

DRAGON USER



The independent Dragon magazine

August 1987

Contents

Letters

DeflateDOS ... Thoughts on Basic 42 ...
author's solution ... and plenty more...

News

Dragon show in September ... 48' Dragons
in interview ... new light pen software ...
modern for Dragon 64.

Dragonsoft

Prism modem package for the Dragon 64,
by Hapton Software.

The Answer

Not only do we have the answer to the May
competition, but we have the answer to the
April competition.

Puzzle FORTH

J. R. Singer investigates FORTH by un-
tangling paid Gordon Lee puzzles.

Winners and losers

Gordon Lee looks at some of the solutions
to the March competition, and compares
his time — from 13.85 seconds to 5 hours.

The Show goes on

Ken G. Smith finally makes it back from the
8889 Show and reports to HQ.

Down in the dumps

We asked for dumps, you sent us dumps.
These three are for the 64-bit FX-85 C40-88
Morrisone and Sandy GMP-88 poster.

Dragon Answers

Sound to the cassette port ... connecting an
Amadeus printer ... Fax or GS-87.

Super SAM

Test your Synthesious Address Multi-
player gently — it doesn't mind being inter-
rupted, but it hates being poked, says
Matthew Lodge.

Electronic author

Philip Good writes up Quickbeam's Elec-
tronic Author.

Two special offers

Three popular books of Dragon programs,
no longer available from their original
publishers, and a FREE GAMES with the first
450 orders. Also Dragon Movie Accounts
for the Dragon 32, at less than half the
original price.

Write: ADVENTURE

Peter Gerard goes back to his pen and
digresses into a little marketing advice.

Adventure trail

Retelling Basic games (why they don't),
picks for Keys of the Wizard.

Competition

Gordon Lee is looking for a mysterious
factor, while Paskoff are shelling out
T-shirts.

Editorial

WALLS, home of the Dragon. The very
place for a summer holiday — as long
as you stay indoors. No, seriously, it's
not like that. It only rains in the morn-
ings. Anyway, I'm off next week, so if
you see a puff of smoke and flame along
Hay Bluff, it won't be me, 'cos I'm not
telling anyone where I am...

...just like The Expert, in fact. We
would like to think that he's lost in the
post, the postal strike started after he
should have been here...

The best news this month is that
Pulver Software are planning a show in
Reigate late in the year. Somebody
else has an eye on a London venue, and
some of the MDUGs are putting on an
industrial show on the south coast. The
future of the Dragon lies in bringing
together suppliers, users and software
writers in a setting which is substantial
enough to draw people from all over
Britain, and informal enough to accom-
modate moderate numbers in comfort.
No more echoing halls.

And don't miss our special book of-
fer. Dragon books are getting harder to
find as big publishers look elsewhere.
Now is a good time to buy out of print
books at remainder prices. Have a look
and see.

Telephone number
(All enquiries)
437-4343

Editor
HELEN ARMSTRONG

Production Editor
BARBARA HALEK

Editorial Secretary
ANNE MARIE ALLEN

Advertisement Manager
ATHENA PERHAM

Administrative
CARMELLA PICCINI

Marketing Manager
HELEN PERRY

Managing Editor
PETER WORLOCK

Publishing Director
JERRY IRELAND

Subscriptions
UK (for 12 issues)
Overseas (for 24 issues)
US\$4.95/£3.95/£1.95
Dragon (over 1212 Little Newport Street,
London WC2H 9PP)

24 address: 100 Business Press International,
285 East 42nd St, New York, NY 10017

Published by BOD Press Ltd.

© Bod Press 1987

Typesetting and Production by Annet Limited,
London W8H

Printed by Rowley Brothers Ltd, Ashford, Kent
Registered at the Post Office as a newspaper.
Dragon and its logo are trademarks of
Burlford Ltd.

How to submit articles

The quality of the material we can publish in
Dragon User each month will, of course, great-
ly depend on the quality of the documents that
you can make with your Dragon. The Dragon
computer was launched on the market with a
powerful version of Basic, but with very poor
documentation.

Articles which are submitted to Dragon User
for publication should not be more than 3000
words long. All submissions should be typed.
Please leave wide margins and a double space
between each line. Programs should, whenever
possible, be computer printed on plain white
paper and be accompanied by a tape of the
program.

We cannot guarantee to return every submit-
ted article or program, so please keep a copy. If
you wish to have your program in the microform
include a stamped addressed envelope.

Letters

Basic extension

HAVING just read Roy Coates's review of Basic 42 again, I decided to add some comments.

Firstly, Roy's review and the ads might lead some to suppose, on looking at the system, that you could expand more memory space available to Basic than when operating in 32K mode. This is not the case, in fact, it is exactly the same. There are 3.75K extra available for machine code or data storage, but this is the area used by the utilities, and it cannot be used if you want to use the non-destructive window.

Another point is that even with Compuserve's Hi Res, once a character address on the screen is in memory, so a high resolution screen dump is required if you want to keep a copy of your information. This is not a problem if you have a real timecode dump routine, as the start and finish addresses of the current window are readily available and can be sent to the dump routine so that just the current window is printed. The keyboard response seems to be slower and I find that I lose characters. This seems to be alleviated by shortening the delay before autorepeat.

Because the system recognises PRINT as a printout relative to the window being used, running existing programs on Basic 42 can give some unexpected screen display. The best way round this is to use the PRAMB command and define a window 32 characters wide and 18 deep, use this window and the display is restored. The user window gives a much more professional look to your programs, making it possible to display a program title at the top of the screen while the action takes place below. If a command window is then used at the bottom of the screen for requesting inputs, keypresses etc., the result can be very impressive.

Finally, accepting commands in lower case letters may be very desirable when programming, but it can be frustrating when INPUT used LINE INPUT is not representative of the answer, converted to upper case. Unfortunately, this is exactly what happens. Help

Every month we will be shelling out a game or two, courtesy of Microdeal, to the reader/s who send the most interesting or entertaining letters. So send us your hints and your opinions, send us your hi-scores and suggestions. Send us your best Dragon stories. What if you think we are, mind readers?

EXTRA
PUFF

Delta users

DUDE stands for Delta Users Data Exchange. I initiated it after spending a long time struggling without information, professional software etc. As time passes it becomes increasingly obvious that the Dragon community is forming into specific user groups and it seems that Delta users should also have a common base. The group is still young and needs definite structure, and an e-mail exchange list if there is a good response from the details given in DU.

It seems there is a steady state of MEMBERSHIP to be put together, as the DU 80 group have done. This keeps costs down for the members and allows easy transfer of address, having already contacted the five people, the initial response seems favourable and I have made several new friends (mainly over the phone). I don't realise just how helpful some people can be, not even to someone as enthusiastic as I have for Delta (is it the best D03)? The group will only thrive if other Delta users get contacted and show some interest, so write a memo phone on (0703) 36040 and let me know what you're thinking if you don't need something now, you may do in the future, and you could make a valuable contribution.

Dave Martin, 44 Pignose Road, Freetown, Southampton,
Hants. SO7 3BN.

P.S. To all the people who "registered" at the show, but have not contacted me since... you should be ashamed, get your wings clipped and NOW!

THE user groups are the backbone of specialist Dragonisation, just as Dragon User is the backbone of general Dragonisation. They form the networks which keep users in touch. I strongly suggest that Delta users get in touch with Devotee to find out what is going on, even if they don't see how a group would be of immediate benefit. Then, if nothing else, contacts would have a register where Delta people could get in touch.

It's hard here, for when I contacted Bob Harris he advised that PQMB 57885, 41 patches (including 40) and PQMB 57884, 15 parts it took on again. Once these were incorporated at the beginning and end of the program, the problem is solved.

After reading this you might think that I am somewhat disappointed with the package, but you would be wrong. Using the system is simplicity itself, a quick browse through the excellent manual and I was ready to go, and the overall effect was to make me work look good and is doing so, made me look good, and who could ask for more than that? I shall have to get the spirit willing though, just

present for me to make things while the machine dumps the forms screen.

Age Smith
30 Black Road
Deal
Kent
CT14 6AD

Dragon chunnel

WITH regard to above, has anyone thought of using Devote?

I know we have many Dragon owners in Belgium and Holland, not to mention France. If we now add those from the south of

England we could have an international Dragon Show. I suppose it is all just a dream but it might make people take notice of us.

Karl G. Smith
30 Black Road
Deal
Kent
CT14 6AD

Curve warped

THE euphoria of finding my name mentioned three times in Gordon Lee's Minors and Losers page (June 1987) soon evaporated when five mistakes were spotted in my Dragon Curve programming. If anyone has tried the program then it would have crashed almost instantly. The following corrections will rectify matters:

Line 80 had a bracket missing and should have read:

```
80 G=0+(360/2)*SIN(C/2)+1:  
POKECH:SGC=G/2
```

Line 150 included the number 570, thereby 5703 known as a basic number, perhaps that was a hint, it should have read:

```
150 IF SGN=0 THEN MD=0:  
GOTO 90
```

D.J. Gray
570 Acklam Road
Skeochrough
Cleveland
TS5 6BE

AFTER everything else, too. Never mind. Let's hope this correction puts the curve straight, so to speak.

Revoked

RECENTLY my Dragon 32 lost its voice. Occasionally it does return for brief periods, but is unpredictable. Since all the software still worked OK, as it was (as the manual is required in case it "died" on the opening table.

I had resigned myself to its silence until last week. I use a Sanyo CRTD data recorder

which has a switchable monitor and LED indicator. I had just loaded a game, and forgot to press the STOP button, and started playing. I was surprised to see that the LED indicator was jumping in time with the expected sound effects. I switched on the monitor and was surprised to find the Dragon was using the cassette to speak to me. I have since played around and found the sequence to get the new voice to of having loaded press STOP to release the head, b) put a blank tape in to keep the sensor pressed and press RECORD only, c) switch the monitor to on, and reduce the volume considerably. The Dragon sends all sound output to the data recorder.

I managed to find someone else with a fully working Dragon, who tried to find out if it worked in this also. In fact the sound came both from the television and the cassette recorder, so you can have a stereo Dragon. I believe the original problem with my Dragon is simply a dry joint somewhere, but now I don't have to worry about it.

Thank you for your excellent magazine, Long Live the Dragon.

M. Hall
150 Mandarin Way
Wyman's Brook
Chesham
Glos.
GL50 4RS

PS I persuaded my friend with the working Dragon to subscribe to Dragon (can he do it get commission?)

WELL, or not as such, but if your friend finds he doesn't like us you can come anytime in the filing cabinet here for a bit. Well, I am your type, by any chance?

Listings I am blasted

I have delayed renewing my subscription as I have become rather disappointed with the magazine recently as it appears to contain little but long pages of listings. The June editorial indicates something, so I will now renew.

A.J. Collins

STRENGTH It's a tough life trying to run a computer

magazine without putting the ad out straight listing in, y'know.

OK, let's get back to work: Original printer issue is over and done, although we still have a stack of error dumps which we shall be running in chunks. HOW we would like to hear from anyone who would like to write a review, description or educational piece on disc drives or disc operating systems. The hardware doesn't have to be new, but it must either be currently available or widely owned. Drop me a line and I will write back with details.

Autorun

LOOKING back in the life of the Dragon, people have always read hard work about running a Basic program. Here is my short solution.

- 1) Load basic program as normal.
- 2) To save type:

POLAR 1 : CLR:RG28,
"LONGMID"
CLEARM "Name", 180, 250 :
PEEK (27) = 255:0
POKE 26, 12 : POKE 27, PEEK
(26): POKE 26, PEEK (27):
POKE 31, PEEK (27): RUN

This works by saving the direct command alongside the Basic, so when released it carries on with the command from where it left off causing the location to be reset and a RUN command.

Paul Morgan
16 Moorfield Road
Fulwood
Sheffield
S10 4SS

Malta message

THROUGH the medium of your magazine Dragon owners have remained united, and I hope to stay. It's with this in mind that I'm writing this letter. Others helped me, so I should help them.

In the book inside the Dragon on page 207, I found a very important point. How to print on

the printer exactly what is on the screen, regarding printer width, but it has a small mistake.

In short, the following Basic Direct (without line number) statements set up the printer width to be the same as the Dragon's display width, of 32 characters per line.

POKE \$H1460
POKE \$H5832
LUYST

That has a mistake in it in the book, with a 5 instead of the correct 4.

I would like to take this opportunity to thank Peter Dancy who is the Dragon user of Sep. 88 pointed out another small mistake on page 215 of Inside the Dragon, the cassette gap type being 801 for continuous stream, whereas it is actually 8801.

In my opinion inside the Dragon is the best book for serious programmers, but to clear out the bugs and get an OK message, one still needs to take a good look at the Dragon over. My compliments to the Editor and all the staff.

Joe Giriat
73 Anzac Avenue Street
Haveron
Mala

JOE is another of these Dragoners who is a really useful human being, and writes us nice letters, too.

Main man

I have read your magazine since it first came out, and I am amazed at the amount of detrimental rubbish stated about both the Dragon and Family Computers.

I have been using both machines for some years and apart from a very small amount of component failures (I modulate, I joystick input chips, I have had no other problems).

The main trouble seems to be that your readers only like to play games. The Dragon and Family are among the most powerful of the 8 bit computers I have ever seen. I have been running random access sorts (several mega languages) and other business

and control programs including using the Scientific word processor programs.

I have four discs of BASIC each on each Dragon, and have not even had a single error on a run program in six years. I wish I could say the same of Family. Does it show how a computer lead or write?

The reason for this low error rate is that I run the so called obsolete Delta Dos which runs like a good maintenance unit. It allows sharing and other important features, and does not have any of the bugs that I keep reading about in your journal. I also run both Intell and Intell II Epson printers and a Juki daily sheet printer. The Juki has only two problems: it takes and it throws away any characters over one line length, you cannot wrap as with the Epson. I feel it is time someone spoke up for these very high powered but small computers. By the way what has happened to the company that bought out the Dragon? Are they still in business? If so can Dragons still be obtained from them (even if the ROMs have to be changed to English ones)?

I worked for many years on mainframe computers and can run the same sort of jobs in the same sort of time on a Dragon. I have not seen it done on any other small computer.

A.C. Barnham
23 Giffels Crescent
Croyley
Macclesfield
CH64 2ET

WELL, er, thanks for the tips, A.C. Actually most of the tips, I know who work on mainframes spend their lunch breaks playing 'Colonial Adventure'.

As far as we know, Barnham are still in business, but no-one over here seems to be able to buy anything from them, so we aren't entirely clear what 'in business' consists of.

Footnote

Footnote: Don't forget to write to us if you would like to write about your disc drives for Dragon User — at last Mark Smith got his Basic 42 later printed — even if it catches up with the next one.

News desk

If you have any new products for the Dragon — software or hardware — ring the News Desk on 01-437 4343

Dragon Show in September

PULISARI Software are organising a Dragon show in Rochdale, Lancashire, in September.

The show, titled the North of England Dragon Show and Convention, will take place on Saturday 12th September at the Bishopscote Upper School, Rochdale.

"We have been in touch with all the software dealers," says Pulari. "and they all want to come, apart from Microdeal who weren't too sure, it's too early to finalise anything yet, but we have a poster of experts from the local user group, and people to demonstrate. We want to make this a full round table

sitting than a normal show."

"There will be a computer clinic, where anyone can come and ask about any computing problems they are having, and we'll be having a forum where people can ask questions."

"We've organised a snack bar and we are supplying for a bar licence." The show will be open from 10am to 5pm, and the gate charge will be £1.50 for adults and £1.00 for children.

Subject to the normal limitations imposed by doing three jobs at once, the editor of Dragon User is looking forward to finding out where Rochdale is at time for September.

New Trojan light pen software

MACDOWAN Consultants have rewritten the previously disappointing software which accompanied the Trojan Light Pen. The new version is available on tape for £700 or disc for £800. The Light Pen itself is also available for new users.

Among the improvements in the new software are full screen availability, all machine-code programs, and PMODE 3 and 4 screen dumping for 14 different printers.

"All our products are printer orientated," says Bill MacDowan. "I am the country's leading expert on printers now, and as far as I am concerned,

anybody can ring us up for advice on printers at any time with no obligation. I'd rather they did, because we can then tell them what's wrong with the printer they want to buy."

MacDowan, who is also writing screen dumps for Commodore, BBC and Atari micros, claims that their products are unconditionally guaranteed and "are up to the standard which our customers have come to expect." MacDowan consultants are at 5 Ashburn Drive, Claythorpe, Nr. Grimsby, Lincs NG32 5DQ (tel. 0406 70885), business products can also be obtained from Harris Micros.

Versions are available for the Brother Hi-8 FTR, Seikosha GP100A, and Epson FX compatible printers. Custom versions can be supplied — call for details. Price £400.

The Seikosha Seisaku Computer Display Unit, dumping 16 characters 0-2 and 0 in mini or large (roughly A4) sizes to the Seikosha GP100A printer. Price £200.

Fontastic Compo-Net is a program for use with MacDowan Consultants' Printer

Control word processor. It gives a new text font in a squared, semi-graphic style, useful for display or for short texts like instructions or messages. Price: £2.00.

These programs can be obtained from Dragonfire Services, 10 Perry Jones Close, Maun, Gwent, NP23 3AH.

High adventure

COWEN Software called to say that Adventure Writer, reviewed in last month's DU, is now available only as an improved version at the higher price of £16.95, and is being distributed by Packsoft.

Modem package

A complete modem package for the Dragon 64 is being marketed by Haydon Electronics for £45, including post and packing.

The package includes a Proton Modem 2000, a connecting lead, software to enable the computer to access variable type boards (Proton), for example, sending ASCII boards which run on 12075 baud, and of over 200 output boards.

67 Dragons wedged in a letterbox

DELIVER of Lancashire, computers — other computers, that is — are not your letter's favourite subject just at the moment, for are they the favourite subject of reader P Clayton of Chorley, Lancs.

The red dust on the computer when Dragon User doesn't arrive on time. Mr Clayton is blaming the computer because Dragon User did turn up on time — about 10 years.

In fact, at the time of writing, Mr Clayton has received 60 and

countrywide, and full instructions.

The software comes on tape or disc for Dragon/DOS, only Haydon Electronics is at 36 Laurel Drive, Warrington, South Wirral, L84 1TW.

South coast show

PHILIP Reed, Dragon person and NDUO regatta, is putting around the word that he and some south coast friends are organising a non-commercial "show and get-together" in the late autumn or early winter. The likely venue will be in Gosport or Portsmouth, and obviously suit numbers.

Philip says that his intention is primarily to get people together to discuss and demonstrate aspects of computing, but that if sufficient numbers are interested they may be able to organise discounts on some computer products. The cost of mounting the meeting should, he reckons, be between £1 and £2 per head.

All those interested in taking part in going along should contact Philip at 27 Pinston Road, Ebbw, Newport, Hants PO12 4BP with a stamped self-addressed envelope for the latest update.

Printer progs

DRAGONPRINT Services have three new utility bundles, all under DOS. Printer Prompt 3 (version 2.5) is a new version of that program giving a 64 character-per-line screen display. WYSIWYG, double width letters and an address block routine.

is awaiting the final fix.

"I couldn't believe my eyes when the postman brought the sack out," he said.

Comments after supreme Anne Marie Allen: "The bug must have been waiting since his subscription was delayed a year ago. Fortunately, he hasn't received 67 copies every month."

So if your Dragon User was a little late this month, check the postmark — it might have been to Lancashire.

News desk

If you have any new products for the Dragon — software or hardware — ring the News Desk on 01-437 1433

New company sells fifth generation

METASOFT is a new software house launched by Chris Jolly, who's already well known for his work on Dragon software.

Metasoft's first production is *Prolog*, which runs under OS-9 on a Dragon 16.

It is "an extremely powerful

5th generation language for applications demanding built-in intelligence".

Prolog programs consist of facts, rules and relationships, and execution consists of a controlled deduction through the facts and rules to arrive at a conclusion.

Sales programmer Alan Cartmeliver "the programmer

who comes to *Prolog* after using a classical language will experience the same sort of revelation as someone who moves from arithmetic to the first steps of algebra."

Prolog costs £29.95, including post and packing and an 180 page manual, from Metasoft at 4 Pinewood Walk, Ovington, Kent BR6 6DD. —

Personal privacy as a matter of concern was ranked fourth after law and order, unemployment and education. People were most concerned about the misuse of financial information and medical records.

Surprisingly perhaps, considering the publicity which is given to these issues from time to time, there was a high level of trust in the police, health services, and banks, while mail order companies and credit reference agencies were regarded with more suspicion.

While we must assume that the DPP's effort wants to show a level of public support for the Act after the consternation that it has caused among computer data users, there is a doubt that people are concerned that information about them might be stored and used without any means of cross checking its accuracy.

If the Data Protection Act can prevent this to any degree, it will be doing a good job.

Reprocess

Dr. Anthony Daniels writes to say that there is a small bug in last month's *Graphics Screen Store/Processor*.

The correct line is:

```
578 IF PEEK(341)=247 AND
L5=1 THEN A6=6:R=L3:
S=L5:R=0:G=0:R=0:G=0:
IF A6=1 THEN M3=A:
M3=L5:R=0:G=0:R=0:G=0:
X=Y:POKE M3=X,143:
H6=X:G=0:G=0:R=0:G=0:
Y=H6:M=L3:X=X-4:R=0:
PPEM1 G=0:R=0:G=0:
PPEM1 G=0:R=0:G=0:
00000000 ELSE 500
```

Data protection survey reveals public approval

A survey commissioned by the Data Protection Registrar shows that seven out of ten people surveyed wanted to be able to gain access to information about themselves held on computer files. There was also support for companies changing access to data on request.

Most interestingly, a majority of those asked wanted to see

legislation to control the type of data which can be kept, to establish the right to have personal data added or removed from a file where relevant, and to extend these protections to paper records as well as magnetic records. The most important right was felt to be that of inspecting records, and having them corrected where necessary.

Dragonsoft

New software for review should be sent to Dragon User
12-13 Little Newport Street, London WC2E 6PP.

Great modem, shame about the software

Package: Dragon 88 Phone Modem 2800 (cable package)
Supplier: Haydon Electronics, 28 Laurel Drive, Willesden, S. West M4 1TW
Price: £45 including post and packing. Disc or tape version available.

WHEN a complete communications package, leads and all, is on offer at this price, the first thing you ask yourself is "Where's the catch?". Well, I have to say that after two months, I have not yet found one.

The modem is a very neat Prism 8080, which operates in 1200/2400 (V22) mode or, for one way communication, 1200 baud half duplex.

All the input/output sockets are at the rear leaving the front clear for the three LEDs and the "make and take" buttons. The three indicators are for Make On, Line On and Signal, which

shows when a carrier wave is established, and information can be exchanged.

Using the system could not be easier. Once the software has loaded, it is simply a matter of dialling the relevant number, using an ordinary phone plugged into the back of the modem. As soon as the high pitched carrier tone is received, press the data button and replace the telephone receiver. You could then be online, and receive data from the host computer. The telephone can be left plugged in and will still function normally when the computer is not using the line. The hardware works very well, I spent admirably without noisy phone line while connecting the incoming data. In fact the only time this has occurred was when someone tried to use the extension.

The software consists of four programs on a cassette and is the weak point of the whole

package. The instructions can be described as vague and the programs themselves have an amateur feel to them.

One example of this is in the 'V1200DATA' program. If you are not going to provide a save to tape routine, and I agree it is impractical, then there should be a DCCM detection routine to stop the system crashing when a tape user presses the wrong key. Likewise, in the bulletin board program, what excuse can there be for allowing the buffer to overflow and corrupt the program? A 'buffer full' routine which rerouted the user back to the main menu could have saved a lot of aggravation.

Even the MENU program has a problem; it lacks a CLEAR command. Any attempt to load back the information stored on disc without first running the ASCII routine results in a system crash. There is also no provision for reviewing saved V1200DATA screens. These may be fairly

insignificant faults to an experienced programmer, but to anyone relying solely on the software supplied, it could be enough to make them give up altogether.

On the plus side, Tim Haydon is always ready to enter into correspondence to sort out your problems and, so far as backup is concerned, you will not find better. He moved heaven and earth to find me a F00E to cure the problems I experienced with my SUPER-DOS, and helped me out with finding local boards.

Taken overall, the package represents outstanding value for money, the price really is unbelievable. If only the software had been a little more professional, it would have been truly superb.

Ken G. Smith



The April Answer

Remember how we couldn't fit last month's Answer in? Well, here it is. Along with some of their poetry . . .

Starter	Final Score	Cube root of X
20	4096	16
1	1	1
16	15625	25
4	1728	12
12	512	8
8	216	6
15	27	3
15	343	7
2	64	4
17	1000	10
3	64	4
19	13000	20
7	125	5
18	5832	18
9	8	2
11	1000	10
14	2744	14
6	625	25
12	4096	16
5	343	7

be achieved if it is realized that on the first circuit of the board, from each starting point, there is a series of twenty ascending subtotals. For example, from sector 20 the series would run 20, 21, 26, 43, 64, . . . and so on up to a total of 2744 after a circuit of the board. Each subsequent circuit of the board would result in running totals of these scores plus successive multiples of 270.

Therefore it is simpler, and mathematically a lot safer, to take an ascending series of values and find each one by determining a possible target score in the series.

$$270n = 4096$$

where n is any integer (including zero) and 4096 is one of the subtotals for each starting score.

In the program, the array sector holds the 20 scores from the card-based in clockwise sequence. These are read from the DATA lines. The loop (FOR) takes each of the twenty starting scores in turn and computes the twenty subtotals from the first circuit of the board. These are held in array RESIDUE (lines 110 — 165).

An ascending series of values is then tested to determine if it could be the opposite score. This is done by dividing the cube by

270 and taking the remainder. If this remainder is one of the residues in the array then a possible result has been found. In practice, the remainder is found by repeatedly subtracting 270 from the number under test.

```

10 DIM SECTOR(20), RESIDUE(20)
20 RESIDUE=0
30 FOR S=1 TO 20 READ SECTOR(S)
40 GOTO 100
100 FOR START=1 TO 20
110 P=SECTOR(S)
120 RESIDUE(P)=0
130 FOR I=1 TO 20
140 RESIDUE(P+I)=P
150 IF P=0 THEN P=SECTOR(S)
160 P=P+270
170 IF SECTOR(S)=P THEN PRINT
180 NEXT I
190 P=0
200 IF SECTOR(S)=P THEN PRINT
210 P=P+270
220 P=P+270
230 P=P+270
240 P=P+270
250 IF SECTOR(S)=P THEN PRINT
260 NEXT S
270 DATA 20,1,16,4,12,6,15,21,5,17,
280 DATA 3,14,9,18,7,11,8,10,2

```

Solution

The final totals can be obtained by repeatedly summing up the scores as described, checking the running total at each stage to see if it is a perfect cube. However, for the larger scores this could be a very lengthy process. A faster result can

The winners (and non-winners) of the April Competition came up with some rilly amazing poems. Barely daunted by the fact that they had to rhyme with 'tangle', our came the Thesauri (a distant relative of the Dragon) and the poetsavers.

Ode to despair

My wires are in a tangle
My dongle will not tangle
And life's one-tangled tangle
While I'm looking for the angle
To this competition tangle.

Alan Thomas

The stacks of ribbon cables too
They'd make a lovely tangle
Next help from Alan, I'd love to have
The buffers as they tangle.

The bus has lost control it seems
It makes an awful tangle
But pointers from an OS FROM
Still part of any tangle.

Two ships are dry, they ought to be —
I've put them through a tangle
Which also wrecks my faith in HEX
When fashioned like rectangles.

My Dragon's roar is muted now
I'll use my last try-angle
For me the prog, for it the prog,
That's in my greatest tangle.

Fred Wilton

As his new bill fell down
"I guess it will just have to tangle."

Tony Pithy, who saves the world the spiral.

Has bin

A Dragon I thought with a grin
Will help me these puzzles to win.
But their logical tangle
Just makes my brain tangle.
Well, just one more try and then — bin.

Mr. Armstrong

Dreadlock

Let us tangle with a tangle
Or tangle with a tangle
Those who come from us such verse
Deserve this curse or something worse.

AD Dreadlock, who ran his poem straight on from his solution, and nearly lost points for writing such a long poem before we realised that it generated the poem as well!

Gentlemen, the Vogens will be at your doors in a week.

Keep it short

A long man am I
And stuck in a tangle.
Better perhaps
Than stuck in a tangle!

Richard Lang

Epic

My poor old Dragon's all confused
It's really in a tangle
It's full of bits, the keyboard fits
As such a funny tangle.

Criss Cross Crash

There was a young Scot from Kilmange
Whose braces got caught in a tangle.
He said with a frown

Puzzling over FORTH

J. B. Singer investigates the powers of the FORTH language
by untangling some past Gordon Lee puzzles

WHEN I heard that Sir Cudge was writing an article on the FORTH language, I thought that it might interest some readers to read about my experience with the language. I said that I only have a limited experience and thus my viewpoint is not far removed from that of a reader who might be wondering whether to take up the language.

I bought a cassette version and as such it does not have the facilities of the full FORTH system but it has given me a great deal of fun. Accounts of FORTH usually pay great attention to the use of the STACK, often to the extent of failing to mention that there are facilities for the values of variables and constants (as in BASIC). It is, in fact, quite easy to translate a BASIC program directly into FORTH and this is how I started.

I had to remember that FORTH uses integer arithmetic: at first this was a frightening concept, but it has not turned out to be a real problem. In fact, it has often been beneficial as I will illustrate in the examples below but before doing so I will return to a brief comment on the use of the STACK.

In my (limited) experience, most of the fun in using FORTH comes from careful planning to use the STACK to the limit of one's nerve at the time, making judicious use of variables to simplify stack manipulation.

Systematic paperwork is crucial. I favour working in columns on A4 paper. In the left hand column I write the stack contents, with the most accessible item to the right (that is to say nearest to the centre of the page). Incidentally, it is conventional to describe the most accessible item as being on 'top' and I shall do so in the centre column, in the next line, I write the operating instructions. In the left hand column of the third line, I write the stack contents after the instructions have been carried out. The effect is that each group of instructions has the 'before' and 'after' stack contents shown above and below it. If there are RETURN STACK instructions (see below), they are written in the right hand column with the 'top' item nearest to the centre of the page; most often the Return Stack is used as a temporary store for items from the normal stack so this orientation is logical in use. It is as if the items flip across the centre of the page. Alternatively the third column is used for remarks. The examples that I have chosen are from Mr Lee's puzzle page.

June puzzle

This was to find a four-figure number which repeated itself after four cycles: in each cycle, the number has to be squared and the central four digits have to be isolated for the next cycle. The key routine is that which isolates the central digits. In BASIC it can be done with 'string' functions or by division and taking the integer parts of the quotient or remainder (as appropriate). FORTH arithmetic is simpler because it is, by definition, in integer form. Here, the square has to be held as a 'double number' (or a 32-bit number) because it is too big for a normal 16-bit FORTH number. For this reason I used the operator `U` which multiplies two single length numbers to give a double length one. The procedure for manipulating the square is as follows:

Notes:

The article by Brian Cudge, *Dragon FORTH*, appeared in Oct/February 1981. Brian Cudge was using *ADRAM FORTH*, available from the National Dragon User-Group, 41 Alameda Road, Worthing, Sussex. Tel: Worthing 203445. J.B. Singer is using *Deafstar* (an cassette from Microdroids) on a Dragon 32 without the disk controller in place. A page of programs used in the three examples is printed at the end of the article.

CYCLE: The purpose is as described. It needs an initial single length number on the stack and finishes with another single length number in its place.

STACK	OPERATION	REMARKS
seed	<code>CYCLE</code>	initial value
seed	<code>DUP U</code>	beginning no change
square		duplicate & square
remainder & quotient	<code>180 MMOD</code>	double length divides by 180
quotient	<code>NOT DROP</code>	single length double length
remainder & quotient	<code>1000 MMOD</code>	drop remainder double number div. by 1000
remainder	<code>2048P</code>	drop quotient
		new seed
		end of routine

As will be shown below `CYCLE` is part of a larger routine. In that sense `CYCLE` is a sub-routine but it is also a complete program. The authors of FORTH recognised that there was a problem of nomenclature; they kept their options open by giving the name `WORD` to refer to such mini-programs.

Before moving on to talk about the larger routine, it is prudent to check that `CYCLE` does do what it is supposed to do. There are two things to check. The first is whether it does, in fact, isolate the central four digits of a square. The second may be unexpected to a BASIC programmer but it is vital to FORTH: does the routine leave 32-bit numbers on the stack, or does it deal numbers of the stack? If either happens, it is a disaster.

The way to check `CYCLE` is to type a number, a space, then `CYCLE`, a space, a dot (which means 'print a number'), and then `ENTER`. The computer should display the answer and its accuracy can be checked for a whole range of numbers. It is necessary to check two and three digit numbers because such can arise (because of leading zeros) in the main routine. The stack effect can be checked by typing a dummy number before the seed, repeating the test, and then typing another 'ver'; all being well the computer should display the dummy number.

Assuming that all is well with `CYCLE`, the main word `JUNE-PUZZLE` can be developed as follows:

JUNE-PUZZLE: Find numbers which are repeated after 4 cycles. It does not need a number on the stack and does not leave one behind.



STACK	OPERATION	REMARKS
empty	: JUMP-PUZZLE	start
empty	10000 1000 DO	(start of loop (the FOR I = 10000 to 9999)
empty	10	(3 copies of (loop counter see above)
counter test result	CYCLE CYCLE CYCLE CYCLE	
counter test result	=	(tests if 2 top (numbers equal)
empty	IF "YES" OR	
empty	ELSE DROP	(print answer when drop counter)
empty	THEN LOOP	(end of it else RECALL-STEP
empty		end of word

The "word" takes about five to twelve minutes. It prints (between sections, but twice) can be eliminated by visual inspection using some additional information given in the original article. The routine could be improved by use of the JUSTPUZZLE routine instead of the DO LOOP. As a final comment, it is easy to life a similarity with Basic but the Forth CYCLE is much faster than a Basic sub-routine would be. There is very little stack manipulation because very little is needed. There is more in the next example.

July puzzle

This involves division to 4080 odd decimal places. Each successive digit in the quotient is used as a direction indicator for a step. At first sight, the integer arithmetic of Forth would appear to be at a disadvantage compared with the floating point arithmetic of Basic but the reverse is true. With Basic, one has to take great care to avoid rounding-off errors by using a sequence such as:

```
DIGIT = INTEGER OF DIVISION
REMAINDER = DIVIDEND - DIGIT * DIVISOR
NEW-DIVIDEND = INTEGER OF 10 * REMAINDER
```

In Forth, the sequence

OLD-REMAINDER TO 1080 1000

gives the new remainder and the digit directly onto the stack. The digit is on top and available for determining the movement. The procedure is long division.

A easy to describe the movement into that of a graph. A horizontal (East-West) movement is a change in the X-coordinate, a vertical (north-south) movement is a change in the Y-coordinate, and a diagonal movement is a change in both.

Imagine that the initial coordinates were 0,0 and wished to see if either had reached 50 after each move. Note that the coordinates were not variables as in Basic, they were simply numbers on the stack distinguished only on my piece of paper by their relative position on the stack.

In designing the word, the most important decision was the order of the various numbers on the stack. The procedure involves moving top numbers from the normal stack to the Return Stack to gain access to less accessible numbers as in the game Towers of Hanoi; this is said to be much quicker than putting numbers into storage as variables.

The symbols for transfer to and from the Return Stack are R: and R: (imagine arrows). There is also an instruction to make a copy of the top numbers from the Return Stack to the top of the stack: : (it is R: in my version of Forth but I believe that R:) is more usual.

The consequence of this is that the part of the word which has the return stack transfers is difficult to break up into small pieces for testing purposes. So another technique has to be used: this involves use of the "Screens". These are a feature of Forth whereby it is possible to type a word into a screen where it holds a typed initial form. The contents can be "loaded" (or compiled), and tested. Afterwards the screen's contents can be recalled and edited as necessary. This is a most useful feature. In the present instance, one could print the loop structure and the long division initially. The rest could be typed in the body of the screen, and checked, that the remainder could be added and the whole loaded and checked.

The word is in six parts. The first part merely puts a word remainder and 3 zeros onto the stack. From the top, the zeros are the X-coordinate, the Y-coordinate and an optional counter. The second, third, and fourth parts of the word are in a loop and are repeated until one of the coordinates reaches an absolute value of 50. In the second part, the coordinates are shifted onto the Return Stack. The counter is incremented and shifted across also. These actions expose the old remainder as the top element of the stack. The long division is the third part of the word. It leaves a new remainder and a quotient on the stack. The latter is on top and handy for determining the direction of the step.

The next major part of the word is a whole series of IF ELSE THEN tests. It is necessary to make a duplicate copy of the quotient for each test; when a match is found, the original quotient is dropped as no longer required. Depending on the value of the quotient, the counter and the coordinates are brought back onto the stack and changed as required.

It is usual to show a series of IF ELSE THEN clauses in a formatted form so that one can check which ELSE matches which THEN.

However in this case I decided to use a different presentation to make it easier to see how the routine works and, more important, to make it easier to check the stack effect. Each case must leave the Return Stack empty, and the stack with, from the top, X-coordinate, Y-coordinate, counter, new remainder.

In the fifth part, the two coordinates are duplicated, OVER OVER, and the copies are tested to see if either has reached the critical value. If so the result is printed, if not the calculation is repeated. The presentation of the result is not as specified in the Dragon Over but can be converted into the correct form.

JULYPUZZLE: The routine does not require an initial value to be put on the stack, and does not leave one behind.

STACK	OPERATION	RETURN STACK (& remarks)
empty	: JULYPUZZLE 10000 0 0 0	(beginning (values)
word counter Y-coord X-coord	BEGIN	(begin loop)
no change	R: R: 1 + R:	COUNTER Y-X- COORDS
word = remainder remainder quotient	TO 1080 1000 remainder quotient	(long div'n (no change) (of RET. STACK)
	DUP 1 = IF DROP R: R: 1 + R:	
	ELSE DUP 2 = IF DROP R: R: 1 + R:	
	ELSE DUP 3 = IF DROP R: R: 1 + R:	
	ELSE DUP 4 = IF DROP R: R: 1 + R:	
	ELSE DUP 5 = IF DROP R: R: 1 + R:	
	ELSE DUP 6 = IF DROP R: R: 1 + R:	
	ELSE DUP 7 = IF DROP R: R: 1 + R:	
	ELSE DUP 8 = IF DROP R: R: 1 + R:	
	ELSE DROP R: R: R:	
	THEN THEN THEN THEN THEN THEN THEN THEN remainder counter 100000 X-coord	(empty)

STACK	OPERATION OVER OVER	RETURN STACK (see list)
as above + copies of coords		
as above except flag instead of copy of R-coord	ADD 50 =	(end X-coord)
	SWAP	(copy of Y-coord top)
remainder counter Y-coord R-coord flag flag	ADD 50 =	(rest Y-coord)
	OR	(either true?)
remainder counter Y-coord R-coord flag	UNTIL	(end of loop)
remainder counter Y-coord X-coord		(structure)
empty		(print out)
		(end of word)

Before leaving the description of the **LUCKPUZZLE**, it is worth explaining what "MULTO" does. It multiplies two single length numbers to a double length number which is then divided by a third (single length number); there would be an overflow error with the combination of " " and "MOD" as this only has a single length intermediate product (10 times 10000 would be too big for it).

August puzzle

This is a much trickier problem altogether. The idea is that the words **ORAGON** and **USER** represent two numbers. Each different letter is a different digit.

We've said that if the one is divided by the other, it will give a two figure integer quotient. The cube of this will produce an English word if its digits are substituted by the letters used in the original code-words.

Conceptually there are two ways to proceed. Either one can synthesise all possible **ORAGON** and corresponding **USER** words and check whether the product meets the criterion of different digits.

Although the first is easier to understand, it actually involves five times more combinations and hence takes five times longer. So I will describe the other method.

The key part is how to indicate whether a particular digit is being used or not. In Basic, one could use an array to store flags. The same is true of Fort, but one has to define the array first. In English, an array could be defined as a group of storage addresses starting at a known address. This means that any location can be described by quoting an index number from the starting address. In my version of Fort such a structure for 10 locations, can be defined by:

DIMENSIONABLE NAME IS ALLOT

UNARIABLE reserves storage for two bytes and **ALLOT** reserves storage for eight more. The start address is given by "NAME". So **NAME** = **OR** NAME **IS** **ALLOT** address of the sixth storage; more simply it is a number which equals the address about to be used as an address-in this article **NAME** could be almost anything, but I use **ZERO** because **ZERO** **IS** **ALLOT** is suggestive of the way in which it will be seen to be used.

The storages in the array are used to store flags to show whether the corresponding digit is being used. I use the number 45 to show vacant, as this is the ASCII code for " "; and the ASCII code for the letter equivalent to the digit for "occupied". Thus if U is equivalent to 4, then the address **ZERO** **IS** contains 45. The advantage of this system is that Fort has a most useful word **EMIT** which prints the letter equivalent to an ASCII code. Thus

45EMIT + CR GET

would print U. **CR** is the instruction to fetch (read) a single byte from an address. Jumping ahead to the ultimate decoding of the cube numbers, this is done by breaking the number into a list of digits and using each one to **EMIT** the corresponding letter. It will be noticed that I use ASCII codes for upper case U S R and I used lower case codes for the other letters.

This is because there are many devices jobs to be done for each value of the **USER** number and it is easy to write a word **PURGE** to remove the lower case codes from the array without affecting the upper case codes.

Two other general points need to be explained. First it is pointless to multiply a **USER** number by a 2 figure number which is too small to give a 2 figure quotient; the word **LOW-LIMIT** calculates a minimum multiplier.

The second point is that the most searching criterion for possible **ORAGON** numbers is to check whether the second digit is the same as the last digit of the **USER** number. By trial only 10% of the possibilities pass this test so it is efficient to make it the first in the testing routine.

The word **TEST** begins with the **USER** number and the multiplier on top of the stack. They are copied and multiplied to a double length number (using **UP**); for the purposes of reference I call it **DRAGON**. It is first divided by 10000 to break it into a remainder **ADON** and a quotient **DR** which is then divided by 10. The remainder is checked against the true value of R which is recalled from the variable **R-TRUE**. It is a reverse check. If the values are different, the test ends in failure. **ELSE** tests if further checks. In the event that any check after the first gives a satisfactory result, the appropriate storage receives the appropriate lower case ASCII code. If all checks are satisfactory, control passes to the word **DECODE**.

DECODE begins with a print-out of the multiplier, **USER**, **DRAGON**, and the cube. The cube is decoded as explained previously. A feature to note is that the order of the digits is reversed twice so that the word and result are right way round. Control then reverts to **TEST**.

The combination of **TEST** and **DECODE** makes sure that the array is **PURGE**d of lower case codes when necessary. The combined routine requires the **USER** number and the multiplier to be on the stack at the outset, and returns to the main routine without the multiplier. The overall stack effect is:-

other items, **USER**, multiplier (—)

other items, **USER**

The other items come from the number selection routine.

The number selection routine is four nested **DO-LOOP**s. The outer one selects the U digit and has first choice. It marks the array as explained. The other digits are selected one by one. In each case there is an **IF** test of whether the call in the array is occupied. **IF** it is the number is rejected. **ELSE** the array cell is reserved and control is passed to the next **DO-LOOP**. It leaves a copy of the digit on the stack which is used for calculation purposes and for releasing the array cell on the outgoing side of the loop. For the sake of clarity I show the **DO-LOOP**s separately but there is no objection to typing them in one structure.

After the R digit has been selected, the **ELSE** clause changes the variable **R-TRUE**, and calls the word **CALL** to calculate the value of the **USER** number. Then it calls the word **LOW-LIMIT** to calculate the minimum value of the multiplier.

The main bulk of the routine is done in a central **DO-LOOP**. The loop counter is placed on the stack and the word **TEST** is called. The workings of **TEST** and its dependent words **PURGE** and **DECODE** have already been described. The whole routine is summarised in a diagram using a sort of "pigeon English/Fort". Elements of each structure are directly underneath each other to show how they fit together. Such an overview is useful because the sub-words have to be typed first. Even before the sub-words, one defines the three variables as follows:

UNARIABLE ZERO IS ALLOT

UNARIABLE R-TRUE

UNARIABLE ? DIGITS

STACK	OPERATION	REMARKS
usermult,ag,addrflag	IF DROP DROP DROP PURGE ELSE NO SWAP C	(reserve)
usermult,ag	HMADD SWAP	o on top
usermult,ag	ZERO + DUP C@ 58)	calc add (occupied?)
usermult,ag,addrflag	IF DROP DROP DROP PURGE ELSE 11 SWAP C	(reserve)
usermult,ag	11 MOD SWAP	g on top
usermult,ag	ZERO + DUP C@ 58)	calc add (occupied?)
usermult,ag,addrflag	IF DROP DROP DROP PURGE ELSE 10 SWAP C	(reserve)
usermult,ag	10 MOD DROP ZERO + DUP C@ 58)	o on top calc add (occupied?)
usermult,addrflag	IF DROP DROP PURGE ELSE 17 SWAP C DECODE PURGE THEN THEN THEN THEN THEN THEN	(reserve) see text
user		word end

R-SELECT Selects the R-digit, & places on stack. Uses it to calculate address and look at contents. If address occupied, digit is rejected. ELSE it is rejected by ASCII code for R. Changes variable RTRUE ; calls CALC to calculate the "USER" number. Calls LOWLIMIT to calculate the minimum value of the multiplier. For every value of the multiplier, places the loop counter on the stack. At this point the top elements on the stack are a copy of the R-digit, the user number, and the loop counter. Then TEST is called for each value of the multiplier. Afterwards when the loop has been completed, the USER number is dropped ; the array address is released using the value of the R digit from the top of the stack.

STACK	OPERATION	REMARKS
user	: R-SELECT 10 0 DO I ZERO + DUP	initial values (start) (loop-counter) (calc add)
user,ag,addrflag	C@ 58)	(occupied?)
user,ag,addrflag	IF DROP DROP ELSE 82 SWAP C	reject it reserve it
user,ag	DUP RTRUE 1	18 variable
no change	CALC	user number
user,loop	LOW LIMIT 180 SWAP	limits (for loop)
user,loop,low-limit	DO 1 TEST LOOP DROP	inner loop see TEST
user,ag	ZERO +	calc. add.
user,addr	45 SWAP C	free number

U.S.E.	THEN LOOP ;	end of word
--------	-------------	-------------

E-SELECT As R-SELECT picks next free digit and reserves it by storing 89 (ASCII code for E) in the array. Places value of digit on the stack; calls R-SELECT. Uses the value on the stack to release the array address. E-SELECT is identical except that it stores 89 in the array and calls E-SELECT.

STACK	OPERATION	REMARKS
U.S.	: E-SELECT 10 0 DO 11 ZERO + (DUP C@ 58)	already there start (loop counter) calc add (occupied?)
U.S,addrflag	IF DROP DROP ELSE 88 SWAP C	reject it reserve it
U.S,R	R-SELECT	see above
U.S,R	ZERO + 18 SWAP C	release it
U.S.	THEN LOOP ;	end of word

U-SELECT Reserves a digit between two and nine by storing 88 (ASCII code for U) in corresponding address of array. Places digit on stack. Calls E-SELECT. Afterwards uses the digit to release the address of the array. The overall stack effect is neutral.

STACK	OPERATION	REMARKS
empty	: U-SELECT 10 0 DO I	start of word start loop
counter = i, counter	ZERO +	calc add.
U,addr	88 SWAP C	(reserve)
U	E-SELECT	see text
U,		ZERO + 18
SWAP C		
empty	LOOP ;	(free add) (end of word)

AUGUST PUZZLE Main body of routine. It empties the array and calls U-SELECT. All the routines have already been defined and call one another as required with the result that the final definition is very brief.

STACK	OPERATION	REMARKS
empty	: AUGUST PUZZLE ZERO 10 45 FILL CR	see above
empty	U-SELECT ;	end of word

The word takes about 50 minutes which compares very favourably with the running time of its BASIC counterpart. The latter took between three and four hours.

Concluding observation, I have had a great deal of fun planning the programs described above and I hope that I have interested one or two other readers to try FORTH. In writing the account I have drawn heavily on the Manual that came with my cassette and upon a book called The Complete Forth by Alan Winkler, published by Sigma Technical Press. I should like to thank the authors of the two sources.


```

: CYCLE DUP 0# 100 R-MOD ROT
DROP 10000 R-MOD 2DROP ;
: JUNE-PUZZLE 10000 1000 DO
  | | | CYCLE CYCLE CYCLE CYCLE
  = IF , , " YES " OR ELSE DROP
  THEN LOOP ;

: ALLT-PUZZLE 10000 0 0 0 BEGIN
  >R >R 1+ >R 10 17300 4/MOD
DUP 1 = IF DROP R> R> 1 + R>
  ELSE
DUP 2 = IF DROP R> R> 1 + R> 1 +
  ELSE
DUP 3 = IF DROP R> R> R> 1 +
  ELSE
DUP 4 = IF DROP R> R> 1 - R> 1 +
  ELSE
DUP 5 = IF DROP R> R> 1 - R>
  ELSE
DUP 6 = IF DROP R> R> 1 - R> 1 -
  ELSE
DUP 7 = IF DROP R> R> R> 1 -
  ELSE
DUP 8 = IF DROP R> R> 1 + R> 1 +
  ELSE DROP R> R> R>
THEN THEN THEN THEN THEN THEN
THEN THEN OVER OVER ABS 50 =
SWAP ABS 50 = OR UNTIL

```

```

1 1 1 2
2 LOAD IS
0 VARIABLE ZERO 0 ALLDT
0 VARIABLE R-TRUE
: PURGE 10 0 DO 1 ZERO + DUP
  C0 00 < IF DROP ELSE 45 SWAP
  C1 THEN LOOP ;
: CALC 3R >R >R DUP 1000 0 0 100
  1 + R> SWAP R 10 1 + R> SWAP
  R + R> SWAP ;
: LOW-LIMIT 1000 OVER 100 SWAP
  R> ;

```

```

0 VARIABLE YODIGITS

: DECODE DUP , OVER DUP , OVER
  0# D.
DUP 32 < IF 4 YODIGITS 1 ELSE
DUP 47 < IF 5 YODIGITS 1 ELSE
5 YODIGITS 1 THEN THEN
DUP DUP 0 0# 2DUP D.
YODIGITS 0 0 DO 10 R-MOD LOOP
2DROP
YODIGITS 0 0 DO ZERO + C0 EXIT
LOOP OR ;
3 LOAD IS

```

```

: TEST OVER OVER 0# 10000 R-MOD
DROP 10 /MOD SWAP R-TRUE 0 =
ABS 0 >
IF DROP DROP DROP
ELSE ZERO + DUP C0 50 >
IF DROP DROP DROP
ELSE 100 SWAP C1 10 /MOD SWAP
ZERO + DUP C0 50 >
IF DROP DROP DROP PURGE
ELSE 110 SWAP C1 10 /MOD SWAP
ZERO + DUP C0 50 >
IF DROP DROP DROP PURGE
ELSE 111 SWAP C1 10 /MOD SWAP
ZERO + DUP C0 50 >
IF DROP DROP DROP PURGE
ELSE 103 SWAP C1 10 /MOD DROP
ZERO + DUP C0 50 >
IF DROP DROP PURGE
ELSE 07 SWAP C1
DECODE PURGE THEN THEN THEN
THEN THEN THEN ;
: R-SELECT 10 0 DO 1 1
  ZERO + DUP C0 50 >
  IF DROP DROP ELSE 02 SWAP C1
  DUP R-TRUE 1 C0 0 LOW-LIMIT
  100 SWAP DO 1 TEST LOOP DROP
  ZERO + 45 SWAP C1 THEN LOOP 1

```

```

1 1 1 2
2 LOAD IS
: E-SELECT
10 0 DO 1 1 ZERO + DUP C0 50 >
IF DROP DROP ELSE 09 SWAP C1
R-SELECT
ZERO + 45 SWAP C1 THEN LOOP 1

```

```

: S-SELECT
10 0 DO 1 1 ZERO + DUP C0 50 >
IF DROP DROP ELSE 03 SWAP C1
E-SELECT
ZERO + 45 SWAP C1 THEN LOOP ;

```

```

: U-SELECT
10 2 DO 1 1
ZERO + 05 SWAP C1
S-SELECT
ZERO + 45 SWAP C1 LOOP ;

```

```

1 1 1 2
2 LOAD IS
: AUGUST-PUZZLE
ZERO 10 45 FILL OR
U-SELECT ;

```


Winners and Losers

Every month, Gordon Lee will look at some prize programming points from a previous month's competition.

W1 was positively received under with replies to the March competition, many from readers entering for the first time—as who said it couldn't be done! Admittedly, the problem was a bit easier than many as requested by a number of Dragon Users who had thought that our usual competitions were a bit on the steep side. In fact, a number of readers found the problem a bit simpler than I had intended—An echo format! But more on that later.

The only real difficulty that I can envisage was the 'ghost' space which is placed at the beginning of a string variable which is created by the computer when utilizing the STR\$ command. Thus `X$=STR$(3)` will produce a string variable, X\$, which is 40 characters long, the first character of the string being a space in which there is room for an invisible + (positive) sign. In relation to the competition, if string variables are created in order to store the digits in certain positions, it has to be remembered that the first digit of the number is in the second position of the string. The easiest way of dealing with the problem is to remove the leading space immediately that the string has been created, by the use of the line `X$=MID$(X$,2)`. This will then restore each digit in the string to its regular position. Only one entrant actually admitted having been caught out by this at first, after his original program failed to produce an answer.

Most readers realized that it was only necessary to test for numbers in the range 1000 to 3162—that is those numbers whose squares have seven digits. However, many readers discovered simple proofs that the upper limit need be no greater than 1999, the final digit of the square must be 1; and that the final digit of the number is not greater than 5 (I can do no better than quote from G.A. Hunt of Cardiff who writes:

"Let M be the four figure number and P be its square. Then since P is a seven figure number its square root must be less than 3162, and configurations of P must be 1, 2, or 3. But only 1 is a possible last digit for a square. So the first figure of M must be 1. From this it follows that the first figure of P must be 1, 2, or 3, and hence the final figure of M must be 1, 2, or 3. A perfect square ending in 1 can only be produced by being the square of a number ending in 1 or 9; hence the last digit of M must be 1 or 9."

This reduction in the number of possibilities requiring to be tested was a frequently expressed idea, with many entrants giving the running times of programs. The following is a representative sample:

John Gomerally: 3 hours
Haji Blahm: 2 hours
H. Crowther: 8.75 minutes
Simon Hargrave: 49.5 seconds
James Benfield: 38 seconds
T.H. Denton: 1006 seconds



[The "+" denotes the use of the 'typed space' in the program.]

The apparent simplicity of the competition certainly brought out the competitive spirit in many readers who, flustered with heady realization of so simple a task, came up with a variety of novel approaches. From H. Crowther of Rotherham came a solution in only one program line, while Chris Fry of West Wickham found a couple of alternative solutions were possible if the numbers were reversed in the grid. Fibre squares is reversed the only solution is `N=1001`, while if `N=2020` the grid centred off H is reversed and is entered as 5262.

Regular competitor Paul Weedon decided to achieve the conventional IF `MID$(X$,2)` = `MID$(X$,1)` approach, and, with the wisdom standardised on who prefers to avoid despite the fact signs are provided, came up with an alternative program utilizing the powerful INSTR function. For readers who are not familiar with this command which searches for a

'target' string within a longer string, Paul has provided the following analysis of his routine:

It is the number and JS lefts square. Line 4 checks for the occurrence of the first digit of JS starting from the third position in B. If it is not there it goes to the NEXT 1, otherwise it passes on the lines 5, 6, and 7, which check for the other three digits. Line 7 is a cross-check which if the other requirements are satisfactory, will give the required answer.

This goes to demonstrate that there is more than one way to skin a cat—or is it 'boil an egg'? Line 1 in this program is the 'typed' space which should be omitted if your computer is not able to handle it. The INSTR command that Paul uses returns the position of the target string within the longer string. This was demonstrated as follows:

```
B$="ABCDEFGHIJKLMN"
Z=INSTR(B,"F").PRINT Z
```

This would give variable Z the value of 6 since the 'F' is at the 6th position in the string B. A third parameter can be specified when using this function if you wish to search for the 'target' starting from a position inside the strings to be searched. So, for example, `Z=INSTR(7,B,"F")` would start the search at character number 7 in the string; in the above example, the target 'F' would not be found and the computer will attribute a value of zero to variable Z.

Finally I would like to thank the following for their good wishes: Mark Griffin of Dulin, Chris Fry of West Wickham, Graham Barber of Sutton Colliery, Austin Henderson of Birmingham, and Lindsay Edwards, Chris Bouslet, Muri, and Paul the cat, of Huntingdon.

H. Crowther's solution is one line:

```
1:FORN=1000TO3162:IFN=STR$(N+N)=FOR
2:STR$(N+N)=N$D(4)N$,1,1)ANDN$D(4)=N$D(
3:NEXT:PRINTZ=8889)ANDN$D(2)=888(4)ANDN$D(4)=88(2
4)ANDN$D(5)=88(3)PRINTN$D(1)N$D(2)N$D(3)
5:PRINTN$D(4)N$D(5)N$D(6)N$D(7)N$D(8)N$D(9)
6:PRINTN$D(10)N$D(11)N$D(12)N$D(13)N$D(14)N$D(15)N$D(16)N$D(17)N$D(18)N$D(19)N$D(20)N$D(21)N$D(22)N$D(23)N$D(24)N$D(25)N$D(26)N$D(27)N$D(28)N$D(29)N$D(30)N$D(31)N$D(32)N$D(33)N$D(34)N$D(35)N$D(36)N$D(37)N$D(38)N$D(39)N$D(40)N$D(41)N$D(42)N$D(43)N$D(44)N$D(45)N$D(46)N$D(47)N$D(48)N$D(49)N$D(50)N$D(51)N$D(52)N$D(53)N$D(54)N$D(55)N$D(56)N$D(57)N$D(58)N$D(59)N$D(60)N$D(61)N$D(62)N$D(63)N$D(64)N$D(65)N$D(66)N$D(67)N$D(68)N$D(69)N$D(70)N$D(71)N$D(72)N$D(73)N$D(74)N$D(75)N$D(76)N$D(77)N$D(78)N$D(79)N$D(80)N$D(81)N$D(82)N$D(83)N$D(84)N$D(85)N$D(86)N$D(87)N$D(88)N$D(89)N$D(90)N$D(91)N$D(92)N$D(93)N$D(94)N$D(95)N$D(96)N$D(97)N$D(98)N$D(99)N$D(100)N$D(101)N$D(102)N$D(103)N$D(104)N$D(105)N$D(106)N$D(107)N$D(108)N$D(109)N$D(110)N$D(111)N$D(112)N$D(113)N$D(114)N$D(115)N$D(116)N$D(117)N$D(118)N$D(119)N$D(120)N$D(121)N$D(122)N$D(123)N$D(124)N$D(125)N$D(126)N$D(127)N$D(128)N$D(129)N$D(130)N$D(131)N$D(132)N$D(133)N$D(134)N$D(135)N$D(136)N$D(137)N$D(138)N$D(139)N$D(140)N$D(141)N$D(142)N$D(143)N$D(144)N$D(145)N$D(146)N$D(147)N$D(148)N$D(149)N$D(150)N$D(151)N$D(152)N$D(153)N$D(154)N$D(155)N$D(156)N$D(157)N$D(158)N$D(159)N$D(160)N$D(161)N$D(162)N$D(163)N$D(164)N$D(165)N$D(166)N$D(167)N$D(168)N$D(169)N$D(170)N$D(171)N$D(172)N$D(173)N$D(174)N$D(175)N$D(176)N$D(177)N$D(178)N$D(179)N$D(180)N$D(181)N$D(182)N$D(183)N$D(184)N$D(185)N$D(186)N$D(187)N$D(188)N$D(189)N$D(190)N$D(191)N$D(192)N$D(193)N$D(194)N$D(195)N$D(196)N$D(197)N$D(198)N$D(199)N$D(200)N$D(201)N$D(202)N$D(203)N$D(204)N$D(205)N$D(206)N$D(207)N$D(208)N$D(209)N$D(210)N$D(211)N$D(212)N$D(213)N$D(214)N$D(215)N$D(216)N$D(217)N$D(218)N$D(219)N$D(220)N$D(221)N$D(222)N$D(223)N$D(224)N$D(225)N$D(226)N$D(227)N$D(228)N$D(229)N$D(230)N$D(231)N$D(232)N$D(233)N$D(234)N$D(235)N$D(236)N$D(237)N$D(238)N$D(239)N$D(240)N$D(241)N$D(242)N$D(243)N$D(244)N$D(245)N$D(246)N$D(247)N$D(248)N$D(249)N$D(250)N$D(251)N$D(252)N$D(253)N$D(254)N$D(255)N$D(256)N$D(257)N$D(258)N$D(259)N$D(260)N$D(261)N$D(262)N$D(263)N$D(264)N$D(265)N$D(266)N$D(267)N$D(268)N$D(269)N$D(270)N$D(271)N$D(272)N$D(273)N$D(274)N$D(275)N$D(276)N$D(277)N$D(278)N$D(279)N$D(280)N$D(281)N$D(282)N$D(283)N$D(284)N$D(285)N$D(286)N$D(287)N$D(288)N$D(289)N$D(290)N$D(291)N$D(292)N$D(293)N$D(294)N$D(295)N$D(296)N$D(297)N$D(298)N$D(299)N$D(300)N$D(301)N$D(302)N$D(303)N$D(304)N$D(305)N$D(306)N$D(307)N$D(308)N$D(309)N$D(310)N$D(311)N$D(312)N$D(313)N$D(314)N$D(315)N$D(316)N$D(317)N$D(318)N$D(319)N$D(320)N$D(321)N$D(322)N$D(323)N$D(324)N$D(325)N$D(326)N$D(327)N$D(328)N$D(329)N$D(330)N$D(331)N$D(332)N$D(333)N$D(334)N$D(335)N$D(336)N$D(337)N$D(338)N$D(339)N$D(340)N$D(341)N$D(342)N$D(343)N$D(344)N$D(345)N$D(346)N$D(347)N$D(348)N$D(349)N$D(350)N$D(351)N$D(352)N$D(353)N$D(354)N$D(355)N$D(356)N$D(357)N$D(358)N$D(359)N$D(360)N$D(361)N$D(362)N$D(363)N$D(364)N$D(365)N$D(366)N$D(367)N$D(368)N$D(369)N$D(370)N$D(371)N$D(372)N$D(373)N$D(374)N$D(375)N$D(376)N$D(377)N$D(378)N$D(379)N$D(380)N$D(381)N$D(382)N$D(383)N$D(384)N$D(385)N$D(386)N$D(387)N$D(388)N$D(389)N$D(390)N$D(391)N$D(392)N$D(393)N$D(394)N$D(395)N$D(396)N$D(397)N$D(398)N$D(399)N$D(400)N$D(401)N$D(402)N$D(403)N$D(404)N$D(405)N$D(406)N$D(407)N$D(408)N$D(409)N$D(410)N$D(411)N$D(412)N$D(413)N$D(414)N$D(415)N$D(416)N$D(417)N$D(418)N$D(419)N$D(420)N$D(421)N$D(422)N$D(423)N$D(424)N$D(425)N$D(426)N$D(427)N$D(428)N$D(429)N$D(430)N$D(431)N$D(432)N$D(433)N$D(434)N$D(435)N$D(436)N$D(437)N$D(438)N$D(439)N$D(440)N$D(441)N$D(442)N$D(443)N$D(444)N$D(445)N$D(446)N$D(447)N$D(448)N$D(449)N$D(450)N$D(451)N$D(452)N$D(453)N$D(454)N$D(455)N$D(456)N$D(457)N$D(458)N$D(459)N$D(460)N$D(461)N$D(462)N$D(463)N$D(464)N$D(465)N$D(466)N$D(467)N$D(468)N$D(469)N$D(470)N$D(471)N$D(472)N$D(473)N$D(474)N$D(475)N$D(476)N$D(477)N$D(478)N$D(479)N$D(480)N$D(481)N$D(482)N$D(483)N$D(484)N$D(485)N$D(486)N$D(487)N$D(488)N$D(489)N$D(490)N$D(491)N$D(492)N$D(493)N$D(494)N$D(495)N$D(496)N$D(497)N$D(498)N$D(499)N$D(500)N$D(501)N$D(502)N$D(503)N$D(504)N$D(505)N$D(506)N$D(507)N$D(508)N$D(509)N$D(510)N$D(511)N$D(512)N$D(513)N$D(514)N$D(515)N$D(516)N$D(517)N$D(518)N$D(519)N$D(520)N$D(521)N$D(522)N$D(523)N$D(524)N$D(525)N$D(526)N$D(527)N$D(528)N$D(529)N$D(530)N$D(531)N$D(532)N$D(533)N$D(534)N$D(535)N$D(536)N$D(537)N$D(538)N$D(539)N$D(540)N$D(541)N$D(542)N$D(543)N$D(544)N$D(545)N$D(546)N$D(547)N$D(548)N$D(549)N$D(550)N$D(551)N$D(552)N$D(553)N$D(554)N$D(555)N$D(556)N$D(557)N$D(558)N$D(559)N$D(560)N$D(561)N$D(562)N$D(563)N$D(564)N$D(565)N$D(566)N$D(567)N$D(568)N$D(569)N$D(570)N$D(571)N$D(572)N$D(573)N$D(574)N$D(575)N$D(576)N$D(577)N$D(578)N$D(579)N$D(580)N$D(581)N$D(582)N$D(583)N$D(584)N$D(585)N$D(586)N$D(587)N$D(588)N$D(589)N$D(590)N$D(591)N$D(592)N$D(593)N$D(594)N$D(595)N$D(596)N$D(597)N$D(598)N$D(599)N$D(600)N$D(601)N$D(602)N$D(603)N$D(604)N$D(605)N$D(606)N$D(607)N$D(608)N$D(609)N$D(610)N$D(611)N$D(612)N$D(613)N$D(614)N$D(615)N$D(616)N$D(617)N$D(618)N$D(619)N$D(620)N$D(621)N$D(622)N$D(623)N$D(624)N$D(625)N$D(626)N$D(627)N$D(628)N$D(629)N$D(630)N$D(631)N$D(632)N$D(633)N$D(634)N$D(635)N$D(636)N$D(637)N$D(638)N$D(639)N$D(640)N$D(641)N$D(642)N$D(643)N$D(644)N$D(645)N$D(646)N$D(647)N$D(648)N$D(649)N$D(650)N$D(651)N$D(652)N$D(653)N$D(654)N$D(655)N$D(656)N$D(657)N$D(658)N$D(659)N$D(660)N$D(661)N$D(662)N$D(663)N$D(664)N$D(665)N$D(666)N$D(667)N$D(668)N$D(669)N$D(670)N$D(671)N$D(672)N$D(673)N$D(674)N$D(675)N$D(676)N$D(677)N$D(678)N$D(679)N$D(680)N$D(681)N$D(682)N$D(683)N$D(684)N$D(685)N$D(686)N$D(687)N$D(688)N$D(689)N$D(690)N$D(691)N$D(692)N$D(693)N$D(694)N$D(695)N$D(696)N$D(697)N$D(698)N$D(699)N$D(700)N$D(701)N$D(702)N$D(703)N$D(704)N$D(705)N$D(706)N$D(707)N$D(708)N$D(709)N$D(710)N$D(711)N$D(712)N$D(713)N$D(714)N$D(715)N$D(716)N$D(717)N$D(718)N$D(719)N$D(720)N$D(721)N$D(722)N$D(723)N$D(724)N$D(725)N$D(726)N$D(727)N$D(728)N$D(729)N$D(730)N$D(731)N$D(732)N$D(733)N$D(734)N$D(735)N$D(736)N$D(737)N$D(738)N$D(739)N$D(740)N$D(741)N$D(742)N$D(743)N$D(744)N$D(745)N$D(746)N$D(747)N$D(748)N$D(749)N$D(750)N$D(751)N$D(752)N$D(753)N$D(754)N$D(755)N$D(756)N$D(757)N$D(758)N$D(759)N$D(760)N$D(761)N$D(762)N$D(763)N$D(764)N$D(765)N$D(766)N$D(767)N$D(768)N$D(769)N$D(770)N$D(771)N$D(772)N$D(773)N$D(774)N$D(775)N$D(776)N$D(777)N$D(778)N$D(779)N$D(780)N$D(781)N$D(782)N$D(783)N$D(784)N$D(785)N$D(786)N$D(787)N$D(788)N$D(789)N$D(790)N$D(791)N$D(792)N$D(793)N$D(794)N$D(795)N$D(796)N$D(797)N$D(798)N$D(799)N$D(800)N$D(801)N$D(802)N$D(803)N$D(804)N$D(805)N$D(806)N$D(807)N$D(808)N$D(809)N$D(810)N$D(811)N$D(812)N$D(813)N$D(814)N$D(815)N$D(816)N$D(817)N$D(818)N$D(819)N$D(820)N$D(821)N$D(822)N$D(823)N$D(824)N$D(825)N$D(826)N$D(827)N$D(828)N$D(829)N$D(830)N$D(831)N$D(832)N$D(833)N$D(834)N$D(835)N$D(836)N$D(837)N$D(838)N$D(839)N$D(840)N$D(841)N$D(842)N$D(843)N$D(844)N$D(845)N$D(846)N$D(847)N$D(848)N$D(849)N$D(850)N$D(851)N$D(852)N$D(853)N$D(854)N$D(855)N$D(856)N$D(857)N$D(858)N$D(859)N$D(860)N$D(861)N$D(862)N$D(863)N$D(864)N$D(865)N$D(866)N$D(867)N$D(868)N$D(869)N$D(870)N$D(871)N$D(872)N$D(873)N$D(874)N$D(875)N$D(876)N$D(877)N$D(878)N$D(879)N$D(880)N$D(881)N$D(882)N$D(883)N$D(884)N$D(885)N$D(886)N$D(887)N$D(888)N$D(889)N$D(890)N$D(891)N$D(892)N$D(893)N$D(894)N$D(895)N$D(896)N$D(897)N$D(898)N$D(899)N$D(900)N$D(901)N$D(902)N$D(903)N$D(904)N$D(905)N$D(906)N$D(907)N$D(908)N$D(909)N$D(910)N$D(911)N$D(912)N$D(913)N$D(914)N$D(915)N$D(916)N$D(917)N$D(918)N$D(919)N$D(920)N$D(921)N$D(922)N$D(923)N$D(924)N$D(925)N$D(926)N$D(927)N$D(928)N$D(929)N$D(930)N$D(931)N$D(932)N$D(933)N$D(934)N$D(935)N$D(936)N$D(937)N$D(938)N$D(939)N$D(940)N$D(941)N$D(942)N$D(943)N$D(944)N$D(945)N$D(946)N$D(947)N$D(948)N$D(949)N$D(950)N$D(951)N$D(952)N$D(953)N$D(954)N$D(955)N$D(956)N$D(957)N$D(958)N$D(959)N$D(960)N$D(961)N$D(962)N$D(963)N$D(964)N$D(965)N$D(966)N$D(967)N$D(968)N$D(969)N$D(970)N$D(971)N$D(972)N$D(973)N$D(974)N$D(975)N$D(976)N$D(977)N$D(978)N$D(979)N$D(980)N$D(981)N$D(982)N$D(983)N$D(984)N$D(985)N$D(986)N$D(987)N$D(988)N$D(989)N$D(990)N$D(991)N$D(992)N$D(993)N$D(994)N$D(995)N$D(996)N$D(997)N$D(998)N$D(999)N$D(1000)N$D(1001)N$D(1002)N$D(1003)N$D(1004)N$D(1005)N$D(1006)N$D(1007)N$D(1008)N$D(1009)N$D(1010)N$D(1011)N$D(1012)N$D(1013)N$D(1014)N$D(1015)N$D(1016)N$D(1017)N$D(1018)N$D(1019)N$D(1020)N$D(1021)N$D(1022)N$D(1023)N$D(1024)N$D(1025)N$D(1026)N$D(1027)N$D(1028)N$D(1029)N$D(1030)N$D(1031)N$D(1032)N$D(1033)N$D(1034)N$D(1035)N$D(1036)N$D(1037)N$D(1038)N$D(1039)N$D(1040)N$D(1041)N$D(1042)N$D(1043)N$D(1044)N$D(1045)N$D(1046)N$D(1047)N$D(1048)N$D(1049)N$D(1050)N$D(1051)N$D(1052)N$D(1053)N$D(1054)N$D(1055)N$D(1056)N$D(1057)N$D(1058)N$D(1059)N$D(1060)N$D(1061)N$D(1062)N$D(1063)N$D(1064)N$D(1065)N$D(1066)N$D(1067)N$D(1068)N$D(1069)N$D(1070)N$D(1071)N$D(1072)N$D(1073)N$D(1074)N$D(1075)N$D(1076)N$D(1077)N$D(1078)N$D(1079)N$D(1080)N$D(1081)N$D(1082)N$D(1083)N$D(1084)N$D(1085)N$D(1086)N$D(1087)N$D(1088)N$D(1089)N$D(1090)N$D(1091)N$D(1092)N$D(1093)N$D(1094)N$D(1095)N$D(1096)N$D(1097)N$D(1098)N$D(1099)N$D(1100)N$D(1101)N$D(1102)N$D(1103)N$D(1104)N$D(1105)N$D(1106)N$D(1107)N$D(1108)N$D(1109)N$D(1110)N$D(1111)N$D(1112)N$D(1113)N$D(1114)N$D(1115)N$D(1116)N$D(1117)N$D(1118)N$D(1119)N$D(1120)N$D(1121)N$D(1122)N$D(1123)N$D(1124)N$D(1125)N$D(1126)N$D(1127)N$D(1128)N$D(1129)N$D(1130)N$D(1131)N$D(1132)N$D(1133)N$D(1134)N$D(1135)N$D(1136)N$D(1137)N$D(1138)N$D(1139)N$D(1140)N$D(1141)N$D(1142)N$D(1143)N$D(1144)N$D(1145)N$D(1146)N$D(1147)N$D(1148)N$D(1149)N$D(1150)N$D(1151)N$D(1152)N$D(1153)N$D(1154)N$D(1155)N$D(1156)N$D(1157)N$D(1158)N$D(1159)N$D(1160)N$D(1161)N$D(1162)N$D(1163)N$D(1164)N$D(1165)N$D(1166)N$D(1167)N$D(1168)N$D(1169)N$D(1170)N$D(1171)N$D(1172)N$D(1173)N$D(1174)N$D(1175)N$D(1176)N$D(1177)N$D(1178)N$D(1179)N$D(1180)N$D(1181)N$D(1182)N$D(1183)N$D(1184)N$D(1185)N$D(1186)N$D(1187)N$D(1188)N$D(1189)N$D(1190)N$D(1191)N$D(1192)N$D(1193)N$D(1194)N$D(1195)N$D(1196)N$D(1197)N$D(1198)N$D(1199)N$D(1200)N$D(1201)N$D(1202)N$D(1203)N$D(1204)N$D(1205)N$D(1206)N$D(1207)N$D(1208)N$D(1209)N$D(1210)N$D(1211)N$D(1212)N$D(1213)N$D(1214)N$D(1215)N$D(1216)N$D(1217)N$D(1218)N$D(1219)N$D(1220)N$D(1221)N$D(1222)N$D(1223)N$D(1224)N$D(1225)N$D(1226)N$D(1227)N$D(1228)N$D(1229)N$D(1230)N$D(1231)N$D(1232)N$D(1233)N$D(1234)N$D(1235)N$D(1236)N$D(1237)N$D(1238)N$D(1239)N$D(1240)N$D(1241)N$D(1242)N$D(1243)N$D(1244)N$D(1245)N$D(1246)N$D(1247)N$D(1248)N$D(1249)N$D(1250)N$D(1251)N$D(1252)N$D(1253)N$D(1254)N$D(1255)N$D(1256)N$D(1257)N$D(1258)N$D(1259)N$D(
```




The Show goes on

Late but lively, Ken Smith returns from the London 8809 Show

THERE are times when you can be really pleased to be wrong. After all, wasn't I the one who said that the last 8809 Show probably would be just that. Yet only four months later there we were for yet another time. The Dragon and its devotees never cease to amaze me.

It is interesting to see the style of the show changing. This one seemed to me to have a more technical bias than previous shows, with probably more bargains than ever.

I cited the last report telling of how I wanted a modern, only to find the substituted bare. Well here it is, this time my needs were supplied in the shape of a neat little Prism, supplied by the National Dragon User Group (I must join someday), complete with software for only forty pounds. Just proves my point, see? and ye shall find.

At the other end of the size scale Microsil had a huge one, undoubtedly 386-GP0 and bigger than the Dragon itself. Still at the silly prices they were asking you could almost use it as a combined modem and bench seat. Maybe that is what is meant by sitting on information.

Bargains for grabs

Bargains come in all shapes and sizes and plenty were up for grabs on John Peck's stall.

One was of particular interest to prospective OS-8 users, namely the operating system itself which John is selling at cost price.

Anyone who has seen the ads from the gentleman in Denmark who has the licence to produce Osix will know the typical price he charges. Well, John's was a fraction of that and his only condition is that he would like you to buy his other software from him. With Pascal at only seven pounds, where else would we get?

Games were taken up on the stand taken by Biosmart. A company that seems to deal mainly in adventure games, this was their first show and I hope to see more of them.

Unfortunately the larger houses and dealers seemed to get first bite at the

cherry and a lot of people were spent out before they found Biosmart, which is a pity, as they may never know what they missed. (They might if they enter the 8809 Java competition. Biosmart have done a some-classic versions as prizes. — Ed.)

Keyboard basics

I would have been in a similar position had it not been for Rob Harris.

Rob marketed an excellent BASIC48 with which I have been having a few minor problems which Rob thought could be down to my keyboard. This prompted me to go to the Peckish stand and try one of their replacement units. For those of you who, like myself, thought that the old Dragon 32 keyboard was good enough then you must try this.

The improvement is real and responsive is appreciable and it seems to have cured my problem. The trouble is it doesn't spell any better than the old one. (No later in my last conversation with Peckish revealed that they have sold the last one and can't obtain any more stocks. A pity as there is apparently a demand.)

Computape were offering a wide variety of software and peripherals all at knockdown prices (and in fact I can't remember a sale with so many people on offer).

Microsil had a bunch of Quickdos, to be sold on an "as seen" basis for one pound each. Some of their software was cheaper still. My youngest son managed to buy eight games for only four pounds.

Anyone with a little knowledge could have assembled a double drive disc system for a fraction of the normal price. NEOS had the drives. Computerer supplied the tapes and DOS. They also had a lot of the disc based software, mainly PLUX (a variant of course but as the Dragon DOS) and also some hardware, like an Epson burner and their famous Plus five (if that's you in the 128k league).

As I walked from stand to stand I was surprised by the enthusiasm still offered by both the makers and the customers. This long after the collapse of Dragon Data, you can't expect to see many new developments for these users. From conversions to

give you a built in speaker and inverse screen at the flick of a switch, is the new game Spectrum from Quickdos.

One big surprise is that there are still some Dragoners who don't subscribe to DL. (If you are reading a borrowed copy I hope you feel thoroughly ashamed of yourself. Dragon User is our strongest link and we must preserve it.)

The delightful Helen was there as promised, thrusting subscription forms at anyone who came within signing distance. Helen, by the way, did not disappear at twelve as I expected which explains the team of white mice and the saggy pumpkin I noticed on the way out. (The silly shape came to the wrong door. Well, I had to flag them passing by, I was driving a couple of cybicycles. — Ed.)

Revitalised

I had arrived at 12.00, when the doors opened, with a sinking feeling. Why? Well they poorly? — No, again, I left just after 1.30, broke but revitalised. Please, oh please, let there be another show in November. I will need another fix of enthusiasm. Just to be in the company of people who really do believe in the Dragon might stimulate going to work those other inferior machines.

When you hear quotes like "after all it is still the only home micro capable of running the new education standard, namely OS-8," from someone who really believes in what he is saying, it makes all the effort worthwhile. See you in November.

Although it looks doubtful that there will be another show at the conference halls, we have heard from at least two lots of people who are thinking of putting on smaller shows in the autumn at easily accessible venues out of London — avoiding the swingeing entry prices. The success of the latest show in April, shortly after the 8809 Show, demonstrates that this format can exploit to anyone's advantage. We publish any news as soon as we get it.

Down in the dumps

Dragon User frequently gets requests for screen dumps. Here we present some specimens from our collection.

OK, you guys. You asked for it. The response to our appeal for screen dumps has come in a flood. We can only

reproduce them in a book by comparison, but we're starting here. They aren't in any particular order, and they

Don't guarantee to work, but numbers of you should find what you are looking for eventually.

Lower EL-AE

This is written in assembly language for the Epson FX-88. The routine handles what I prefer to call a half-size preview of a PMODE4 (3) screen display using a two-pass technique, not unlike the hi. Output port.

College Publications Department

[illegible]

DEI-NO Microlina

This was very complicated to make, because you have to turn the screen to have it fit the paper. Secondly, the screen bytes and the printer bytes are configured differently.

1000

4271	W400	LOC	4000	4274	ST05	MODE	0000
4272	W400	LOC	4145-8	4275	ST05	MODE	0000
4273	W400	LOC	4145-8	4276	ST05	MODE	0000
4274	W400	LOC	4145-8	4277	ST05	MODE	0000
4275	W400	LOC	4145-8	4278	ST05	MODE	0000
4276	W400	LOC	4145-8	4279	ST05	MODE	0000
4277	W400	LOC	4145-8	4280	ST05	MODE	0000
4278	W400	LOC	4145-8	4281	ST05	MODE	0000
4279	W400	LOC	4145-8	4282	ST05	MODE	0000
4280	W400	LOC	4145-8	4283	ST05	MODE	0000
4281	W400	LOC	4145-8	4284	ST05	MODE	0000
4282	W400	LOC	4145-8	4285	ST05	MODE	0000
4283	W400	LOC	4145-8	4286	ST05	MODE	0000
4284	W400	LOC	4145-8	4287	ST05	MODE	0000
4285	W400	LOC	4145-8	4288	ST05	MODE	0000
4286	W400	LOC	4145-8	4289	ST05	MODE	0000
4287	W400	LOC	4145-8	4290	ST05	MODE	0000
4288	W400	LOC	4145-8	4291	ST05	MODE	0000
4289	W400	LOC	4145-8	4292	ST05	MODE	0000
4290	W400	LOC	4145-8	4293	ST05	MODE	0000
4291	W400	LOC	4145-8	4294	ST05	MODE	0000
4292	W400	LOC	4145-8	4295	ST05	MODE	0000
4293	W400	LOC	4145-8	4296	ST05	MODE	0000
4294	W400	LOC	4145-8	4297	ST05	MODE	0000
4295	W400	LOC	4145-8	4298	ST05	MODE	0000
4296	W400	LOC	4145-8	4299	ST05	MODE	0000
4297	W400	LOC	4145-8	4300	ST05	MODE	0000
4298	W400	LOC	4145-8	4301	ST05	MODE	0000
4299	W400	LOC	4145-8	4302	ST05	MODE	0000
4300	W400	LOC	4145-8	4303	ST05	MODE	0000
4301	W400	LOC	4145-8	4304	ST05	MODE	0000
4302	W400	LOC	4145-8	4305	ST05	MODE	0000
4303	W400	LOC	4145-8	4306	ST05	MODE	0000
4304	W400	LOC	4145-8	4307	ST05	MODE	0000
4305	W400	LOC	4145-8	4308	ST05	MODE	0000
4306	W400	LOC	4145-8	4309	ST05	MODE	0000
4307	W400	LOC	4145-8	4310	ST05	MODE	0000
4308	W400	LOC	4145-8	4311	ST05	MODE	0000
4309	W400	LOC	4145-8	4312	ST05	MODE	0000
4310	W400	LOC	4145-8	4313	ST05	MODE	0000
4311	W400	LOC	4145-8	4314	ST05	MODE	0000
4312	W400	LOC	4145-8	4315	ST05	MODE	0000
4313	W400	LOC	4145-8	4316	ST05	MODE	0000
4314	W400	LOC	4145-8	4317	ST05	MODE	0000
4315	W400	LOC	4145-8	4318	ST05	MODE	0000
4316	W400	LOC	4145-8	4319	ST05	MODE	0000
4317	W400	LOC	4145-8	4320	ST05	MODE	0000
4318	W400	LOC	4145-8	4321	ST05	MODE	0000
4319	W400	LOC	4145-8	4322	ST05	MODE	0000
4320	W400	LOC	4145-8	4323	ST05	MODE	0000
4321	W400	LOC	4145-8	4324	ST05	MODE	0000
4322	W400	LOC	4145-8	4325	ST05	MODE	0000
4323	W400	LOC	4145-8	4326	ST05	MODE	0000
4324	W400	LOC	4145-8	4327	ST05	MODE	0000
4325	W400	LOC	4145-8	4328	ST05	MODE	0000
4326	W400	LOC	4145-8	4329	ST05	MODE	0000
4327	W400	LOC	4145-8	4330	ST05	MODE	

Tandy CGP-115 plotter

They don't go anywhere for hours.

1000

41 LAST HERE IS A GARDEN GUT PROBLEM FOR ALL DRAGON DARTS AND POSSES A TARTY COP-113 A PEN PLOTTER PLOTTER. THIS IS ALSO A LATE RESPONSE IN REPLY TO REQUESTS ON THE columns 114 and 115 PAGE OF C. USER EDITIONS JUNE & OCTOBER 1980, BY C. CROFT OF EASTBOURNE BROVE, CROFT AND D.S. PHILIPS OF BRONSTONE ROAD SOUTH, CHESHIRE.

THE PROGRAM IS SET UP FOR MODE 4 WITH
EITHER SCREEN ALSO ACQUIRING THE 1ST 4
GRAPHICS PAGES ARE USED.
HERE ARE A FEW PROGRAM NOTES,
LINE 301-SETS PROMPT TO GRAPHIC PAGE
WITH THE ORIGIN SET AT 400,0
LINE 300-DISPLAYS SCREEN.

Answers

LINES 30-50--INVERT THE SCREEN ONLY IF THERE IS MORE BLACK THAN WHITE 50 SWAP ON YOUR BLACK PEN;
LINE 80--LOOP EQUAL TO THE NUMBER OF VERTICAL DOTS ON SCREEN. THIS MAY BE ALTERED FOR OTHER RESOLUTIONS.
LINE 70--AS DO BUT FOR THE NUMBER OF HORIZONTAL DOTS.

LINES 90--PUTS COLOUR OF DOT LOOKED AT IN THE TWO LOOPS INTO VARIABLE CFI.
LINES 100--PRINTS OUT A 2 DOT LENGTH LINE FOR EACH BLACK DOT ON THE SCREEN

NOTE THE COLOUR CODE MAY BE ALTERED FOR OTHER MODES, USUALLY BEST TO CHANGE TO THE MAIN COLOUR USED, AND DO NOT FORGET TO CHANGE YOUR PRINTER'S PEN COLOUR!

LINE 100--END OF HORIZONTAL DOT LOOP.

LINE 110--MOVES PAPER BACK READY FOR NEXT LINE TO BE PRINTED ALSO MOVES PEN.

LINE 120--END OF VERTICAL DOT LOOP.

LINE 130--FILLS REMAINDER DUMP TO YOU.

If you want a darker screenage than listed the following directly without a line number or remark.

For this to work correctly YOU SHOULD NOT have to save the paper by handfeed, you can repeat this as many times as you wish to obtain your required dump.

```
1 REM:0=10-2,"0"-2,"0-1,0"-2,"1-000 00
```

FURTHER NOTE THAT IN MODES 1&3 LINES 30-50 MAY BE EITHER REMOVED OR TURNED INTO REMARKS.

IN MODE 0 LINES 30&50 SHOULD BE:-

```
FOR K=1530 TO 2020:ETC. LINE 40 BECOMES  
IF 0=1530&0 THEN 00
```

IN MODE 2 CHANGE 30&50 TO BECOME

```
FOR K=1530 TO 4007:ETC. LINE 40 BECOMES  
IF 0=300000 THEN 00
```

```
00 PRINT0=2,CHR(16):PRINT0=2,"000,0"0  
PRINT0=2,"1"
```

```
20 PPOD0,1:SCREEN,1
```

```
00 FOR K=1530 TO 2000:0=0:PEEK(K):NEXT K  
40 IF 0=000000 THEN 00
```

```
00 FOR K=1530 TO 2000:0=K,255-PEEK(K)  
NEXT K
```

```
00 FOR Y=0 TO 100
```

```
20 FOR X=0 TO 255
```

```
00 P=PEEK(X),Y
```

```
00 IF P=0 THEN PRINT0=2,"00,-1" ELSE PRINT0=2,"00,-2"
```

```
100 NEXT X
```

```
110 0=0:2=PRINT0=2,"0"-5:0=0:0
```

```
120 NEXT Y
```

```
130 PRINT0=2,"000,-000"-END
```

Special Holiday Offers

Order any 3 of our £1.99 games
and you only pay

£5
plus 75p p. + p.

Discount

BORG THE WILD PURY ROSSA, STARWOLF ROMERO, MAYE SELL,
MODDUGALAST DING, THE BELLS, MURRAY, DEEPFRODO DING,
STARWOLF, PERDUS PT, JACOB, KINGDOM, MOPIS, MARCH,
GARY, BURGLES AND MESSON, ATTACK.

Only available during the month of August

Special Offer

JUSTICE OF THE PEACE

At £1 - only £2.99 incl. p. + p.

DRAGON JUSTICE OF THE PEACE (The Justice)

only £2.99 incl. p. + p.

DRAGON FOR A FREE CHALLENGE

Special Offer for Saturday 10th September

only £1.99 incl. p. + p.

(only valid at this Show)

R. & A. J. Preston

1000 HALL COURT, ST. BRIDGES, MAJOR AND ST. BRIDGES
Telephone 3000 0000

ADDRESS

VGA

THIS COUPON ALLOWS

30p off

a £1.99 game

OR

50p off

a higher price game

IF PRESENTED AT THE DRAGON SHOW
IN ROCHDALE
ON SATURDAY, 12TH SEPTEMBER
(only valid at this Show)

If you've got a technical question write to Brian Corder.
Please do not send a SAE as Brian cannot guarantee to
answer individual inquiries.

Dragon Answers

A sound start

I AM involved in a sports club and we want to use my Dragon on time teams etc. These written Basic programs to do this task would also link the computer to all the scores with time. Can you explain how these can send sound to the cassette port and how to connect a synthesiser to it. How will I be able to incorporate sounds to the TV and PA system?

Doc Lister
Reading

This output from the cassette port is suitable for directly driving a standard hi-fi or cassette system input. Looking at the Dragon's cassette socket and an oscilloscope the pins will show pins 1 to 5, you need to connect pin 2 to the input line of your amp and pin 3 to the ground line.

All sound produced by the SOUND and PLAY commands is automatically sent to this port so you need no mixers. It is possible to prioritise between the Tascam Portastudio Portastudio Port only, however, this cannot be done from Basic as the SOUND/PLAY commands always enable TV sound.

Flex or OS-9?

As a student presently studying a computer course, having owned a Dragon for four years and using it mainly for assembly language programming I find the machine as useful for my needs but I would like to obtain a 'commercial' operating system for it. The alternatives available to me are Flex or OS-9. Flex, although preferable, is the more expensive. I therefore would like to obtain OS-9 but this seems uncertain.

C.A. Goodwin
5 South Devon Rd
Plymouth
Bristol

THERE has been much heated debate over the years in Dragon User about the relative merits of OS-9 and Flex. Both are now available once again for the Dragon 64. Flex from Compuserve is, in fact, the cheaper of the two at £99.99 which includes an excellent editor/assembly package plus a disk version of Basic. OS-9 from R.C. Anderson also



comes with an excellent editor/ assembler and debugger (but no Basic) and is somewhat more expensive at £124.

It must be up to the individual to weigh up the relative aspects of these two very different operating systems, but Flex is cheaper and does have a larger software base.

Colour runs out

MY Dragon 3C loses its colour transmission after 5 to 10 minutes running time. It functions correctly in other respects, can you please advise me?

John Kemp, Leicester

Two most common causes of this problem is that, along with many other home computers, the Dragon tends to 'wander' slightly off channel during operation. Since TVs are more sensitive to this than other forms of the more modern ones will keep re-tuning themselves to the strongest signal.

Try to find that by using a second remote the colour then the channels are further Dragonised, especially if the modulator is at fault. This is relatively simple to explain yourself with a colouring line and a little skill or alternatively your local TV engineer will probably do it for you.

Cure for Anadex

I RECENTLY acquired an Anadex printer and I was wondering if it is possible to connect it to my Dragon 3C. It is a Model DP-8000 dot matrix. I have two connections at the rear of the printer and 3 sets of 6p switches inside. There are also four switches and a light on the front, another linked. I'd also have knowledge of whatever of printer.

Richard Ball
16 Sunnybank Rd
Hartpool, Grant

THE two sockets at the rear of the printer are for serial and parallel interfaces. The former socket is its pin female type is a standard Centronics interface as used on the Dragon. Any standard Dragon printer lead should work with this printer.

In the first bank of dip switches, 1-5 control the form length in increments of 0.5 inch; 6 controls regular/condensed font; 7 controls the print time-out function; 8 controls the selection of 'C' or 'I' in the character set.

In the second bank of dip switches, 1-2 control the skip over particular lengths measurements at 0.5 inch; 4 sets 4 or 8 lines per inch; 3 controls wrap-round/line advance and should be OFF for the Graphic 8 controls the rate linked setting

and should be ON for the Dragon.

The third bank of 3-dip switches is concerned with setting the baud rate of the serial port and should all be 'off' when using the parallel port.

Stood him up

I AM writing a basic program to run from disc. I want the program to display the date at which it was last run, but cannot see any way of adding the date (DATE) with the program as when I save it to disc and then reload, the variables appear to be cleared.

Paul Owen
Birmingham

ONLY solution to this would be to save the date in a disc file and reload it at the start of the program.

Another more interesting solution, to store the date permanently in your program which can then be overwritten before saving. Assuming the variable DT holds the date then the following code could be used:

```
100 DT="THIS PROGRAM HAS  
LAST RUN ON DRAGON"  
110 PRINT DT  
120 INPUT "ENTER TODAY'S  
DATE (DDMMYY)";DT  
130 DT=VALPDT()  
140 DT=PDAT(DT+1)  
150 DT=DT+1  
160 FOR I=1 TO 10:PRINT DT+1  
NEXT I  
170 SAVE "PROGRAM"  
175 REM NEXT DT PROGRAM...
```

This works by putting the new date directly into the DT address to be found by using the VALPDT function. When a string is added on a random basis in line 160, Basic does not waste time space as it had to do to find it to print directly in the program text, hence it is permanently changed and saved along with the rest of the program in line 170.



Super SAM

Matthew Lodge is here to tell you about a double-dealing chip

THE Synchronous Address Multiplexer chip number 6803 or 94LS030, is given its full title, has never been fully explained in one article. However, it is one of the most important chips in the Dragon, and unfortunately one often overlooked piece of technology. This technology can be accelerated by fast software practice, so beware, the virtuous speed-up pole WILL DAMN YOU! YOUR COMPUTER. The local Dragon know-all will tell you that it doesn't matter, and some software even uses it to achieve faster running speeds but IT WILL KILL YOUR DRAGON. The Dragon was not designed to run at the faster speed and it will cost you dearly in time and money if you use the poke (Compuserve supply the 6803/94LS03 for £23 a time).

Basically, SAM provides most of the address decoding for the 6803 chip or its very simple terms, what goes where. SAM starts at \$FF00 and finishes at \$FFFC. Some of the facilities are useful on their own, others are used in conjunction with the two PLA chips on the Dragon. First it explains how to access SAM.

Each register in SAM has two locations allocated to it. One when on-line and the odd location turns off the register, the even one turns it off. This dual location system also has the side effect that you can't tell how SAM is set up by peering to it, because all you get back is 0FE (\$FE).

For example, let's take the memory map toggle, locations \$FFDF and \$FFDE. The location \$FFDE is even (if you can't see why, convert it to decimal), \$FFDF is odd. When you type

POKE \$FFDF

you are setting the map type to one. The number you poke doesn't matter at all. When you type

POKE \$FFDE

you reset it to zero. But don't do it yet! You see, you did it! That's OK for all with a Dragon 32, but if you have a Dragon 64 then you've got a train. This is because on the Dragon 64, map type one is RAM mode, that's 64K of continuous RAM, and map type zero is the normal 32K mode. And because you were running Basic, all the interrupts were pointing to areas in the Dragon ROM. When you changed the map type, the ROM disappeared, and the interrupts had nowhere to go. Also, you were running Basic, which is a program just like any other, and the program itself also disappeared so wait! you have a crash. On the Dragon 32, map type one does not apply, so nothing happened. (unless you have a very rare and early Dragon 32 which has 64K of RAM fitted, but only 32K used).

This register is useful, as 64 owners can utilise it from assembly language programs. In order to prevent a crash, all you have to do is turn off the interrupts (ORCC

\$FE), then change modes. If you want to change back, all you have to do is change modes back, then re-enable them (ANDCC \$EAF). Thus, it is possible to copy 32K Basic and DOS into RAM where they can be backed, copied, altered (a program's dollars is limited as program time). This is the method that Harris's Basic 42 users employ in all of its goodies.

The 64 owners among you may now be wondering "but what happens when I type EXEC 49000 to go into 64K mode?". What happens is that a small machine-code program is copied into the cassette buffer, then switches in the second ROM (which appears at \$8000 to \$FFFF) and copies some code from it. This it switches in map one, which is 64K RAM mode. This still leaves the cassette buffer and all low SAM (the first 32K) intact. It stores this code in its free position (ie \$8000 from where I got it), then repeats the operation until all of the Basic is transferred, then it "loads" the new Basic (that is not a machine code load, but a term meaning supply the same as PLD).

Look at figure one. This provides a sketch of what each toggle means to SAM. I'll describe what each means, from \$FFC0 up.

\$FFC0 to \$FFC5 is the video graphics mode. This is specific conjunction with PLA, and I am not going to explain how to access all the modes, as it takes a long time. If you really want to know the contact Motorola for a SAM68407 application guide (free) which gives all the necessary diagrams, or get a copy of Roy Cooper's article on the Graphic, dated in November '88. Dragon user: Having said that, I'm going to try to explain it in simple terms from:

These locations control the VDD memory modes, or if you like, bytes per page.

MODE BINARY REAL TERMS

0 000 000

1 001 000

Program 1:

START	ORCC	\$FE	INTERRUPTS OFF
	LDX	\$8000	START OF BASIC
LOOP	STA	\$FFDE	32K MODE
	LD	,1	LOAD NEXT WORD FROM ROM
	STA	\$FFDF	64K MODE
	STD	,1+4	STORE IN SAME PLACE, BUT IN RAM
			END OF DOS?
	CHPX	\$FFFF	NOPE, SO DO NEXT WORD
	BLO	LOOP	INTERUPTS ON
	ANDCC	\$EAF	FINISH (NOTE: EXITS IN 64K RAM MODE)
	RTS		

3 BYTES SAM

\$200 0000
\$201 1000
4 100 0000
5 101 0000
6 100 0000
7 100 0000

7 100 0000 Watch-out Process!

Don't mix pages in this sense with pages in the Dragon Basic sense. A page here means the number of bytes used by the screen. A page in Dragon Basic means 16Kbytes, so the 64K bytes control for BASICs 3 and 4, using the binary columns from the table above (working from left to right or down in memory), and the SAM diagram, poke the odd location to set one bit, and poke the even location to clear a bit. So if you wanted to set 1024 bytes per page, you would:

POKE \$FFC0,POKE \$FFC3,0
POKE \$FFC1,0

— which are the actual registers, \$1,1 are the binary equivalents.

However, don't think that poking to \$FFC0, \$FFC3 and \$FFC1 will get you mode 4. You haven't set the PLA up, or the display offset. The latter just happens to be zero in memory.

The locations from \$FFC6 to \$FFC9 control the start of the VDD mode you are currently in, that is the address at the top left hand corner of the screen. Write in Basic it's 1024, but it can be anything from 0 to \$FFC0. You notice that there are only seven bits, so only addresses in multiples of \$200 (\$200 is zero). There post you had hardware scroll (a technique often used on the BBC, which uses the 6847's sister chip, the 68450). And before you rush off and write an animation program which flips through frames by moving the display offset, note that some Dragons have a hardware error, so that pages over \$4000 cannot be assigned. All you get if you try is lovely green/brown garbage. The machine and all the RAM is OK, it's just that your display is different. I've just killed two potentially good ideas stone dead, but moving the

display data have its own. If you have a machine code graphics program which needs all the space it can get, try locating the screen at \$B04, or even out at \$B12. If it doesn't use any ROM calls or only a few simple ones, then why not locate it at 0, if you're using text modes, you can fit quite a few separate text screens in \$B000 bytes.

Now we come to locations \$FF04 and \$FF05, which is labelled page on the diagram, but it doesn't do anything when in dets mode. I must confess I don't know anything of value about this location. All I has in the proto diagram is "MPU addresses from \$0000 to \$7FFF apply to page 0 if PF1 = 1".

Next we come to the legendary MPU rate. This is the thing all the bad software writers twiddle when they want to turbocharge their Dragons. The Dragon operates in slow mode normally, but three other modes exist. The first is just fast, the second is superfast and the third is turbocharged. You will notice that you lose the display if you try superfast or turbo. This is because the machine is running so quickly it doesn't give the poor VDG any processor time to do its work, so it is left bewildered, with the swirling green garbage. Under-

neath all that, the machine is still working although the keyboard won't be, as it was designed to work at a slower speed.

Next is \$FF04 to \$FF05 which is memory size. There are four possible values (2 bits), all of which are marked in the diagram. The Dragon uses the third mode all the time. If you try setting the computer it has 64K of static RAM, interesting things result, with a crash which kills the system.

One last comment, which is that all registers in SRAM are cleared when the machine is reset. That concludes SRAM, and I hope this has provided a helping hand to any programmer bothered by the SRAM chip. I should like to express my thanks to Compuserve, without whom, etc., etc.

Glossary

Interrupts - Every so often the CPU is interrupted. That is, it finishes executing the current instruction, then services a small machinecode routine. When it reaches the RTI instruction, it returns to whatever it was doing before. On the Dragon interrupts happen every 50th of a second, and are caused by the VDG chip which is signalling

that it is starting to scan the TV screen. The Basic ROM uses this to update the software clock or timer so it can keep track of real time (useful when PLANNING scores, to see when to stop).

MPU - Micro Processing Unit
SAM - Synchronous Address Multiplexer
Static RAM - Normal RAM (as used on the Dragon) needs what is called a refresh signal every few milliseconds. SAM provides this, and it serves to refresh the contents of the RAM so make it remember what it holds. Static RAM does not need a refresh signal. The price you pay is cost and size, as static RAM chips cost more and there is less memory for a chip (the silicon directly on the square of silicon takes up quite a bit of room).

VDG - Video Generator

6847 - VDG used in the Dragons. Designed to be used in conjunction with SAM, which explains why some circuitry it needs is on SAM.

6845 CRT controller (different to a VDG) used in the BBC micro and the BBC PC (Sport).

6883 - SAM's chip number

HELEND - SAM's more common chip number (one of a series of 74,xxx chips).

Figure 1 - SAM proto diagram

			C = CLEAR BIT B = SET BIT				All bits cleared when SAM reset				
FFC0	C	VB	VDG	1	0	1	0	1	0	1	0
FFC1	S										
FFC2	C	V1	MODE	1	1	0	0	1	1	0	0
FFC3	S										
FFC4	C	V2	(SAM)	1	1	1	1	0	0	0	0
FFC5	S										
FFC6	C	FB		0	0	0	0	0	0	0	0
FFC7	S										
FFC8	C	F1		0	0	0	0	0	0	0	0
FFC9	S										
FFCA	C	F2	DISPLAY	0	0	0	0	0	0	0	0
FFCB	S										
FFCC	C	F3	OFFSET								
FFCD	S										
FFCE	C	F4	(BINARY) Address of upper left-most display element								
FFCF	S										
FFD0	C	F5									
FFD1	S										
FFD2	C	F6									
FFD3	S										
FFD4	C	P1		MPU ADDRESSES FROM \$0000 TO \$7FFF APPLY TO PAGE #1							
FFD5	S										
FFD6	C	R0	MPU	1	0	1	0				
FFD7	S								0000	0000	0000
FFD8	C	R1	RATE	1	1	0	0				
FFD9	S										
FFDA	C	R2	MEMORY	-FAST	-FAST	-FAST	-SLOW	1	0	1	0
FFDB	S										
FFDC	C	M1	SIZE					1	1	0	0
FFDD	S										
FFDE	C	TY	MAP TYPE								
FFDF	S										

LITERAL PRINTING The problem with justified text is it does not allow you to line up items of text. The answer to this is to switch off justification for text where this is necessary; the manual does not make it clear you can do this. For example to produce a list of items as in Figure 4 use `CL` after the items, remembering you will have to take care to prevent word wrap yourself so do not exceed the line length, and add a `CB` at the end of each line. Remember also my tip for using `ENTER` to line things up. On completion use `CU` to return to justified mode. `CL` does not affect any of the parameters you have set, so margins, page-length etc. will remain correct.

BUGS Many of these are quite serious and not what one would expect from a program costing \$100.

REPLACE FILE For some reason the replace file does not leave a backup file; this is not a problem in itself but for some reason the replace file disappears completely. I say "completely" as it often appears correctly, but on many occasions it reports "FILE EXISTS" then gives you DOS ERROR 18 then quietly backs the text file screen, but while doing so kills the file on disc. There is no indication to the user that this has been done except a slight whir of the drive head. Anyone not paying 100% attention could easily be fooled into thinking the file had been written to disc. NOW the file, and there save a subsequent file to the disc, losing the new version and the old version, which could well represent many hours work.

WRITE PROTECT If you try to write your file to a protected disk, an error message appears if you have taken a hard file from a protected disk, the error trapping reports a problem but then says EA losing everything for you. There is in fact a way out of this (and other crashes or if you press **ESC**, **IOBC** **IOBC** menu program, then press **BP&K** and use the old command to remove your file, I would then suggest saving the file immediately and reloading. EA&K may well have corrupted. This may not work for all crashes and should not even be necessary — a little bit of more inspiration would have been to that.

FILENAME.C EA will allow you to save filenames beginning with C but then refuses to load them as it thinks you want to load a cassette file. The answer is to exit EA rename the file, a better answer would be for EA not to allow filenames beginning C.

PROGRAM DEFICIENCIES EA lacks in the following ways. It does not allow disc users to save files in cassette, users for it should be backup, and sending to other users with the tape version. Directory limits off the top of the screen without giving the user a chance to see the top files, and with no pause facility. Disc output is restricted to single drive use which is extremely limiting for users with a dual drive system. The manipulation can take a unnecessary amount of time in disc swapping while drive two remains idle. Lack of space cannot be used as an excuse as a separate disc control routine could easily be hooked in by the main program. No option for exiting to Basic is provided.

■ FIG. 4. TO PRODUCE A NEATLY ALIGNED LIST OF ITEMS
WE NEED TO OVERLAP SOME OF THE

- | | | |
|--------------------|----------------------|--------------|
| 1. JETSET MILLY | = PLATFORM, | FIVE DRAGONS |
| 2. SPEED RACER | = GRAND PRIX, | FIVE DRAGONS |
| 3. WORLD OF FLIGHT | = FLIGHT SIMULATION, | FOUR DRAGONS |

FIGURE 1. *Effect of temperature on the growth of *Hyphomycetes* in the presence of 100 ppm of copper.*

1. JETSET WILLY - PLATFORM, FIVE DRAGONS
2. SPEED RACER - GRAND PRIX, FIVE DRAGONS
3. WORLDS OF FLIGHT - FLIGHT SIMULATOR, FOUR DRAGONS

FIG. 4. Linear relation of the film used to produce monomers.

*****JANUARY 1980***** FOR 1. *****
***** GOOD TITLE *****
***** FIG 3.81 THIS TEXT IS IN NORMAL MODE AND
IS AT 12 CHARACTERS PER INCH WITH A MARGIN OF TEN
CHARACTERS LEFT AND WRITE=RECURSIVE WE ARE
IN COMPRESSED MODE WITH 41 CHARACTERS PER INCH BUT
AS YOU CAN SEE THE MARGINS REMAIN THE SAME AS IN
THE FORMAT CHANGES TO ACCOUNT FOR THE MORE
COMPACT PRINTING,RECURSIVE AND NOW WE RETURN
TO NORMAL MODE. IT IS VERY IMPORTANT TO DO THIS IF
YOU DO NOT WANT YOUR TEXT TO BE MISALIGNED WHEN USING
DIFFERENT FONTS,FIG 3.81 THE NEXT DEMONSTRATION
ON IS OF INHERENT TEXT, A GOOD EXAMPLE OF THIS WOULD
BE TO INCORPORATE A GATE IN YOUR BODY OF TEXT=RECURSIVE
GOOD WORD PROCESSORS SHOULD INCORPORATE
EASE OF USE WITH FLEXIBILITY AND FULL CONTROL OVER
ESSENTIAL TEXT LAYOUT ON PAPER,RECURSIVE NOW WE
RETURN TO THE MAIN BODY OF THE TEXT AGAIN, THIS CAN
BE VERY USEFUL FOR MAKING PARTS OF THE TEXT STAND
OUT, NOW FOR A TO PRODUCE A NEATLY ALIGNED LIST
OF ITEMS WE NEED TO ENTER LITERAL MODE 4=LITER 1, J
ETTER WILLY - PLATON, FIVE DARGONS
2. SPEED WAKER - GRAND PRIS, FIVE DARGONS
3. 4. MONIES OF FLIGHT - FLIGHT SIMULATOR, FOUR
DARGONS, NOW SEE WHAT HAPPENS IN JUSTIFY MODE
4=J1 1, BETTER WILLY - PLATON, FIVE DARGONS
5. 2. SPEED WAKER - GRAND PRIS, FIVE DARGONS
6. 3. MONIES OF FLIGHT - FLIGHT SIMULATOR, FOUR
DARGONS

This could be very useful for such things as renaming files or loading other utilities in conjunction with EA (eg address book) without having to switch off the Daemon.

Not really a fault but an idea for a future product is a spell-check routine to assist party EA, again on disc this could be a separate program which would be useful.

Mail file check the spelling and formatting codes, allowing it to be loaded back into Eudora for editing.

So how about it Mayra, why not cancel the bugs, and wake up on my suggestion for your next product?

Program: Electronic Author
Supplier: Quickbeam
Price: \$14.95 (Discounted trial no money back)

SPECIAL OFFER!

Exclusive to Dragon User readers

Dragon User and Abbott Computer Products have stocks of three well known Dragon books which are now no longer available from the original publishers. This means that we can offer them to readers at much reduced prices while stocks last.

FREE GAME
with each order



Load and Go with your Dragon by John Phillips and Trevor Toms. A popular starter's collection of programs with educational hints, a section on debugging, games, and a Whoopie. 128 pages. £1.95

Dragon Magic by John Sharp and David Bolton. A guide to programming the Dragon, undertaken through a collection of 30 entertaining programs from games to VAT calculations. 176 pages. £3.75



The first 400 orders will also receive a **FREE GAME** — Banks is a leaders-and-followers game for the Dragon 32. It's not one of the great DRAGON classics, it's not even new. But it's a great way to kill off those idle half hours when you feel like a break from reading or programming ... and it's free.

These books have been travelling around looking for Dragon owners for some time, so although they are in perfect nick inside, most of the covers have a tiny dent, or sticker mark, or other small token of life's experience. These, like the low prices, are all part of the natural beauty of cut-price books, and do not affect their performance.

Dragon Magic — Your first programming book by Richard Whisman. This is a programming primer, appeared as a Dragon teaching book by Dragon Club, ideal for youngsters (or not so young!) getting started out. With line and colour illustrations. 96 pages. £1.90



Please send me the following DRAGON BOOKS:

LOAD AND GO, £1.95 each.....

THE POWER OF THE DRAGON, £3.75 each.....

DRAGON MAGIC, £1.90 each.....

TOTAL.....

NAME..... ADDRESS.....

POSTCODE.....

Please make our cheques and postal orders to the order of Dragon Books

Bank account empty? Don't know where the money goes? Can't afford an accountant? Or even a lock-up wallet?

You should have DRAGON HOME ACCOUNTS for the Dragon 32.

- A 4pp booklet describes the options for storing your monthly household transactions on tape, making forward estimates of expenditure and reviewing your budget to date or month by month.
- The accompanying program was originally marketed by Gemini Marketing at £16.95. DRAGON USER's special offer price is £8.50, INCLUSIVE of postage and packing. The program comes on cassette only — we regret there is no disc version available.
- DRAGON HOME ACCOUNTS can handle up to 192 transactions a month of up to £10,000 each. The program can display a bar graph plot of monthly expenditure on any particular item, and store fixed values for regular monthly outgoings. Information from earlier months stored on separate tapes can be loaded into the program.

Please send me copies of Dragon HOME ACCOUNTS at £8.50 each, RPIAL.....

NAME.....

ADDRESS.....

Please make our cheques or postal orders to the order of Dragon Books

IMPORTANT: When ordering special offer items, you can send your orders for BOOKS and/or HOME ACCOUNTS to us in the same envelope if you wish, but please make our payments for BOOKS and for HOME ACCOUNTS on separate cheques, and send them with the separate coupons provided (photocopy coupons are acceptable). This helps us to keep the flow of orders separate, and send the goods out more quickly.

Send your orders to Dragon User at 12/13 Little Newport St., London WC2H 7FF, marking the envelope DRAGON SPECIAL OFFERS.

Write: ADVENTURE

Peter Gernard talks about words and anniversaries

ADventure last month, we'll be spending some time examining a more detailed version of our parser. We might not have a parser that compares with Magister Dornik's, but it *can* produce any kind and class of letter that we can find. It will be able to type them all in (I guess bug in at least one computer version of *The Pawn*).

By way of a slight diversion, after all the programming is finished for the month, someone recently was asking how to go about getting ideas for an adventure so that you can have a game ready when a famous anniversary comes around. Here is, in other words, how the right adventure ready for the right time and thus hopefully increase your sales.



But first, the programming, and as usual the listing will tell you what a lot it is to do.

Some things, you'll be pleased to note, remain pretty much the same as before. In particular, the routine to accept an input from the player hasn't changed by very much. Lines 6000 to 6003 take care of all this, and I've managed to understand the explanation in the June issue that you'll have no trouble in working your way through this lot. The only real difference is (a) we keep track of the previous entry from the player in case they want to repeat a command and (b) we allow for keys like quotation marks for talking to other characters and commas, which are used for separating commands.

This means that we can happily accept inputs of the form:

**SAY TO GEORGE "WATCH THE TROLL"
SMASH THE DOOR AGAIN WALK
THROUGH ISO SOUTH EXAMINE
THE GROUND**

or whatever turns you on, I suppose.

Now all we have to worry about is the code that makes sense of all this lot. As with accepting the input, some of it will be quite familiar, and it all really works on the same principle as the first **VERB**, **NOUN** parser covered in the June issue. If you read the July one you should have got most of your vocabulary sorted out by now, or at least all the essentials parts of it, but the program includes just a few words to enable us to test you've entered this month's section of program correctly.

We'll take each new addition in turn, and conclude at first of all on the section that looks for not only verbs and nouns, but also link words, action words, and additional nouns (if you'd typed in something like **PLACE KEY UNDER DUSTY BOOK**).

Lines 4010 to 4042 split the sentence up as appropriate, taking out spaces and references to the words 'that' and 'is' in lines 4020 to 4025. Note that the presence of a verb is checked in lines 4016 to 4018 before we start removing parts of sentences. It's so easy checking out all the additional words (I've haven't even managed to find a single space in the player's input. A space, after all, is what separates the verb from the other words).

Lines 4044 then translates everything into a list letter word, so that if a player had entered as his verb **EXAMINE** then **V55** would just contain the letters **EXAM** after passing (or is that parsing) through this list. If you want to look for more or less letters than just after this line accordingly.

Lines 4046 to 4075 then do all the checking to know verbs, link words, nouns and so on. Variables like **V5**, **N5** and so on are returned as being the verb number, the noun number (L) being the link word number and **AG** the action word number. Nouns are split into ordinary nouns and additional nouns, and have the variables **N5** and **N4** attached to them.)

Assuming everything is found to be a word that the program has been taught to recognize, the program translates through lines 4081 and returns from this section of the game. Now, having got numbers to represent verbs, nouns, and so on, you can easily include code of the sort:

ON V5 GO TO 1080,1052,1050

and so on, so that everyone would take us to line 1050. Here, if you use **GP** to represent the player's current position in the game, we could have something like:

**1050 IF GP=1 AND L1=13 AND NQ=21
AND ORP(L1) THEN PRINT "You
throw the book at the king in disgust
and stare out." GP=L1 GP=22=22.
RETURN**

Here we have assumed that the **throw** verb in word is a book, the twenty first noun is the king, the first verb is the word 'throw' and **ORP** contains the position of all the objects (this is one indicating that it's been carried) while **22** is the total number of objects that the player is carrying at the time. The **throw** book is thrown, it is placed in the room (by the statement **ORP(L1)=GP**) and the number of objects being carried is decreased by one. All this in response to the single sentence from the player:

THROW THE BOOK AT THE KING

What if the player wasn't carrying the book? Well, line 1050 could read:

**1050 IF GP=1 AND L1=13 AND NQ=21
AND ORP(L1) THEN PRINT "You
aren't carrying the book." RETURN**

Perhaps you'd like to expand on the response and have the king get about beating up the player for his impudence in throwing something at a mighty ruler. Or even attempting to throw something, as is the case with line 1052.

Unknown words

All very well if the program understands every word in every known language, but also there will come a time when the program comes across something that it doesn't understand or which, regrettably, it recognizes as a bad word. So, we have some code to take care of that.

Lines 4076 and, more importantly, 4082 to 4114, look after the latter of these options, and ensure that should wholesome people as readers of *Dragon* User cannot begin to imagine what rude words are being replaced by asterisks here. This is a fairly tame game, come on, we don't want to get into trouble. Fiddle Star as my ever-faithful.

Words that aren't recognized are covered in lines 4086 to 4095, where **UNQ** contains the word that the program doesn't know. Okay, so it should be **UNQ**, we can't all be perfect. **UNQ** itself can come from a variety of different places, including lines 4046, 4054, and so on. Anyway, in short, when an unknown word can be found:

The little routine in lines 4086 to 4090 comes into play if the player is attempting to use a verb as a noun, or vice versa perhaps, and tells the player that the word is a known one but isn't recognized in the context that he is trying to use it in. Otherwise, he just gets told that the word is an unknown one.

Repeated instructions

If you use the word **AGAIN**, or more simply the abbreviation **A**, a player can get the computer to repeat the last instruction. Unless, of course, the last instruction contained an unknown word, in which case line 4086 might come into play. Assuming that all has been understood (including something like **OPEN DOOR CLIMB IT**, where the 'it' is translated back to be a 'door' again by line 4084) then lines 4095 and 4096 look first of all to see if the first thing encountered in a sentence is an **AGAIN** or **AKA**, and if so, then set out the input strings accordingly. Now you know why the input routine was modified slightly so as to keep track of the previous entry by the player.

Lines 4097 to 4180 are also used by the **AGAIN** routine, but only if something is found that can be **AGAINed**, if you'll excuse the phrase. Here we are also look-



WHAT is quite unassuming for you Dragon adventure fanatics are. Apart from British Telecom engineers with Dragons who use British Telecom stamped envelopes to get in touch, and people who write on pink paper or yellow paper or (ugh!) green paper, something like ninety per cent of the letters I get are growling pleas for help with various adventures. Of the rest about half are offering solutions to vast numbers of different games, and the other half are asking on about Jet Set Willy and other things only obliquely related to the business of solving adventures. Something strikes me as odd here, somehow. Does it mean that nine out of ten Dragon owners, who expressed a preference, are incapable of solving adventures? I shouldn't think so. Does it mean that nine out of ten Dragon owners are like the writer of this column and are a wee bit on the lazy side? Ahah, much more like it, so let's have some more solutions please, chaps.

Solving everything

Once someone who has probably solved more Dragon Adventure games than have been written, and is now going about writing this column just to redress the balance. Young Hargrave, for it is he: a letter from him is like a stroll through a post office. Wads of paper, jiffy bags, stamps, more envelopes, tapes, all sorts of interesting things. Plus, and this is what they want, solutions. For 25 pence, to cover printing costs (it says here) he can offer solutions to *Engelwood*, *Witch Factor*, *Colossal Cave*, *Pelican*, *Aqueduct* 471, 471 *Chadwin*, *Black Sanctuary*, *Madness* and the *Minotaur*, *Adventure*, *Don't Panic*, *Rat* (part one), *Geogic*, *Starman*, and something called *Others*. By which of course he means every other adventure in the known universe, I have no doubt. His address is Ormsley Hill Farm, Uxley, Dunley, Gloucester GL11 5BN, although that could be 5BN I have to admit. Simon Hargrave, may your postman curse this column.

The game he sent me, *Starman*, was mentioned by big brother in the January column, so even if you didn't get told about it here, except to say that I had been updated somewhat since then, it includes a very useful verb, *WALK*, which prints out a list of all known verbs. Very handy when you're convinced that the program writer has been watching too many episodes of

Call My Bluff. While trundling about an unknown planet awash with alien lifeforms and acid... which is a bit like going into Wetney's pub in Birmingham, I was reading Simon's letter accompanying letter or script for the text six months of *Starman*, whichever is the shorter. Here trying to make this game, but says the other software houses will not themselves take on the responsibilities of marketing a Basic (with machine code additions of course) adventure.

Now the software houses can naturally enough please themselves about what they decide to take on or ignore, but Simon's point is worth pursuing. Most of the retailers will not stock any adventure that doesn't contain graphics, unless it's by Infocom. Which is not, by the way, a complaint about Infocom. Good luck to them. The worst of most of these retailers is exactly the same one as Simon gets when trying to make his Basic adventure, namely that there's no demand for them. Of course there's no demand for them if nobody stores the blessed thing! A butcher who didn't stock meat would probably come to the conclusion that there was no demand for it because no-one came into his shop. How do they know? Market research on people who came into the shop? If you know you're only going to be able to buy overpriced and over-typed (generally) graphic adventures and you don't like graphic adventures then you won't go into the shop, and so the results of any market research done in that shop will be totally incorrect.

What can be done about this sort of situation? After all, if W.H. Smith won't stock a game you can hardly persuade them to do so. If Crichton Software won't sell your game you can't tell them that they must open up to dealers the Spanishquisition. After all, no-one expects etc. etc. Personally I liked Simon's *Starman*, but haven't got the clout to market it. I can't rush around Wigan (of which more later) throwing copies into every shop I come across. Perhaps you'd like to tell me your views on the subject. Have you tried marketing a game of your own? What sort of problems have you had, and more importantly, did you find any solutions to them? Simon Hargrave, and I await your letters.

Back to this letter of the young boy

Hargrave again. He offers us some *POKE* for the *Days of the Wizards*, if you're having problems with this particular task. Three little goodies in all, and they are:

POKE 4=DOOF AND
POKE 4=DOOAH

This disables death, which would be most convenient for a lot of us I imagine. According to Simon this takes care of most death situations, and all you have to do is press 'W' to re-incarnate yourself. I shall try and remember that next time I visit the Seven Stars.

POKE 5=HIELD-ON

This governs the strength of all the creatures in the game. However, set it too low and you'll manage to stop the Unicorn helping you, which is definitely not what they want.

POKE 6=HIDELAND

This allows you to rest wherever and whenever you want, so that you can get your strength back whenever it's needed.

Unfortunately even with the help of this bit Simon hasn't managed to finish the game, and seems to be convinced that it's impossible. As it

One more mention for him, and then we'll ignore him for the rest of this month, and possibly next month as well. Not that I think he'll go away.

Tips on *Engelwood*, and no backwards writing because all the blood needs to my head. To reach the fireflies you must go north-east, west, north-west, to go into the southern entrance to the marsh at night.

To get *Beasts* on the island you must drink in the Western Well and then jump on the springboard.

To get *Beast* into *Scars Castle* you'll have to wait until the end of the game. You can enter by using the fire drill, but only if you can find "the silly thing" as the Hargrave pen has it. He also doesn't mention where it is, guessing about people who can't find it, talking about certain animals who like to bury their heads in the sand... yes... yes... yes!

Northwards

Enough of him, let's leap about the coast-trip and head further north even than Wigan. However, John Goagins, up in Aberdeen, tells us that he had managed to sort out the *St-Ants*, who seem to have appeared in almost as many games as the legendary Goagins, and who pick up as

many mentions as the renowned dead Dini! Gding. Anyway, to get past these whoever had the archangel should simply LUGAAHCPA, DOW, which seems a bit drastic but works. Beyond that our foot-fair-haired runs into problems, although he does tell us the *Advent* of the citizen's advice bureau (an ordinary tale of everyday adversity) *advent* (yourself) and a magic carpet, and that working on the waiting wall will transport you to the other waiting wall, nearly always the citizens who dispense advice.

By the way, Mr. Resque, the verb *ARE* as contained in the Marchpous refers to *ARE* but not *ARE* the, and hence the response about violence, please. We all make mistakes. Finally, thanks for his sheets on a number of adventures, but as I only buy sheets of paper in units of 500 at a time you'll have to wait a wee while.

Text compressor

Continuing our countrywide forays and currently heading past Mowat's Magazine past to Sheffield, wherein lives John Foster, at 64 The Oval (personal contact given), must be well off. With that, Sheffield 55 BSR being an about larger friends of yours does not guarantee a mention, John, especially when they're called Arnold and Sylvester. The latter, as we all know, is about as tall as the aforementioned (Dini! Gding, but I dare say it Arnold (if I can't spell his surname) split my chin (I probably hit him another one. John has written a *Basic* text compressor for Dragon adventure writers, which costs £2.00 plus 50 pence post and packaging, and this magazine (as he refers to as *System One*).

The instructions for use are about as long as a good issue of *Dragon* (and it isn't that difficult to use. We at *Chai* General managed quite successfully, even under the strain of trying to watch the Botham smash several cricket balls out of the ground. To work at all you'll need to have at least 2K ahead in your game, which should be a very great strain for most of us, and although it does take quite a while to hunt through everything and compress it down in size, if you haven't got access to a routine like this then it's worth the wait. Not the greatest saving of memory that I've ever seen, although any saving at all is not

to be sneered at, and it will reduce 18K of text, the maximum allowed, down to about 14K, in other words, about a 22% saving.

Despite my comments the instructions are actually very good and very clear, and give enough details and examples for anyone to be able to make use of the program. Saved £2.50 to John and the program, which works and does the job it is intended to do, will be yours. And the program can have higher prices than that, really, 22% compression seems like average for whatever amount of text you use, between the limits of 2K (under savings) the less text you have, I think, and 18K. So send Arnold of the unspelt surname off to John and it reduces him by 22% I'll have my chin back, thank you very much.

Whoever started off this desktop publishing task has also to answer for whoever coined the phrase has even more to answer for, since the idea of reading the tops of desks is not one that has ever appealed to me personally. Still, it has prompted a lot of people to start producing magazines of varying quality, and too many for all you adventure heads to start sending off for (most produced in pairs) for letters, lists of trumpets, Lancelot Hope and Glory etc.) to Wigan.

One of the advantages of living in Wigan, you, there is more than one that over can try to meet the producers of these magazines, and despite the fact that one of them beat me at *pool* (I was losing under

the auspices of *pool*, however) I shall still mention them. First of all we have *Adventure Probe*, which is produced by Sandra Sharkey from 79 Myron Road, Wigan Lancashire WMB 6AT, closely followed by *Adventure Contact*, and this one is produced by the Wizard of the pool (as Pat Winklesley, the address to write to this time being 13 Hollington Way, Wigan Lancashire WMB 6LS. Both magazines are over a year old, which says something in itself, and both are full of useful adventure hints and tips. Not exclusively *Dragon* of course, but you should find something to interest you in them somewhere. Sample issues are a quid each, by the way, for which you'll get about 40 or so pages of adventure-mania. *Adventure Probe* seems to cater more for the adventure player, while *Adventure Contact* tends to look after the adventure writers amongst us, so take your pick. These ideas are not, however, mutually exclusive. If I ever run a game of *pool* against one or the other either of you let me know.

Oh well, out of space again, so I'll leave you with something to ponder on until next time. We're all used to collective terms for things, like a flock of sheep, a herd of cows, a crowd of people, or silver ones like a column of accountants, an abbreviation of estate agents and a box of matches. But what would you call a collection of adventure enthusiasts? A text of adventures, a problem of adventures, a school of adventures? What do you think?

Adventure Contact

To help puzzled adventures further, we are realising an Adventure Hotline — simply fill in the coupon below, stating the name of the adventure, your problem and your name and address, and send it to Dragon User Adventure Helpline.

1010 Little Newport Street, London EC2H 7PP. As soon as enough entries have arrived, we will start printing them in the magazine.

Don't worry — you'll still have Adventure Mail to write to us with!

Adventure
 Problem

 Name
 Address

Adventure Contact

Adventure: *Position*
Problem: How to get across the Pool of Fire on the 3rd level.
Name: D. Beckwith
Address: 30, Ratchall Road, Leyton, London E7 5JF.

Adventure: *The Room*
Problem: How do I get light, so that I can see something inside the tree?
Name: Hanning Johnson
Address: Gammel Sanddalvej 20, 4845 Holstenborg, Norway.

Adventure: 1) *Master Foster 2)* *Justaposition*

Problem: 1) I've done everything but haven't finished 2) Where's the oil for the wily dwarf?

Name: James Wood
Address: 48 Pinewood Crescent, Woodhead, Pudsey, W. Yorks LS28 5AD.

Adventure: *Templewood*
Justaposition
Problem: How do you get the fishing rod? How do you get the false tube in the hydrophonic dome?

Name: Neil Davies
Address: 4, Grinstead Drive,

Crescent, Sharnbury, Shropshire SY2 5JE.

Adventure: *Sharnbury*
Problem: How can I get from the igarthen? What does the location want? Where is the? Help!
Name: Dario Palmieri
Address: 110 Molino 181 Pinerolo 12, 40031 Bologna, Italy.

Adventure: *The Labyrinth*
Story
Problem: How do I open the cabinet? What do I do at the

10th Shrine to get the sword in the statue's hand? How do I get the puzzle from the humming generator without being electrocuted? Help!
Name: Mr. S. Alport
Address: 22, Little down Orchard, Newton Poplington, Wiltshire, Devon.

Adventure: *Arbit*
Problem: Can I pick up more than one crate at a time, if so, how?
Name: Nicholas Curtis
Address: 60 Hale Lane, Watling, London NW7 5PZ.

Tomorrow the world!

Gordon Lee narrowly misses filling the whole earth with numbers

PIERRE de Fermat (1601-1665) was one of the two leading mathematicians of the first half of the 17th century (Descartes was the other). His wide-ranging discoveries in various branches of mathematics — geometry, number theory, differential calculus, and probability — are all the more remarkable, since it is realised that he was an amateur mathematician, being a lawyer by profession, serving as a councillor in the local parliament at Toulouse. He also studied the sciences of optics, especially in relation to the laws of refraction, and his theories are the speed of light, though coinciding with the views held at that time, was later shown to be incorrect — but not until after his passing of two centuries!

Many of his mathematical theorems were also not proven until long after Fermat's death, one of these, known as Fermat's last theorem, is still unproven to this day. This theorem states that the expression:

$$x^n + y^n = z^n$$

has no whole number solution when n is greater than 2. In other words there are, for example, no two cubes which sum to form a cube, no two 4th powers which sum to another 4th power, and so on. Remember that each of the values must be a whole

number otherwise there are, of course, an infinity of solutions. When n is the power of 2 we have the Pythagorean relationship, with an infinite number of integral solutions:

$$2^2 + 4^2 = 2^2, 2^2 + 10^2 = 12^2, \text{ etc.}$$

Although this apparently simple theorem has been around for over three hundred years, no-one has yet been able to demonstrate a proof, neither has it been shown to be wrong — for example, by the discovery of a single instance of a solution containing a higher power. As with other of Fermat's conjectures, he evidently felt intuitively that they were correct. Hundreds of examples to support this theorem, and was unable to find any examples which disproved them.

A number of his other conjectures were in relation to the interesting class of numbers known as primes. One of these theorems states that every prime number of the form $4n + 1$ (eg 5, 13, 17, 29, 37, 41, etc.) is expressible as the sum of two squares in one way, and one way only.

Another states that p is a prime number and a binary positive integer, then $p - 1$ is exactly divisible by p , if both of these theorems are correct, then they could be useful tools in the search for the higher primes. In fact the proofs for both of these

theorems have been established, but again, not until after Fermat's death. Two of the mathematicians who worked on them were Gotfried Leibniz and Leonhard Euler.

However, there is one instance in which Fermat has been shown to have been mistaken. Since the impact of prime numbers was recognised, mathematicians have sought (in vain) for a formula which will generate exclusively prime numbers. One such, is the well known $x^2 + x + 41$, which works obligingly for values of x from zero to 39, but thereafter produces composites as well. Fermat, while studying the possibility of such formulas, came up with one of his own which, he maintained, produced only prime numbers. This formula was:

$$2^{2n} + 1$$

That is, 2 raised to the power of $2n$, plus 1, where n is any positive whole number (including zero).

The following table gives the first nine values:

$n =$			
0	=	$2^1 + 1$	= 3 (prime)
1	=	$2^2 + 1$	= 5 (prime)
2	=	$2^4 + 1$	= 17 (prime)
3	=	$2^8 + 1$	= 257 (prime)
4	=	$2^{16} + 1$	= 65537 (prime)
5	=	$2^{32} + 1$	= 4294967297
6	=	$2^{64} + 1$	= (a 20-digit number)
7	=	$2^{128} + 1$	= (a 39-digit number)
8	=	$2^{256} + 1$	= (a 78-digit number)

Now here Fermat, heavily misled a belief in his formula since the calculations needed to establish the larger numbers in the series were not possible at that time. The numbers he used were a mixture of 3, 1, 2, 3, and 4 were easily shown to be prime, and so Fermat stated that it was probable that the rest of the series were also, including the number 4294967297, found when $n=5$. In fact, this number is composite and has a factor of 641. It is surprising that he was unable to determine this factor, especially as there is record of him factoring much more difficult numbers such as 100893208169 in its two prime factors 686423 and 145593.

It has now been proved that all values of n that are greater than 4, produce numbers that must be composite — almost the exact opposite of Fermat's original theorem! The formula still presents a challenge to some mathematicians to discover factors of some of the larger numbers in the series, and one of the largest numbers for which a factor has been found is when $n=33$. The number itself has over 3,325 digits (that is 3 followed by 21 zeros), but one of its factors has been shown to be:

$$2^{11} \times 5 \times 1$$

In everyday terms, this factor is equal to 2 raised to the 76th power! This number is then multiplied by 5, and 1 is added to the result.

The competition this month is to devise a program which computes exactly the value of this factor. NB Don't try to compute the actual number that this is the factor of! The whole earth is not large enough to contain it!

Prize

NOW that summer is with us (and it isn't with us by the time you read this, I shall probably have emigrated to Australia, what would be a more ideal place than an elegant, sunny island, proceeding to all and sundry that one is a Dragon? Well, we are giving us twenty of our Dragon Systems as prizes for the August competition, so this will be the first Dragon Over competition ever to ask you to state your size as well as your solution. Small, medium, large or extra large. You have your already? Come over now for the postcard confirmation. Let them see where you stand.

Rules

Firstly, do as we suggest. Stick to calculating the factor and not the actual number. While Dragon Over makes every effort to ensure that the contents of its articles are accurate, it cannot be held responsible for the rest of the world as we know it.

Secondly, send us your solution, any value you want to include, a printout of your program (on paper, please), and your name and address in an envelope marked AUGUST COMPETITION.

This month, not so much a tie breaker as the tie breaker. Tell us what you will be doing with the last five bits, it doesn't have to be true, only imaginative. And printable.

May winners

The number of three-letter word sequences discovered by entrants varied hugely.

from a couple to nine pages of words, and I haven't got time to go through all of them, but some competitors do devise a program, not to read a dictionary, the actual number of sequences wasn't the crucial factor. However, thank for all the words we've never heard of...

Copies of Pete Gerrard's *Following Adventures on the Dragon*, published by Duckworth, go to Richard Lang of Cambridge, C. Hickman of Middlesbrough, T. Paswell of Hendon, Terry Peter of Chislehurst, S.A. Seddon of Greenwich, Terry Smith of Harlow, S.A. Newman of Addlestone, Phil Sayers of Liverpool, P. Morgan of Bristol, Paul Mason of Wotton-Under-Edge, Jake Anderson of Solihull, D. Dennis of Twickenham, R. Banks of Southampton, Keith David of Crawley, P.G. Maddocks of Maidenhead, A.J. Westwood of Northampton. Since Hargrave & Gurney John Orpington at Harpenden, Pete Hoadley of Warrington and G.R. Barber of Sutton Colfield.

The tie-breakers had a heavy printing bill, but we liked the chap who had actually done some research. 'Communication with the Internet Data Manufacturers Association and the BSI's interperson-D. Geoff Eas (produced a different version). However, the oldest local solution, recalled a clock work life itself (Gerrard, 1993).

Solution

This month's solution appears on pages 6 and 7. So does last month's.

100