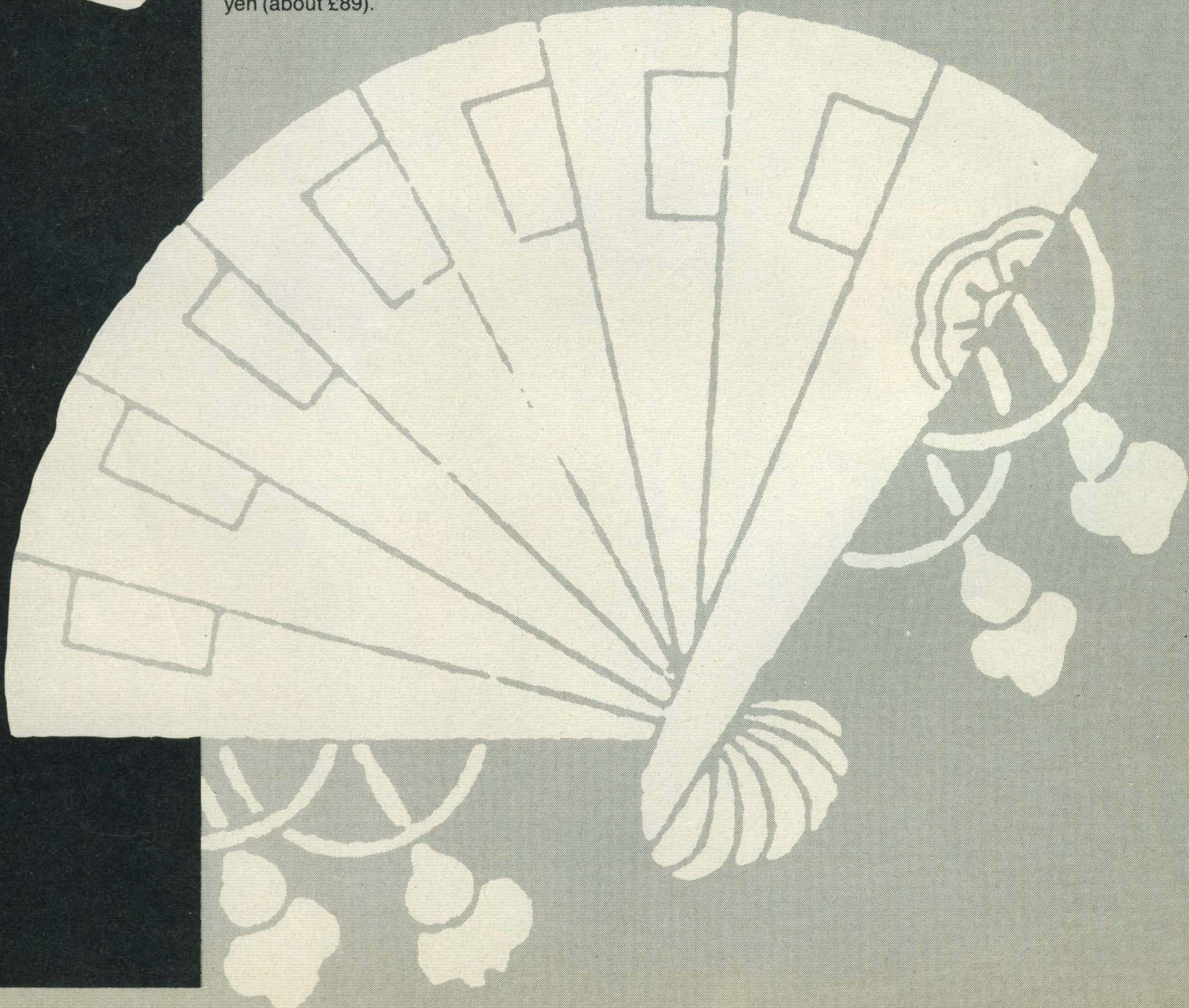
The price breakthrough has been made possible by the use of a cheaper, rubber key type of keyboard and by supplying it with just 8K of user RAM memory. Casio has a great deal of experience at manufacturing keyboards and although this one is easy to use it does not compare well with the standard MSX keyboard.

Japan has a large market for low-cost family computers. Tomy, Namco, Sega, Bandai and the Max from

Commodore have all been successful suppliers of computer products for the low end of the Japanese market.

The Casio is aimed at the home entertainment market where it will mostly be used with cartridge software which does not require large amounts of RAM memory. The lack of RAM is not a problem if the users progress to programming as almost all the MSX manufacturers sell memory extenders which simply plug into the cartridge slot. These 16K RAM memory packs cost about £27 in Japan.

The new Casio has proved to be a steady seller in the last eight weeks. Many customers are obviously buying it purely for entertainment and often leave the shop with a Casio and a handful of Konami cartridges. Until now the Fujitsu MSX at 49,000 yen



(about £150) has been the cheapest MSX micro. Casio are definitely now the MSX price leaders with a computer which is itself cheaper than some data cassette recorders.

Potential

The re-election of Rajiv Gandhi as Prime Minister of India was the cause of some celebration at the Tokyo headquarters of ASCII-Microsoft. Kay Nishi, the inventor of MSX, is on very friendly terms with Mr. Gandhi and has been paying frequent visits to Delhi to establish MSX as a standard for India. A team at ASCII-Microsoft has been developing a special version of MSX BASIC which incorporates special language symbols.

Mr. Nishi knows that with a population of over 200 million there is tremendous potential for MSX in India. The new Prime Minister is familiar with digital electronics and is a well known radio amateur – like Mr. Nishi.

New standard

In Japan there are about 20 computer magazines and about an equal number devoted to hobby electronics. Many of the articles in the two types of magazine seem to be crossing over into the other hobby's interest area. Many of both types of magazine are now carrying articles about using computers for communications. Most of this type of work is being done via bi-directional RS232 interfaces connected to telephone modems or radio apparatus. To feed this growing communications market ASCII-MICROSOFT have extended the MSX specification to cover RS232 interfaces.

The only RS232 interfaces previously available for MSX have been uni-directional serial devices which were only suitable for outputting data to an external device – usually a serial printer. The new addition to the specification allows manufacturers to produce bi-directional devices to a

common technical standard and using the same MSX BASIC commands. The new RS232 units simply plug into the cartridge slot and have a built-in ROM which adds the new BASIC commands.

JVC and Toshiba have been the first manufacturers to actually produce these new units. They are slightly taller than a games cartridge, and the bi-directional interface allows users to link two MSX micros and to send and receive data from mainframe computers. The MSX can also be used in a terminal mode as a display console for another computer. The new MSX commands are:

CALL COMINI selects the RS232 mode;
COMTERM puts the MSX micro into terminal mode;
COMDTR signals to the host computer that the terminal is ready to transmit data;
COM GOSUB, COM BREAK, COM STAT, COM STOP are all commands which check the status, enable and disable interrupt signals from the RS232 port;
LOAD receives a program via the RS232;
SAVE sends a program via the interface;
MERGE merges the received program with an existing program loaded from cassette;
LOC and LOF calculates the number of bytes in the receive buffer and those remaining free.

The interface is very flexible. The character lengths can be 5, 6, 7 or 8 bits and it allows full duplex operation. The communication speed can be software selected at 14 different speeds from 50 up to 19,200 bauds per second. It is possible to send and receive at different baud rates by setting the parameters of the COMINI command.

One of the most interesting applications allows one MSX computer to receive a program sent by another MSX micro and to then run that program. Data can be

transferred and received from non-MSX computers as ASCII files. The interface also has many applications for use with the Captain system – a Japanese equivalent to our Prestel and Teletext networks.

Initial supplies of the interfaces from JVC and Toshiba have been limited and the demand over the New Year holiday period seems to have soaked up all the available interfaces. Sanyo are expected to release their bi-directional RS232 in the next few weeks. It will, of course, perform according to the additional MSX specification.

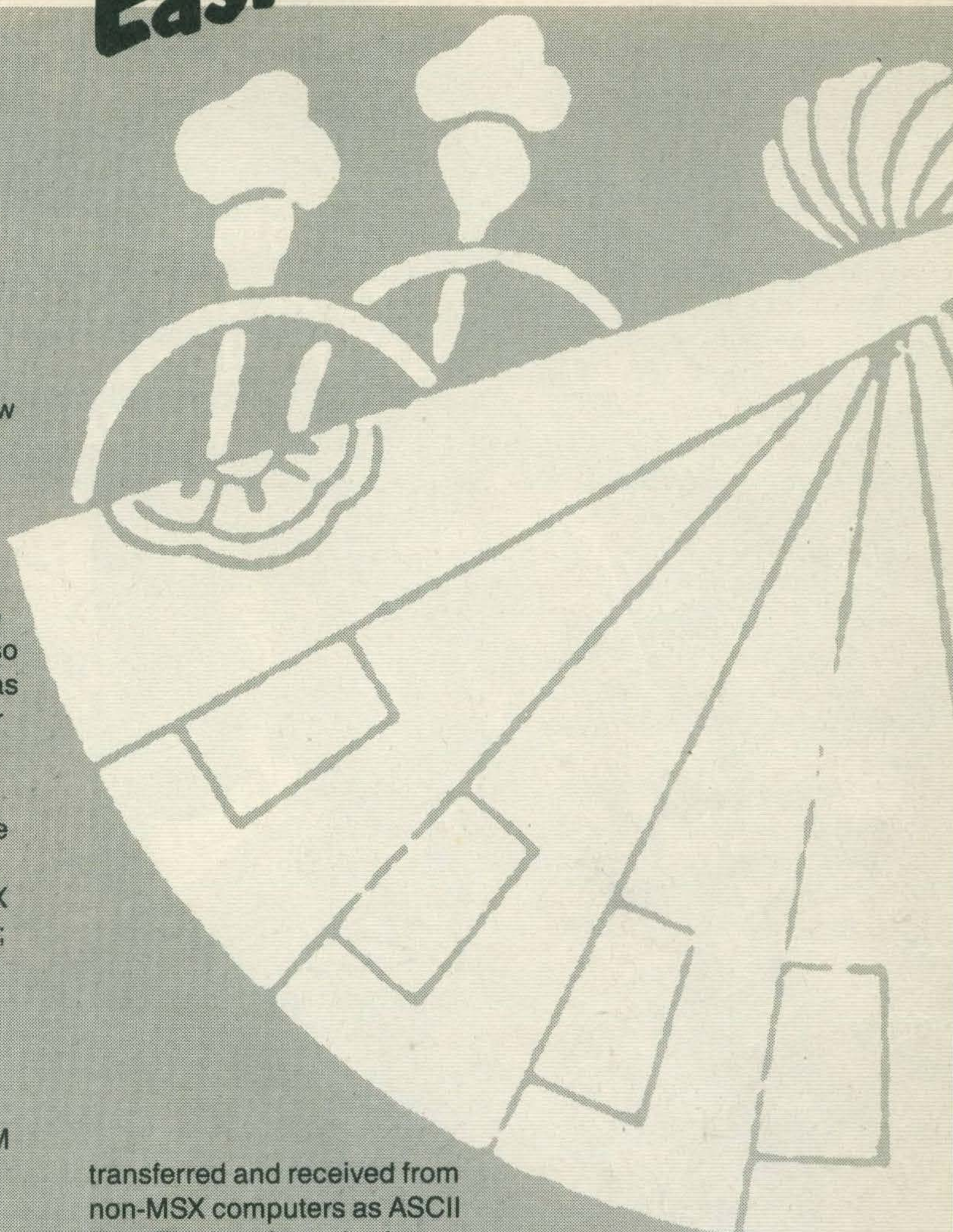
The good thing for users about the MSX agreement is that all the manufacturers will use the same memory area for the interface ROM and as always the interfaces are not machine-dependent but will work with any manufacturer's MSX micro.

MSX-net reaches Japan

MSX-net is a UK-based computer network which

offers electronic mail and telex facilities and provides access to a number of international databases. The communications system used is a combination of Private Switched Telephone System (PSTN) and Packet Switchstream (PSS) utilising the Telecom Gold Service.

With the current upsurge in interest in communications via RS232 interfaces a number of very important Japanese computer people have joined the UK MSX network. If any UK users leave messages on the MSX noticeboard they may well be read by the engineers who designed your computer! The engineers are using the MSX network to send messages to Europe and America. One of the computer stores in Osaka plans to have UK MSX net facilities on display at the forthcoming MSX fair.



Cheaper disk units?

The Nagoya local newspapers were recently full of stories about the success of the local Sony and Toshiba factories in producing thousands of MSX computers for the export market. The same papers also carried a story about how Toshiba had beaten a number of Japanese manufacturers for a contract to supply thousands of disk units to IBM of America.

These disks will be in the 3.5 inch format already popular with MSX users. At present this section of the MSX market is supplied by Toshiba and Sony who market similar units – each with 360K of formatted storage on every disk. The fact that Toshiba had won this huge IBM order prompted me to ring and enquire whether these volume orders would help to lower the price of the MSX disk units. The encouraging reply was that they would introduce a new cheaper 3.5 inch disk in April but the actual price still has to be fixed.

Quick Disk

Yuji Kudo, the President of Hudson Corporation, the largest independent software house in Japan, recently showed me the Quick Disk system connected to an MSX using a Hudson operating system.

The problem with Quick Disk has been that the single spiral track meant that all access to files was sequential. Hudson's software engineers have now developed an operating system which can give random access to any file on the disk. Mr. Kudo hopes that Quick Disks will become available in the UK in the next few months.

TV printers

Mitsubishi have recently started selling three new printers. One is called the TV printer mainly because it is a permanent part of a TV set – it prints whatever is on the

screen on a high resolution printer. As it prints the complete screen it is possible to get both a print of a TV picture as well as from a computer display.

Mitsubishi's second printer is about the size of a radio and it can also print screen pictures. The third printer is the most expensive but it is a jet ink print and can be programmed to reproduce pictures which are very detailed and are in vivid colours. The printer is equipped with tubes of coloured ink which are gently squeezed to spray the character onto the paper in a fine jet. The resolution is very high and it is almost silent in operation.

Toshiba winner

Toshiba were kind enough to invite me to their headquarters to see a new model, the HX-22. This model is to be upmarket of the popularly priced HX-10 and will probably cost around £350. While most MSX micros are very similar in technical specification, the new Toshiba is certainly very different.

It retains all the features of the HX-10 but has an additional 32K of ROM memory and has a bi-directional serial interface

built in as standard. It also has a special RAM DISK feature, a printer buffer, a word processor and two cartridge slots.

The RAM DISK is certainly not available on other MSX micros and it is called by entering the command CALL MEMINI. Up to 16 programs can then be loaded from tape or disk into a 32K area of RAM. Each program is then treated just like a disk file. Ask for a directory of the RAM DISK and the names of the programs in memory appear on the screen together with the amount of free memory available on the RAM DISK.

All the programs can then RUN instantly – far faster than even waiting for a disk to find and load the program. Imagine the possibilities – a bunch of your favourite games available without reloading tapes. None of this RAM DISK storage affects the operation of the computer as there are still 28815 bytes free in BASIC at all times. It is possible to RENAME, KILL or MERGE the RAM disk files as if they were on an ordinary floppy disk.

Another feature is that a 32K area of memory can be set aside as a printer buffer – this is called by the command SPOOL ON. A 28K BASIC program can be loaded into the ordinary user RAM, SPOOL ON called, and then the program can be RUN and printed at the same time.

This SPOOLING facility will be familiar to users of mainframe computers but is unusual on small micros. It will make the HX-22 very suitable for a variety of business applications. The operator can be printing one invoice whilst simultaneously entering the information for the next.

The built-in software for the word processor is totally unsuitable for use in Europe as it only handles the Japanese Kanji characters. If the HX-22 were to be released in Europe the space occupied by the wordprocessing software could hold a spreadsheet or any other application program. This is the first real upgrade from the standard MSX format and Toshiba look certain to have produced another winner.

Chuck away those old-fashioned paper cards and gamble electronically with your MSX computer.

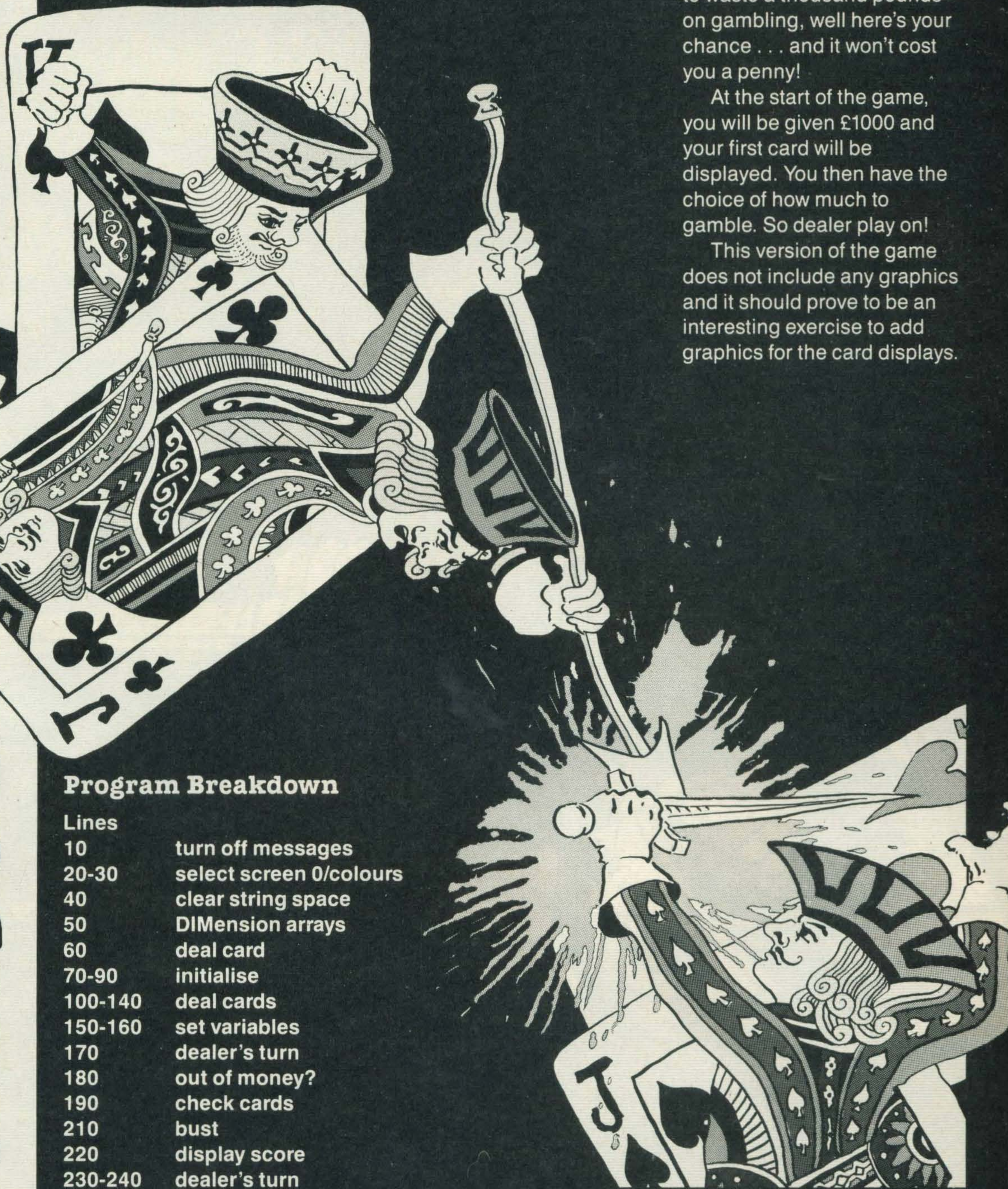
This is a computerised version of the familiar card game where your object is to compete with the computer to get the score closest to 21. If you have always dreamt of being able to waste a thousand pounds on gambling, well here's your chance . . . and it won't cost you a penny!

At the start of the game, you will be given £1000 and your first card will be displayed. You then have the choice of how much to gamble. So dealer play on!

This version of the game does not include any graphics and it should prove to be an interesting exercise to add graphics for the card displays.

jack

Black



Program Breakdown

Lines	
10	turn off messages
20-30	select screen 0/colours
40	clear string space
50	DIMension arrays
60	deal card
70-90	initialise
100-140	deal cards
150-160	set variables
170	dealer's turn
180	out of money?
190	check cards
210	bust
220	display score
230-240	dealer's turn
250-290	end game?
300-380	deal card
390-430	display cards
440-470	test score
480-500	stick or twist
510-710	check cards
720-800	check dealer's cards
810-860	bust

Main Variables

sc	score
c	random card
c\$	card number/letter
ym	money left
b	bet

Other variables are used as flags.

```

10 KEY OFF
20 COLOR 15,4,8
30 SCREEN 0
40 CLEAR 1000:YM=1000
50 DIM Y$(20),D$(20),Y(20)
60 GOSUB 300
70 CLS:PRINT"You have :-":YM;" pounds in cash"
80 YN=1:Y$(1)=C$:D$(1)="":NG=17
90 Y(1)=C
100 SC=0
110 PRINT:PRINT"Your First card = ";;GOSUB 390
120 PRINT"How much do you want to bet ";;INPUT B
130 PRINT:PRINT" Dealer's card = ";;GOSUB 440
140 PRINT" Your card = ";;GOSUB 300
150 Y$(2)=C$:YN=2:Y(2)=C:OS=16
160 B=ABS(B)
170 GOSUB 390
180 IF B>YM THEN 820
190 GOSUB 470
200 PRINT:PRINT
210 IF SC>21 THEN LOCATE 5,20:PRINT"You're bust !":YM=YM-B:GOTO 250
220 PRINTN$;"You scored :- ";SC;" ";
230 PRINT"Dealer's throws :- "
240 GOSUB 720
250 PRINT:PRINT"You now have :-":YM;" "
260 PRINT:PRINT"Do you want to play again <Y/N> ?"
270 A$=INKEY$:IF A$="y" OR A$="Y" THEN 60
280 IF A$="" THEN 270
290 PRINT:PRINT"Goodbye...":END
300 C=RND(-TIME)

```

```

310 C=INT(RND(1)*13+1)
320 IF C>1 AND C<10 THEN C$=CHR$(C+48):RETURN
330 IF C=10 THEN C$="10"
340 IF C=11 THEN C$="J" ELSE IF C=12 THEN C$="Q"
350 IF C=13 THEN C$="K"
360 IF C>10 THEN C=10
370 IF C=1 THEN C$="A":C=11
380 RETURN
390 IF D$(1)="dealer" THEN OS=16 ELSE OS=0
400 I=YN
410 C$=Y$(I)
420 GOSUB 690
430 PRINT:PRINT:RETURN

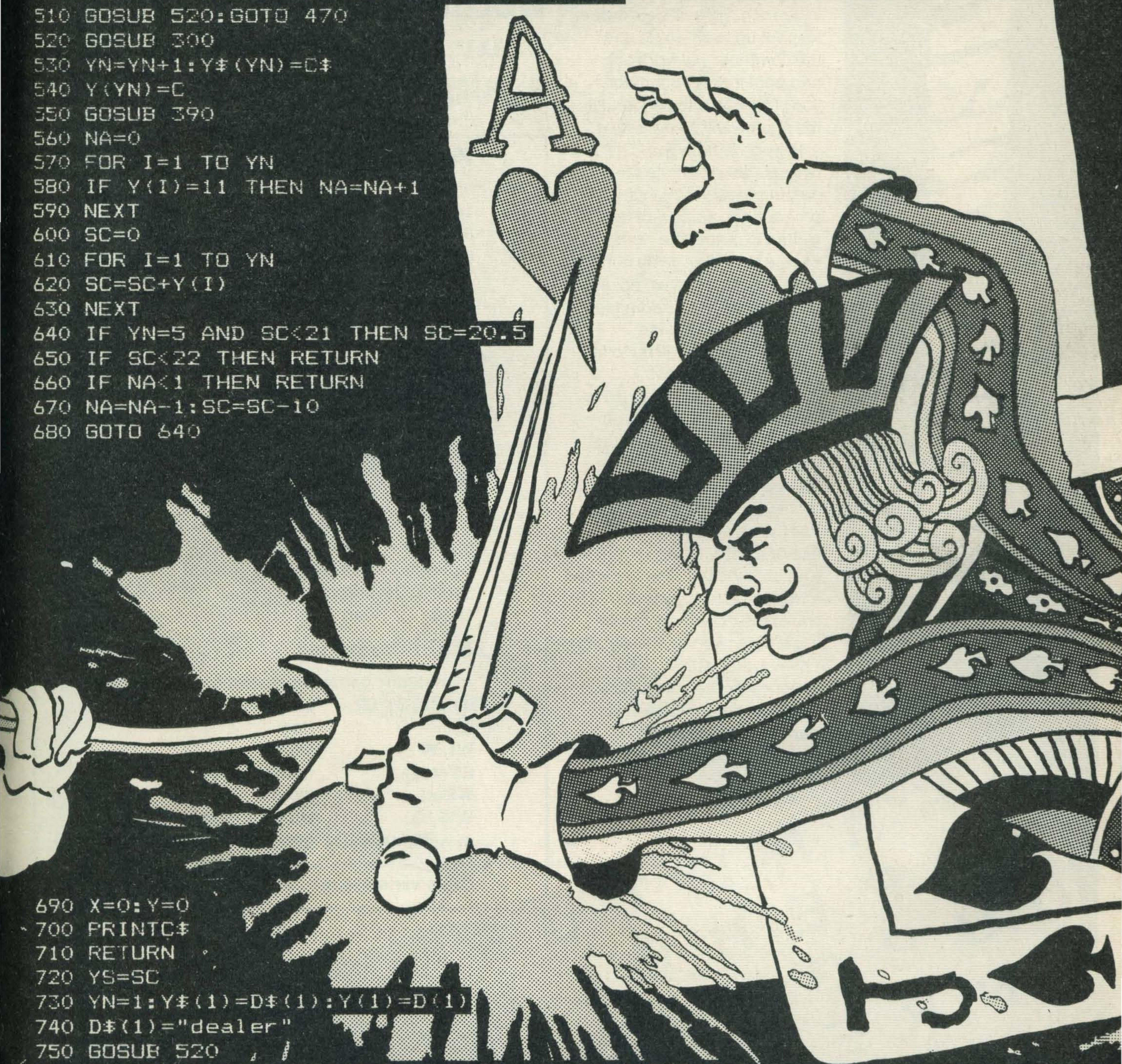
```

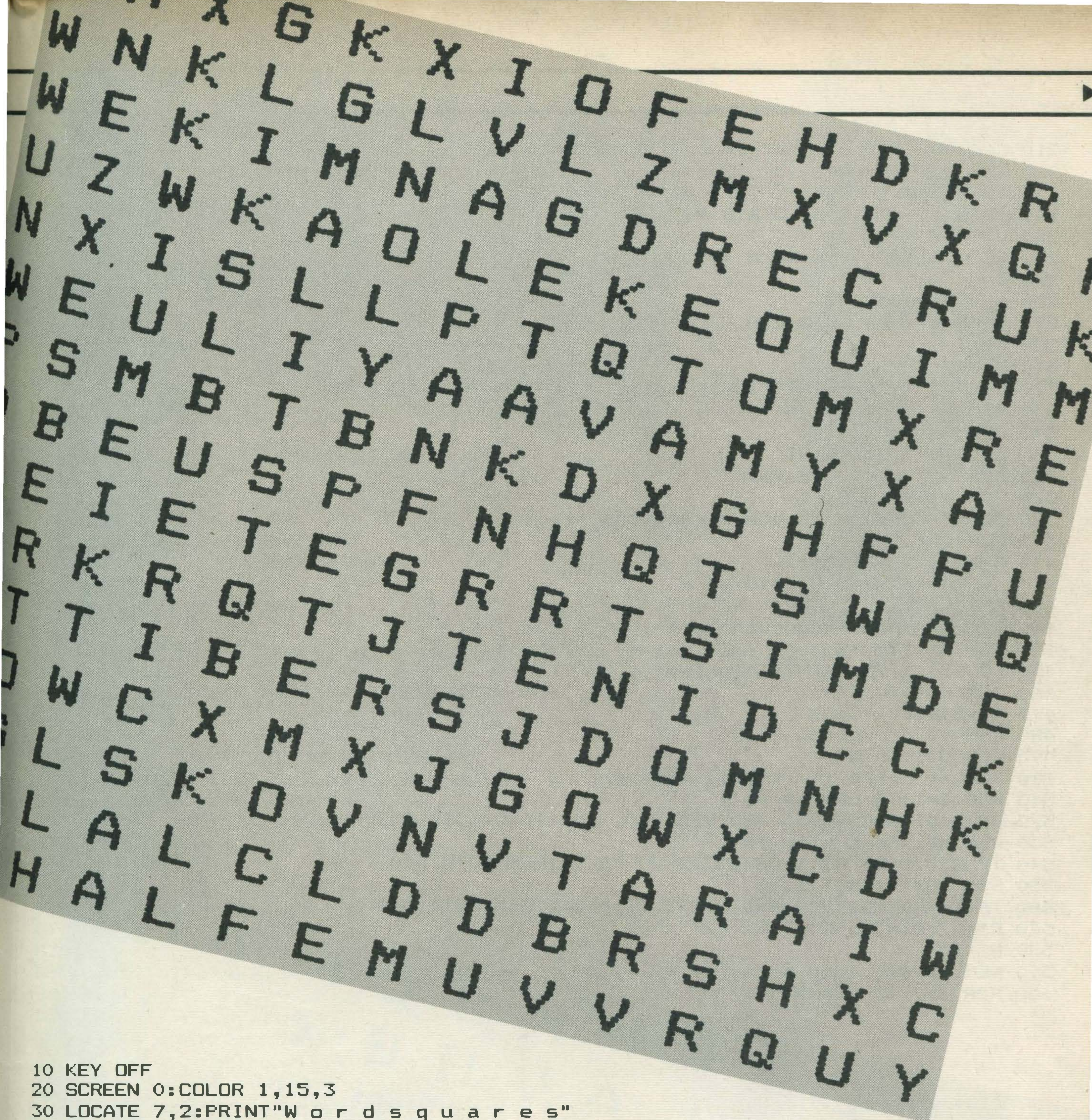


THERAPY

```
440 GOSUB 300
450 DN=1: X=18: Y=2: D(1)=C
460 GOSUB 700: RETURN
470 IF SC>21 THEN RETURN
480 PRINT: PRINT: PRINT "Stick or Twist <S/T> ?"
490 Q#=INKEY#: IF Q#="s" OR Q#="S" OR Q#="t" OR Q#="T" THEN 500 ELSE 490
500 NG=NG-1: IF Q#="s" OR Q#="S" THEN 560
510 GOSUB 520: GOTO 470
520 GOSUB 300
530 YN=YN+1: Y#(YN)=C#
540 Y(YN)=C
550 GOSUB 390
560 NA=0
570 FOR I=1 TO YN
580 IF Y(I)=11 THEN NA=NA+1
590 NEXT
600 SC=0
610 FOR I=1 TO YN
620 SC=SC+Y(I)
630 NEXT
640 IF YN=5 AND SC<21 THEN SC=20.5
650 IF SC<22 THEN RETURN
660 IF NA<1 THEN RETURN
670 NA=NA-1: SC=SC-10
680 GOTO 640
```

```
690 X=0: Y=0
700 PRINT C#
710 RETURN
720 YS=SC
730 YN=1: Y#(1)=D#(1): Y(1)=D(1)
740 D#(1)="dealer"
750 GOSUB 520
760 IF SC=YS THEN PRINT "Evens bet again ": RETURN
770 IF SC<YS THEN 750
780 IF SC>21 THEN PRINT "Dealer has bust ": YM=YM+B: RETURN
790 PRINT "Dealer wins!": YM=YM-B
800 RETURN
810 REM ** BROKE **
820 CLS: LOCATE 10,5: PRINT "You ran out of money !"
830 PLAY "cdededededec"
840 LOCATE 1,20: PRINT "Press <Space Bar> to play again."
850 A#=INKEY#: IF A#<>" " THEN 950
860 RUN
```





```
10 KEY OFF
20 SCREEN 0:COLOR 1,15,3
30 LOCATE 7,2:PRINT"Wordsquares"
40 LOCATE 7,3:PRINT"=====
50 CLEAR 1000
60 DEFINT A-Z
70 PRINT:PRINT:PRINT"This program allows you to enter a list of up to 16 words
. It will then produce a wordsquare of the sort found in puzzle books."
80 DIM C2(20),C(20),R2(20),R(20),D(20),S$(20,20),W$(20)
90 PRINT:PRINT"You will need a printer connected to your computer for this progr
am to work."
100 PRINT:PRINT"If you want to type in less than 16 words, you should press the
'*' key"
110 PRINT:PRINT"The maximum number of letters in any word is 15, but if you use
too many long words, the computer may not be able work out a square for you !"
120 REM ** shuffle the coordinates **
130 FOR X=1 TO 15:R(X)=X
140 C(X)=X:IF X<9 THEN D(X)=X
150 NEXT X
160 P=RND(-TIME)
170 FOR X=1 TO 15
180 R1=INT(RND(1)*15)+1:R2=INT(RND(1)*15)+1:X1=R(R1):X2=C(R2):R(R1)=R(X)
190 C(R2)=C(X):R(X)=X1:C(X)=X2:NEXT
200 REM ** enter the words **
```

```

210 CLS:PRINT"Now enter your list of words "
220 FOR W=1 TO 16
230 INPUT S$:IF S$="*" THEN W%=17:GOTO 300
240 IF W=1 THEN CLS:LOCATE 2,2:PRINTS$
250 W$(W)=S$
260 REM ** test the word **
270 S=0
280 GOSUB 890
290 IF S=1 THEN GOTO 230
300 NEXT
310 W=W-1
320 LOCATE 1,25:PRINT"Please wait for me to sort this out!";
330 REM ** sort words
340 GOSUB 1050
350 LOCATE 1,25:PRINT"
360 REM ** next word **
370 FOR W1=1 TO W
380 REM ** choose random directions **
390 FOR A=1 TO 8
400 R=INT(RND(1)*8)+1
410 X=D(R):D(R)=D(A):D(A)=X
420 NEXT
430 REM ** choose coordinate **
440 FOR R=1 TO 15
450 LOCATE 1,25:PRINT"Thinking!";
460 FOR C=1 TO 15
470 R1=R(R):C1=C(C)
480 REM ** direction **
490 FOR D=1 TO 8
500 REM ** check if it will fit **
510 FOR I=1 TO LEN(W$(W1))
520 ON D(D) GOSUB 950,960,970,980,1000,1010,1020,1030
530 REM ** check size **
540 IF R2>15 OR R2<1 OR C2>12 OR C2<1 THEN GOTO 640
550 S$=MID$(W$(S1),I,1)
560 IF S$(R2,C2)<>" " AND S$(R2,C2)<>S$ THEN GOTO 640
570 R1=R2:C1=C2:R2(I)=R2:C2(I)=C2
580 NEXT
590 REM ** fit into array **
600 FOR I=1 TO LEN(W$(W1))

```

B T I B
J O W C X
U G L S K
X L L A L
M W H A L

ERIL
FRED
LIZ
SUE
TIM
SID
TOM
ALF
AMY

H A R E T U P M O C R H Y J
P P E M B L A V H G X Q R U
U S O R Y M K L L O O I Y Q
P R E U F E I Z P S O V A H
S O L R T Y N K S G U B F Z G Y
R L A A B E A T Q G J B R L G V
V M C O Y R A X T F E M Q A O Z M V
D B A L R N X T F E H S C W R
I R E A S X T F P O T S S H T M
7 D F P O T S S H T M B

R S J D O M N H K
 X J G O W X C D O
 V N V T A R A I W
 L D D B R S H X C
 E M U V V R Q U Y

ARRAY
 GOSUB
 STOP
 PSET
 LINE
 PLAY
 GOTO
 ROM
 RAM
 CLS

```

610 S$(R2(I),C2(I))=MID$(W$(W1),I,1)
620 NEXT I
630 GOTO 730
640 NEXT
650 REM ** fit failed **
660 LOCATE 30,25:PRINT"Hard!";
670 NEXT
680 LOCATE 1,25:PRINT"
690 NEXT
700 PRINT:PRINT"Sorry I can't cope with this. Press the <Space Bar> to start again."
710 AA$=INKEY$:IF AA$<>" " THEN 710
720 RUN
730 NEXT W1
740 REM ** PRINT OUT TO PRINTER **
750 CLS:PRINT"What is the title of your wordsquare";:INPUT T$
760 LPRINT TAB(25);T$
770 LPRINT TAB(25);STRING$(LEN(T$),"=")
780 LPRINT
790 FOR R=1 TO 15
800 FOR C=1 TO 15
810 IF S$(R,C)="" THEN LPRINT CHR$((INT(RND(1)*26)+1)+64);" "; ELSE LPRINT S$(R,C);" ";
820 NEXT
830 LPRINT TAB(48);W$(R)
840 NEXT
850 LPRINT TAB(48);W$(R)
860 FOR X=1 TO 4:LPRINT:NEXT
870 RUN
880 REM ** test input **
890 IF LEN(W$(W))>15 THEN S=1:PRINT"This word is too long...try again!":RETURN
900 FOR A=1 TO LEN(W$(W))
910 S$=MID$(W$(W),A,1)
920 IF S$<"A" OR S$>"Z" THEN S=1:PRINT"capital letters only":RETURN
930 RETURN
940 REM ** horizontal and vertical **
950 R2=R1+1:RETURN
960 R2=R1-1:RETURN
970 C2=C1+1:RETURN
980 C2=C1-1:RETURN
990 REM ** diagonal **
1000 R2=R1+1:C2=C1+1:RETURN
1010 R2=R1+1:C2=C1-1:RETURN
1020 R2=R1-1:C2=C1+1:RETURN
1030 R2=R1-1:C2=C1-1:RETURN
1040 REM ** SORT **
1050 S=0
1060 FOR W1=1 TO W-1
1070 IF LEN(W$(W1))<LEN(W$(W1+1)) THEN S$=W$(W1):W$(W1)=W$(W1+1):W$(W1+1)=S$:S=1
1080 NEXT
1090 IF S=1 THEN GOTO 1050
1100 RETURN
  
```

It's easy to complain about advertisements. But which ones?

Every week millions of advertisements appear in print, on posters or in the cinema.

Most of them comply with the rules contained in the British Code of Advertising Practice.

But some of them break the rules and warrant your complaints.

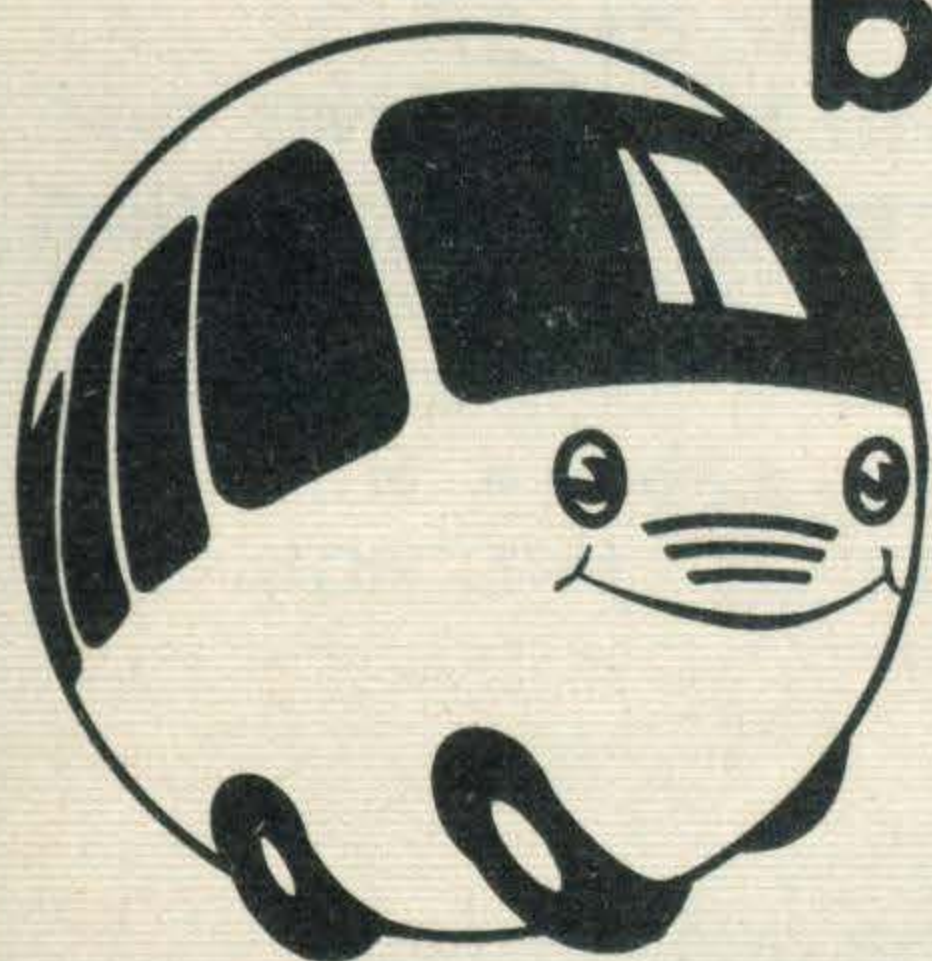
If you're not sure about which ones they are, however, drop us a line and we'll send you an abridged copy of the Advertising Code.

Then, if an advertisement bothers you, you'll be justified in bothering us. ✓

The Advertising Standards Authority.
If an advertisement is wrong, we're here to put it right.

ASA Ltd, Dept 2 Brook House, Torrington Place, London WC1E 7HN

This space is donated in the interests of high standards of advertising.



bubble bus software

Announces its first two MSX games

Hustler only £6.99

also available on Commodore 64 and Spectrum

and **Boardello** only £6.99

MORE GAMES

TO FOLLOW



Available from all good retail outlets or:

Dept. MSX
Bubble Bus Software
87 High Street
Tonbridge,
Kent TN9 1RX

£70 of MSX SOFTWARE - FREE
£12 MSX JOYSTICK - FREE
with every MSX computer

MITSUBISHI MSX MICRO
SANYO 112K MSX MICRO
TOSHIBA 112K MSX MICRO
TOSHIBA PLOTTER-PRINTER
TOSHIBA 105 CPS PRINTER

RING FOR OUR LATEST PRICES

THE BEST MSX SOFTWARE TAPES

DEMONSTRATOR-1 shows all the MSX features	£5.20
DEMONSTRATOR-2 watch it, then list and learn	£5.20
MSX GRAPHICS shows high resolution graphics	£5.20
TEACH TYPING speed up your keyboard skill	£5.20
JUNIOR MATHS teach kids to add and multiply	£5.20
MSX SMASHOUT addictive, maddening, multi-level	£5.20
VICIOUS VIPER eat men but not your own tail	£5.20
MSX OHELLO the classic game now on MSX	£5.20
EXPLODING ATOMS two player strategy game	£5.20
GATE CRASHER try the slalom - 9 levels	£5.20
HELP ME take a deep breath before answering	£5.20
CAVE ADVENTURE extremely complex adventure	£5.20

ALL 12 PROGRAMS ABOVE FREE WITH EVERY MSX PURCHASED FROM KNIGHTS

MAX BASIC TUTORIAL 3 tapes - 20 programs	£15
BUDGET & FORECASTING ACCOUNTS	£25
COMPLETE STOCK CONTROL SYSTEM	£25
MSX WDPRO WORD PROCESSOR	£26

KONAMI MSX CARTRIDGES

TRACK & FIELD 1 same as Konami's arcade	£13
TRACK & FIELD 2 superb graphics	£13
HYPER SPORTS diving, trampoline, springboard	£13
SUPER COBRA the best flying fighter game	£13
CIRCUS CHARLIE lions, fire, tightrope, trapeze	£13
TIME PILOT smart bombs, UFOs, two player action	£13
COMIC BAKERY tragic, funny, cruel, amazing	£13
MONKEY ACADEMY great arcade graphics	£15
ATHLETIC LAND ropes, fountains, pitfalls	£15
ANTARCTIC ADVENTURE the famous penguin game	£15

Dear Microfans,

We are acknowledged as experts in Japanese computers by TOSHIBA, SANYO AND MITSUBISHI who all buy KNIGHTS programs. We are also enthusiasts and will always help you - we *guarantee* our deals are unbeatable. Ring or write for full details.

KNIGHTS RUSH SERVICE - ring us with your Access or Visa number as we can usually deliver anywhere in the UK next day. In the last ten years we have sold thousands of Japanese micros worldwide and have never charged for a single repair. Ring us for personal service.

happy computing
Graham Knight and Neil Hunter

U.K. customers - delivery is free but add Maggie's 15% VAT.
EXPORT customers only - no VAT just add £10 for freight anywhere.

KNIGHTS TV & COMPUTERS (est. 1937)
108 ROSEMOUNT PLACE, ABERDEEN
PHONE 0224 630526 TELEX 739169

We've decided you need sorting out, and this is just the program to do it.

If you are writing a program which stores DATA in an ARRAY, such as the

Address Book program in this issue, you will probably want to extend your program to sort the contents of the array into alphabetical order. There are several ways of doing this. Probably the simplest method to program is the 'bubble sort' which, unlike this shell sort, actually sorts the contents of the array into alphabetical order. Although this would seem to be the logical way

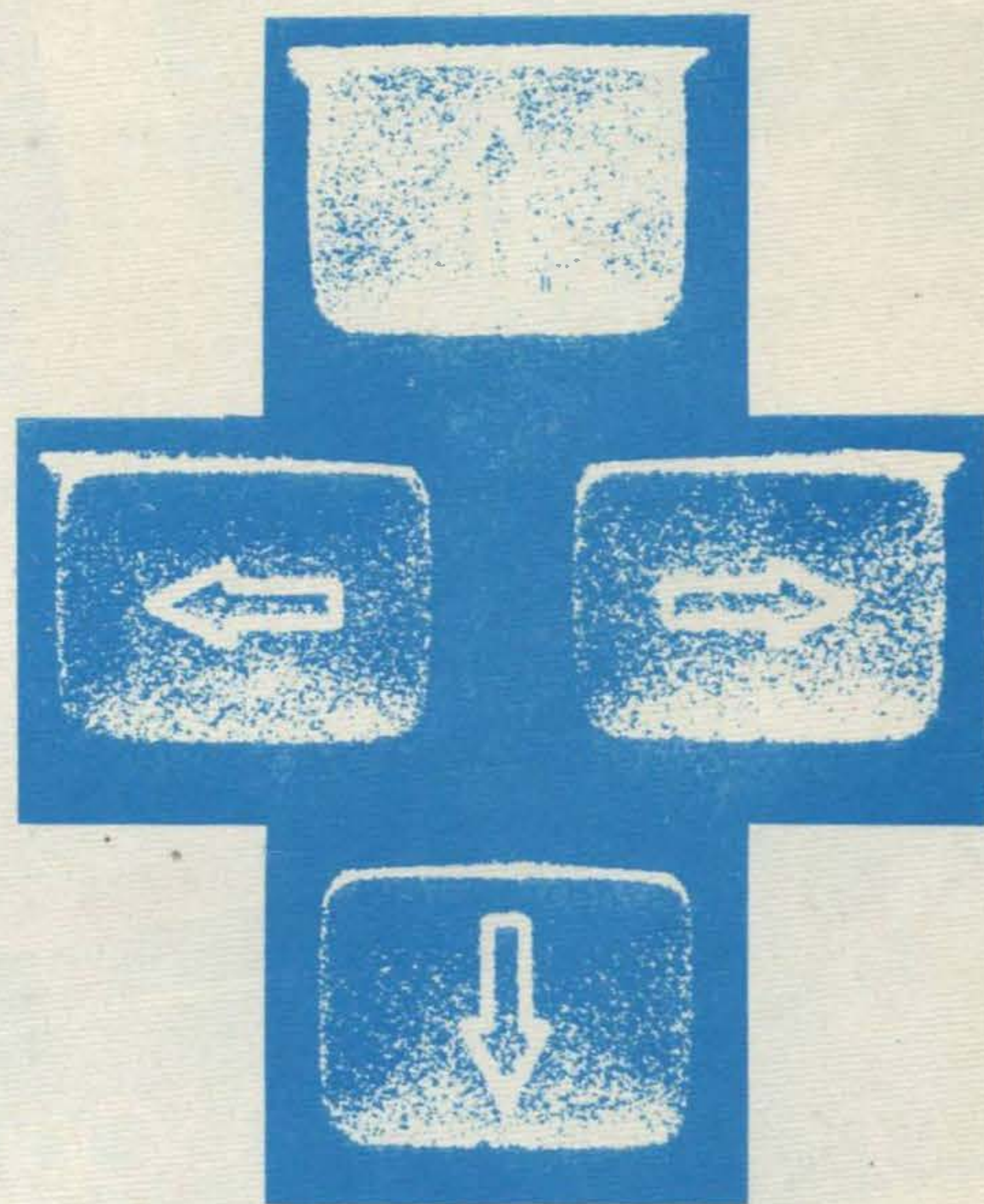
of doing it, it is terribly slow if you are sorting more than 20 words.

This program does not actually sort the contents of the array A\$(x) which is used to hold the words, but instead it returns the array SP(X) as a pointer to the string array A\$. This will probably sound very complicated, but using this method, it isn't necessary to actually move the records around and this dramatically increases the speed. Note the actual time taken to sort an

array with N elements is proportional to $2 * \text{LOG}(N) * N$.

The actual shell sort routine lies from line 430 onwards and if you examine the rest of the program carefully, you should see how the sort subroutine is called. If you are writing a database program, you can type in the subroutine at the end of your program and call it from the main program when you want to sort your DATA files into alphabetic order.

Shell Sort



Program Breakdown

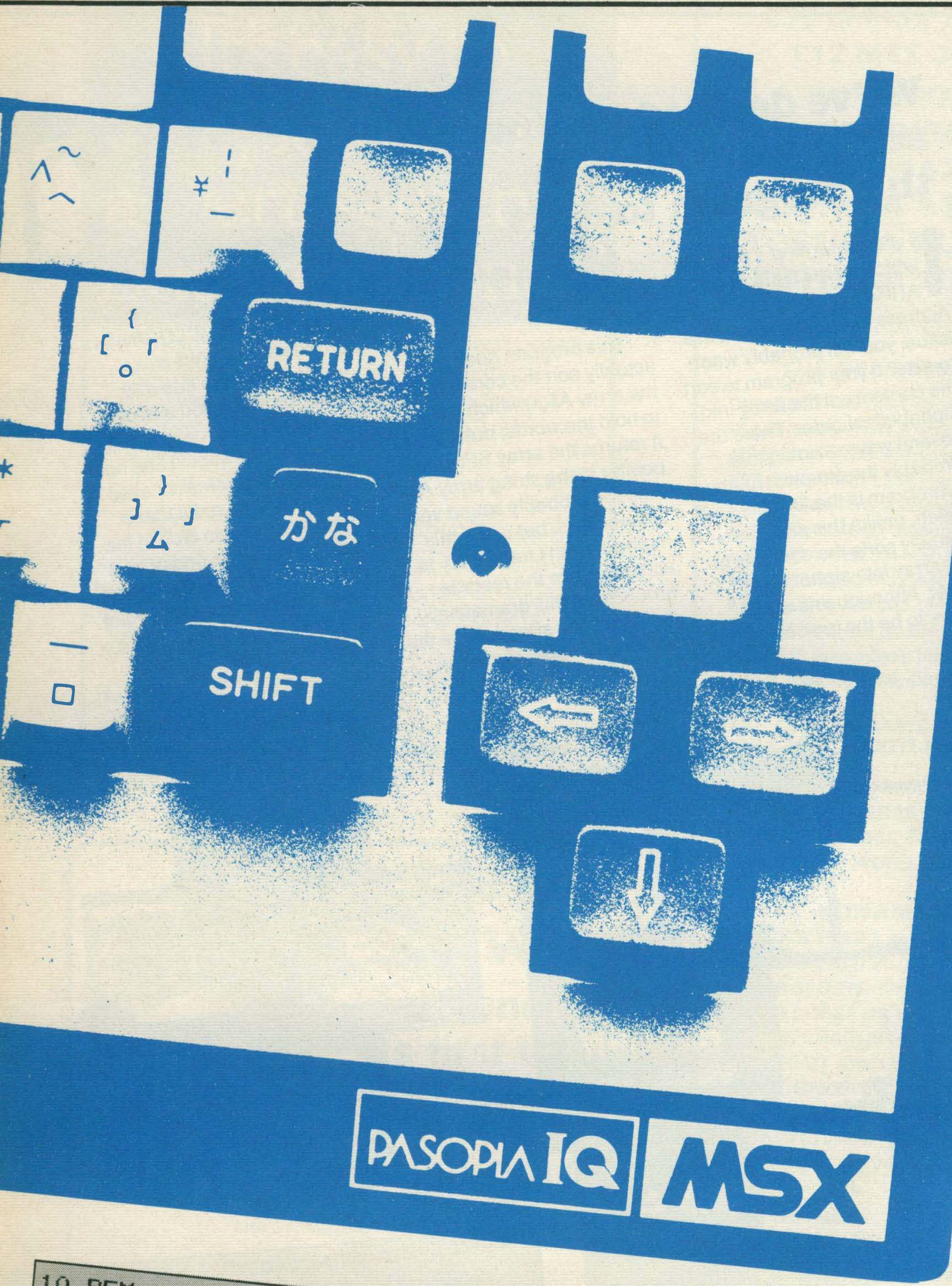
Lines	
10-60	titles
70	clear string space
80	DIMension array to be sorted
90-240	type in DATA into the array A\$(x)
250-330	ready to sort array
350-420	print output
430-710	** SHELL SORT **routine

Major Variables

A\$(x)	array to be sorted
SP(x)	pointer to sorted array
S	check for end of sort
SK, SP, SN, SI	used in sorting
SS	number of steps for sorting
SB	block size
CT	number of words entered

Notes

1. If you want to print the output on the printer change the PRINT statement in line 350 to LPRINT.
2. The DIM statement 80 allows only 25 to be entered. Changing this will allow you more words.



SAMPLE RUN

*
 ALBERT
 ANNE
 BILL
 CLARE
 ELIZABETH
 ELSIE
 ERIC
 FRED
 HELEN
 JAMES
 KATE
 KEVIN
 LORA
 LYNN
 MARGARET
 MARY
 MIKE
 PAUL
 PETE
 PETER
 RAYMOND
 SIDNEY
 STEVE
 SUSAN
 TIM

```

10 REM ** shell sort routine for MSX Computers **
20 KEY OFF
30 SCREEN 0
40 COLOR 11,8,9
50 LOCATE 5,2:PRINT"Shell Sort   Demonstration"
60 LOCATE 5,3:PRINT"=====
70 CLEAR 1000
80 DIM A$(255):EN=255:CT=0
90 REM ** set array A$ to maximum **
100 FOR I=0 TO 255
110 A$(I)=""
120 NEXT I
130 PRINT:PRINT:PRINT:PRINT"Please enter the names into the array"
140 PRINT:PRINT"when you are ready to sort the array type '*' and press <RETURN>"

```

```

150 PRINT:PRINT
160 REM ** Input data into the arrays **
170 PRINT"Press the <Space Bar> to start."
180 A$=INKEY$:IF A$<>" " THEN 180
190 CLS:PRINT"There is room for ";EN;" more names"
200 INPUT A$(CT)
210 IF A$(CT)="*" THEN 260
220 CT=CT+1
230 IF CT>254 THEN 260
240 EN=255-CT:GOTO 190
250 REM ** finished entry of data...ready to sort array **
260 SN=CT-1
270 GOSUB 440
280 REM ** line output **
290 PRINT"Hit any key for list."
300 K$=INKEY$:IF K$="" THEN 300
310 LP=0:SL=18
320 FOR P=LP TO LP+SL
330 IF P>CT THEN END
340 REM ** the array sp(x) is used as a pointer to the contents of a$(x) **
350 PRINT A$(SP(P))
360 NEXT P
370 PRINT:PRINT"** Hit any key to continue **"
380 PRINT:PRINT"** Press '*' KEY TO END **"
390 K$=INKEY$:IF K$="" THEN 390
400 IF K$="*" THEN END
410 LP=LP+SL+1
420 GOTO 320
430 REM ** Shell Sort Routine **
440 SS=(INT(LOG(SN)/LOG(2))+1)
450 SN=2^SS
460 DIM SP(SN),SQ(SN)
470 FOR S1=0 TO SN-1
480 SP(S1)=S1
490 NEXT S1
500 SB=1
510 S=S+1
520 IF S>SS THEN RETURN
530 CLS:LOCATE 6,2:PRINT"Sorting : blocksize = ";SB
540 S=S+1
550 SJ=SN/2:S1=0:SK=0
560 SL=SB+S1:SM=SB+SJ
570 IF (SJ>=SM) AND (S1>=SL) THEN 650
580 IF (SJ>=SM) THEN 610
590 IF (S1>=SL) THEN 630
600 IF A$(SP(S1))>A$(SP(SJ)) THEN 630
610 SQ(SK)=SP(S1)
620 SK=SK+1:S1=S1+1:GOTO 570
630 SQ(SK)=SP(SJ)
640 SK=SK+1:SJ=SJ+1:GOTO 570
650 IF SK>SN-1 THEN 670
660 SM=SM+SB:SL=SL+SB:GOTO 570
670 FOR S1=0 TO SN-1
680 SP(S1)=SQ(S1)
690 NEXT S1
700 SB=SB*2:SP=SP+1
710 GOTO 520

```

Auto Racer

An arcade style game for MSX machines, which shows just what's possible in a short program.

In this short program, I have attempted to illustrate how it is possible to create an enjoyable arcade game using only a few lines due to the sophisticated commands available in MSX BASIC. You are a racing driver and must try to drive as far as possible without colliding with the other drivers in the race, driving off the track or hitting the fences in the middle of the track. You can move left or

right using the cursor left and right keys and should be able to dodge between the fences, with practice. Accelerator and Brake are applied using the

cursor up and down keys respectively.

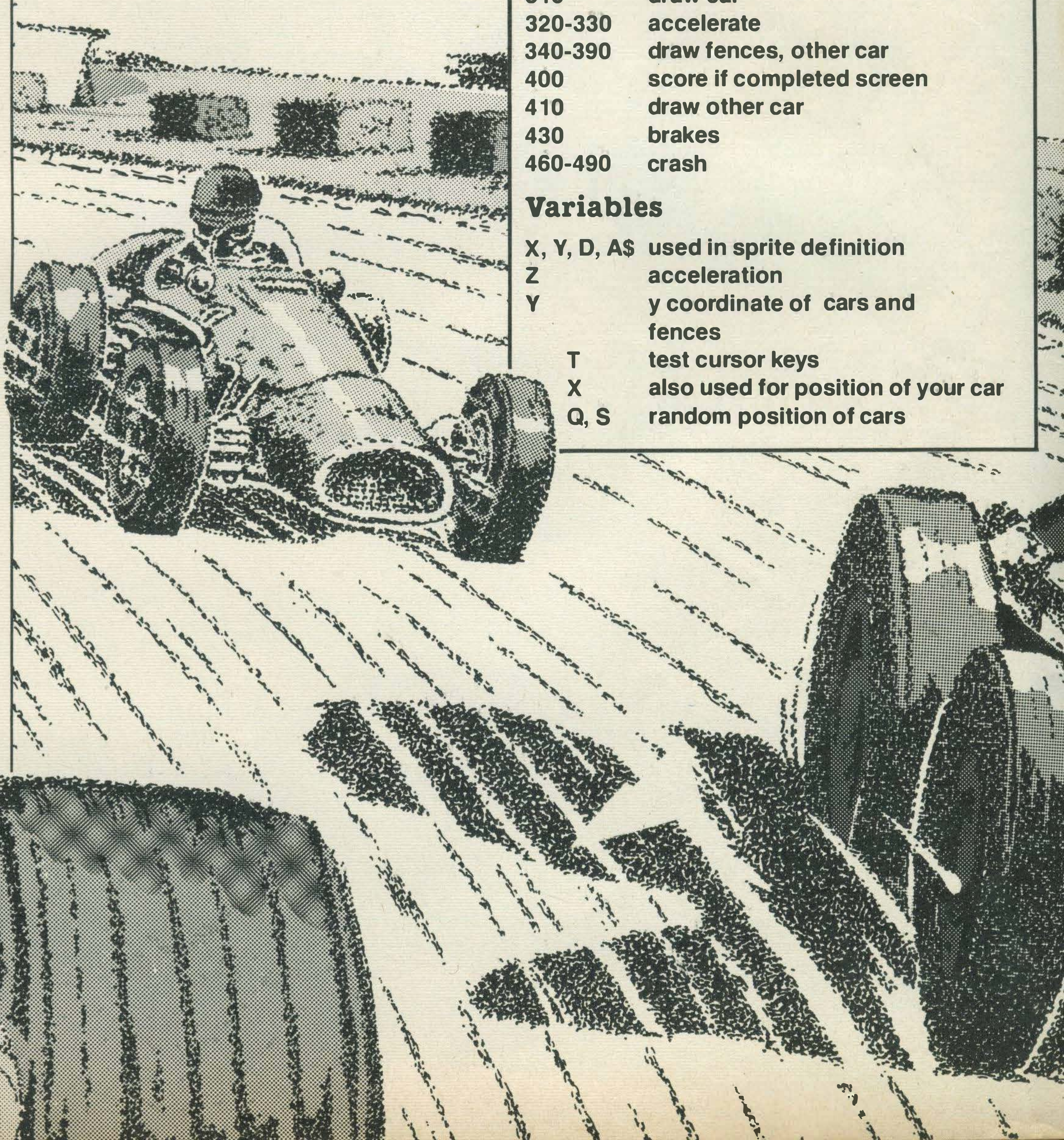
The further you drive before colliding, the higher your score!

Program Breakdown

Lines	Description
20	set speed at start
30-50	set screen/ messages off/colour
60-180	sprite definition
190-220	draw graphics
230-240	sprite collision
260	random seed
270	random position of other cars
280	test cursor keys
290-300	move car
310	draw car
320-330	accelerate
340-390	draw fences, other car
400	score if completed screen
410	draw other car
430	brakes
460-490	crash

Variables

X, Y, D, A\$	used in sprite definition
Z	acceleration
Y	y coordinate of cars and fences
T	test cursor keys
X	also used for position of your car
Q, S	random position of cars



```
10 REM ** AUTO RACER ... a car race game for MSX **
20 Z=1
30 KEY OFF
40 COLOR 15,4,7
50 SCREEN 2,2
60 DATA 3,7,14,109,253,236,111,12
70 DATA 12,108,239,252,239,103,7,3
80 DATA 192,224,112,54,255,183,246,48
90 DATA 48,54,247,63,247,230,224,192
100 FOR Y=1 TO 2
110 A$=""
120 FOR X=1 TO 32:READ D
130 A$=A$+CHR$(D)
140 NEXT X:SPRITE$(Y)=A$:NEXT Y
150 DATA 32,32,32,32,32,32,32,32,32,32
160 DATA 32,32,32,32,32,32,32,32,32,32
170 DATA 0,0,0,0,0,0,0,0,0,0
180 DATA 0,0,0,0,0,0,0,0,0,0
190 LINE (85,0)-(20,191),2
200 PAINT (20,20),2
210 LINE (185,0)-(235,191),2
220 PAINT (200,20),2
230 SPRITE ON
240 ON SPRITE GOSUB 460
250 X=134
260 S=RND(-TIME)
270 S=INT(RND(1)*30)+70
280 T=STICK(0)
290 IF T=3 THEN X=X+2:IF X>220 THEN GOSUB 460
300 IF T=7 THEN X=X-2:IF X<30 THEN GOSUB 460
310 PUT SPRITE 1,(X,170),13,1
320 IF T=1 THEN Z=Z+.1
330 Y=Y+Z
```

```
340 PUT SPRITE 2,(125,Y),15,2
350 PUT SPRITE 3,(125,Y+50),15,2
360 PUT SPRITE 4,(125,Y+100),15,2
370 PUT SPRITE 5,(125,Y+150),15,2
380 PUT SPRITE 6,(125,Y+200),15,2
390 PUT SPRITE 7,(S+INT(RND(1)*4)-2,Q),14,1
400 IF Y>255 THEN Y=0:S%=S%+1
410 Q=Q-INT(RND(1)*10)+1
420 PUT SPRITE 8,(S+80+INT(RND(1)*4)-2,Q+120),14,1
430 IF T=5 THEN Z=Z-.2:IF Z<0 THEN Z=0
440 GOTO 280
450 GOTO 450
460 SPRITE OFF
470 SCREEN 0
480 LOCATE 10,4:PRINT"C R A S H"
490 LOCATE 8,10:PRINT"You scored :- ";S%
```



Possible Improvements

1. Adding sound effects should prove an interesting exercise.
2. Try adding extra cars in the race (sprite 1).
3. Change to joystick by replacing STICK (0) with STICK (1) in line 280.

MSX

Don't those aliens ever give up? Here they come again to test your dodging ability.



Dodger

The aliens are on the attack again. Can you dodge their bombs as wave after wave of them pour down from above onto your spaceship? Each successive wave of bombs becomes faster and more furious. The longer you survive, the higher your score becomes. Good luck, Commander.

Cursor left and cursor right move you left and right respectively. When writing this game, I wanted to see just how few lines were needed to make an enjoyable game using the sophisticated sprite handling facilities of MSX BASIC. You could try adding a background display to the program, adding more sound effects or changing the shape of the sprites.

Program Breakdown

Lines	
20	initialise
30-50	turn key definitions off, select colour and screen 2
60-130	data for sprites
140-160	define sprites
170-190	decide on coordinates
200-210	detect sprite collision
220-430	control loop
220	poll cursor keys
230-240	move left/right
250-400	move sprites
410	change coordinates
420	increase speed of bombs
440-500	lose game

Variables

S%	score
X, X1, Y	coordinates of bombs
Z%	speed of bombs
A\$	test keyboard
D	data for sprites
X%, Y%	coordinates of spaceship

```
10 REM ** astrobomb dodger .. an arcade game for MSX **
20 Z%=2:S%=0
30 KEY OFF
40 COLOR 15,4,5
50 SCREEN 2,2
60 DATA 1,3,3,3,15,25,56,58
70 DATA 0,0,0,0,0,0,0,0
80 DATA 192,224,224,224,248,204,14,174
90 DATA 0,0,0,0,0,0,0,0
100 DATA 24,24,24,24,24,24,24,24
110 DATA 126,125,126,126,126,60,24,0
120 DATA 0,0,0,0,0,0,0,0
130 DATA 0,0,0,0,0,0,0,0
140 FOR Y=1 TO 2
150 A$="":FOR X=1 TO 32:READ D:A$=A$+CHR$(D):NEXT X:SPRITE$(Y)=A$
160 NEXT Y
170 X%=125:Y%=183
180 X=INT(RND(1)*64)+1
190 X1=INT(RND(1)*64)+1
200 SPRITE ON
210 ON SPRITE GOSUB 440
220 T=STICK(0)
230 IF T=3 THEN X%=X%+2:IF X%>241 THEN X%=241
240 IF T=7 THEN X%=X%-2:IF X%<0 THEN X%=0
250 PUT SPRITE 1,(X%,Y%),10,1
260 PUT SPRITE 2,(X,Y),11,2
270 PUT SPRITE 3,(X1,Y+10),11,2
280 PUT SPRITE 4,(X1+50,Y+90),11,2
290 PUT SPRITE 5,(X+50,Y+110),11,2
300 PUT SPRITE 6,(X1+121,Y+141),11,2
310 PUT SPRITE 7,(X+95,Y+123),11,2
320 PUT SPRITE 8,(X+194,Y+165),11,2
330 PUT SPRITE 10,(X+155,Y+34),11,2
340 PUT SPRITE 9,(X+110,Y+75),11,2
350 PUT SPRITE 11,(X+130,Y+129),11,2
360 PUT SPRITE 12,(X1+100,Y+19),11,2
370 PUT SPRITE 13,(X+138,Y+59),11,2
380 PUT SPRITE 14,(X+28,Y+59),11,2
390 PUT SPRITE 15,(X+172,Y+59),11,2
400 PUT SPRITE 16,(X-16,Y+191),11,2
410 Y=Y+Z%
420 IF Y>255 THEN Z%=Z%+1:Y=0:S%=S%+1
430 GOTO 220
440 SCREEN 0
450 LOCATE 10,2:PRINT"C r a s h ! ! !"
460 PLAY"cdecde"
470 LOCATE 10,10:PRINT"You scored :- ";S%
480 LOCATE 1,22:PRINT"Press <Space Bar> for another game."
490 A$=INKEY$:IF A$<>" " THEN 490
500 RUN
```

Clock

Digital

If you want to know the **TIME**, ask an MSX computer. Provided you've typed in our Digital Clock program, of course.

Variables Used

HR	hours
MI	minutes
SE	seconds
AS	erase character
T	time
X, Y	used to locate graphics blocks

This program was written to illustrate how the TIME facility available in MSX BASIC can be used to drive a clock. When the program is run, you will be asked to input the current time. The computer will then draw a digital clock on the screen and will continuously display the time. As this is driven by the TIME facility, it should prove to be very accurate!

Although SCREEN 0 doesn't offer the superb graphic facilities of SCREEN 2, this short program shows that it is possible to produce a good display using the block graphics available in the character set provided.

Improvements

1. It should prove an interesting exercise to try to draw an analogue clock. This would need to be done in SCREEN 2, using the variables HR, MI, SE to test hours, minutes and seconds.
2. To convert it to a 24-hour clock, change the number 12 to 24 in line 300.
3. Try adding an alarm facility to the program.

Program Breakdown

Lines	
10-40	initialise screen
50	call routine to set time
60-210	draw graphics using character set
230	set time variable to zero
240-320	update variables HR, MI, SE and print them on screen
340-420	set time

```
10 TIME=0
20 KEY OFF
30 COLOR 1,15,15
40 SCREEN 0
50 GOSUB 340
60 LOCATE 6,10:PRINTSTRING$(23,CHR$(255))
70 FOR Y=10 TO 20:LOCATE 6,Y
80 PRINTCHR$(255)
90 LOCATE 29,Y
100 PRINTCHR$(255)
110 NEXT Y
120 LOCATE 7,18:PRINTSTRING$(22,CHR$(255))
130 LOCATE 6,19:PRINTSTRING$(25,CHR$(204))
140 LOCATE 5,19:PRINTSTRING$(26,CHR$(204))
150 FOR Y=11 TO 17
160 LOCATE 14,Y:PRINTCHR$(222)
170 LOCATE 21,Y:PRINTCHR$(222)
180 NEXT
190 LOCATE 8,12:PRINT"Hours"
200 LOCATE 16,12:PRINT"Mins"
210 LOCATE 23,12:PRINT"Secs"
220 A$="  "
230 TIME=0
240 T=TIME
250 IF TIME>60 THEN TIME=0:SE=SE+1
260 IF SE>59 THEN MI=MI+1:SE=0:LOCATE 24,14:PRINT"  "
270 LOCATE 24,14:PRINT SE
280 IF MI>59 THEN HR=HR+1:MI=0:LOCATE 16,14:PRINT"  "
290 LOCATE 16,14:PRINTMI
300 IF HR>12 THEN HR=1:MI=0:LOCATE 9,14:PRINT"  "
310 LOCATE 9,14:PRINT HR
320 GOTO 250
330 END
340 CLS:PRINT"What is the current time"
350 LOCATE 10,4:PRINT"Hours ";
360 INPUT HR
370 LOCATE 10,8:PRINT"Mins ";
380 INPUT MI
390 LOCATE 10,12:PRINT"Secs "
400 INPUT SE
410 CLS
420 RETURN
```

Alien Planes

This short games listing shows just what can be achieved with the graphics commands of MSX BASIC in only a few lines. Your task is to prevent the alien planes from landing by moving your sights over the target plane and firing. Each time that you shoot a plane down, another one will be sent at even higher speed to test your reactions.

Use the cursor keys to move your sights – they can only be moved vertically or horizontally! The space bar is the fire button.

Program Breakdown

Lines	
10	set variables
20-40	turn off messages, select colours and hires screen
50-140	data for sprites
150-240	sprite definitions
250-330	draw screen background
340	set coordinates for gun sights
350-360	detect sprite collision
370-460	main control loop
370	test cursor keys
380-410	move sights
420-450	move sprites
470-520	shoot down plane
540-590	plane lands . . . lose game

Gunnery officers to the fore, with this short shoot-em-down program for MSX computers.



THERAPY

```
10 Z%=2:S%=0:X=150:Y=180
20 KEY OFF
30 COLOR 4,15,5
40 SCREEN 2,2
50 DATA 0,0,15,24,127,255,24,24
60 DATA 0,0,0,0,0,0,0,0
70 DATA 0,0,192,48,252,255,24,24
80 DATA 0,0,0,0,0,0,0,0
90 DATA 0,128,143,241,255,127,3,2
100 DATA 0,0,0,0,0,0,0,0
110 DATA 0,0,0,8,232,252,8,8
120 DATA 0,0,0,0,0,0,0,0
130 DATA 32,32,32,254,32,32,32,0
140 DATA 255,1,1,1,1,1,1,1
150 FOR Y=1 TO 2
160 A$=""
170 FOR X=1 TO 32
180 READ D
190 A$=A$+CHR$(D)
200 NEXT
210 SPRITE$(Y)=A$
220 NEXT Y
230 FOR X=1 TO 8:READ D:S$=S$+CHR$(D):NEXT X
240 SPRITE$(3)=S$
250 LINE (248,191)-(255,189),7,BF
260 LINE (0,191)-(6,189),7,BF
270 LINE (2,189)-(4,184),6,BF
280 LINE (250,189)-(252,184),6,BF
290 LINE (6,191)-(127,100),3
300 LINE (250,191)-(127,100),3
310 PAINT (50,189),3
320 CIRCLE (190,40),20,11
330 PAINT (190,40),11
340 X=120:Y=180
350 SPRITE ON
360 ON SPRITEGOSUB 470
370 T=STICK(0)
380 IF T=1 THEN Y=Y-2:IF Y<1 THEN Y=1
390 IF T=5 THEN Y=Y+2:IF Y>191 THEN Y=191
400 IF T=3 THEN X=X+2:IF X>255 THEN X=255
410 IF T=7 THEN X=X-2:IF X<0 THEN X=55
420 PUT SPRITE 3,(X,Y),1,3
430 X%=X%+Z%:IF X%>255 THEN X%=0:Y%=Y%+10
440 PUT SPRITE 1,(X%,Y%),13,2
450 IF Y%>180 THEN GOSUB 540
460 GOTO 370
470 SPRITE OFF
480 AA$=INKEY$:IF AA$<>" " THEN BEEP:SPRITE ON:RETURN
490 BEEP
500 S%=S%+1:PLAY"cdedec":X=100:Y=180:Z%=Z%+1
510 X%=0:Y%=0
520 SPRITE ON
530 RETURN
540 SCREEN 0:A$=""
550 LOCATE 10,10:PRINT"They landed !"
560 LOCATE 10,15:PRINT"You scored ";S%
570 LOCATE 1,22:PRINT"Press the space bar for another game";
580 A$=INKEY$:IF A$<>" " THEN 580
590 RUN
```

Variables

S%	score
X, Y	coordinates of sights
X%, Y%	coordinates of plane
Z%	speed of plane
AAS	test for space bar
T	test for cursor keys
D	data for sprites
A\$	sprite definition

MSX

Address Book

Here's a utility program for MSX machines that shows you how to store data files, in this case address data.

This short program was written to illustrate how to use two-dimensional arrays to hold data. When run, it displays a menu which asks whether you want to load a data file, save a data file, enter data, query the file or quit the program. Pressing the appropriate key will take you to the subroutine chosen.

After this has been completed, the computer will return you to the main menu.

This program will allow you to store the names, addresses, telephone numbers and birthdays for up to 200 people on tape, so you won't have any excuses for forgetting Uncle Bill's birthday!

Program Breakdown

Lines	turns off messages
30	titles
40-80	initialise variables
90-110	options
120-190	select option
200-250	load data file
270-360	save data file onto tape
370-460	enter data
470-640	query file
650-710	print data
720-810	

Variables Used

F	number of person in file
B%	flag
x, y	elements of array
A\$(x, y)	holds data
B\$, A\$	keyboard input

```

10 REM ** Address Book for MSX computers **
20 REM ** version 1 December 1984 **
30 KEY OFF
40 F=1
50 SCREEN 0
60 COLOR 1,14,7
70 LOCATE 6,2:PRINT"A d d r e s s   B o o k"
80 LOCATE 6,3:PRINT"=====
90 CLEAR 1000
100 DIM A$(200,4)
110 F=1
120 LOCATE 10,5:PRINT"Do you want :-"
130 B%=0
140 LOCATE 1,10:PRINT"1. To load a data file."
150 LOCATE 1,12:PRINT"2. To save a data file."
160 LOCATE 1,14:PRINT"3. To enter data into the file."
170 LOCATE 1,16:PRINT"4. To query a data file."
180 LOCATE 1,18:PRINT"5. To end the program."
190 A$=INKEY$:IF A$="1" THEN GOSUB 270
200 IF A$="2" THEN GOSUB 370
210 IF A$="3" THEN GOSUB 480
220 IF A$="4" THEN GOSUB 650
230 IF A$="5" THEN CLS:PRINT"Goodbye. Thank you for using me.":END
240 IF B%=1 THEN CLS:GOTO 120
250 GOTO 190
260 CLS:GOTO 120
270 CLS:PRINT"Place tape in recorder and press play";

```

THERAPY

```
280 OPEN "cas:tel" FOR INPUT AS #1
290 FOR X=1 TO 200
300 FOR Y=1 TO 4
310 INPUT #1,A$(X,Y)
320 NEXT Y
330 NEXT X
340 B%=1
350 CLOSE
360 RETURN
370 CLS:PRINT"Place tape ready to record. "
380 OPEN "cas:tel" FOR OUTPUT AS #1
390 FOR X=1 TO 200
400 FOR Y=1 TO 4
410 PRINT #1,A$(X,Y)
420 NEXT Y
430 NEXT X
440 B%=1
450 CLOSE
460 RETURN
470 REM ** enter data into the file **
480 CLS:LOCATE 1,2:PRINT"Enter name of person number ";F
490 INPUT A$(F,1)
500 B%=1
510 LOCATE 1,4:PRINT"Enter their telephone number "
520 INPUT A$(F,2)
530 LOCATE 1,6:PRINT"Enter their address "
540 INPUT A$(F,3)
```

```
550 LOCATE 1,8:PRINT"Enter their birthday "
560 INPUT A$(F,4)
570 LOCATE 1,10:PRINT"Is this correct <Y/N>?"
580 B$=INKEY$:IF B$="n" OR B$="N" THEN 480
590 IF B$="y" OR B$="Y" THEN 600 ELSE 580
600 F=F+1
610 LOCATE 1,12:PRINT"Do you want to enter any more names?"
620 A$=INKEY$:IF A$="n" OR A$="N" THEN RETURN
630 IF A$="y" OR A$="Y" THEN 480
640 GOTO 620
650 CLS:LOCATE 1,2:PRINT"Who do you want to find ":PRINT:PRINT
660 INPUT B$
670 FOR X=1 TO F
680 IF LEFT$(B$,4)=LEFT$(A$(X,1),4) THEN GOSUB 720
690 NEXT
700 B%=1
710 RETURN
720 CLS:LOCATE 1,2:PRINTA$(X,1)
730 LOCATE 1,4:PRINT"Telephone number "
740 LOCATE 1,6:PRINTA$(X,2)
750 LOCATE 1,8:PRINT"Address "
760 LOCATE 1,10:PRINTA$(X,3)
770 LOCATE 1,12:PRINT"Birthday"
780 LOCATE 1,14:PRINTA$(X,4)
790 LOCATE 1,20:PRINT"Press <Space Bar> to continue"
800 A$=INKEY$:IF A$<>" " THEN 800
810 RETURN
```

Looking for the latest software? So are we. We want software houses to tell us about their latest titles to add to our current list below.

A

ACTIVISION,, 15 Harley House, Marylebone Rd, London, NW1:

Beamrider – game
Decathlon – game
Pitfall 11 – game
River Raid – game
Space Shuttle – game
Zenji – game – £11.99 each

A&F SOFTWARE, 8 Camalside Ind. Estate, Woodbine St East, Rochdale:

Chuckie Egg – game – £7.90

ALLIGATA, 1 Orange St, Sheffield, S1 4DW:

Blogger – game – 7.95
Contract Bridge – game – £9.95
Disc Warrior – game – £7.95

AMPALSOFT, Ampal Computer Services, Woodbridge Rd, Derby Green, Surrey:

Fun Words – educational (four progs) – £9.95

ANIROG, 8 The High St, Horley, Surrey, RH6 7AY:

Flightpath 737 – game – £7.95

ARTIC, Brandesburton, Driffield, North Humberside, YO25 8RL:

Mr Wongs Loopy Laundry – game – £6.95

A.S.K., London House, 68 Upper Richmond Rd, London, SW15 2RP:

Number Painter – educational, 5-14 years.

B

BUBBLE BUS, 87 High St, Tonbridge, Kent, TN9 1RX:

Hustler – game – £6.99

C

CDS MICROSYSTEMS, Silver House, Silver St, Doncaster, Sth Yorkshire:

French Is Fun – educational
German Is Fun – educational
Italian Is Fun – educational
Spanish Is Fun – educational – £7.95 ea.

COMPUTER MATES, PO Box 2, Stockbridge, Hampshire:

Word Processor – business – £49
Cards System – business – £49
Cash Accounts System – business – £99
Double Entry Accounts – business – £99
Word Processing & Cards System combined – business – £75

CRL, CRL House, 9 Kings Yard, London, E15 2HD:

Glug Glug – gme – £7.95
War Of The Worlds – game – £7.95

D

D.K. TRONICS, Saffon Walden, Essex, CB11 3AQ:

Minder – game
Popeye – game
Hagar The Horrible – game – £5.95 ea.

G

G.S.T./ELECTRIC SOFTWARE, 8 Green St, Willingham:

Buzz Off – game – £8.95
Shark Hunter – game – £9.95

Le Mans – game – £9.95
Norseman – game – £8.95
Backgammon – game – £9.95

H

HEWSON, 56B Milton Trading Estate, Milton, Abingdon, Essex, OX1 4RX:

Bazam – game

HI-SOFT, 180 High St Nth, Dunstable, Beds, LU6 1AT:

Devpac – utility – £19.95
Pascal Compiler – utility – £29.95

K

KEMP, 43 Muswill Hill, London, N1D 3PN:

Stock Control – business – £34.95

KONAMI, Television House, 269 Fieldend Rd, Eastcote, Middlesex:

Super Cobra – game
Athletic Land – game
Circus Charlie – game
Antarctic Adventure – game
Cosmic Bakery – game
Monkey Academy – game
Time Pilot – game
Hyper Olympic 1 – game
Hyper Olympic 2 – game
Hyper Sports 1 – game – £14.99 ea.

KUMA, 12 Horseshoe Park, Pangbourne, RG7 7JW:

Binary Land – game – £8.95
Eric & The Floaters – game – £8.95
Hyper Vipers – game – £7.95
Spooks & Ladders – game – £6.95
Holdfast – game – £5.95
Home Budget – business – £14.95
WDPRO-Wordprocessor – business – £29.95
Database – business – £19.95
Starting With The MSX – utility – £5.95
The MSX Red Book – utility – £8.95

Star Avenger – game – £8.95
Super Chess – game – £8.95
Stop The Express – game – £6.95
Ninja – game – £6.95
Mean Streets – game – £6.95
Colour Fantasia – game – £9.95
Logs Turtle Graphics – £19.95
Spread Sheet – £29.95
WDPRO – business
Spread Sheet – business disc
Zen Assembler – business disc
Data Base – business disc
Kuma Forth – disc – £39.95ea.

L

LLAMASOFT, 429 Mt Pleasant, Tadley, Hants:

Grid Runner – game – £5

M

MASTERTRONIC, Park Lorne, Park Rd, London NW8 7JL:

Space Walk – game
Magic Carpet – game – £1.99 each

MICRO-AID, 25 Fore St, Praze Camborne, Cornwall, TR14 0JX:

Cashbook – business – £14.95
Memo-Calc – business – £14.95
Payroll – business – £29.95

MR MICRO, 69 Partington Lane, Swinton, Manchester, M27 3AL:

Punchy – game
Humphrey – game

Cubit – game
Crazy Golf – game
Zakil Wood – game – £6.90 ea.

MIRRORSOFT, Holborn Circus, London ECP 1DQ:

737 Flight Simulator – game – £9.95

MORWOOD, Maple Walk, Bexhill, East Sussex:

Cannon Fighter – game
Panic Junction – game
Super Mind – game
Super Maze – game
Super Puzzle – game – £6.95 ea.
Intro. To Numbers – educational
Calculation 1 – educational
Calculation 2 – educational
Memory – educational
Reasoning – educational
Reflexes – educational – £14.95 ea.

P

PREMIER MICROSYSTEMS, 208 Croydon Rd, London, SE20 7YX:

Wordmate – business – £24.95
Cribbage – game – £6.95
MSXMON – utility – £9.95
Home Accounts – business – £7.95
Spelling – educational – £6.95
Plant – educational – £6.95
French Verbs – educational – £7.95
Gambling Pack – game – £6.95
Maths Games Pack – educational – £7.95
House Of Horrors – game – £6.95
Cabins of Doom – game – £6.95
Dragon Tower – game – £7.95
Dungeon of Death – game – £7.95
Adventure Plus – game – £7.95

PSS, 452 Stoney Stanton Rd, Coventry, CV6 5DG:

Champ – utility – £12.95
Maxima – game – £7.95
Time Bandits – game – £7.95*
Les Flics – game – £7.95

Q

QUICKSILVA, Palmeston Hse, Palmeston Rd, Southampton, SO1 1LL:

Games Designer – utility – £9.95
Ant Attack – game
The Snowman – game
Fred – game
Bugaboo – game – £7.95 ea.

S

SOFTCAT, Woodbridge Rd, Derby Green, Blackwater, Camberley, Surrey:

Challenge My Bluff – educational – £8.95
Kriss Kross Kwiz – educational (twin pack) – £8.95

T

TOSHIBA, Toshiba Hse, Frimley Rd, Frimley, Camberley, Surrey:

3D Golf – game
Polar Star – game
Pinball – game
Pyramid Warp – game
Battleship Clapton 11 – game – £7.95ea.

TASMAN, Springfield House, Hyde Terrace, Leeds, LS2 9LN:

Tasword – business – £13.90
Tasprint – utility – £9.90

In the last issue we created the basic framework of an adventure game but without a SAVE facility such a game can be very frustrating. Imagine reaching a crucial point in the adventure when the 'phone rings or you have to go out – on your return you're back at the start! So, here we look at adding a SAVE game routine and graphics. If you've taken up the challenge and started to write your own adventure, you'll realise how time consuming it is. These two extras can take just as long (that's right, Steve, encourage our readers – Sarcastic Ed)!

There are many ways of adding SAVE facilities to a game, but they all involve writing a data file to tape or disc containing the current position, your score, the items carried, information about all the problems solved so far and the position in which all items have been left. Many players argue that you can judge the quality of an adventure game by the length of time it takes to SAVE a game and because the time taken to SAVE a game depends on how many problems there are, this is usually true.

There are two approaches which can be adopted at this stage. The first involves listing, on a piece of paper, all the variables whose value will change as the game is played and then saving those as a data file. The second involves simply saving the contents of all the arrays and variables onto tape. This is by far the easiest to code and it's the method we'll look at. As with all additional commands, we need to insert a statement into the main control loop immediately after the INPUT statements to recognise our instruction and call the appropriate subroutine. Listing 1 illustrates how we can add a SAVE game routine and a routine to load it back in again.

LISTING 1

```

445 IF C$="sav" THEN GOSUB 5000
450 IF C$="loa" THEN GOSUB 6000
.
.
4999 REM ** Save game routine **
5000
.
.
.
5999 REM ** Load 'saved game in ' **
6000

```

LISTING 2 shows how I would write the save game

subroutine, and LISTING 3 illustrates how a game saved on tape can be read back in again.

LISTING 2

```

4999 REM ** SAVE GAME ROUTINE **
5000 OPEN"CAS:data" FOR OUTPUT AS #1
5010 FOR X=1 TO 25
5020 PRINT #1, Q$(X)
5030 FOR Y=1 TO 4
5040 PRINT #1, S%(X,Y)
5050 NEXT Y
5060 NEXT X
5070 FOR X=1 TO 16
5080 PRINT #1, G$(X), B%(X), N$(X), N%(X)
5090 NEXT X
5100 PRINT #1, P%, S%, AA, AB, AC, AD, AE.....
5110 FOR X=1 TO 3
5120 PRINT #1, V$(X)
5130 NEXT X
5140 CLOSE #1
5150 PRINT"DATA FILE CREATED"
5160 RETURN

```

LISTING 3

```
5999 REM ** Routine to read in data
      from tape **
6000 OPEN"CAS:data" FOR INPUT AS #1
6010 FOR X=1 TO 25
6020 INPUT #1,Q$(X)
6030 FOR Y=1 TO 4
6040 INPUT #1,S%(X,Y)
6050 NEXT Y
6060 NEXT X
6070 FOR X=1 TO 16
6080 INPUT #1,G$(X),B%(X),N$(X),N%(X)
6090 NEXT X
6100 INPUT #1,P%,S%,AA,AB,AC,AD,AE,...
6110 FOR X=1 TO 3
6120 INPUT#1,V$(X)
6130 NEXT X
6140 CLOSE #1
6150 PRINT"DATA FILE READ IN"
6160 RETURN
```

Notice that these listings are almost identical, the only difference being the replacement of PRINT #1 with INPUT #1. Lines 5010 and 6010 show how you would save a game with 25 locations, while lines 5070 and 6070 show how to deal with 16 objects. The game described in the last issue use P% to hold the current location, S% to hold the score and the variables AA-AZ to act as flags. These are saved in line 5100 and read back in with line 6100. The final array to be saved is V\$, which holds the items you are carrying and, as you can carry three

items, is set to 3 lines 5110 and 6110. These lines will need to be modified to deal with the correct number of locations and objects used in your game.

Adding graphics to your game is not a very difficult task in theory, but in practice will probably take at least as long as writing the rest of the game. In principle, all you need to do is to have a subroutine for the graphics for each location and call this subroutine from the main control loop. The simplest way of doing this is to insert a number of lines into the main loop as indicated in LISTING 4.

LISTING 4

```
201 IF P%=1 THEN GOSUB 7000:
      REM ** graphics for location 1
202 IF P%=2 THEN GOSUB 8000:
      REM ** graphics for location 2.
```

Part 89 of a 20th Century MSX production

.... "Writing an adventure program". Presented by Steve Lucas.

Writing the calls to the graphics subroutines in this way is going to take a lot of space in the computer's memory and will take you a great deal of time to type in. Fortunately, the MICROSOFT BASIC used in MSX machines has an elegant way of doing the same thing, as shown in LISTING 5.

LISTING 5

```
201 ON P% GOSUB 7000,8000,9000,
```

This is interpreted by the machine so that if P% has the value of 1, it will call the subroutine at line 7000. If P% has the value of 2, then the subroutine at line 8000 will be called, and so on.

Unfortunately, we cannot just call a graphics routine on

MSX computers because we have been using the computer in the TEXT mode of SCREEN 0. Therefore, before we use these subroutines, we will need to change the mode to SCREEN 2, to allow hi-resolution graphics to be drawn and then to OPEN channel #1 to allow text to be printed onto the graphics screen. LISTING 6 shows how this can be done. When the graphics have been drawn, the

computer will return to line 204, which moves the cursor, prints the message asking you to press the space bar and then returns to SCREEN 0.

Now you'll need to spend a lot of time to create a good graphics display. The facilities available in MSX BASIC are the best around, but it will

LISTING 6

```
201 SCREEN 2
202 OPEN"GRF:" FOR OUTPUT AS #1
203 ON P% GOSUB 7000,8000,9000
204 PRESET 1,1
205 PRINT #1,"Press <Space Bar> to c
206 AA#=INKEY#:IF AA#<>" "THEN 206
207 SCREEN 0
```

take you at least an evening to draw the graphics for each location if they are to be good. I prefer a totally text adventure. The mind is

been carefully planned. LISTING 7 is not intended to show superb graphics (it doesn't), but just to give you a place to start.

LISTING 7

```
6999 REM ** graphics for location 1
7000 CLS:LINE (100,100)-(200,102),7,BF
7010 PUT SPRITE 1,(175,45),1,1
```

capable of creating a much more vivid picture of a setting than any graphics can do, especially if the description of the locations and objects has

If you intend to use sprite graphics in your adventure, you are, unfortunately, likely to rapidly run out of room in the computer's memory. Don't forget that you'll need to include a sprite definition routine together with the appropriate DATA before trying to put the sprites on the screen.

10000 DATA in a small barn

If we shift this up the ASCII codes by one, this line would become

10000 DATA jo!b!tnbmm!cbso

The ASCII code for a space is 32, shifting that up by one gives us 33 (!). In a similar way, a becomes b, b becomes c etc. Using this method makes it extremely difficult for anybody looking at the listing to solve the game without actually playing it, but it does set us the task of converting this into normal text. LISTING 8 shows how this can be done.

LISTING 8

```

130 FOR X=1 TO 25:READ Q$(X)
131 T$="": FOR T=1 TO LEN(Q$(X))
132 T$=T$+CHR$(ASC(MID$(Q$(X),T,
1) ) -1)
133 Q$(X)=T$

```

The important line in this listing is line 132, which looks up the ASCII code for each letter contained in the array Q\$(X), subtracts one and then adds the appropriate character to T\$. This is a very effective method of confusing anyone trying to cheat, but it does slow the program down to a very great extent and it will also take you much longer to write the game. I have never been particularly fond of making it impossible to cheat because it becomes much more difficult to debug the game when you are developing it.

Many commercial games now use data compression techniques which are not within the scope of this article. They do mean, however, that the game can be much more efficient in its use of memory and this allows the game to be far more complex than a game written in standard BASIC could ever hope to be. However, with about 28K free to play with, it is still possible to write a very extensive game

in BASIC which is fun to play and provides a challenge to the player.

This series has been an attempt to introduce some of the techniques which can be used when writing an adventure, but no amount of fancy programming can improve a dull plot! It is at the planning stage that a game will either succeed or fail and it is absolutely essential that you choose a plot very carefully and only when you are completely sure of your ideas should you sit down and write your game.

Even if you never sit down

and write a game, this series should have given you a few ideas about playing games. An experienced adventurer will always plot a progress map as they proceed through the various locations, but one problem seems to baffle players and that is finding your way out of mazes. In so many games you will find yourself in a maze of twisty passages all alike and no matter which way you go, you always seem to end up back where you started. The solution to mazes is always easy once you know it, but finding the right combination of directions to go in can prove to be difficult to the novice. The best technique is to make sure that you are carrying plenty of items with you when you enter the maze and then drop these items at different locations. You should then write down the directions you move in and make a note of the objects you come across. Very soon, you should have a map of the maze and a good idea of how to get out of it.

In this short series, I've attempted to take you quickly through all the steps needed to write your own adventure. Once you have created your program, it may just be good enough to try marketing. The market for adventure games is not as lucrative as that for arcade games and your game will have to be extremely good if it is to be saleable. There are, however, a few points worth mentioning. Most commercial adventures are not written in BASIC for two reasons. BASIC is often slow and the player can cheat and LIST the program. We can, in fact, do something about both of these faults in BASIC games.

Using integer variables throughout your game will result in the response times to your actions being much faster. There is another benefit to be gained from this due to the way in which the BASIC interpreter works. Integer variables (X% rather than X) also use less memory and this allows you to fit more locations and/or problems into your games.

There are a number of methods of preventing a program from being listed, but they can nearly all be cracked by a determined enthusiast. One method of stopping people from cheating is to shift the DATA up or down the ASCII codes. As an example, consider the description of a location



This game is distributed in
"HYPER OLYMPIC '84" in Japan.

VIDEO GAMES

HYPER SPORTS™

HYPER SPORTS

MSX AVAILABLE NOW ON ROM CARTRIDGE MSX

Track and Field I
Track and Field II
Hyper Sports I
Hyper Sports II

Konami's Tennis
Time Pilot
Super Cobra
Athletic Land

Antarctic Adventure
Comic Bakery
Monkey Academy
Circus Charlie

Konami

KONAMI LTD., TELEVISION HOUSE, 269 FIELD END RD, EASTCOTE, MIDDLESEX HA4 9LS,
® ENGLAND. TEL: (01) 429 2446 TLX 883978 KONAMI G © KONAMI 1984

MSX

Therapy

Master

It's your starter for 10 - a program that turns your MSX machine into an electronic Bamber Gascoigne. Terrifying . . .

This program was written to show just how easy it is to use data files to save information onto tape and load it in again at a later stage. When run, the program will display the main menu and you must press the numbers 1 to 5 to make your selection from the choices offered. They are:

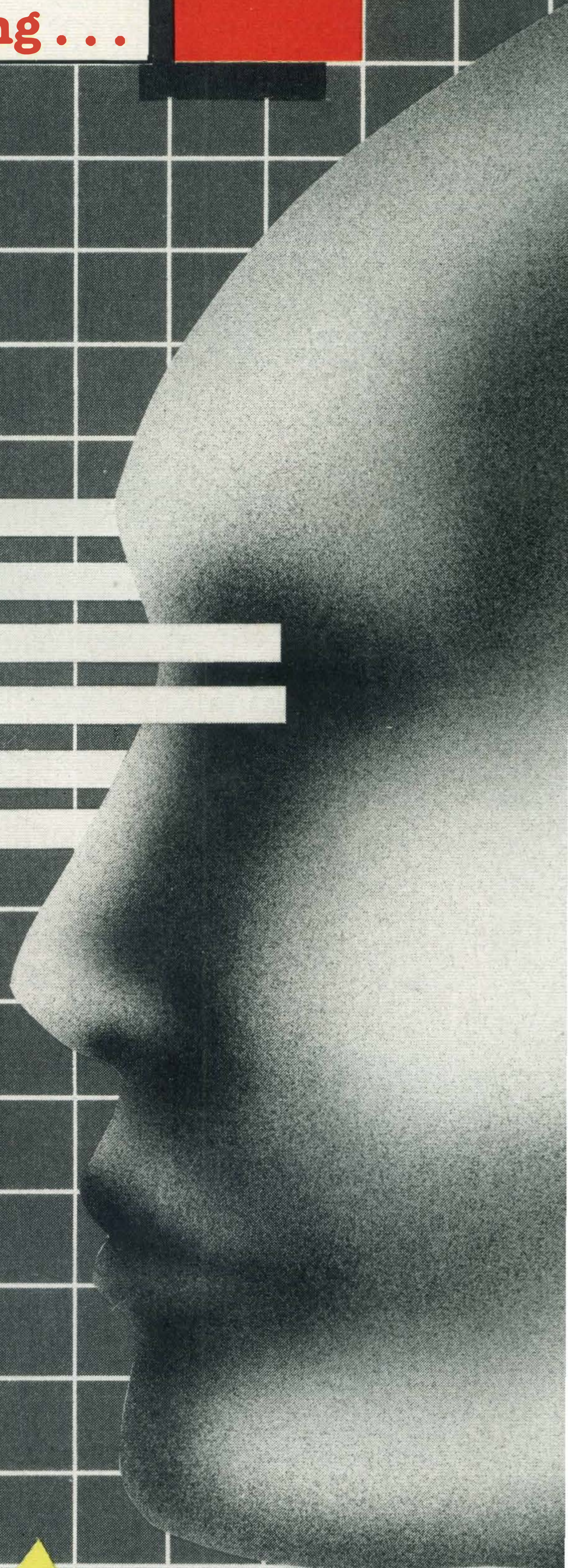
1. To type the questions into the computer.
2. To load the questions from a data file held on tape into the computer.
3. To save the questions onto tape.
4. To set the quiz.
5. To exit the program.

Before choosing options 2, 3 or 4, you must have run option 1 at least once!

When you have chosen option 1, you will be asked to enter 100 questions. After typing in the question, you will be asked to enter four alternative answers, only one of which will be correct. Finally, you should enter the number of the correct one from 1-4. Before going on to enter the next question, the computer will check that this question is OK, and after you have typed in all 100 questions it will return you to the main menu. This time you have the choice of saving the questions to tape or setting the quiz. If you choose to take the quiz, you will be given a score as a percentage at the end of the quiz.

You will see from the listing that the program is only short and there is, therefore, plenty of room to add extras to it. You could, for instance, add extra optional answers, graphics, sound, a choice of the number of questions etc etc . . .

Quiz



```
10 REM ** Quizmaster .... a utility program for MSX computers **
20 KEY OFF
30 CLEAR 1000
40 DIM A$(100),B$(100,4),A%(100)
50 COLOR 15,1,8
60 SCREEN 0
70 LOCATE 12,2:PRINT"Quizmaster"
80 PRINT:PRINT:PRINT"Do you want :--"
90 PRINT:PRINT:PRINT"1. want to enter questions"
100 PRINT:PRINT"2. want to load a data tape"
110 PRINT:PRINT"3. want to save a data tape"
120 PRINT:PRINT"4. want to use the quiz"
130 PRINT:PRINT"5. want to quit"
140 A$=INKEY$: IF A$="5" THEN CLS:PRINT"Goodbye":END
```

```
150 IF A$="1" THEN GOTO 200
160 IF A$="2" THEN GOTO 350
170 IF A$="3" THEN GOTO 470
180 IF A$="4" THEN GOTO 590
190 GOTO 140
200 CLS
210 FOR X=1 TO 100
220 PRINT "What is question number ";X
230 PRINT:INPUT A$(X)
240 FOR Y=1 TO 4
250 PRINT"What is alternative answer number ";Y
260 INPUT B$(X,Y)
270 NEXT Y
280 PRINT:PRINT"Which is the correct one 1-4 "
290 INPUT A%(X)
300 PRINT "Is this correct <Y/N> ";
310 INPUT AA$:IF AA$="Y" OR AA$="y" THEN CLS:NEXT X
320 IF AA$="n" OR AA$="N" THEN CLS:GOTO 220
330 NEXT X
340 GOTO 70
```

```
350 CLS:PRINT"Place the tape with the data file in the cassette recorder and
press the play button."
```

```
360 OPEN "CAS:quiz" FOR INPUT AS #1
370 PRINT"READING DATA FILE NOW "
380 FOR X=1 TO 100
390 INPUT #1,A$(X)
400 FOR Y=1 TO 4
410 INPUT #1, B$(X,Y)
420 NEXT Y
430 INPUT #1,A%(X)
440 NEXT X
450 CLOSE
460 GOTO 70
```

```
470 CLS:PRINT"Place the tape with the data file in the cassette recorder read
y to record the DATA file."
```

```
480 OPEN "CAS:quiz" FOR OUTPUT AS #1
490 PRINT"SAVING DATA FILE NOW "
500 FOR X=1 TO 100
```

```
510 PRINT #1,A$(X)
520 FOR Y=1 TO 4
530 PRINT #1, B$(X,Y)
540 NEXT Y
550 PRINT #1,A%(X)
560 NEXT X
570 CLOSE
580 GOTO 70
590 CLS
600 FOR X=1 TO 100
610 LOCATE 10,2:PRINT"Question ";X
620 LOCATE 10,3:PRINT"======"
630 LOCATE 1,5:PRINTA$(X)
640 LOCATE 1,8:PRINT"Is the answer "
650 LOCATE 1,10:PRINT"1. ";B$(X,1)
660 LOCATE 1,12:PRINT"2. ";B$(X,2)
670 LOCATE 1,14:PRINT"3. ";B$(X,3)
680 LOCATE 1,16:PRINT"4. ";B$(X,4)
```

```
690 AA$=INKEY$: IF AA$<"1" OR AA$>"4" THEN 690
700 AA%=VAL(AA$)
710 IF AA%=A%(X) THEN S%=S%+1:LOCATE 1,20:PRINT"Well Done." ELSE LOCATE 1,20:PRI
NT"Wrong. It was ";A%(X)
720 LOCATE 1,23:PRINT"Press the <Space Bar> to continue"
730 AA$=INKEY$: IF AA$<>" " THEN 730
740 CLS:NEXT
750 CLS:PRINT"You scored ";S%;"%"
760 LOCATE 1,13:PRINT"Press the <Space Bar> to continue"
770 AA$=INKEY$: IF AA$<>" " THEN 770
780 CLS:GOTO 70
```


Variables

AA\$	keyboard input
X	number of question
S%	score
A% (x)	number of correct answer
A\$ (X)	number of question
AA%	number of answer chosen
B\$ (x, y)	four alternative answers

Program Breakdown

Lines	
10-60	initialise variables, DIMension arrays and choose screen/colours
70-190	main menu
200-340	enter the questions
350-460	load data file from tape
470-580	save data file onto tape
590-780	set quiz



Unless you order

Home Computing WEEKLY

from your newsagent

this is what you're in danger of missing:

Latest software reviews — our experts tell you what to watch for
... and what to avoid

Quality-tested listings for all popular home computers
— games utilities and educational

Up-to-date news of the micro scene

The liveliest letters page around

Software charts — see what's selling best for your computer

U.S. Scene — our man in California brings you regular reports

Spot the bargains on our classified pages

Free supplements on things you need to know

Free-to-enter competitions.
Our readers have won prizes worth tens of thousands of pounds

You can't afford to be without Home Computing Weekly. And all you have to do to make sure you get your copy is to fill in the form below and hand it to your newsagent.

Dear newsagent

Please deliver/reserve for me a copy of Home Computing Weekly every week

Name

Address

.....

Newsagent trade distribution of Home Computing Weekly is by Argus Press Sales and Distribution,
12-14 Paul Street, London EC2A 4JS. Phone: 01-247 8233

On sale
every
tuesday
only 45p



Jeremy Vine helps readers make the right connections.

This month's postbag has concentrated on connecting computers to a monitor and in particular the Sony HIT BIT micro. Whether this is an indication of the present sales of the computer I'm not sure but my mailbag has been full of queries concerning this aspect of the machine, so here are a few of the problems you have encountered.

Kicking off is E. Hitchcock of Gillingham Kent:

I have just bought a Sony HB75-B computer along with a Fidelity CTM1400 Monitor TV, both with the 21 pin RGB Euroconnectors. I have been unable to purchase such a lead. Would it be possible to let me know the

pin connections, or if not where I could purchase such a lead?

The Euroconnector is a relatively new standard (yes, yet another standard!) and has been in some televisions for about three years. The connections for a 21 pin SCART to a TV are as follows:

- Pin 2 – Audio in (right)
- 4 – Audio earth
- 5 – Blue earth
- 6 – Audio in (left)
- 7 – Blue in
- 8 – Composite video status
- 9 – Green earth
- 11 – Green in
- 13 – Red earth
- 15 – Red in
- 16 – RGB status (to Pin 16)
- 17 – Composite video

18 – RGB status earth
21 – Socket earth
Pins 2, 4, 5, 6, 7, 9, 11, 13, 15, 16, 18 and 21 should be made straight through, ie Pin 2 to Pin 2. The remaining pins are not implemented and are intended for future expansion purposes. This will be for an inter-communications line for domestic control devices.

I would point out however that making up your own lead may be a bit tricky and could cause problems as our next reader's letter shows. A good electrical shop or your dealer should be able to make you up the lead if given the connections of both monitor and micro.

Now on to Dave Craig of Camberley, Surrey, who

writes:

I noticed your plea for questions in the December issue, so try this one. I have a Sony HIT BIT and have been trying unsuccessfully to connect its RGB output to my Ferguson MC01 RGB input. Sony's use of the so-called Euro standard connector was the first obstacle. Most electrical shops I visited had never heard of it. A helpful Sony engineer supplied the pin settings and I used the pin diagram on the back of the MC01. I get sound (with lots of noise) and a nice white screen. Any suggestions?

I suggest that if you've checked the pin connections a visit to your local dealer with

the pin diagrams is worthwhile. It's hard to say where you may have gone wrong (if you have at all) but your local computer dealer should be able to sort out the problem if you provide him with both monitor and micro. Failing this, a call to Sony may be necessary. Do check your connections against the list above.

Still involving the Sony machine, J Crane wrote the following:

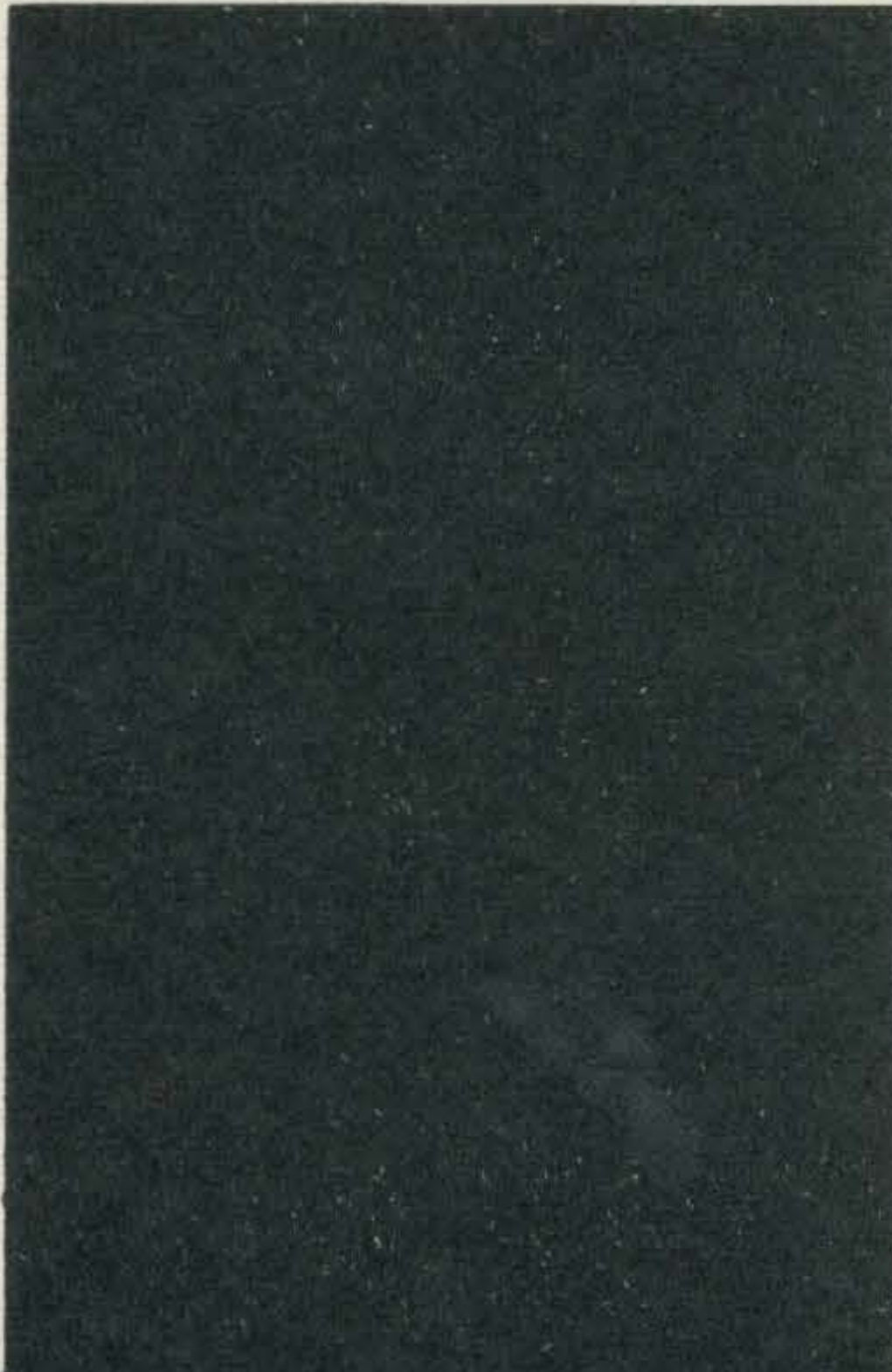
Your wonderful magazine might solve my dilemma. (Ed: Crawler!)

I have a HIT BIT which has output sockets, 21 pin, and 5 pin for the VDU, also the usual coaxial socket for the television.

I have seen a Fidelity AVS 1600 which has a 21 pin socket and a coaxial PAL socket. Also a

the HIT BIT.

You should be able to use either the AVS 1600, 21 pin



Ferguson TX which has a 5 pin din socket, a 7 pin RGB and a PAL system socket.

I want to use the 21 pin socket on the HIT BIT to get RGB quality screen output. Would either of these VDUs be OK for my purpose?

Your article on making your own leads (page 25, January) prompted me to seek your advice because I can get the wire and 21 pin ends in my local shop.

One good report. Thank you for the large page numbers – try to get one on every page. Thank you.

Finally I have had a Spectrum computer for 2 years and got nowhere in learning to program. Thanks again to MSX User for my rapid progress on

(see above) or the Ferguson. In the case of the latter you will need to know the following connections for the 7 pin socket:

- Pin 1 – RGB status
- 2 – Earth
- 3 – Blue

- 4 – Earth
- 5 – Red
- 6 – Composite video status
- 7 – Sound (left and right)
- 8 – Green

A ready made SCART 7 pin RGB lead can also be obtained from Philips Service on 01-686 0505. The lead number is: 32120648 code 67200/05. Philips can also supply a host of other leads by post and probably can supply the 21 pin leads as well.

Now you may be thinking that the only micro our readers have is the Sony but there are other micros you've purchased. Take Pat Fitzpatrick from Cork, Ireland for instance. His problems are as follows:

I have a Spectravideo 328 – as you know not fully

MSX compatible. But it is 'almost' MSX so I'm delighted with your magazine. The manufacturer's guide to SV BASIC is probably the worst such manual ever published!

I know some of the main differences between SV BASIC and MSX – the graphics and the STICK/STRIG parameters for example. But would it be possible for you to give me a full list of the differences? What makes conversion so difficult is that terrible manufacturer's guide – it doesn't even mention half the commands never mind give examples of how to use them.

One of the idiosyncracies of my

machine is that once the SCREEN is set the COLOR command will not change the background colour – as I'm sure it's supposed to do. For example after
10 COLOR 15, 1, 4
20 SCREEN 1
30 COLOR 12, 6, 10:
GOTO 30

the foreground colour is 12, border colour is 10 but background colour is still 1. The machine seems to ignore the 6 in line 30. I wonder if this is true on all SV's? Is it a design fault? How can I get around this problem? Keep up the good work.

Not having a 328 in the office I contacted Spectravideo who weren't able to give me a full list of the differences between MSX and SV BASIC as they haven't analysed these as yet. However they were able to offer the following points.

Firstly the CLEAR command doesn't work on the 328 or 728. The same applies to the KEY ON/OFF statements even though the 328 will accept it as a legal command.

The CLICK ON/OFF statements are peculiar to SV BASIC and are not MSX standard implementation. Finally the SCREEN 1 & 2 commands are the wrong way round in SV BASIC, so you will need to implement the opposite number if you wish to be in line with MSX BASIC.

As for the problem with the SCREEN and COLOR commands, Spectravideo are not aware of any such bugs and I suggest you look at the SCREEN command, as this does seem to be a source of differences. I hope by next month to be able to give a full list of those differences for all Spectravideo owners.

And that's it for this month. Keep those letters rolling in, no matter what the problem is, and we'll do our best to help. Or indeed if you have any useful hints or tips to pass on to other readers we'd be delighted to hear from you.

ZAXXON FORT APOCALYPSE BEACH HEAD
 POLE POSITION CONGO BONGO
 SPITFIRE ACE SOLO FLIGHT
 UP 'N DOWN SPY HUNTER
 BRUCE LEE DROP ZONE
 BLUE MAX RAID OVER
 STELLA 7 MOSCOW
 PAC MAN FIS STRIKE
 SENTINEL EAGLE MR.
 FLAK NATO DO DREIBS
 COMMANDER KNIGHTS OF
 BUCK ROGERS THE DESERT
 INDIANA JONES CONAN SLINKY
 TAPPER FORBIDDEN FOREST DALLAS QUEST



GOLD STRIKE

**GUARANTEED PAYMENTS
FOR FIRST CLASS ASSEMBLY LANGUAGE
PROGRAMMERS**

Translation work on the chart-topping
U.S. Gold Commodore games to the

MSX



Become part of the best motivated team in the U.K.
... one-off or long term contracts available
and original game or software concepts also required
for World Wide publication.



For fame and fortune from U.S. Gold
write in confidence to:
Software Development, U.S. Gold,
6 Central Street, Manchester M2 5NS,
or phone 061-832 6633 for details.



Garry Marshall reviews two useful guides for the MSX owner.

There has been a tradition of computer exposure and revelations for almost as long as there have been micros. At first, this was a very exciting and rewarding activity, for often the manufacturers of the early micros had no idea of the ins and outs of computer design and production. Their ignorance could have been any one of several causes but, typically, the design team left the company before any proper documentation on the micro was produced, or they moved on to other activities within the same company without telling anyone else all the details of the previous project. This left a perfect gap for enthusiast writers and publishers to fill, and they often did so with gusto.

MSX Exposed

published by Melbourne House
ISBN 0-86161-182-9
Author - Joe Pritchard
Price - £7.95
240 pages

But MSX doesn't have these kinds of loose ends to tie up, and so **MSX Exposed** can't convey the excitement and promise of illicit goodies that earlier exposés could. Indeed, it starts with a neat account of the MSX standard, which is a tacit admission that it is dealing with a well-defined system, and not with a system barely under control.

Since the book can't really be in the tradition of the micro exposures, what kind of a book is it? Well, it's simply a rather nicely structured guide

to the various aspects of MSX, with a few unexpected insights into the hardware and its capabilities. There is also some fairly mundane stuff on BASIC and programming style, but the author seems to be much more at home with hardware than software.

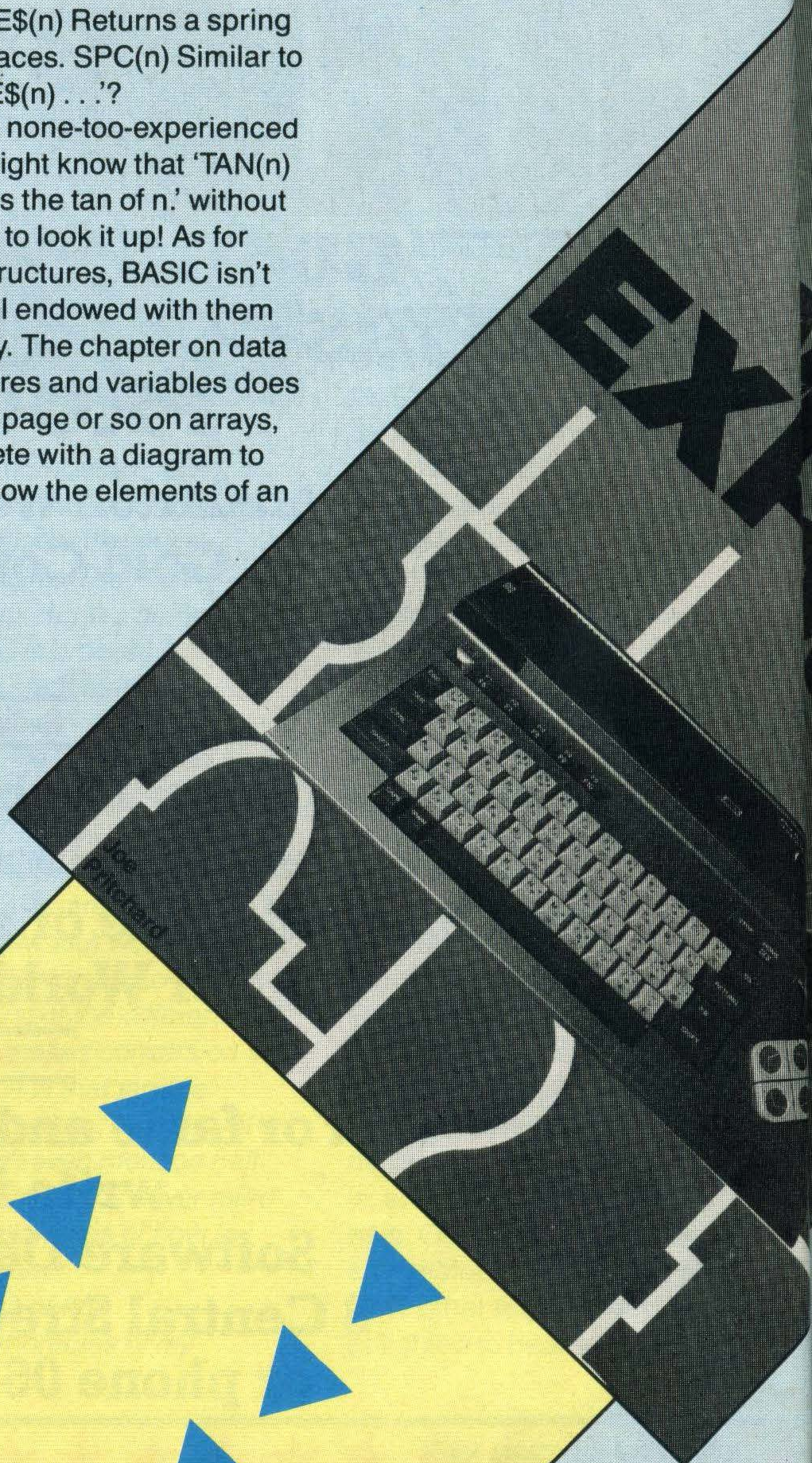
After detailing the MSX standard, there are chapters on the core BASIC and 'data structures and variables'. The BASIC chapter is little more than a brief run through the keywords. It's done at a level that is too brief to help the beginner, and too superficial to be of value to the experienced user. For instance, what can the beginner make of:

'SPACE\$(n) Returns a string of n spaces. SPC(n) Similar to SPACE\$(n) . . .?'

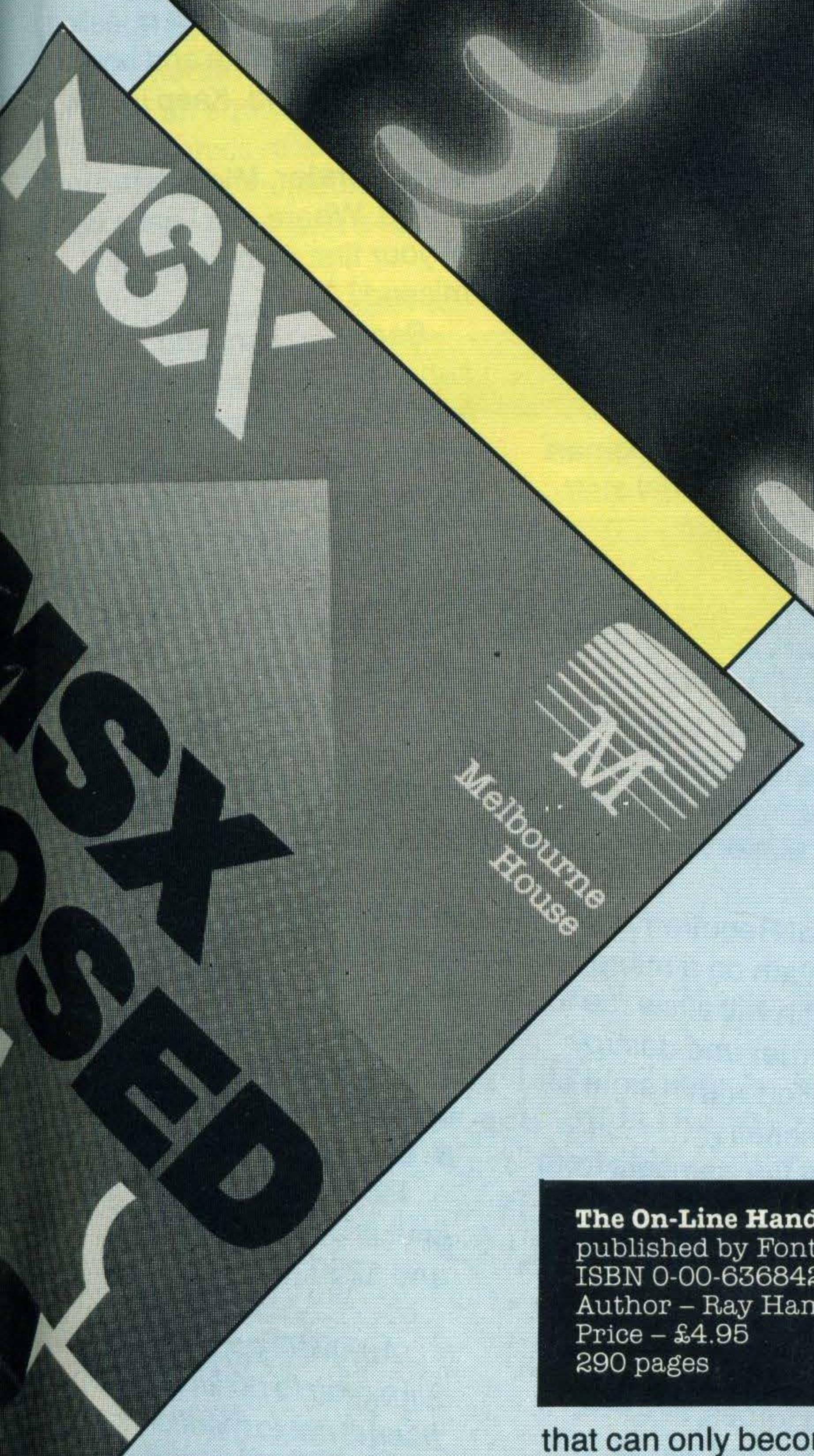
Even a none-too-experienced user might know that 'TAN(n) Returns the tan of n.' without having to look it up! As for data structures, BASIC isn't too well endowed with them anyway. The chapter on data structures and variables does have a page or so on arrays, complete with a diagram to show how the elements of an

array are held in memory which would be more helpful were it the other way up, but it is mainly about constants, variables and operators.

After this, there is a good chapter on cassette usage. This explains how programs are represented in the computer and the different ways in which they can be stored on cassette, as well as how to deal with files. Later chapters on the video display processor and the programmable sound processor are also good, providing more information than usual about an area that generally seems to be poorly understood. Having said that, the chapter on joysticks is a bit of a joke at two pages of text, including diagrams, and the treatment of the memory map is short on hard information.



So, all in all, **MSX Exposed** is a bit of a mixture: it is good in parts and pretty ordinary in others. If you are looking for a book that treats some aspects of MSX in depth, then this one might be worth a look. If your topic is hardware you could be lucky, but if it's software, I wouldn't give much for your chances.



Although **The On-line Handbook** isn't specifically to do with MSX, it explains how any micro can be used to advantage, something

The On-Line Handbook
published by Fontana
ISBN 0-00-636842-5
Author - Ray Hammond
Price - £4.95
290 pages

that can only become more important as time goes on. Its sub-title is 'How to connect your micro to the outside world', and **The On-line Handbook** explains how you can use your micro for banking, for on-line shopping,

to access Prestel, to obtain information and even how to communicate with a main-frame computer in order to get it to carry out computations well beyond the capabilities of your humble PC. Ray Hammond has used his micro for all these things, and explains with clarity and enthusiasm how it's all done.

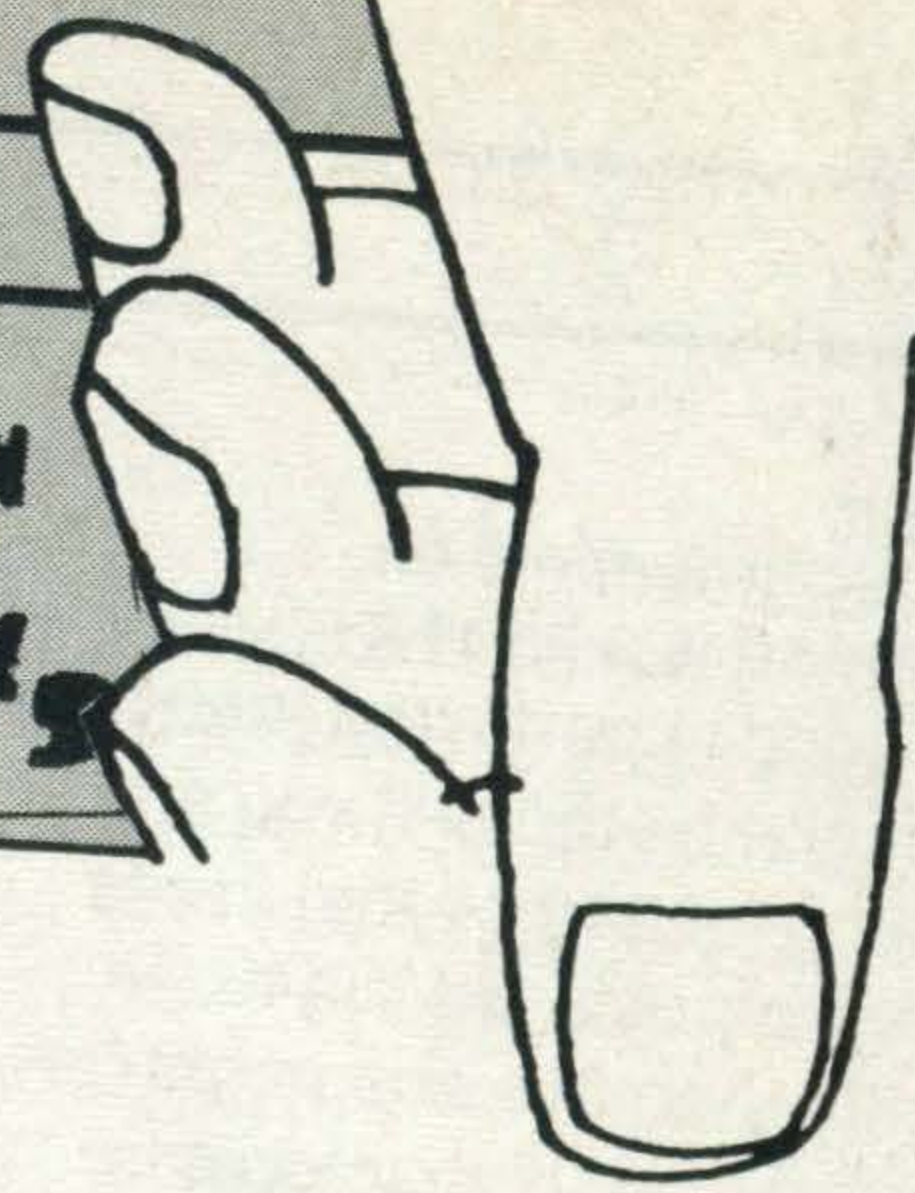
He starts by explaining the workings of the technology. Modems, hardwired and acoustic; communications signals; communications software; protocols; all are explained in a straightforward and accessible fashion. Having explained the technology, he then proceeds to tell us what we can do with it. You can exchange

information via bulletin boards at one extreme or, at the other, find the reference to any book that has ever been published in this country from BLAISE, the British Library's on-line system.

Hammond is an enthusiast. He assures us that all this is fun, and I hasten to say that I am sure that it is. He also states that it doesn't cost much to access all these on-line systems. I must say that I am not so sure about that. It's not too difficult to imagine some hefty bills accumulating after joining all the various services and networks, and then paying for information from them. In fact, I have a sneaky feeling that Hammond didn't write this book entirely for our benefit?

If you want to get into the global village and start to link with the rest of the world, then this is the place to begin. It tells you exactly what there is and exactly how to get at it. It is a good, informative read in its own right. Well, what more d'you want?

YOU WRITE



MSX USER
 Argus Specialist Publications
 No 1 Golden Square
 London
 W1R 3AB

Dear Sir,

I have recently said goodbye to my much loved TI 99/4A because of the difficulty and expense of expanding it. So I started looking around for a replacement before the withdrawal symptoms of not having a computer became too severe.

MSX machines appear to have everything going for them - except the support of

my local software shops, whose comments range from 'you're wasting your money because they will finish up like the TI' to 'we intend to carry software if they catch on, come back in six months'.

Out of curiosity I bought your January edition of MSX User and I am now totally sold on the idea of MSX after seeing the Sony Hit Bit demonstrated by a salesman who, with your editorial staff,

know what they are talking about. It's now just a question of which machine and how much to spend. Keep up the good work.

T. M. Forster, Middlesbrough
 P.S. Where can I get a copy of your first edition which I missed?

Back numbers and subscriptions are available from Infonet, Times House, 179 The Marlowes, Hemel Hempstead, Herts UP1 1BB.

Lizzy

P.S. Got any spare t/writear corection ribbons?

1 Golden Square, London W1R 3AB. Registered in England No. 1269520
 P. Lewis, R. Moulton, A. T. Patten, M. J. McKenna (Secretary).

Feedback

Dear Sir,

Personal File

- A) 35 year old Aussie musician, permanently residing in UK
- B) Hobbies - spending all my hard earned gig money on arcade games, (and cans of Fosters).
- C) Knowledge of micro-computers - absolutely none.
- D) Recent illuminations - four weeks ago I bought a mag called 'Your 64', coz my copy of 'Melody Maker' had not yet arrived at the newsagents. (Very interested).
- E) Within five days, bought another fourteen mags about micros. (Totally hooked now. To hell with 'Melody Maker').
- F) Present state of mind -

confused, cracking up, almost suicidal, prone to outbursts of tears, and on the verge of going back to 'Melody Maker'.

HELP!!

Personal Requirements Information on a micro, any micro, which will allow me to:

- A) Kill, murder and destroy every alien within sight of my crosshairs.
 - B) Rescue fair damsels from 'orrible grillas.
 - C) Fly through 3D time and space and save the universe as we know it.
 - D) Fight the entire German armed forces, single-handed, and win WWII.
 - E) Program, draw pretty pictures, etc, etc, and write games programs so as I can earn lotsa dough.
- Please help! You may save

a life (namely our cat, who can no longer take the ravings and rantings, or physical abuse of a confused computer-crazed madman).

PS Please print this letter so I can have one of J.J.'s fantastic, incredible, megatriffic MSX badges to weld onto my saxophone.

Yours with sincerity and desperation,

B. D. Everingham, Romford

P.S. My micro spending limit is between £200 and £250.

Any MSX computer will allow you to do all of those things, not to mention save your cat. Your spending limit narrows the choice down to the GoldStar, at £199, or the Toshiba HX-10 at £239.

Badge follows.

Mike,

PS I've got a box of correction ribbons but no one can find a Tek 1444 888 an engineer to put them in.

Dear Sir,

Congratulations on producing the first MSX magazine around. (How's that for a sycophantic opening!) The first issue wasn't too bad, but the second was really jolly good!

At the moment I have a BBC micro and printer with a black and white portable and am considering upgrading to an MSX machine.

Yes, upgrading. The reasons for my decision are six fold:

- 1. There are 32 easily definable and controllable sprites on the MSX micro.
- 2. More memory is available, 28K in hi-res mode without a disk drive or 23K with one, whereas the BBC micro in an equivalent graphics mode leaves only 8K available to the user or 6K with a disk drive!
- 3. The sound is easier to

use (the PLAY command is very versatile and aids conversion from proper musical notation).

- 4. The games are better, especially the ones from 'Konami' and 'Ant Attack' from Quicksilva. The promised mega-game is also an enticement to change micros.
- 5. A GoldStar machine with a Sony disk drive for example costs £550 at the moment, with an equivalent BBC system coming in at least £600.
- 6. The MSX machines have standard Atari-type joystick ports.

There are three disadvantages that I can see in changing to an MSX machine:

- 1. If I did swap software with my friends at school, (which of course I don't), I could no longer.
- 2. Software and hardware availability is low (at the

moment anyway).

- 3. Failure of software houses and publishers to support the machines.

You may feel my fears are unfounded, but witness the excellent Memotech MTX series which seems to have been abandoned by the majority of software houses.

I hope to become an MSX owner sometime in the early part of this year. If anyone wants a BBC micro, 12 months old, excellent condition, dust cover with a year's supply of 'Acorn Users' and several books, contact me now! Sorry about the length of this letter but I'm hoping to win one of the most highly prized accolades in the world of computing . . . a MSX User badge! (Hint, hint).

**Russell Jefford
143, West Close,
Medmenham, Marlow,
Bucks, SL7 2EH**

You've got the badge.

Feedback

Dear MSX User,
Britain: 1, France: 0.
Nobody here had the idea to promote the MSX devices in the 'Republic Francaise'.

More seriously, we won't miss one issue, even though

it's quite a job to get a copy in gay Paris!

Your reports about the Canon, JVC, Hitachi, etc . . . (plus the ads) will certainly help us to choose the MSX we need.

We are a pop music duet; our single will be out soon and we work on a Yamaha FM DX7 and RX11 system.
Bon baisers de Paris.
**Regine and Michel
Paris**

raw prawn and say you gave them to me to file I'll search on the Holy Thesaurus that you didn't!

Secondly, I had . . .

MSX USER
Argus Specialist Publications
No 1 Golden Square
London
W1R 3AB

Dear Sir,
I read MSX User every month, and find it very interesting. However, when I look at these MSX machines running loads of different peripherals, I can't help thinking what a mess they look. I think that MSX manufacturers ought to standardize the colour of their hardware etc. If you can imagine a GoldStar with a Sony disc drive plugged in, together with a Toshiba printer and a Canon joystick, you'll begin to see what I mean!

**Steven Kendrick
Sutton Coldfield**

Correction, ribbins but no one can engineer to put them in.

WE REPLY

ANIROG

*The Name
For Quality
And
Innovation*

Flight Path 737



ADVANCED PILOT TRAINER

NOW AVAILABLE For The MSX £6.95

TRADE ENQUIRIES: ANIROG SOFTWARE LTD. 29 WEST HILL DARTFORD KENT (0322) 92513/8
MAIL ORDER: 8 HIGH STREET HORLEY SURREY 24 HOUR CREDIT CARD SALES HORLEY (02934) 6083
PAYMENT BY CHEQUE P.O. ACCESS/VISA 50p POSTAGE & PACKAGING

