

DRAGON USER



May 1987

The independent Dragon magazine

Contents

Letters

DS-B news, disabling Scientist, DS-B Blues update, yet more Fire Force

News Desk

R & AJ Preston now selling Blazy games; another Analogue update; what became of our Wizard's Guests?; 6800 Show

Simon

Isle Agi-Gayen brings for a simpler name and comes up with Simon — a version of the electronic game which makes use of PCOPY for speed

Dragon Soft

Dislike from H.C. Andersen, an update to P.O. Scott's DOSPlus, and Microdesk's new one, Stone Reader II

Searching techniques

Dene Beddington explores a jungle of searching routines, and provides some answers to the problems of unsorted lists

Introduction to wordprocessing

In the first of two articles, Roger Marais starts from first principles to examine the needs of word processing

Communication

Readers ask, readers answer

Into the directory

Paul Daggleth takes a long rope and lowers himself into the DragonDOS directory, with directory access programming in mind

Soft extra

Hi-Text from Microvision

Write: ADVENTURE

Peter Gerard begins a column on writing your own adventures from scratch

NDUG

Introducing the National Dragon User Group, by member Philip Reid

Winners and Losers

Gordon Lee looks at his own and other solutions to DU's December competition, with programming tips

Dragon Answers

Brian Cudge answers your questions

Adventure trail

Peter Gerard takes up his leatherhock and lantern and sets off down the trail — what awaits him?

Competition

Gordon Lee shuffles cards to see what they spell, and Duckdoffler copies of Exploring Adventure on the Dragon for the winners

No Expert this month

Did someone drive a stake through his heart? No such luck — he just got lost in the pen! He'll be back

Editorial

THE 6800 Show was small but busy, and I had the opportunity to match up quite a few names and faces. I left late and still didn't get to talk to all the people I wanted to talk to, concentrating as I was on the ones who are hardest to get hold of, and could quite happily have kept myself occupied till wrapping up time if circumstances had permitted.

Cheers again to everyone who came along, and special thanks to Jenny Pope who organised the show. Jason for answering tricky questions and scavenging everything he could lay his hands on, Helen Perry and David Owen for fetching and carrying and personally penning the stall in the afternoon, Tim Lomas of NDUG for offering me a pint which I didn't have time to drink, Alan Preston of R & AJ Preston for contributing some Boulder Crashes above the call of duty to help us out of an embarrassing situation, and above all to EVERYONE WHO RE-SUBSCRIBED TO DRAGON USER. In the heat of enthusiasm or the depths of despair it is easy to forget that this year you've got to remember your sub, or we won't know you're there, and we won't be. Don't delay.

Telephone number
(All departments)
020-10403

Editor
HELEN ARMSTRONG

Production Editor
BARBARA HULKIC

Associate Editor
JOHN COOK

Editorial Secretary
ANNE MARIE O'BRYEN

Advertisement Manager
DAVID COEN

Administration
GABRIELLA FISCHIA

Marketing Manager
HELEN MORRIS

Managing Editor
PETER WOODCOCK
Publishing Director
JENNY IRELAND

Subscriptions
UK: £14 for 12 issues
Overseas (surface): £20 for 12 issues
DASH CODE 01. Tel: 02020
Dragon User, 15/16 Little Newport Street,
London WC2H 7NP
US address: Cox Business Press International,
305 East 42nd St, New York, NY 10017
Published by Scott Press Ltd.
© Scott Press 1987
Typesetting and Production by Aries Limited,
London W9W
Printed by Headley Brothers Ltd, Juddville, Kent
Registered with the Post Office as a newspaper.
Dragon and its logo are trademarks of Eurocom Ltd.

How to submit articles

The quality of the material we can publish in Dragon User each month will, of course, be entirely dependent on the quality of the contributions that you can make with your Dragon. The Dragon computer was launched on to 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 2000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, wherever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return any submitted article or program, so please keep a copy. If you want your own program returned please include a stamped addressed envelope.

Letters

This is the chance to air your views — send your tips, compliments and complaints to Letters.
Page, Dragon User, 12-13 Little Newport Street, London WC2N 7PP.

Just a thought

THE shiny black, new state of the art 0208 850286 Cambridge Computer manufactured by Clive Sinclair is advertised as being able to run its ROM word processor with RAM of 'around 15K available'.

My levels, levels, out-dated, bottom-of-the-range Dragon 32 runs the Cambridge Super Writer word processor with 12876 bytes (12,308) of available RAM — or 28608 bytes (28,224) if the graphics capabilities DUMPed.

Mike Hosker

Hazebrook

Portsmouth

Hampshire

Bury St Edmunds

Suffolk IP20 9AU

And it has a decent screen size, and you can't fit it in a briefcase, which means that nobody can get on all you to work on the train.

More OS-9 news

In answer to recent letters, I can report that it is possible to obtain the original Dragon OS-9 plus the manual from: Kanten & Ludwig Data Consulting, Deichenschwanz 27, D-68521 Dietzenbach, Germany.

Alternatively it is possible to order the Tandy Color Computer OS-9 which includes the assembler, editor and other pile of manuals. CoCo OS-9 does not look straight on the Dragon, you have to reformat the disk files, disk, printer and keyboard to the Macintosh with the versions for the Dragon OS-9. The address of Tandy is: Radio Shack, 305 One Tandy Center, Port Worth, Texas 76182, USA.

OS-9 users in the UK should contact the HOUG's OS-9 guru, Jason Shouler, at 71 Victoria Road, Peckham, Poole, Dorset BH12 5AJ.

We have in Finland have Dragon OS-9 patches which allow a terminal to be used through the serial port at baud 9600, 7 + 8 bits, no parity, and actually with the fastest typing

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 finds and your opinions, send us your tips and suggestions. Send us your best Dragon stories. What if you think we are, mind readers?

EXTRA
PLUFF

What a pain!

DO I detect some longstanding animosity between Messrs. Hitchman and Orban? It's necessary to submit Dragon User readers to this kind of petty carrying? Surely the job of a games reviewer is to review games and not to try to become kind of literary superset? I have played *Fire Force* and find it very interesting. As a former programmer myself I can imagine I began this review will have inflicted on the poor author who must have spent many hours writing the program.

I was under the impression that DU was designed to promote the Dragon, but this type of journalism is contrary to this ideal. If you discourage the author and Dave Hitchman's world (and there are frighteningly few) you destroy the Dragon. Long live DU, long live DU and return JCRC full time accuracy. If you need a replacement, I'm available!

Ralph Fowler, 1 Victoria Road, Inter, Manchester

MR. Fowler, your letter touches on a number of important points. The irony of your review is that it has been shown to have been the instigator of a long term feud in order to find himself participating in one. I am not going into a list of name checks, but Jason Orban was not at all the only person who was unhappy about *Fire Force*. Now, we haven't printed letters from everyone who has written to us, because, as you so rightly pointed out, it would not be DU's policy to pick on suppliers for individual mistakes or misinterpreted findings. But it is our job to give a voice to readers' negative views as well as positive, whenever these views are fairly supported. If a number of readers have bought a game which they feel is poor value, then their views must be put forward. It is a political local explosion 'carrying' because it is negative.

For instance, one frequent complaint about *Fire Force* is that the collision detection is inconsistent. Some people are very irritated by this feature in a game, while others ignore it. If a programmer does not take this spread of opinions which is used as computer gaming sufficiently into account then he is going to have a number of seriously disappointed customers, no matter how perfect his programming is, or how often it tanks. This is a fact. Games don't come free, people are paying with their hard earned money.

The controversy about *Fire Force* is about to continue. Jason believes that facets of the game which make it difficult to play are 'bugs'. Dave Hitchman stresses that they are part of the game's design. A bug is programming or a design fault in a program, and there comes a point where it becomes a matter of opinion whether you are dealing with a mistake or a challenge. Jason Orban looks at games loosely, and thereby makes unsupported statements, however he chooses them up.

Dragon User would very quickly lose credibility if it only talked about the good news. We reject Jason's point of view, after all, we are not about to get angry to us before publishing, and run other reviews putting pro and con points.

Jason's review has not frightened Dragon users away from the game in drives — Quickbeam report good sales since the review appeared. Which demonstrates what we already know, that readers are prepared to use their own skill and judgement. On that subject, let me say again that anyone who wants to send a review in is welcome to do so. We are only print a selection, but if you want to show DU your stuff, do what writers do — write something!

note on the upper drive, and a 'European' that shows ABCs 91-93 and 193-199 as in parentheses.

Another useful address is the US OS-9 user Group, Dale Pickett, P.O. Box 7588, Des Moines, Iowa 50322, USA. Membership was \$25 a year at the last count.

Commercial OS-9 software is available from Compimage, 27 Colborne Road, Southminster, Essex CM9 7AH, and in the USA through Tandy or from Frank Hogg Lab, Inc., The Regency Tower, Suite 915, 770 James St., Syracuse, NY 13203, USA.

Marin Hovner
Dragon User Club, Finland
Panttilantie 4 B 71
SF-00820 Helsinki 87
Finland

THIS letter has been in the system for a while, but the information is useful and probably still up to date. If in doubt, write to them and enquire.

Each way, John?

HAS anyone noticed that if you hold down the keys J, O, and P while playing *Moon Crests*, a message comes up on the screen.

Maybe other software houses will catch on from incentive and have a Dragon and Tandy colour version of the game on the game cassette.

Pear Carter
1110 Stairway Road
Romford, Essex

Video titles

CAN anyone who has used Rainbow Writer for making video titles, please let me know the way they went about it?

D. W. Clarke
41, Somerville, Fulwood,
Preston, Lancs., PR2 3EQ

PS This would make a useful article for a future magazine.

IS anyone interested in taking up one or both of these guarantees?

Stuck up a tree

With reference to *Fire Force*, Jason Orlowski reviews the game and does have one or two inaccuracies.

The gun does work properly when you get the knack of it, and the game is not completely bug-ridden.

The only drawback that I can see is getting stuck in trees, and you cannot always jump over them by zigging the joystick backwards and forwards.

Fire Force is an excellent piece of programming with good graphics and sound.

A plea: does anyone know the Password to the game and what it does? I can get past the Authorization code to the Password but no further.

Daughard Field
14 Camwood
Hessle
H. Humberside
HU10 0PT

Next, please

I READ the review of *Fire Force* with sad sighs, after watching the game on Christmas Day. I loaded it with anticipation of seeing a 'command' style game at last for my Dragon.

At first sight the game looked very good. Excellent sound and graphics, but when it came to the gameplay I was very disappointed.

The figures move jerkily about, and it is almost impossible to last over fifteen seconds with one man.

I do not agree that it is a challenge 'for the serious game player', more an impossible task for anyone.

I am not intending Quixote for the fact of it, *Shogun Master* is one of my favourites. I look forward to seeing *Shogun Master* hoping it is to Shogun standards.

David Humphreys
Coombe Road
Loughborough
Leicestershire
LE11 1JH

PS Please expand the Expert's Arcade Arena, perhaps by getting rid of some of the listings or without staff.

Telewriter disable

OCCASIONALLY with Telewriter, it is advantageous to disable the program's control over the number of printed characters per line. An example would be using it with a printer that has proportional spacing and better built-in justification capability than the word processor. To use such features Telewriter must be tricked into thinking each paragraph is a single line. This required more than just a <CLEAR> which only disables wordwrap on the screen. For the text to be printed properly the <Carriage> per line is set to 128 or larger and all carriage returns left at the end of the paragraph to the end are

removed. The margins and line lengths are controlled by printer escape codes rather than Telewriter (Margin and <Character> commands).

As writing and editing without wordwrap is cumbersome I leave it enabled until the text is ready to be printed. Wordwraps that disabled with a <CLEAR>. Carriage returns, which are not displayed, are found by positioning the cursor at the start of the paragraph and executing a <CLEAR> (right arrow). The process is repeated until the only carriage return left is at the end of the paragraph. Carriage returns terminating blank and single lines are left in place. At

this point one must be careful not to use the <CLEAR> (right arrow) command, as it will remove all the carriage returns which were removed.

Printer escape codes for setting left and right margins are defined at the start of the text. Also the <Character> and <Margin> commands, eg 'MO C20', are defined there. The <Margin> can be set to a non-zero value for affecting short lines. One drawback to this whole procedure is that Telewriter no longer keeps track of the number of printed lines. Scope breaks and numbering will have to be done manually printer escape codes.

Allen Alderson
2044 Hockings Street
Lincoln, Notts LN6 5JA

Stylistic changes

THANK you at last for an article on the OS-9 operating system, and in particular how to modify <Stylog> to remove the dead-end line feed and other impracticalities.

However, I must take either your Mr. Pathery to task on the inaccuracies in this otherwise excellent article. As it stands, any attempt to follow the 'change a byte' instructions merely results in an error after error. I only tried to alter 'byte' but discovered the following:

'To change a byte'

- 1) DEBUG = 1 (ok)
- 2) Load <Stylog> (ok)
- 3) Lstyle is ok
- 4) The printout was as expected (ok)
- 5) Do command = ok
- 6) There was a ahead of the = in its copy, there should be no colon, just enter =00 (or whatever)
- 7) C is ok
- 8) The line is quite wrong, it should read:

SAVE <STYLOG> mod
<STYLOG> (see pages 7 to 44 of the OS-9 manual)

9) For another error in this line, which should read:

VERIFY U <STYLOG> mod
<STYLOG> (see pages 7 to 44 of the OS-9 manual)

Note: before carrying out steps 1 to 5 it is necessary to save the original <STYLOG> from disk. If otherwise call the file '<STYLOG>' or something and rename it later.

10) AFTER <STYLOG> a (a is ok subject to the note above.

I didn't try any more at the time, except the remedy for SPELL, I discovered that the empty file was only one of the problems and as far as I went the article solved this. However, for some reason my copy of SPELL will not modify the last file, nor will it display words in context after the saving process. Attempts to make it do so simply result in an error message.

I thank Mr. Pathery for an otherwise fine article and I have already written to the OS-9 User Group for details of membership.

D.L. Jones
Corbett
Queens Road
Brixton
W 8 7NR
LONDON SW2

Here follow edited highlights from David Pathery's reply to Mr. Jones and to DA:

'The errors were mostly caused, I think, by the precise way OS-9 instructions must be given if they are to work, and spellchecking to wilderness and space control would have been very difficult. However, I take full responsibility for misquoting the <SAVE> instruction and forgetting to tell people to delete the old file.

'However, all the locations to be changed should be accurate, and with a file reference to the manual most people would still find the article of use.

'With reference to the problem with SPELL, did Mr. Jones transfer to a separate disk? If

so you will need to copy DEL, REPAIR and possibly COPY because SPELL uses these to change the file.

'To change a byte should have read:
DEBUG = 1
Load <Stylog>
Lstyle is ok
= 00
make further changes the same way
0
SAVE <Stylog> mod stylog
DEL <Stylog>
VERIFY U <Stylog> mod
<Stylog>
AFTER <Stylog> a (a)

The comments on each line were correct, as I hope were all the tables of alterations.

'Also, in case anyone thinks I am repeating myself, myself the letter on the same subject was an alternative, and not intended to be published as well.

'I have had lots of nice letters and phone calls about my article OS-9 Blues, thanks a lot for publishing it.'

David Pathery
7 Heath Road
Chesham
Bucks HP81 3JY

OUR pleasure, David. As you can see, we are repeating repeating you again in a manner of speaking, with alternative versions of the 'correction'. The letter was one of those deliberate mistakes to which the letters page is peculiarly prone — but I hope it means that everybody got the message.

which is added in lines 1070 to 1100 each time the player successfully enters the correct sequence of numbers. It is necessary to take the second character in the string in line 1080 because every number generated by the RND command is preceded by a space when this number is converted to its string representation by the STR\$ command. The loop between lines 1110 and 1160 copies the appropriate page to the top of the screen and produces the corresponding row. The length of time for which the number appears is controlled by the length of the row which gets progressively shorter as the player manages to remember longer sequences. The shuffling around of pages in lines 1170 and 1180 is necessary in order to avoid loading any instructions and to

ensure that the correct pages appear the next time around. This shuffling occurs at the other places as well for the same reason.

The response of the player is handled in lines 1190 to 1260. A correct answer produces a suitable display on the text screen. Since it would be a pity to return to the ordinary text mode at this stage, and since also there are no spare graphics pages available, this and a different approach was considered desirable. The display is generated by FORKING in line 1200 (values contained in the GOF statements held on lines 1280 to 1290). The squares come in pairs, the first telling the location and the second being the code for a graphics character. Lines 1300 to 1330 read all pages and values and send a look to 1350 for another value to be added

to the sequence and the display to be again, this time a little faster.

Incorrect answers are dealt with in lines 1340 to 1360, giving the number of correct answers until particular win. The time taken to write out the response 'on screen' is felt to be an important part of the PCOPY command is interesting of this game. Finally should any player reach the grand total of a sequence of 20 correct answers, then the programme congratulates and wishes gives the champion a chance to repeat the feat!

Should anyone not fancy the task of typing in all this program then I would only be too pleased to send them a cassette of it if they send me four pounds (i. sp. Gwyn, Ples Talley, Llanybydder, Aberystwyth, Dyfed, SY23 4HD, WALES).

```

410 'DRAW BORDER
420 LINE (1,1)-(255-1,95-1),PSET,B
430 LINE (1,1+95)-(255-1,91-1),PSET,B
440 RETURN
450 'DELAY
460 FOR I=0 TO 255:GOTO 470
470 RETURN
480 SQR=" "
490 CLR:PRINT##(15+32),"Simon"
500 PRINT:PRINT:PRINT">>>wait for
  up>>>wait!"
510 PRINT(15+32),"c L.ap gwynn,9/83"
520 J=4:FOR I=1 TO 4
530 PROBE1,I
540 PCL83:COLOR3
550 STB=STB+1:STB=MIN(STB,2,1):
  B=ASC(STB)-32
560 AN=AN+1
570 X=150:Y=0:S=32:GOSUB460
580 J=J-1
590 NEXT
600 'PREP INSTR
610 PROBE1,5:PCL83:COLOR2,3
620 FOR I=0 TO 4
630 GOSUB410
640 NEXT
650 L1B="READY"
660 X=20:Y=70:R=32:R5=40:Y5=40
670 GOSUB520
680 L1B="Simon says..."
690 X=30:Y=130:R=0:R5=12:Y5=20
700 GOSUB520
710 L1B="Look & Listen!"
720 X=30:Y=160
730 GOSUB520
740 PROBE1,7:PCL83:COLOR3,3
750 FOR I=0 TO 4
760 GOSUB410
770 NEXT
780 L1B="Simon says..."
790 DATA 33,129,34,146,35,137,36,134,37,133,38,132,39,131,40,129,41,128,42,127,43,126,44,125,45,124,46,123,47,122,48,121,49,120,139,138,137,136,135,134,133,132,131,130,129,128,127,126,125,124,123,122,121,120,119,118,117,116,115,114,113,112,111,110,109,108,107,106,105,104,103,102,101,100,99,98,97,96,95,94,93,92,91,90,89,88,87,86,85,84,83,82,81,80,79,78,77,76,75,74,73,72,71,70,69,68,67,66,65,64,63,62,61,60,59,58,57,56,55,54,53,52,51,50,49,48,47,46,45,44,43,42,41,40,39,38,37,36,35,34,33,32,31,30,29,28,27,26,25,24,23,22,21,20,19,18,17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0
800 DATA 120,140,131,136,135,140,134,136,136,139,136,142,140,143,136,145,142,144,136

```

```

1410 DATA 148, 172, 149, 136, 151, 132, 142, 132, 197, 131, 179, 130, 207, 129, 208, 130, 230, 13
2
1420 DATA 237, 131, 233, 130, 230, 131, 236, 130, 238, 131, 239, 138, 260, 133, 261, 129, 262, 13
3, 263, 133
1430 DATA 265, 132, 266, 133, 260, 133, 269, 133, 271, 133, 272, 132, 273, 131, 274, 130, 275, 13
2, 276, 131
1440 DATA 277, 134, 278, 132, 279, 131, 300, 134, 301, 132, 302, 131, 303, 138, 305, 129, 306, 13
5, 311, 129
1450 DATA 308, 149, 309, 146, 351, 145, 353, 150, 354, 156, 375, 156, 376, 156, 377, 147, 400, 14
8
1460 DATA 420, 149, 421, 148, 422, 146, 423, 149, 425, 154, 426, 154, 429, 149, 431, 146, 432, 14
8
1470 DATA 432, 149, 434, 148, 435, 151, 437, 154, 460, 154, 461, 149, 462, 150, 463, 148, 464, 15
3, 466, 147
1480 DATA 468, 145, 469, 146, 471, 147, 484, 148, 487, 148, 489, 148, 490, 156, 491, 156
1490 DATA 493, 148, 495, 148, 498, 154, 500, 148, 501, 152, 503, 154
1500 GOSUB 650
1510 PCOPYV105:PCOPYV108
1520 PCOPYV107:PCOPYV106
1530 SIN="":IN="":GOTO1500
1540 :INCORRECT ANSWER
1550 LINE 10, 0) = (255, 95), PRESST, BF
1560 LINE "INCORRECT!"
1570 X=0:Y=25:G=0:X=16:Y=25
1580 GOSUB 520
1590 LINE "Start again!"
1600 X=0:Y=25:G=0:GOSUB 520
1610 LINE "Start" + STR$(LEN(ANS) - 1)
1620 X=0:Y=25:G=0:GOSUB 520
1630 FOR=0 TO 20: SOUND 1, 1: SOUND 2, 1: NEXT

```

NON IN STOCK — COWEN'S AMAZING ADVENTURE WRITER

This is the incredible program which turns YOUR ideas into an exciting machine code adventure.

You dream up the plot... Adventure Writer does the work! Only from Peaksoft — supplied on cassette with full instructions. £14.95 post free

KEYBOARDS: If we have any left, there won't be many, so please phone to check before ordering. Dragon 64-type, but suitable for 32s or 616. Simple timing: £19.95.

An SuperSmooth Dragon POWER SUPPLY: 17.95 complete. 5 years guarantee plus life-time service warranty. Optional control switch: £2.50. Direct from Britain's leading manufacturer of replacement power supplies.

JCMYCKOS: General purpose, precision potentiometer Pro-Slides £3.95 (pair), £5.45 (each). Crystal Quartzline arcade-type, with Dragoniser interface £19.95 (each). Chesskit 128 Auditive arcade-type, with Dragoniser £14.95 (each).

BOOKS: Including Inside the Dragon (THE book) £7.95. Plus Dragon User, Dragon Monthly and Dragon's Software numbers. LEADERS: Aerial £2.95, cassette £3.95, printer £11.95.

All prices include UK post and packing.

This is just a small sample of our stock of over 150 Dragon lines. We probably have the world's largest range of GAMING Dragon accessories — chess boards, carrying cases (fast feet, T-shirts, sweat shirts, light pens, stickers, pens, etc. etc. etc. Whatever you need, please phone.

PEAKSOFT DRAGON SUPPLY CENTRE,
48 QUEEN STREET, BALDERFON, NEWARK,
NOTTS. NG24 3ND.

Access/Via orders or inquiries: Phone 0608

762233/Peafax MEX 21896246.

0000



H.C. ANDERSEN COMPUTER A/S

HIGHLY RELIABLE

SOFTWARE AND HARDWARE

WE ARE THE ONLY ONES IN THE WORLD
LICENSED TO DISTRIBUTE OS-9 OPERATING
SYSTEM LEVEL 1 FOR THE DRAGON COMPUTER
INCL. EDIT ASSEMBLER, DEBUGGER, AND THE
SYSTEM PROGRAMMERS MANUAL, ETC. (LO)
SOFTWARE FOR THE OS-9 OPERATING
SYSTEM.

CAS DRAGON COMPILATION PROGRAM FOR TRANSLATING ON
SPEECH AND MODERN CASE DRUGS (BEN MARCH 1987) £29.50

CAS DRAGON MONITOR WITH ORIGINAL OPERATING SYSTEM (LO)
FILE TO AND CHARACTERISTICS PH (LO) IF YOU WANT PLUS DRAGON
£14.95

CAS DRAGON DRAGON DOUBLE BUILT-IN TRACK DRIVER AND RESISTOR
UPGRADES YOUR 48 TRACK OS-9 OPERATING SYSTEM FLOPPY (LO)
TO 48 IN TRACK DOUBLE 16000 105-5 OPERATE TWO SYSTEM FLOPPY
£29.50

CAS DRAGON REAL TIME CLOCKS/EN £14.95

SOFTWARE FOR THE DRAGON:

CAS SUPERWRITER 1 (TEXT PROCESSOR) UPGRADE TO OVER 2200
COL. SUPERWRITER 2 (TEXT PROCESSOR) ON CARTRIDGE £19.95

INCL. CAS SUPERWRITER CASE UPGRADE £19.95

NO ERROR SOFTWARE AND HARDWARE FOR OS-9 FILES AND BUILT-LO
PROLOG AND NO. 10 MONTH GUARANTEE ON SOFTWARE AND
HARDWARE (LO) and the only way to get HARDWARE (LO) £2
MAX. £20 AND/OR FOR SOFTWARE (LO) £2 MAX. £20 (NO 1000-0000
MADE FLOPPY TO)

H.C. ANDERSEN COMPUTER INC.
ENGLANDS/ELJ 388 DR-3775 KASTRUP
TELEPHONE: 01-52 44 04 TELEX: 31916
DRAGON DISTRIBUTOR IN DENMARK.

OS-9 REGISTERED TRADEMARK AND SERVICE USER/USER LICENSE
REGISTERED BY THE DENMARK SYSTEM OPERATORS (LO) (LO) IS
REGISTERED TRADEMARK AND SERVICE USER/USER LICENSE

0000

Dragonsoft

New software for online should be sent to Dragon User,
12-13 Little Newport Street, London WC2H 7PP.

Thinking person's boulders . . .

Program: Stone Raider II
Supplier: Microdeal
Price: £3.95

THE instructions supplied with this game bear some resemblance to the correct instructions, which scroll across the bottom of the title screen. Inso Microdeal can clear up this slackness.

Stone Raider II is a similar game to Baby's a Boulder Crash. The aim of the game is to collect all of the gems in 25 different caves. The number of gems on each screen ranges from one to 125, but the number of gems to collect does not reflect the difficulty of the screen. Once all the gems have been collected you have to find your way to the finishing post.

You control your man using the right joystick. Generally each screen becomes slightly more difficult as you progress through the game, but some screens later on seem easier than earlier ones. The 25th screen is extremely difficult, and I am yet to conquer it. To pick up a gem all you need

to do is walk over it. There are many nasty monsters to stop you in your quest. (By making a fool over you? — Ed.), as well as boulders and slimes.

Different monsters behave in different ways, some always preferring to move left, others moving right. When certain monsters are killed by boulders or by gems, they will turn into mine gems. On some screens there are more gems than you actually need to complete the screen. In order to trap the monsters you need to be some clever, but laying so that the monsters will follow your track is where you can drop boulders on them, which all goes to make the game far more enjoyable.

Falling boulders are another problem. Boulders fall when they are pushed, or when the ground is pulled away under them. The falling boulders will kill you if you don't move fast enough.

Stone Raider II is not the usual shoot'em up and knock'em down game. It requires skill, and it is usually necessary

to think about your next move, as moving the wrong way, moving too soon or too late could either get you trapped in a rock fall or killed on a boulder.

There are 25 screens, and considering you only start with three lives the extra lives included as a cheap facility which enables you to change the number of lives, start screen and speed levels. This facility is useful for people who are not too adept at games playing, and also enables each screen to become a game in its own right.

Obviously a good thought has gone into each screen to make it more difficult than the first appears. I would recommend this game to any arcade or even adventure fan who enjoys a bit of fun. Stone Raider II is the sort of game you can play and play until you have completed a screen, just to prove to yourself it can be done.

Robert Margrave



Tracks per pound

Program: Diskette
Supplier: H. C. Andersen Computer A/S, Englandsvej 380, DK-2770 Kastrup. Price: £85.00 plus p.p.h.

A COUPLE of months ago I reviewed one of H. C. Andersen's new utilities. The second is Diskette.

This is not, as it may first sound, yet another disc doctor program. In fact it provides support for 80 track drives to be used with your standard version of DOS-2. When run the program will allow you to select one or two sectors, 48 or 16 tracks. It then formats the disc to your specification and creates new track drivers to the disc. (It creates a new version of OS-2 boot for those that want to know).

If 80 track drives selected then double buffers are sent to the drive for track movement — is the 80 track drive more than twice the tracks for every one of a 48 track disc. Of course, it goes without saying that you must have the hardware for support all of this, is a single or double sided 80 track disc drive; you cannot use a 40 track disc drive (such as the standard Dragon Data drive) with this software.

Once the disc has been formatted and the drivers installed, the Diskette program is not needed to use the 80 track discs. The advantage is that (with a suitable drive) you get four times as much space on the disc as you would using the standard Dragon drive and normal program. The disadvantage is price — £85.

If you're an OS-2 user and have 80 track drives then you must decide for yourself on the value of this software. It certainly does exactly what it claims to do. If you don't already have a 80 track drive, then the cost of one plus this software would add up to an awful lot of blank discs.

Robin Cadogan



Requires Dragon 64 with disc drive and OS-2 system.

DOS DISC ROM

Program: DOSplus 2.7 Disk ROM
Supplier: P. G. Scott, 1 Redgewood Drive, Primley, Camberley, Surrey GU11 5LP
Price: £15

OH NO! Not another Dragon-DOS compatible ROM? Well, yes, but this one has all the bugs removed! What do you get for your money? A little black thing about 2.5 inches . . . oh, you mean the program. Right. The DOS in this ROM gives all the facilities of Dragon-DOS, plus a number of Dragon-DOS command extensions, plus extensions to other BASIC commands, plus other extensions that I wish it called DOSplus.

Firstly, Dragon-DOS compatibility. All the standard commands are there, though some do give different results if used in the "standard" Dragon-DOS way (your programs are unaffected). These include DIR scrolling slowly to the screen (you can just about read it), with a disc name at the start and a file count at the end; DISKMT checking that you really want to

format the disc in the drive and displaying the track being processed; BACKUP on a single drive stopping the motor quickly; BCP performing the same with cassette and disc. There are also extra error checks — it is now difficult to make your drive stop like a machine gun when the limit stop is reached.

A number of Dragon-DOS commands have been extended to give extra facilities: DIRMT will write disc name when formatting, RENMT will rename discs, DIR can be made to page directory listings and output to any open output stream (ie printer, cassette and disc file), BCP, LOC, and LCP will accept filenames or stream numbers (not just letters). I'll get to that later. LOC can also be used to set the read pointer. LOAD will load segmented binary files and KILL can be used with a special filename to work through a directory asking for files to be deleted.

However, in addition, the following enhancements are thrown in free. RESTORE will accept a line number, DPCN

can be used to open disc files in four modes (input, output, append, empty), allocating a stream number in the range 1 to 15. If you are not sure what I mean, the cassette is stream 1 and printer is stream 25. To complete this, INPUT, LINE INPUT and PRINT have been enhanced to work with disc programs and CLOSE has been extended as well.

Finally, you can select different keyboard modes (invo or repeat) and user standard character input/output routines for disc IO, from machine code, in the same way as for cassette (but easier).

That's not too bad and this is the same size ROM as Dragon-DOS. Now PLEX 1.3.1 doesn't boot, but P.G. Scott will supply the PLEX connections with the ROM on request and the DOS is fully supported, with all programs investigated and your EPROM programmed with the corrections for DCP.

Give yourself a treat and get DOSplus 2.7, the ultimate update of the ultimate DOS for the Dragon.

Robin Warwick



TURBOCHARGE YOUR DRAGON:

With our great value hardware and software:

BASIC 42

Extended BASIC for the Dragon 42

For Dragon42s (please state version) £14.95

Run your Dragon in 84K mode, while retaining BASIC and DOS. Print on 4-line screen, using standard PRINT commands, and a screen layout of 24 rows of 42 columns. Other features include:

Alternative, redefinable character sets, control key for special characters, repeating keys, and commands in lower case, windows, CATCH command for automatic return to window, inverted video (green on black/black on green), true underlining and extra PRINT commands and functions.

LIBRARY lists commands and functions. Automatic startup of BASIC program. TEXT command for software compatibility. Sell 25000 bytes free to BASIC. Packsize for Dragon42s 10. One load in extra UTILITIES from disk.

HELP UTILITY £3.00

Extensions to BASIC 42 include: storage cursor, character, word, double, phrase listing, BREAK, double, improved TRON (allows single stepping), Detailed help and error messages.

SPOOL UTILITY £5.00

Use computer while printing. 32K print buffer. TRYST program runs Dragon 42s type writer.

NEW FOR '87:

ICONS UTILITY £5.00

Put icons in your program! Controlled by cursor or "mouse". Commands to define, clear, load and save icon positions and windows.

STRUCTURE UTILITY £5.00

Another fast Structured BASIC on the Dragon! Allows named procedures, improved loop controls by WHILE...REPEAT, and REPEAT...UNTIL, etc.

DOS UTILITY £5.00

Mate friends with your DOS! Enter all the main DOS commands, plus LIST, EDIT etc, directed files by cursor or "mouse".

WHAT THEY SAY ABOUT BASIC 42:

"probably the best step so far" Dragon User

"an invaluable utility" Dragon Update

HARDWARE

Memory Upgrades	FROM	£55.00
Superior controller (chip only)		£18.00
Peaksoft joystick (ideal "mouse")		£5.50

DISK SOFTWARE FOR DRAGON 32/64/128 WITH DRAGONDOS/ROMANA DOS 2.0

DISKREAM (Greenwood) £18.00
The standard Dragon Diskreamsembler

D.R.S. (Greenwood) £9.00
Machine code database program

SOURCEMAKER (Pencemon) £9.50
Disassembler for use with DISKREAM

DISK-IT (Pencemon) £9.95
Sort out your disk problems

PRINTER CONTROL (MacGowan)* FROM £15.00
A text AND graphics processor

DUMPER (MacGowan)* FROM £15.00
Reliable screen dump program

* Prices vary according to printer: please specify.

HONEYBOX (Harris) £14.95
Home and small business accounts

MAILBOX (Harris) £18.95
Selective-mailing list program

SHAREBOX (Harris) £18.95
Manage your stocks and shares

SALESBOX (Harris) £18.95
Balance B/P Sales Ledger

BILLSBOX (Harris) £18.95
Balance B/P Purchase Ledger

CASHBOX (Harris) £18.95
Double-entry Nominal Ledger

STOCKBOX (Harris) £18.95
Full featured Stock Control

ORDERBOX (HARRIS) £18.95
Invoicing linked to Sales or Stock

Cheques/P.O.'s/Further details/dealer enquiries to:

HARRIS MICRO SOFTWARE

49 Alexandra Road, Hounslow, Middlesex, TW3 4HP

Tel: (01) 570 8335

DM-1

Searching techniques

Dene Bebbington shows you how to find quickly what you're looking for

MANY users have probably used some form of searching while programming as the need to search lists of data arises in many applications — eg databases, mailing lists etc. In these applications searching is often one of the most used processing activities, so it is desirable to keep the time spent searching to a minimum. Searching methods will vary greatly in performance.

When measuring the performance of searching routines we count the number of probes which are made in the list to find the item we are looking for. If we use N to represent the number of items in the list then we find that the number of probes to find an item can vary from N to just 1 depending on the method used. If we choose a 'good' method, searching time can be greatly reduced.

The method chosen depends on whether the list is sorted or not. Most of the time it is much faster to search a sorted list than an unsorted one, but there is an exception in the case of hashing — this will be looked at separately.

Unsorted lists

Here we have an unsorted list to search (there is a simple technique we can use. Start at the beginning of the list and examine each successive item until the one you are looking for is found, or until you reach the end of the list if it is not there).

This method is known as a **LINEAR SEARCH** and is easy to implement. See listing one for an example. Here $R2$ is the item being searched for and the list is held in array $A(1)$ of length N . If $R2$ is not in the list then control is transferred to line 50, otherwise the program jumps to line 70 with its position stored in I .

Listing 1

```
10 I=0
20 I=I+1
30 IF A(I)=R2 GOTO 70
40 IF I=N GOTO 20
50 PRINT "NOT FOUND"
60 END
70 PRINT "FOUND AT " I
80 END
```

This method is not suitable for anything but small lists as in the worst case (where the item is at the end of the list) it needs to make N probes. In the best case (where the item is at the start of the list) it will only need to make one probe. Thus, on average, it will only need to make $N/2$ probes for successful searches. For unsuccessful searches it will always have to make N probes.

Sorted lists

If we have a sorted list then we can make use of its order to improve searching times. We could use the ordinary linear search but this would be inefficient as it does not take into account the ordering of the list, and would make unnecessary probes.

Firstly there is the 'intelligent' linear

Listing 2

```
10 I=0
20 I=I+1
30 IF A(I)=R2 GOTO 70
40 IF A(I)=R2 AND I=N GOTO 20
50 PRINT "NOT FOUND"
60 END
70 PRINT "FOUND AT " I
80 END
```

search, which is essentially the same as the ordinary linear search except that we can stop searching if the item we are looking for is less than the current item being looked at (see listing two). Below is an example. The item we are looking for is the name Henry, and here we would stop searching at the name Henry.

LIST: Alan David Eric Gordon Henry Ian Mark Paul Robert

This search method only has to make $N/2$ probes on average when the item is not in the list or not, which makes it better than the ordinary linear search for sorted lists.

Secondly there is the **BINARY SEARCH** which is a very efficient searching method for any size list. The way it works is to split the list in half and see if the middle item is the one we are looking for. If not, it works out which half it should be in and then repeats the process.

Thus as we halving the size of the list to be looked at after each probe. Below is an example of the probes we would need to make to find the name Paul.

List (1) (2) (3)

Alan			
David			
Eric			
Gordon			
Henry	Henry		
Ian			
Mark		Mark	
Paul			Paul
Robert			

If an item is not in the list then searching is stopped when the length of the sublist being examined is 1.

Using the binary search we find that if an item is in the list then only 1 probe is needed in the best case, and only $\log_2 N$ (base 2) probes are needed in the worst case. If the item is not in the list then $\log_2 N$ (base 2) probes will always be needed. We can see

Listing 3

```
10 B=1
20 T=N
30 H=FIX((B+T)/2)
40 IF A(H)=R2 GOTO 90
50 IF A(H)=R2 THEN T=H ELSE B=H+1
60 IF T=B GOTO 30
70 PRINT "NOT FOUND"
80 END
90 PRINT "FOUND AT " H
100 END
```

that the binary search is significantly better than the other two methods looked at. It is not very difficult to code and is thus recommended in applications where sorted lists are used (see listing three).

Hashing

Another form of searching is hashing, which is not strictly a form of searching but a way of inserting items into a list, leading to efficient searching. A hashing function is applied to the item which generates the position in the list where it is to be inserted. A problem arises when a collision occurs (where the position is already occupied). Some way of handling the collision is needed.

If we can derive a hashing function which results in a low number of collisions then searching will be faster. If we could find a hashing function which leads to no collisions then all searches would only need 1 probe, however, we can't really expect this and so searching using hashing will take more than 1 probe but will tend to be more efficient than the other methods looked at.

Generally we find that the best results are obtained when the size of the list is a prime number, for example if we want to hold about 100 items then it is better to use a list of size 101 which is a prime.

Here I will present a hashing routine which leads to quite good search times, on average. A secondary hash function is used if a collision occurs and if a collision occurs

again then the item is inserted in the next free position in the list — this is known as **LINEAR PROBING**.

The way it works is to add up all the ASCII values of the characters in the item (P3) and then to multiply this by the length of the item, giving T. The value of T mod H is then calculated and H is added; this is then used as the position in the list for insertion. If there is a collision then we use the secondary hash function which is the original hash function plus the length of the item. If there is also a collision at this position then we check each consecutive location until a free position is found, whilst doing this we 'step' to the beginning of the list if the end is reached.

Listing four is an implementation of the method set out here, and it could easily be converted for use in a database, stock control system, etc., while the hashing function could be changed to improve the performance.

As can be seen from the program searching is done by applying the hash function to the item being searched for; if the item is not at this first location then we apply the secondary hash function to the item, if at this next location the item is not found then linear probing is used until it is found, or until we circle back to the first location determined.

Another searching method is available to us but can only be used on sorted lists. I call this the **PARTITIONED LIST SEARCH** as the way it works is to partition the list so that only the relevant partition need be searched. For example, if we are dealing with items consisting of just letters of the alphabet then we can partition the list into 26 parts, each part containing items beginning with a different letter. Thus the first partition contains all items beginning with 'A', the next with 'B' and so on.

We can use anything as a table to hold the partition information, the first entry in the table says where items beginning with 'A' start. The next where items beginning with 'B' start, etc. When searching we can then use this table to find the relevant partition to search.

See below for an example list and table.

LIST PARTITION TABLE

A:1	5:1	14:8
A:2	5:3	15:5
B:3	5:4	16:11
C:4	4:5	17:0
C:5	5:5	18:5
F:6	6:5	19:13
G:7	7:7	20:0
G:8	8:5	21:16
H:9	9:5	22:0
H:10	10:5	23:19
P:11	11:0	24:0
P:12	12:5	25:0
S:13	13:5	26:19
S:14		
S:15		
U:16		
U:17		
W:18		
Z:19		
Z:20		

As can be seen from example, all items beginning with 'A' start at position 1, those beginning with 'B' start at position 3 and so on, ending with those beginning with 'Z' starting at position 18. Any entries in the table with a zero mean that there are no items beginning with the corresponding letter. Also the last non-zero entry in the table represents the end of a partition.

To implement this search we need 2 routines, the first routine creates the partition table (**Listing five**) and the second routine finds the end of the partition which the item being searched for is in (**Listing six**).

Using this method we can apply either

the linear or binary search to the appropriate partition, the improvement in performance is greater with the linear search than with the binary search. The improvement comes from the fact that it only needs to examine fairly few items. For example, if we are searching for the name Paul then we need only look at the items beginning with 'P'.

The searching method chosen for use in an application will depend on whether sorted lists are being used, what performance is required, and also how much space is available. Generally the binary search is a good choice for sorted lists, and hashing for unsorted lists.

Listing 4

```

10 * MAIN PROGRAM WOULD GO HERE-

20 B$=STRING$(255," ")

30 DEF FN H(X)=X-((X\N)+N)+1

100 * ROUTINE FOR INSERTING ITEMS-

110 * ASSUMES DIM A(X) HAS BEEN DONE-

120 T=0

130 FOR I=1 TO LEN(B$)

140 T=T+ASC(RID$(B$,I))

150 NEXT I

160 T=T+LEN(B$)

170 IF B$(FN H(T))<>B$ GOTO 210

180 A$(FN H(T))=B$

190 RETURN

200 * SECONDARY HASH

210 NP=FN H(T)+LEN(B$)

220 IF NP>N THEN NP=NP-N

230 IF A$(NP)<>B$ GOTO 270

240 A$(NP)=B$

250 RETURN

260 * LINEAR PROBING-

270 NP=NP+1

280 IF NP>N THEN NP=NP-N

290 IF A$(NP)=B$ THEN A$(NP)=B$ ELSE GOTO 270

300 RETURN

```

```

300 : BEEPONE FOR SEARCHING-
310 : LB IS ITDN TO BE FOUND.
320 : IF LB POSITION LB FOUND AT.
330 : IF LB IS NOT FOUND THEN P=0.
340 P=0
350 T=0
360 FOR I=1 TO LEN(LB)
370 T=T+ASC(MID(LB,I,1))
380 NEXT I
390 T=T*LEN(LB)
400 IF BEEP=NOT(LB) THEN
    P=0:GOTO:RETURN
410 BEEP=NOT(LB)
420 IF BEEP THEN BEEP=0
430 IF BEEP=LB THEN P=P*RETURN
440 BEEP=1
450 IF BEEP=NOT(LB) GOTO 320
460 RETURN

```

Listing 5

```

100 : BEEP=0 P=0.
110 : BEEP=0 P=0 HAS BEEN DONE.
120 : BEEP=1 IS THE PARTITION TABLE.
130 I=0
140 I=I+1
150 GOTO:LEFT(LB,I,1)
160 BEEP=0:GOTO:400
170 IF T=0 THEN T=0+1
180 IF BEEP=0,AND I=0 GOTO 140
190 RETURN

```

Listing 6

```

100 : LB IS ITER BEING SEARCHED FOR.
210 : E IS THE END OF THE PARTITION.
320 B=LEFT(LB,1)
330 B=RIGHT(LB)
340 IF T=0 THEN B=1:RETURN
350 I=1
360 IF I=0 GOTO 340
370 B=0
380 RETURN

```

STOP PRESS . . . NEW TITLES . . . STOP PRESS

CATACOMB ORIGIN	Can you escape the dreaded labyrinth of Tyne mountain? I doubt it.	£7.95	GA,CA,JB
BEASTALKER	Can you help poor Jack reach his goal in this Pz. 1 game?	£7.95	AA,CA,DR
STARSHIP DESTINY	Only you can diffuse the bomb and save the World. Can't you?	£5.00	GA,CA,DR
DUNGEON DESTINY	Castles are not the most hospitable places, but especially this one!	£5.00	GA,CA,DR
WILD WEST DESTINY	Alone in a strange town with no money. How will you get home?	£5.00	GA,CA,DR
COMPOSER COMPANION	This package provides a graphics interface to Microsoft's 'Composer', no more DATA statements (REQUIRES MICROSOFT'S 'COMPOSER' PACKAGE)	£5.00	UT,CA,DR
HI-TEXT	A collection of screen drivers that give a 64 x 24 display.	£5.00	CA,UT
LIGHT-MAIL	A light-pen driven mailing list program.	£5.00	CA,DR,DO
DEMANATCH	Used with Beassembler load, save and edit the 50 game screens.	£2.00	CA
EL-DREA	A collection of twenty ragtime and jazz compositions.	£3.00	MJ,DR,DO

FLEX SOFTWARE

CURSE OF COMARD	Sheer excellence, no adventurer should be without this one. A classic.	£10.00	AD,FL
SHERLOCK	An immensely powerful sector editor with many additional features.	£10.00	UT,FL
OMON	FLEX Machine code programmers monitor program.	£10.00	UT,FL
FLEXPACK-1	FLEX program collection. Phone Help '90 Diary form and more.	£5.00	UT,FL

CA—Casette DR—Dragon DO—DOS GC—Galle DCS JB—Requires joystick AA—Arcade Adventure
AD—Adventure GA—Graphic Adventure UT—Utility MJ—Music FL—Flex disk

ALL THESE AND MORE! SEND S.A.E. NOW FOR FULL DETAILS
ALL PRICES INCLUDE POSTAGE AND PACKING

MICROVISION SOFTWARE

8 Wansley Road, Liverpool L9 8DW

Introduction to wordprocessing

Roger Merrick checks off the vital points of computerised text handling in the first of two articles

It seems to me that for anyone with a need to do any written work on paper, and preferably some experience with a manual typewriter, the single most useful purpose that a home computer system can easily serve is that of word processing.

For the beginner, or the uncommitted, the computer must be able to demonstrate some advantages over the traditional methods of pen — or typewriter — and paper. Why would I use a word processor when I could scribble something on a piece of paper with a biro?

Given that the hardware and software requirements are met, a wordprocessing system offers the following advantages over the alternatives:

a) high standard of presentation — readable, neat, nothing crossed out, no Tipples.

b) easy reworking of text — if you've typed or written a page, you feel less than keen to rewrite it if a better way of expressing yourself occurs to you (or worse, if someone peering over your shoulder). On the wordprocessor, it's a piece of cake.

c) work can go from rough notes to completed text in one operation. Because it is easy to reformat text, I tend to put my original thoughts straight into the computer in the order that they occur to me. Then I can expand, reorganise, present, edit, reformat.

d) selective typing centred — instead of typing "From A. Bloggs, 21 Railway View" at the start of each letter, you can look from a directory at a couple of keystrokes; standard letters can be stored and customised as required.

e) a word processor can be used as a database, archiving books, articles or records, names and addresses, as a diary, and so on.

For the computer system to stand any chance of competing with the two certain minimum requirements must be met.

The computer system consists of three important components:

- 1) the computer system/hardware
- 2) the wordprocessing software
- 3) the user

First, to consider the user: either user must be committed to using the new system because there will be a latency period during which it will be slower to use the computer than the alternative. Either user must have a need to be met by the wordprocessing system. I mean that if your writing needs per se almost are flat a dozen thanks, you letters to relatives and some apologies to the Bank Manager, then it is hard to justify the cost of the word processor, and it is hard to generate the time/skill necessary to learn to use it efficiently: whereas if the user is a student having to prepare and submit substantial amounts of written material, or uses a word-

processor at work, or has any special interest that involved generating written work, then, given that other conditions are met, a word processor will offer a significant improvement over alternative methods of generating text.

c) the user needs an ability to learn, to follow instructions, to accept his or her own lack of understanding rather than always blaming the equipment, get to infer what the instructions provided are incorrect and, if/when, will have already got past the very worst "hand and peak" stage of typewriter keyboard literacy. You don't need to be a touch typist to use a word processor, indeed, in my opinion it is undesirable (because touch typists are taught to type in a certain way because of the constraints of the mechanical keyboard), but you would find it helpful to have a pretty good idea of where each key is.

Secondly, the hardware:

a) unless the hardware is a self contained portable system (in which case this point goes without saying), the computer system should be always in a condition where all the user needs to do is switch it on (and, I will accept, load the software) — you're in to a lesser if, instead of picking up a biro, you've got to beg the TV off the kids or the loaned one, carry it to the computer room, find a three-way plug adaptor etc., etc., messing about for an hour.

b) the system should consist of a computer, monitor, printer, one or more disc drives.

The computer should have:

- i) upper and lower case,
- ii) at least 48 by 24 character display, more is better,
- iii) near typewriter quality keys,
- iv) standard qwerty key layout (including punctuation keys, shifted numbers, F1/F2/F3) and so forth

v) memory of a size sufficient to allow, after operating system and wordprocessing software, about six A4 pages of text. More free RAM is fine, but not less. You see, I have this idea that there is some statistic to say that a typical document can be contained well under this amount of space (Don't you believe it). The shortest letter effectively fits up 1 or 2K, and anyone doing letters or memos will go because without a disc drive — 60,000 is undesirable to be always filling up or running out of memory space, which adds to the time required, saving a document, breaking it into a number of parts, and slows down the operation of the software as, for example, string clearing operations have to be carried out increasingly often as the memory fills. The word processor I use at work has about 20K free at maximum, and I have never run out of space. Though I write on it every day, most documents I create are far less than six pages in length.

The monitor should be high quality, high resolution, because the user will be spend-

ing quite long periods of time up close to it. The screen should be large enough to display characters that the user can read without strain. Black and white monitors are relatively cheap (£50 to £100), it should be positioned at an appropriate height so that the user does not need to crane his/her neck to look at it, and so forth.

Wordprocessing does not require colour or sound, but whether either of these are desirable will depend on the software. Colour monitors of sufficient quality still cost a lot of money. Amstrad's colour monitors are not, in my opinion, of high enough resolution to be really usable in wordprocessing unless the working situation permits the screen to be positioned at some distance from the user.

[Speaking as someone who spends whole days at help meetings, including a standard Amstrad PCW monitor, as well as cheaper and more expensive ones, I would recommend that anyone using any monitor at all at least looks at and/or preferably tries one far away from it, as the glare from the screen is doing to the eyes when the resolution is first used. Experiment till you find a comfortable position. — Ed.] I must acknowledge that there have been very good results obtained on a 14" black and white TV and a Dragon running Bellerose.

The printer should use ordinary paper:

- not too much expensive, not always easy to get hold of, good quality computer/typing paper

- not narrow width because it limits the use to which the output can be put

The printer should use an ordinary ribbon or a standard cartridge

- not strike cone cartridges or little spin pots because they are too expensive in use

- not highly specialised ink units, eg the solid ink in a glass tube of the Olivetti Spangol printer, because you try getting them when the Spangol is discontinued (dealers will, of course, offer continued support while there is a demand, like floots did for the Dragon)

- which conforms to a current standard — eg IBM, Centronics plug/punch

- which is able to produce the full ASCII range of characters

- which produces print that is of readable quality

In my opinion, the speed that a printer works is not a major problem in word processing — you can always have a cup of coffee while the printer is working. (Once again, this is a time for letters or memos, but it can be a real problem if you are creating longer documents, or writing to a deadline — Ed.) If you buy a standard machine, it may come with, or you may buy, a 24 or larger buffer if you need it. Alternatively, your software may support "background printing", where printing is done without locking up the keyboard. The critical feature of a printer is the quality of the

print — if it's not a satisfactory one you forget that printer. If you used a DMP1101 it would always select a correspondence quality font even though it took a bit longer to print.

The different types of printer can be described in a nutshell:

i) **Dot-matrix.** Epson or compatible. This is the effective current standard in dot-matrix printers. Fast, medium-quality print, widely available ribbons. Some compatibles use ordinary typewriter ribbons that only cost about £1.50.

ii) **Daisy-wheel.** These produce the best quality print currently. It is expected that developments in high quality dot-matrix printers will knock the bottom out of the daisy-wheel market, as it doesn't spend a lot of money and it makes sure you get a standard one.

iii) **Pengliners.** For example the GCP115. A couple of years ago this was the most big thing. I found them unsuitable for the following reasons:

a) narrow, bill-fold type paper
b) pens were of variable quality and short life — they would run out or perform during quite ordinary length documents, at 50p each, they are expensive

c) the character set in the 8-inch version is very poor (it is good on the standard GCP115, however)

d) idiosyncratic operation — commands are unlike any other printing device.

Essentially, the machine is being used out of context — as a plotter it works great, as a printer it's limited.

iv) **Others:** laser printers were great but too costly for home use. Bubble printers are standard in limited numbers and you'll be shocked if supplies stop when production is discontinued, some give poor quality lettering — the ink blurs, the character sets are poor; thermal printers give highly variable quality print and thermal paper is limited both lump — you won't find a good poor impression if presented in this way.

The data storage device could be one or more floppy disc drives. While a fixed or mine where it is managed to do for work on a cassette-based system, it is obvious that the full advantages of the word-processing system will not come about until disc is connected. The disc offers FAST saving and loading, encouraging regular backup — this article leads into whether in a couple of seconds.

Dragon DOS offers an automatic backup facility, discs are far more reliable than cassettes, under normal circumstances the documents saved are automatically catalogued (DPS) — you don't need to load a file before you discover what it is. It is practical to use a letterhead from disc — on cassette it may take longer to load a standard covering paragraph than to type it. The range of advantages available from using disc depends on how many drives you have and what the software you are using offers (Mail merging or Spellchecking, for examples).

The alternatives in mass storage are worth mentioning:

i) 5.25, 3.5 and 3 inch, single or double sided, 40 or 80 track, half or full height, floppy disc drives are all functionally equivalent on the Dragon with Dragon-

DOS. These are the only valid choices for mass storage.

ii) cassette: the Dragon's DOS is not the slowest, not the least reliable, but it does not compare with the speed and reliability of the disc system.

iii) 8-inch drives can, I understand, be connected to the Dragon DOS cartridge. I haven't tried it. 8-inch drives and discs are disproportionately expensive compared to other forms of drive. If you can get one cheap, it might be interesting to play with.

iv) proprietary fast tape drives — eg Burrough's Quickdrive, UltraDrive, Hobbit, Easton Stringy Floppy etc. These are OK if they don't require special tape cartridges (in-house made cassettes), but if they do, then when the model is superseded or the company goes bust, you're stuck. Also, the operating system may not be easily compatible with your main disc operating system or whatever. If you ever upgrade to another machine (I do, for instance), your proprietary storage device is unlikely to be usable with the new system (you can just plug your disc drives into most systems — in my case, when I switched my Olivetti drives from a Video Genie to a Dragon, I literally just disconnected one cable and connected another; they are not as fast or as reliable as disc drives. No fast tape system can offer random access in the way that a floppy disc drive can.

v) hard disc system. These, which not so long ago would have been prohibitively expensive for a home computer system, are now looking to be in the range of the well-off enthusiast, or the home businessman. The snag is that they will require a special operating system and interface before they can be connected to the Dragon. You would need custom software too. An operating system like DOS-II would really come alive with a hard disc (and half a meg of RAM).

There are no other mass storage devices of any relevance.

So how does the Dragon meet the hardware requirements?

It has a near typewriter quality keyboard, and in terms of the number of keys and their layout, closely matches a conventional typewriter.

It offers a standard parallel printer port, TV or composite video monitor output, easy-disc interfacing.

In storage configuration, 32, 'regular' 40, or upgraded 80 to 80, it offers a useful amount of memory.

As it stands, it does not give upper and lower case. There are facilities for software-generated upper and lower case drivers, but obviously they consume RAM. Have you ever wondered, by the way, why the VDG doesn't have upper and lower case? It seems it was to save a small bit of RAM. Remember that the original 'Tandy CoGo' was offered with 4K or 8K, and the handling of bytes involved for upper and lower case made a difference then. If you have an expanded 32, it is possible to hide a high res screen driver in the upper 16K of RAM and still have disc I/O.

With the 64 or 96 screen, the Dragon offers more characters on screen than most other home computers (a quote from the letterbox advertising).

If you are reading this, you are likely to have a Dragon. It connects via a cheap cable (80 new, from Tandy) to a standard printer. You could get a second hand Epson compatible printer for under £400, or not much more, if you don't have discs, the most expensive requirement is the DOS cartridge. Make sure you get the Dragon-DOS or Commodore-DOS versions. Premier-DOS, which also looks good, is now a definite no-no. New drives are costing £30 or so, with PSII and case extra. If you shop mail order, you'll need your own screen. The cost of getting the shoe on the road would buy a hell of a lot of tape.

Finally, the software:

Wordprocessing software should meet certain general criteria:

i) it should be easy to use.
ii) it should prevent the user from doing anything really stupid, or, perhaps preferably, give the user a chance to change his/her mind before proceeding.

iii) it should not stand in the way of the user doing what s/he wants with the text.

iv) it should enable the user to access all relevant computer facilities from within the program.

v) it should take advantage of the facilities of the hardware to provide a fast-acting system.

vi) it seems self-evident to me that the operating system and wordprocessing software together should leave an appropriate amount of memory free to use for text generation. In a disc based system, operations will be significantly slowed down if continual disc accesses have to be made.

vii) it should not attempt to be a jack of all trades — it should be designed to enable the user to generate text quickly and easily, not clog up memory and slow down its speed of operation with functions of limited, specialised, and, irrelevant to text generation.

viii) it should store data (text) in a standard form so that files can be taken from one program to another for specialised applications.

ix) it should be able to be configured to the system being used.

For display, the user should be able to select background and foreground colours (limited only by hardware, not software), for storage, single or twin drives, 40 or 80 track, single or double sided, for printing, the printer should be driven by a user-configurable module, specifying for example, whether or not the program should send linefeeds with carriage returns, for typing, the keys on the keyboard should perform the functions engraved upon them. If the program has been written for use on a number of different machines, it should be possible, as with some games, for the user to select what keys perform what functions. The user configurations should be able to be stored as the user does not have to type them all in every time the program is used.

Essentially, you need to be able to match the 'up and away' convenience of sending a bit of paper into a typewriter and then, in top effort, gain the advantages of the computer system.

Into the directory

Paul Daglish opens up the Dragon DOS directory for programmers

THIS article is intended to give an overall view of the organisation of the Dragon DOS directory. I was prompted into writing it when I read the article by G. Rothery on page 8 of Dragon User, November 1988. He did describe a lot of the information presented here but ignored parts which tend to cause difficulty to programmers trying to write 'directory access' programs for themselves; notably when directory entries have been extended by DOS. I hope this complementary information will help you to understand more clearly how this

program works. I have repeated a fair amount of what he said, but this is because I have aimed at presenting a complete reference document.

As most regular readers of Dragon User now know the Dragon DOS directory is stored starting at the third sector of track twenty and continues to the end of the disk, a total of 18 sectors. The whole directory is organised into 'slots' each of which is 26 bytes long. Thus each sector has room for 18 slots with 8 bytes left over at the end. (It is these free bytes which are used in Mr

Rothery's program to store a directory.) This, then, provides us with a theoretical maximum of 162 files on a disk. Although each file has only one 'entry' in the directory, each entry can be composed of one or more 'slots', thus reducing the possible number of files in the directory. This distinction between 'slots' and 'entries' must be always born in mind.

The first slot of a directory entry always has the same format, as shown diagrammatically here:



Each 'initial' slot is divided into eight fields:

byte 1: Flags
bytes 2 to 9: Filename
bytes 10 to 12: Filetype
bytes 13 to 15: Pointer 1
bytes 16 to 18: Pointer 2
bytes 19 to 21: Pointer 3
bytes 22 to 24: Pointer 4
byte 25: SIZESLPR byte

These fields are described in detail below.

Byte 1: entry status flags

Bit No.	Names	Meaning
0	Entry Type	0 = Header entry (filename) 1 = Continuation (of another entry)
1	Protected Flag	0 = Not protected 1 = Protected
3	Directory end flag	0 = More entries in directory 1 = End of directory
5	Continued flag	0 = file continues 1 = This entry is continued
7	Valid file flag	0 = This file OK 1 = File has been deleted
2, 4, 6		Not used

Bytes 2 to 12, filename and filetype

Bytes 2 to 9 contain the ASCII filename (eg 'THISPROG'), and bytes 10 to 12 contain the ASCII filetype (eg 'SAS'). If the filename is less than eight characters long then the zone for the filename in the directory is padded out with zeros. The filetype zone is padded to fifteen characters in a similar way. Note that the pole character ':' between the filename and the filetype is not stored in the directory. Thus the name 'MYPROG.DAT' will be stored as follows:

Byte No.	2	3	4	5	6	7	8	9	10	11	12
Hex Value	4D	59	59	52	4F	47	00	80	44	41	54
ASCII value	M	Y	P	R	O	G			D	A	T

Bytes 13 to 24, file storage pointers

Data (in the form of files) is stored on disc in 'clusters'. A cluster is simply a block of storage allocated on the disc; it is always an integral number of sectors long. When DOS writes a file to disc it allocates a free cluster and writes the data to this area. However, because of fragmentation of the disc due to file deletion, the original cluster may not be big enough for all of the file to be stored. In this case DOS finds another free cluster and continues storing data there. This process of finding unused 'holes' on the disc and filling them continues until all of the file is stored (or no more holes can be found) — DF EPICURE. Thus to recover the data contained in a file we need to know the location and size of each cluster as well as the in which they were allocated. This is what these fields called the storage pointers. Each storage pointer contains two entries: 1) a two-byte Logical Sector Number (LSN) of a cluster, and 2) a one-byte cluster size in sectors. The location on the disc of each cluster is given by the LSN. This is a number which starts at around increments to the maximum number of sectors on a disc (720 for single sided 40-track disks). It is calculated by multiplying the track number (starting at zero) by the number of sectors per track (10 for DRAGON DOS) and then adding the sector number for that track (starting from zero also). Thus the start of the Dragon DOS directory which is at track 20 sector 3, has an LSN of 360 (20 x 18) + 3. It also follows that the first sector of the second track has an LSN of 18. As an example let us consider a file which is stored in three clusters as follows:

cluster 1: 5 sectors starting at track 3 sector 5
cluster 2: 12 sectors starting at track 36 sector 1
cluster 3: 4 sectors starting at track 11 sector 18

This gives us the following LSNs:

LSN cluster 1 = 3 x 18 + 5 = 59 (003A HEX)
LSN cluster 2 = 36 x 18 + 1 = 723 (028B HEX)
LSN cluster 3 = 11 x 18 + 17 = 219 (00D7 HEX)

Since clusters are contiguous areas on the disc the above clusters will cover the following sectors:

cluster 1: track 3, sectors 5 to 10 inclusive
cluster 2: track 36, sectors 1 to 12 inclusive
cluster 3: track 11, sector 18; and track 12 sectors 1 to 3 inclusive

Byte 25, last sector size

The above example shows that the file occupies 22 sectors, or $22 \times 512 = 11264$ bytes on the disc. It does not mean the file is exactly 5632 bytes long! It may be shorter by up to 55 bytes (any shorter than that would require less sectors). In order to keep track of the logical end of the file, DOS uses the many bytes at the end of the last sector of the last cluster. This value is stored in the last byte of each directory entry. Should the last sector be completely filled then byte 25 will be zero which corresponds to $512 \times 1 + 384$.

Using the above cluster allocation scheme for a file 5632 bytes long called EX-AMPLE.BAS, the directory entry would look like:

```
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
00 48 88 41 40 50 4C 45 00 42 41 53 08 3A 06 02 80 8C 00 07 04 00 00 00 00
```

Note that Pointer4 (bytes 22-24) contains zero. This is a flag to show there are no more clusters for this file entry. It can happen however that more than four clusters are required to store the whole file, especially when the data is heavily fragmented due to frequent use. In this case it is necessary to extend the number of storage pointers for the file list. This is where the "Continued" flag of byte 1 comes in. If this bit is set (=1) it indicates that there is another continuation slot for this file somewhere in the directory. That somewhere is given by byte 25 of this slot. Byte 25 then has changed its meaning depending on the state of the continued flag. If no longer contains the number of bytes in the last sector of the file but the slot number of the continuation of this entry. Directory slots are numbered starting from zero, so since each slot occupies 26 bytes, then the second slot (slot number 1) starts at the 26th byte of the directory, the third at the 51st byte and so on. The second continuation slot may itself be continued which is again indicated by the "Continued" flag of the first byte of the slot. Since the filename is already stored in the first slot for the file, each continuation slot has a different format to the first, apart from the first and last bytes it contains only storage pointers (if of them) plus the unused bytes 22 and 23 as shown below:

gives us no information. This is done using the Bit Type flag of the first byte of each slot. If this bit is set then (and DOS knows it is a continuation slot and we should not attempt to interpret the filename from bytes 2 to 12, if this flag is set then the slot containing it does not contain any useful information but is the logical end of the directory and we go searching for the next slot).

File Protection flag is set in the first byte of the directory entry. This indicates to DOS that the file must not be deleted. This bit is switched by using the PROTECT/CHOFF commands.

One final flag is the "Valid file" flag which indicates if the file has been deleted or not. A newly created file entry will have this flag cleared (=0). If the file is subsequently

deleted using the RLL command then this bit is set to 1, and the bit map of the disc is updated to show that the previously occupied clusters are now free for use. The filename and associated cluster pointers are not removed from the directory, the flag only indicating to DOS that it may reuse this entry if it wishes. Likewise the contents of the newly released clusters are not touched; they are only flagged as free for reuse. It is this mechanism which allows the "unFOLLing" of files by clearing this flag and re-allocating the bit map. It will of course only work if DOS has not written anything to disc in the meantime since it is likely to have overwritten all or part of the deleted file.

A newly formatted disc has all its entries set to 888 signifying (1) an invalid file (deleted), (2) the end of the directory, and (3) a continuation slot (contains no useful information on its own).

A word about file headers. These are not stored in the directory but in the first nine bytes of the file itself. They have the following format:

byte	01	02	03	04	05	06	07	08	09
Hex value	88								AA
			LOAD ADDR		LEN ADDR		EODIC ADDR		
	Type								

The BasicDOS program ReadDD

This program uses all the information outlined above except the bitmap. It was written to enable the transfer of any file from Dragon DOS to OS-9. Should the Dragon DOS file be a Basic program or a memory image file (binary file) then the file load information is displayed and the user given the option of removing the file header from the OS-9 file. If this is done then a new file with 'noheader' added to the name is created. Thus the user has the option of keeping both versions on higher OS-9 disc. A data file which has no header information is copied 'as is'. Should the user possess only one drive then the option is given to use this drive to do both reading from a Dragon disc and writing to an OS-9 disc. This will mean a lot of disc exchanging in response to the prompts but at least a transfer is possible.

Coming to an 'error' in the distributed device descriptor for DDQ and DT in OS-9. DRAGON discs cannot be read directly without a little playing with the device descriptor. This is because the descriptors supplied by Microsoft define the discs as having every file allocated space in 'chunks (clusters)' of 8 sectors at a time whereas Dragon discs are allocated

1-sector chunks. This can be overcome by 'patching' the descriptor for the drive in which you wish to read the Dragon disc in the following way:

```
OS-9>debug          " Call the OS-9 debugger
OS-9>L D1           " Locate the device descriptor
OS-9>...+80         " Open offset 800 (note the space between the two points)
OS-9>+f             " Cluster size = 1
OS-9>q              " Exit the debugger
```

The above five lines (without the prompts) can be added into a file and called as a command file from the keyboard or the OS-9 startup file in which case the patch will be done automatically whenever OS-9 is booted. It is also possible to change the descriptor permanently whether it is in the bootstrap or separate descriptor file, but this is best left to the OS-9 'hackers'. (If you want to contact me and don't mind international telephone calls which are cheaper after 23.00 UK time give me a ring on 090-00-7600008 and I will gladly tell you how it's done). If the drive used for reading the Dragon disc is DDQ then the first command in the debugger would be 'L DQ' instead of 'L D1'. In any case the device descriptor must be in memory at the time of the patch (use LOAD if not), and it does not

update OS-9 to patch entry descriptor in this way since OS-9 normally gets its allocation information from the disc, itself if there is a valid OS-9 disc in the drive. This patch is

```
OS9>ddr dd
or
OS9>dr dd
```

The OS-9 disc can then be removed from the drive and replaced with the Dragon disc. There should be no problem in reading it from then on.

```

PROCEDURE readdd
0000  ("
0001  (" Program to transfer a DRAGON-DOS file to OS9.
0002  ("
0003  (" Written by P.A.Daglish 7-Jan-87
0004  ("
0005  REM Define our special types for easy directory access
0006  TYPE PntType=LONG;INTEGER; Name=BYTE
0007  TYPE DirEntry=flag:BYTE; Name=STRING(8); Ss=STRING(30); Pnt=
0008  (4);PntType; Last:BYTE
0009  TYPE SslType=flag:BYTE; Spn(3);PntType; unseed:INTEGER; Ssl=
0010  :BYTE
0011  TYPE header_Type=first_byte:byte;type:BYTE; Load_Addr,Length,EXEC_Addr
0012  :INTEGER; Last_byte:BYTE
0013  (" Allocate variables using these special types *)
0014  DIM File,Directory(100):DirEntry
0015  DIM Extension:KType
0016  DIM pnt:PntType
0017  DIM header:header_Type
0018  (" Now define the simple variables *)
0019  DIM First_read,Get_Header:BOOLEAN
0020  DIM OS9Name:STRING(12)
0021  DIM Drive,OS9Drive:STRING(1)
0022  DIM OS9Len,OS9n,i,j,k:INTEGER
0023  DIM Filename,Track,Head,Block,last_seek_size:INTEGER
0024  DIM BASIC,BINARY,Head,Locked,DirEnd,Continued,Valid:BYTE
0025  DIM Pnt:STRING(8)
0026  DIM Comp,Reply,FF,NULL:STRING(1)
0027  DIM NumOnline,MaxOnline:INTEGER
0028  DIM Single,Done,More:BOOLEAN
0029  DIM Buffer(255):ByteType:BYTE
0030  DIM File,Filez,Position,Buffer:REAL
0031  (" Initialize some useful constants *)
0032  BufferLen:=SIZE(Buffer)
0033  BASIC:=1
0034  BINARY:=2
0035  MaxOnline:=3
0036  NumOnline:=0

```

```

0308      NULL:=(CHRG(0))
0309      FF:=(CHRG(255))
0310      Rad:=" "
0311      Head:=$00
0312      Locked:=$02
0313      DirEnd:=$00
0314      Continued:=$08
0315      Valid:=$00
0316      Files:=0
0317      (* ----- Start of Main Programme ----- *)
0318      INPUT "Drive number of DRAGON disc? (Default=1):",Drive
0319      IF Drive="" THEN
0320          Drive:="1"
0321      ENDIF
0322      OPEN #0n,"/d"+Drive+"*":READ
0323      SEEK #0n,0
0324      INPUT "Disc number of OSB disc? (Default=0):",OSBdrive
0325      IF OSBdrive="" THEN
0326          OSBdrive="0"
0327      ENDIF
0328      END "/d"+OSBdrive
0329      Single:=0nFile+OSBdrive *(= Single drive transfer ? *)
0330
0331      Track:=0
0332      j:=1
0333      IF Single THEN
0334          PRINT "Insert DRAGON source disc, then hit a key";
0335          (* Wait till he types something on the standard input path *)
0336          GET #0,Empty
0337          PRINT
0338      ENDIF
0339      (* FOR loop reads the whole of the DRAGON directory into our array at
0340      address "Directory" *)
0341      FOR Sect:=0 TO 15
0342          Position:=(Track*16+Sect)*256.
0343          SEEK #0n,Position
0344          FOR i:=1 TO 16
0345              GET #0n,Directory[i]
0346              j:=j+1
0347          NEXT i
0348          NEXT Sect
0349          j:=1
0350          (* The WHILE loop prints out the valid filenames in the directory together
0351          with their entry number *)
0352          WHILE LAND(Directory[j]).flags,DirEnd)=0 OR
0353              FOR i:=1 TO 8
0354                  (* Header requires strings to be terminated by EOF *)
0355                  EXITIF MID$(Directory[j].Name,i,1)=NULL THEN
0356                      Directory[j].Name:=LEFT$(Directory[j].Name,i-1)+FF
0357                  EXIT
0358                  NEXT i
0359                  Files:=Directory[j]
0360                  (* Only print out slots which are valid header entries *)
0361                  IF LAND(Files.flags,Valid+Head)=0 THEN
0362                      PRINT USING "a3r, '-',a0, ',' ,a0",j,File.Name,File.Ext,
0363
0364                      NameOfLine:=NameOfLine+1
0365                      IF NameOfLine=MaxOfLine THEN
0366                          PRINT
0367                          NameOfLine:=0
0368                      ENDIF
0369                  ENDIF
0370                  j:=j+1
0371              ENDWHILE
0372          PRINT
0373      
```

```

0758      (* Ask which file is required *)
0759      REPEAT
0760      REPEAT
0761          INPUT "Select number of file to copy: ",Filenum
0762          UNTIL Filenum<0 AND Filenum<=0
0763          IF Filenum=0 THEN
0764              END "Abort"
0765          ENDIF
0766      UNTIL .AND(Directory(Filenum).Flags,Valid=Head)=0
0767      File:=Directory(Filenum)
0768      OSName:=File.Name+"."+File.Ext
0769      IF Single THEN
0770          PRINT "Insert 525 destination disc, then hit a key"
0771      :
0772      SET #0,Device
0773      ENDIF
0774      CREATE #0BChan,OSName:UPDATE
0775      More:=.AND(File.Flags,Continued)=1
0776      First_read:=TRUE
0777      Got_Header:=FALSE
0778      (* Handle the 1st entry slot *)
0779      FOR i:=1 TO 4
0780          ptr:=File.Ptr(i)
0781          IF i=4 THEN
0782              Done:=File.Ptr(i+1).Headers=0
0783          ELSE
0784              Done:=NOT(More)
0785          ENDIF
0786          Nsects:=ptr.Headers
0787          IF Done THEN
0788              Nsects:=Nsects-1
0789          ENDIF
0790          Position:=ptr.LBN*256, (* NOTE multiply by a read *)
0791          SEEK #0Bn,Position
0792          (* Read any possible header then reread the whole of the first sect
or *)
0793          IF First_read THEN
0794              First_read:=NOT(First_read)
0795              SET #0Bn,Header
0796              (* Re-position to start of the file (header) *)
0797              SEEK #0Bn,Position
0798              Got_Header:=Header.First_byte=256 AND header.Last_byte=
256
0799          ENDIF
0800          RUN readsect(#0Bn,#0BChan,Nsects,File,Buffer,Single)
0801      EXITIF Done THEN "ENDXIT"
0802      NEXT i
0803      (* Now handle the extension entries if any *)
0804      IF NOT(Done) THEN
0805          Done:=.AND(Directory(Filenum).Flags,Continued)=0
0806      ENDIF
0807      IF Done THEN
0808          last_sect_size:=File.Last
0809      ELSE
0810          Filenum:=Directory(Filenum).Last
0811          REPEAT
0812              Extension:=Directory(Filenum)
0813              More:=.AND(Extension.Flags,Continued)=1
0814              FOR i:=1 TO 9
0815                  ptr:=Extension.Ptr(i)
0816                  (* Check if this is the last pointer in the list *)
0817                  (* i.e. the next one is zero or if this is the 9th *)
0818                  (* Is there a continuation entry? *)
0819                  IF i=7 THEN
0820                      Done:=Extension.Ptr(i+1).Headers=0
0821                  ELSE

```



```

0009      Done:=NOT(More)
0010  ENDIF
0011  Nsectors:=ptr.Nsectors
0012  IF Done THEN
0013      (* If we're at the end only read Nsectors-1 complete sectors *)
0014      (* since the last one is normally only partly filled *)
0015      Nsectors:=Nsectors-1
0016  ENDIF
0017  Position:=ptr.LSN*255.
0018  SEEK #DDn,Position
0019  (* Read Nsectors sectors *)
0020  FOR readsect(DDn,DSZchan,Nsectors,File,Buffer,Single)
0021      EXITIF Done THEN \ ENDEXIT
0022  NEXT i
0023  IF Done THEN
0024      last_sect_size:=Extension.Elsest
0025      (* If finished then the last byte given as the number *)
0026      (* of bytes to read in the last sector, otherwise the *)
0027      (* last byte points to the continuation slot *)
0028      (* and the REPEAT loop continues *)
0029      ELSE
0030          Filename:=Extension.Elsest
0031      ENDIF
0032  UNFIL Done
0033  ENDIF
0034  (* Here we read the last (partial) sector *)
0035  FOR readsect(DDn,DSZchan,last_sect_size,File,Buffer,Single)
0036  CLOSE #DDn \(* Finished with #DASDn Size *)
0037  (* Here we put the info found in any file header *)
0038  IF Get_Header THEN
0039      IF header.Type=BASIC THEN
0040          PRINT "File is a BASIC program file"
0041      ELSE
0042          IF header.Type=BINARY THEN
0043              PRINT "File is a binary image file"
0044          ENDIF
0045          PRINT USING "Load address: $'h4,'"header.Load_Addr;
0046          FOR preadr(header.Load_Addr)
0047              PRINT USING "Load length : $'h4",header.Length;
0048          FOR preadr(header.Length)
0049              PRINT USING "EBCD address: $'h4",header.EBCD_Addr;
0050          FOR preadr(header.EBCD_Addr)
0051              (* Allow the removal of the header if required *)
0052              INPUT "Remove the N-Byte header from the DSS copy? (y/n)"
0053              ,Reply
0054              IF Reply="Y" OR Reply="y" THEN
0055                  CREATE #DDn,255Header-"_noheader",WRITE
0056                  File:=File+5.
0057                  SEEK #DSSchan,.5
0058                  WHILE File<=255. DO
0059                      SET #DSSchan,Buffer
0060                      PUT #DDn,Buffer
0061                      File:=File+255.
0062                  ENDWHILE
0063                  WHILE File<=5. DO
0064                      GET #DSSchan,OneByte
0065                      PUT #DDn,OneByte
0066                      File:=File+1.
0067                  ENDWHILE
0068                  CLOSE #DDn

```

```

0004         ENDIF %C= End of header removal *)
0005     ENDP
0006     CLOSE #SOURCE
0007     PRINT
0008     END "Transfer complete"
PROCEDURE readsect
0009 (* Read N sectors from "in", starting at present position *)
0010 (* Output goes to "out" in "buffer"-size chunks *)
0011 PARAM in,out:N:INTEGER; N:INTEGER; files:REAL; buffer(256):BYTE
0012 PARAM single:BOOLEAN
0013 DIM i:INTEGER
0014 DIM dummy:STRING[1]
0015 i:=N
0016 WHILE i>=0 DO
0017     IF single THEN
0018         PRINT % PRINT "Insert DRAGON source disc, then hit a key"
0019         GET #0,dummy
0020     ENDIF
0021     GET #in,buffer
0022     IF single THEN
0023         PRINT % PRINT "Insert DSS destination disc, then hit a key"
0024         GET #0,dummy
0025     ENDIF
0026     PUT #out,buffer
0027     i=i-1
0028 ENWHILE
0029 files:=files+256.*N
PROCEDURE readpartsect
0030 (* Read N bytes from "in", starting at present position *)
0031 (* Output goes to "out" in N_byte chunks *)
0032 PARAM in,out,N:INTEGER; files:REAL; buffer(256):BYTE
0033 PARAM single:BOOLEAN
0034 DIM nbyte:BYTE; Nbyte,i:INTEGER
0035 DIM dummy:STRING[1]
0036 IF single THEN
0037     PRINT % PRINT "Insert DRAGON source disc, then hit a key"
0038     GET #0,dummy
0039 ENDIF
0040 GET #in,buffer
0041 Nbyte:=N
0042 IF N=0 THEN
0043     Nbyte=256
0044 ENDIF
0045 IF single THEN
0046     PRINT % PRINT "Insert DSS destination disc, then hit a key"
0047     GET #0,dummy
0048 ENDIF
0049 FOR i=1 TO Nbyte
0050     nbyte:=buffer[i]
0051     PUT #out,nbyte
0052 NEXT i
0053 files:=files+Nbyte
PROCEDURE Preadr
0054 PARAM address:INTEGER
0055 temp:=address
0056 WHILE temp<=0 DO
0057     temp:=temp+65536
0058 ENWHILE
0059 PRINT " ("; temp; ")"

```

Dragonsoft

New software for review should be sent to Dragon User,
12-13 Little Newport Street, London WC2H 9PP.

Fast characters

Program: Hi-text
Supplier: Information Software,
8 Winsley Road, Liverpool
L6 8DW
Price: £5.00 (casualty)

I have recently purchased a copy of Hi-text from Microvision. I was amazed by the speed at which this program prints characters on a TV screen.

Hi-text offers 64 x 25 display. Unlike many others Hi-text displays which define characters on a 4 x 8 matrix giving flexible 64 column text, Hi-text defines characters on a 3 x 8 matrix (with the last row of matrix left for descenders), giving a crystal-clear display even on an ordinary black and white TV with contrast and brightness properly tuned up. In fact, a 1mm gap is visible between two 'H's on a 14" TV.

Hi-text offers a screenprinting speed noticeable to that of the Amstrad CPC-Microware when used from within Dragon Basic.

Other features of Hi-text include user-definable character

facilities. A formula has been included for this purpose. There are four control codes which may be sent to the driver using the PRINT CHR\$(2) command. The effects generated are as follows:

Value of 2	Effects
10	Clear Screen
8	Make/erase
7	Swap
1	Inverted/True

Being in 64 columns, the PRINT@ command has to be replaced by PRINT X@, where X has a value ranging from 0-63 and Y, 0-24. The CLS command when used with any number will clear the screen (except with values greater than 8) which produce the normal copyright messages. Full Auto-repeat of keys has been implemented. You may have black text against white or green background. Machine code users may use any of the Basic ROM routines to output text such as \$B\$A4 or \$B\$B5. The program takes up only about

2-3K when installed, which is a relatively low figure.

The only drawback of Hi-text is that you cannot use it in the 64K mode if you have a Dragon 64.

Many serious business software packages such as word-processors, report generators and bank accounts can be written with the aid of this program.

The first advantage is that you now have a screen which displays information two and a half times more than before. Secondly, 64 characters per line offset completely matches the width of an A4 size paper. This means that many programs can now be designed on a What You See is What You Get basis. In fact, the popular Roward and The Miller for the Spectrum are two such programs which take advantage of 64 characters.

Those of you have a Dragon 64 may incorporate these Auto-repeat PC statements (given in Dragon's Supplementary Manual) into Hi-text to do this you must first install Hi-text. The result is that you have a fast response keyboard in addition to a fast screen. Words will not

be left out when you type fast. The 'smoothness' of the Auto-repeat action is however affected, because of the already built-in Auto-repeat routine in Hi-text. Microvision should consider adding a fast keyboard routine to Hi-text, such as their Name in Microdrive's Name.

Atwood of \$500 Hi-text gives your Dragon a fast, business-useable kind of screen; this represents good value for money.

Finally, I would appreciate if Microvision would consider producing a 64 characters display wordprocessor which offers facilities and standards similar to those found in Roward and The Miller (such as complete in-screen formatting (this includes justification of text, word wrapping, automatic page breaks, centring line and sideways), scrolling 128 column text) and otherwise features, faster and a second character set for the printer.

Dr. Siew Yee Hong



FREE GAME

when you subscribe

to POPULAR Computing WEEKLY

Your choice of any one of the games listed below is free when you subscribe to Popular Computing Weekly

Please enter my annual subscription to Popular Computing Weekly at £25.00 U.S. \$25.00 Overseas, and send my free game as soon as below:

Commodore 64

- ☐ William Wagon
☐ Dragon Chase

Amstrad

- ☐ Jet 1 Photo
☐ Dragons

Spectrum

- ☐ William Wagon
☐ Dragon Chase

BBC Micro B

- ☐ Jet Man Sam
☐ Microvision

☐ I enclose a cheque made payable to Serative Publications Ltd

☐ Please charge my Visa Access card No

Expiry Date

Name

Address

Computers owned

Age

Return this form together with your payment to: PCW Sales, 12-13 Little Newport St., London WC2H 9PP (allow 28 days for delivery).

Write: ADVENTURE

Peter Gerrard begins a news series on creating your own games

OUTSIDE my window it is another dismal day and the streets of North London are packed with objects of a dubious nature and certainly not good. But do I care? Not a jot. For I have just returned from a voyage of discovery on the inside of a strange satellite deep-in space, trying to deliver a mysterious box to a mad professor (So that's what became of the Boulder Games — Ed.) and dodging all kinds of unusual and unnatural obstructions, aided only by my trusty IMP. And your ordinary companion, is Eddie the IMP. He is a small, translucent being, sitting comfortably under the skin but apt to panic in moments of crisis.

All this is part of an adventure called *Isle Two* for some technical reason that I will not go into here. Written twice by yours truly, and over the course of the next few months I'm going to be sharing some of my adventure writing secrets with you through the pages of this magazine (say this month — Ed.) journal. The object of all this is of course to get you writing your own adventures. I know from my brief acquaintance with the *Adventure* Tiki column that a number of Dragon readers already write their own games, but a much larger number do not and wish to know where to start. Read on.

Why Bother?

Utilities exist to aid you in the writing of adventures, so why not use those? Well, for one thing you will learn absolutely nothing about programming, and for another you will only be producing something that relies almost totally on the skill of someone else. Yours will be the imagination, but the author of the utility will be the one that deserves the most credit. Surely it would be better if you'd written the whole thing yourself? Then you could sell it as your own creation from start to finish, without bothering about such murky things as copyright. You would also have, let's be honest, the immense sense of pride that comes from completing your own game, written in your own way.

I will be providing some programming utilities, you will be doing the actual work, altering those utilities, changing them to fit into your own games and your own particular scenarios. Or, to put it another way, I will provide the cream and numbered pages of a book, you will fill in the pages with whatever you want.

I can still remember the amazing sense of pleasure that I got on completing my first Dragon adventure and receiving a query from someone who couldn't solve a particular problem. It was only a small game, with a limited number of puzzles and short text descriptions, but it still got remembered all these years later. So people really like the best thing I've ever written. You definitely can't write it all. But between you, me and the Dragon, I think we're capable of making more than a few people sit up and take

notice I see from some of the letters received in *Adventure* that that cute a few husband and wife, boyfriend and girlfriend teams of adventure players exist. How about writing an adventure for your companion? Greater love hath no person, etc. etc.

Getting started

In order to think about writing an adventure game, you need to have no programming skills whatsoever! This is not meant to deter people who produce the ideas rather than the programs, but it is a statement of fact. The greatest programmer in the world will not produce a decent adventure without a very strong storyline behind it, and I someone else provides the storyline, all well and good.

And if we want to start thinking about a possible scenario for our game, let us first of all begin by looking at what we won't be doing.

There will be no (or very few) graphics in the adventure-writing technique I am going to concentrate on. For one thing is that the graphic adventures, and although I can admire other people's computers running glorious pictures from games like *The Pawn*,

defeating the point of an adventure, to my way of thinking. There should be problems to solve, not vast underground corridors to explore that go on and on seemingly without end.

We will not have a parser (the part of the program that provides the vital link between player and computer, and which interprets what the player has typed in) capable of understanding the likes of "Turn to page 124 of the red book and read the third paragraph before putting it back under the table". For gosh, if you had spent the last week in context, it would all be a waste of time anyway, so the most advanced thing that we'll be looking at will be something like "Look under the clear window", "stare over the third door", "soak the candle with the gloves", etc. (though, in other words, to produce some fairly detailed problems without resorting to the obviously bigger).

Right, that's what we can't do. Beyond that the world is your oyster for the time being, and we're aiming for a lot of text, a reasonably sophisticated parser, and a rather large vocabulary. Time to sit down in front of some paper.

Mapping out

Before touching the computer and before next month's issue of *Dragon* User appears and we start doing some programming you'll need to have a few things sorted out. Where is the adventure going to take place? Outer space? Inside Earth? The American Wild West? Inside? Up? You, but do get some idea of where the action will be happening and the type of game world in which your player will be operating.

Having decided where we're going to be, what will be the object of the adventure? Survival? Collecting treasures? Completing a dangerous mission? Getting out of bed and having breakfast? Again, it's up to you, but my word that that's all well. Only roughly, it doesn't matter yet.

Next, draw a map on a sheet of paper. Start with, say, fifty locations, because you can always add more later. Just draw the numbers one to fifty on a sheet of paper, and that will become your adventure world. As a brief example, like this:

123456

678910

71



It doesn't make me want to rush to the keyboard and produce something similar then if I could, which I doubt. I much prefer text only games, on the premise that the most vivid imagination, the most superb illustrations, the best of artistic designers, will not compete with the images conjured up inside my own head by a good, detailed, atmospheric description of the game world.

We will not be able to have adventures with three million locations, and a practical limit will be around one hundred. That will suffice to conceal a realistic arena in which to play the game, and for much wandering about without doing anything seems to

and so on. Assuming north to be at the top of the page, room one heads east to room two, which heads east to three and south to six. Room six heads south to two, south to eleven, and east to seven. Any number that links up with another one means that the two rooms will be connected. Now you'll want to set up some problems, doesn't matter

where they are yet, as long as we know where we're going. Try and stick to an average of one problem to every three or four locations, giving the player the chance to explore a bit before getting stuck. In the example above you might circle the numbers 20 and 10 anywhere before solving them. 4, 6, and 8 are useful-looking locations that can progress to other ones.

Say we're going to be inside a satellite in outer space, and locations 2, 3, 4, 6, 7 and 8 obviously appear to form a complete area. They could be, for instance, the control area in which the player starts his mission. Location 10 looks like a corridor going off somewhere, while location 11 could be heading off into the heart of the

satellite. Start labeling areas on your complete map, and pretty soon you'll have about nine or ten distinct areas for your life location adventure (it will grow, believe me). Knowing what's going to be there helps you to start thinking about problems. Location 2 could be the entrance to the control room, a locked door which you would obviously need to get past in order to start the game. A problem straight away, how to open the door. It doesn't take long for the whole game to build up from your simple sketch on paper.

It is also worth drawing a tidier map, as I know to my cost. I once designed a massive adventure game using discs that occupied hundreds of locations, and had

well over sixty involved problems to solve. Looked great on my neat, badly-drawn map. It was also unsolvable, because I'd got two locations numbered the wrong way around, and the player was doomed to go round in circles for evermore. Putting that problem to rights took up an awful lot of time that I needn't have had to waste if I'd got a tidier, legible map to look at.



Conclusion

The Dragon is lurking away in the corner, waiting to get started, and next month we will be powering up and starting to produce the first part of our adventure game. Until then, get that map drawn, think about all the problems that the player might have to come across in your imaginary world, and label them. Clearly! Make sure that the game areas form a coherent whole, and that there are no obvious flaws in it, no problems that are just so ludicrous that nobody will ever solve them. Get the game areas started out, so that you know roughly what every location corresponds to.

Finally, start thinking about the sort of words that the player might have to type in in order to go through the completed game. A vocabulary list in other words, because that is what we will be looking at next time. Bye for now.



National Dragon Users' Group

HAVING written a favourable lead on Dragon User in Dragon Update it's now my turn to do the same thing for the NDUG in DJ. Many people will have noticed that advertising for this group is very low key. This is deliberate policy, as the number of user groups which took money and then vanished without delivering any goods has meant that people are now very wary of such groups. So the National Dragon User Group is run more on a personal recommendation basis, and our high membership which is still growing must speak well for the group.

NDUG is the largest Dragon user group in existence, with members in the UK and abroad. It has been running for well over two years now and is likely to be with us for as long as there are Dragon owners who feel the need for it. While it is not a necessity for a Dragon owner to join, I must urge them to give it every consideration. A strong core of users can only help us to get more out of our machines.

Membership is currently £750 for a year, virtually unchanged since day one. For this you get a monthly newsletter, Update, which tries to give information from all spheres of interest in the group, so there is more technical and news and views content than games information. More important than the newsletter, though, are the group's other facilities. Tapping into the as-

Introduction by Philip Reed

perities of members, the group can aim to answer questions on practically any subject; help can be only a letter or a photocall away. If no-one has actually solved your problem already, then some-one will be soon enough to investigate it. How to buy a disc drive? Screen dumps for a specific printer? Programming difficulties? How to upgrade to 64K? etc.



There is also a full page of classified ads, which are free to users, through which some good buys can be had. The group has issued some programs of its own at a

very reasonable cost: a Forth operating system, a disc editor, and graphics utility, and circuit diagrams for 32s and 64s, as well as the DOS, are available, as well as a detailed manual on how to do your own 32 to 64 upgrade. An at-cost repair service is also available for computers and peripherals, an excellent way of beating the cowboy operators often heard about (how cheap should have mentioned that before - did I). Just recently the group has acquired pages on Minicom (page 70070000), a graphics library has been started, and a direct helpline for various problems has been started.

The most important point about NDUG is that it is a group not a publication, and anyone joining to receive a monthly newsletter may well be disappointed. You will get out what you put in. Everyone no matter what level they are at, or what their interests are, has something to contribute to the group and if they do this they will find that by taking part and helping others they will benefit themselves. I have been a member for 18 months now and wish I had joined earlier. The group has helped me to spend my money wisely on improving my system, has given me a outlet for my feelings on various topics, and helped me to expand my knowledge of the Dragon. I would highly recommend it to anyone who owns a Dragon.

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

Dragon Answers

Ports and pins

I HAVE just bought a Dragon 54 without a user manual, and I don't know how to use the serial port. What are the pins and how do I wire the port from programs?

Paul Smith
Archwood Road
Farnham
Canter

THERE seems to be quite a healthy trade in second hand Dragon kits at the moment. A number of readers have asked where they can get the "Dragon 54 Supplement", as this contains information on using the serial port. Unfortunately, as far as I know it is no longer available from any official source, so I will try to answer all these queries in one go. Here is all the useful information from the elusive booklet:

The basic use of the serial ports can be done using the following:
PORT ADDRESS (PORT ADDRESS AND DATA) 00 1

where "1" holds the code for the required baud rate, one of the following:

1=50, 2=75, 3=150, 4=300, 5=600, 6=1200, 7=2400, 8=4800, 9=9600, 10=19200, 11=38400, 12=76800, 13=153600, 14=307200

The format of serial data is 1 start, 8 data and 2 stop bits with no parity. To select between printer hardware:
Serial port — PORT 0001
Parallel port — PORT 0002
Locations 0001/0002 hold a 16-bit code of hexadec (in 16 millisecond units) for the serial printer port.

To send a character to the serial port directly, use:

10 IF (PORT ADDRESS AND 16) = 0 THEN 10

20 PORT ADDRESS, ASCII(C) : REM C IS CHARACTER TO SEND

To read a character from the serial port use:

10 IF (PORT ADDRESS AND 8) = 0 THEN 10

20 C=CHR\$(PORT ADDRESS)
The pin assignment of the port is as follows:

pin 0=TX Data, pin 1=RX, pin 2=RTS, pin 3=CTS, pin 4=DSR, pin 5=TX, pin 6=CTS, pin 7=DSR, pin 8=TX, pin 9=CTS, pin 10=DSR, pin 11=TX, pin 12=CTS, pin 13=DSR, pin 14=TX, pin 15=CTS, pin 16=DSR, pin 17=TX, pin 18=CTS, pin 19=DSR, pin 20=TX, pin 21=CTS, pin 22=DSR, pin 23=TX, pin 24=CTS, pin 25=DSR, pin 26=TX, pin 27=CTS, pin 28=DSR, pin 29=TX, pin 30=CTS, pin 31=DSR, pin 32=TX, pin 33=CTS, pin 34=DSR, pin 35=TX, pin 36=CTS, pin 37=DSR, pin 38=TX, pin 39=CTS, pin 40=DSR, pin 41=TX, pin 42=CTS, pin 43=DSR, pin 44=TX, pin 45=CTS, pin 46=DSR, pin 47=TX, pin 48=CTS, pin 49=DSR, pin 50=TX, pin 51=CTS, pin 52=DSR, pin 53=TX, pin 54=CTS, pin 55=DSR, pin 56=TX, pin 57=CTS, pin 58=DSR, pin 59=TX, pin 60=CTS, pin 61=DSR, pin 62=TX, pin 63=CTS, pin 64=DSR, pin 65=TX, pin 66=CTS, pin 67=DSR, pin 68=TX, pin 69=CTS, pin 70=DSR, pin 71=TX, pin 72=CTS, pin 73=DSR, pin 74=TX, pin 75=CTS, pin 76=DSR, pin 77=TX, pin 78=CTS, pin 79=DSR, pin 80=TX, pin 81=CTS, pin 82=DSR, pin 83=TX, pin 84=CTS, pin 85=DSR, pin 86=TX, pin 87=CTS, pin 88=DSR, pin 89=TX, pin 90=CTS, pin 91=DSR, pin 92=TX, pin 93=CTS, pin 94=DSR, pin 95=TX, pin 96=CTS, pin 97=DSR, pin 98=TX, pin 99=CTS, pin 100=DSR, pin 101=TX, pin 102=CTS, pin 103=DSR, pin 104=TX, pin 105=CTS, pin 106=DSR, pin 107=TX, pin 108=CTS, pin 109=DSR, pin 110=TX, pin 111=CTS, pin 112=DSR, pin 113=TX, pin 114=CTS, pin 115=DSR, pin 116=TX, pin 117=CTS, pin 118=DSR, pin 119=TX, pin 120=CTS, pin 121=DSR, pin 122=TX, pin 123=CTS, pin 124=DSR, pin 125=TX, pin 126=CTS, pin 127=DSR, pin 128=TX, pin 129=CTS, pin 130=DSR, pin 131=TX, pin 132=CTS, pin 133=DSR, pin 134=TX, pin 135=CTS, pin 136=DSR, pin 137=TX, pin 138=CTS, pin 139=DSR, pin 140=TX, pin 141=CTS, pin 142=DSR, pin 143=TX, pin 144=CTS, pin 145=DSR, pin 146=TX, pin 147=CTS, pin 148=DSR, pin 149=TX, pin 150=CTS, pin 151=DSR, pin 152=TX, pin 153=CTS, pin 154=DSR, pin 155=TX, pin 156=CTS, pin 157=DSR, pin 158=TX, pin 159=CTS, pin 160=DSR, pin 161=TX, pin 162=CTS, pin 163=DSR, pin 164=TX, pin 165=CTS, pin 166=DSR, pin 167=TX, pin 168=CTS, pin 169=DSR, pin 170=TX, pin 171=CTS, pin 172=DSR, pin 173=TX, pin 174=CTS, pin 175=DSR, pin 176=TX, pin 177=CTS, pin 178=DSR, pin 179=TX, pin 180=CTS, pin 181=DSR, pin 182=TX, pin 183=CTS, pin 184=DSR, pin 185=TX, pin 186=CTS, pin 187=DSR, pin 188=TX, pin 189=CTS, pin 190=DSR, pin 191=TX, pin 192=CTS, pin 193=DSR, pin 194=TX, pin 195=CTS, pin 196=DSR, pin 197=TX, pin 198=CTS, pin 199=DSR, pin 200=TX, pin 201=CTS, pin 202=DSR, pin 203=TX, pin 204=CTS, pin 205=DSR, pin 206=TX, pin 207=CTS, pin 208=DSR, pin 209=TX, pin 210=CTS, pin 211=DSR, pin 212=TX, pin 213=CTS, pin 214=DSR, pin 215=TX, pin 216=CTS, pin 217=DSR, pin 218=TX, pin 219=CTS, pin 220=DSR, pin 221=TX, pin 222=CTS, pin 223=DSR, pin 224=TX, pin 225=CTS, pin 226=DSR, pin 227=TX, pin 228=CTS, pin 229=DSR, pin 230=TX, pin 231=CTS, pin 232=DSR, pin 233=TX, pin 234=CTS, pin 235=DSR, pin 236=TX, pin 237=CTS, pin 238=DSR, pin 239=TX, pin 240=CTS, pin 241=DSR, pin 242=TX, pin 243=CTS, pin 244=DSR, pin 245=TX, pin 246=CTS, pin 247=DSR, pin 248=TX, pin 249=CTS, pin 250=DSR, pin 251=TX, pin 252=CTS, pin 253=DSR, pin 254=TX, pin 255=CTS, pin 256=DSR, pin 257=TX, pin 258=CTS, pin 259=DSR, pin 260=TX, pin 261=CTS, pin 262=DSR, pin 263=TX, pin 264=CTS, pin 265=DSR, pin 266=TX, pin 267=CTS, pin 268=DSR, pin 269=TX, pin 270=CTS, pin 271=DSR, pin 272=TX, pin 273=CTS, pin 274=DSR, pin 275=TX, pin 276=CTS, pin 277=DSR, pin 278=TX, pin 279=CTS, pin 280=DSR, pin 281=TX, pin 282=CTS, pin 283=DSR, pin 284=TX, pin 285=CTS, pin 286=DSR, pin 287=TX, pin 288=CTS, pin 289=DSR, pin 290=TX, pin 291=CTS, pin 292=DSR, pin 293=TX, pin 294=CTS, pin 295=DSR, pin 296=TX, pin 297=CTS, pin 298=DSR, pin 299=TX, pin 300=CTS, pin 301=DSR, pin 302=TX, pin 303=CTS, pin 304=DSR, pin 305=TX, pin 306=CTS, pin 307=DSR, pin 308=TX, pin 309=CTS, pin 310=DSR, pin 311=TX, pin 312=CTS, pin 313=DSR, pin 314=TX, pin 315=CTS, pin 316=DSR, pin 317=TX, pin 318=CTS, pin 319=DSR, pin 320=TX, pin 321=CTS, pin 322=DSR, pin 323=TX, pin 324=CTS, pin 325=DSR, pin 326=TX, pin 327=CTS, pin 328=DSR, pin 329=TX, pin 330=CTS, pin 331=DSR, pin 332=TX, pin 333=CTS, pin 334=DSR, pin 335=TX, pin 336=CTS, pin 337=DSR, pin 338=TX, pin 339=CTS, pin 340=DSR, pin 341=TX, pin 342=CTS, pin 343=DSR, pin 344=TX, pin 345=CTS, pin 346=DSR, pin 347=TX, pin 348=CTS, pin 349=DSR, pin 350=TX, pin 351=CTS, pin 352=DSR, pin 353=TX, pin 354=CTS, pin 355=DSR, pin 356=TX, pin 357=CTS, pin 358=DSR, pin 359=TX, pin 360=CTS, pin 361=DSR, pin 362=TX, pin 363=CTS, pin 364=DSR, pin 365=TX, pin 366=CTS, pin 367=DSR, pin 368=TX, pin 369=CTS, pin 370=DSR, pin 371=TX, pin 372=CTS, pin 373=DSR, pin 374=TX, pin 375=CTS, pin 376=DSR, pin 377=TX, pin 378=CTS, pin 379=DSR, pin 380=TX, pin 381=CTS, pin 382=DSR, pin 383=TX, pin 384=CTS, pin 385=DSR, pin 386=TX, pin 387=CTS, pin 388=DSR, pin 389=TX, pin 390=CTS, pin 391=DSR, pin 392=TX, pin 393=CTS, pin 394=DSR, pin 395=TX, pin 396=CTS, pin 397=DSR, pin 398=TX, pin 399=CTS, pin 400=DSR, pin 401=TX, pin 402=CTS, pin 403=DSR, pin 404=TX, pin 405=CTS, pin 406=DSR, pin 407=TX, pin 408=CTS, pin 409=DSR, pin 410=TX, pin 411=CTS, pin 412=DSR, pin 413=TX, pin 414=CTS, pin 415=DSR, pin 416=TX, pin 417=CTS, pin 418=DSR, pin 419=TX, pin 420=CTS, pin 421=DSR, pin 422=TX, pin 423=CTS, pin 424=DSR, pin 425=TX, pin 426=CTS, pin 427=DSR, pin 428=TX, pin 429=CTS, pin 429=DSR, pin 430=TX, pin 431=CTS, pin 431=DSR, pin 432=TX, pin 433=CTS, pin 433=DSR, pin 434=TX, pin 435=CTS, pin 435=DSR, pin 436=TX, pin 437=CTS, pin 437=DSR, pin 438=TX, pin 439=CTS, pin 439=DSR, pin 440=TX, pin 441=CTS, pin 441=DSR, pin 442=TX, pin 443=CTS, pin 443=DSR, pin 444=TX, pin 445=CTS, pin 445=DSR, pin 446=TX, pin 447=CTS, pin 447=DSR, pin 448=TX, pin 449=CTS, pin 449=DSR, pin 450=TX, pin 451=CTS, pin 451=DSR, pin 452=TX, pin 453=CTS, pin 453=DSR, pin 454=TX, pin 455=CTS, pin 455=DSR, pin 456=TX, pin 457=CTS, pin 457=DSR, pin 458=TX, pin 459=CTS, pin 459=DSR, pin 460=TX, pin 461=CTS, pin 461=DSR, pin 462=TX, pin 463=CTS, pin 463=DSR, pin 464=TX, pin 465=CTS, pin 465=DSR, pin 466=TX, pin 467=CTS, pin 467=DSR, pin 468=TX, pin 469=CTS, pin 469=DSR, pin 470=TX, pin 471=CTS, pin 471=DSR, pin 472=TX, pin 473=CTS, pin 473=DSR, pin 474=TX, pin 475=CTS, pin 475=DSR, pin 476=TX, pin 477=CTS, pin 477=DSR, pin 478=TX, pin 479=CTS, pin 479=DSR, pin 480=TX, pin 481=CTS, pin 481=DSR, pin 482=TX, pin 483=CTS, pin 483=DSR, pin 484=TX, pin 485=CTS, pin 485=DSR, pin 486=TX, pin 487=CTS, pin 487=DSR, pin 488=TX, pin 489=CTS, pin 489=DSR, pin 490=TX, pin 491=CTS, pin 491=DSR, pin 492=TX, pin 493=CTS, pin 493=DSR, pin 494=TX, pin 495=CTS, pin 495=DSR, pin 496=TX, pin 497=CTS, pin 497=DSR, pin 498=TX, pin 499=CTS, pin 499=DSR, pin 500=TX, pin 501=CTS, pin 501=DSR, pin 502=TX, pin 503=CTS, pin 503=DSR, pin 504=TX, pin 505=CTS, pin 505=DSR, pin 506=TX, pin 507=CTS, pin 507=DSR, pin 508=TX, pin 509=CTS, pin 509=DSR, pin 510=TX, pin 511=CTS, pin 511=DSR, pin 512=TX, pin 513=CTS, pin 513=DSR, pin 514=TX, pin 515=CTS, pin 515=DSR, pin 516=TX, pin 517=CTS, pin 517=DSR, pin 518=TX, pin 519=CTS, pin 519=DSR, pin 520=TX, pin 521=CTS, pin 521=DSR, pin 522=TX, pin 523=CTS, pin 523=DSR, pin 524=TX, pin 525=CTS, pin 525=DSR, pin 526=TX, pin 527=CTS, pin 527=DSR, pin 528=TX, pin 529=CTS, pin 529=DSR, pin 530=TX, pin 531=CTS, pin 531=DSR, pin 532=TX, pin 533=CTS, pin 533=DSR, pin 534=TX, pin 535=CTS, pin 535=DSR, pin 536=TX, pin 537=CTS, pin 537=DSR, pin 538=TX, pin 539=CTS, pin 539=DSR, pin 540=TX, pin 541=CTS, pin 541=DSR, pin 542=TX, pin 543=CTS, pin 543=DSR, pin 544=TX, pin 545=CTS, pin 545=DSR, pin 546=TX, pin 547=CTS, pin 547=DSR, pin 548=TX, pin 549=CTS, pin 549=DSR, pin 550=TX, pin 551=CTS, pin 551=DSR, pin 552=TX, pin 553=CTS, pin 553=DSR, pin 554=TX, pin 555=CTS, pin 555=DSR, pin 556=TX, pin 557=CTS, pin 557=DSR, pin 558=TX, pin 559=CTS, pin 559=DSR, pin 560=TX, pin 561=CTS, pin 561=DSR, pin 562=TX, pin 563=CTS, pin 563=DSR, pin 564=TX, pin 565=CTS, pin 565=DSR, pin 566=TX, pin 567=CTS, pin 567=DSR, pin 568=TX, pin 569=CTS, pin 569=DSR, pin 570=TX, pin 571=CTS, pin 571=DSR, pin 572=TX, pin 573=CTS, pin 573=DSR, pin 574=TX, pin 575=CTS, pin 575=DSR, pin 576=TX, pin 577=CTS, pin 577=DSR, pin 578=TX, pin 579=CTS, pin 579=DSR, pin 580=TX, pin 581=CTS, pin 581=DSR, pin 582=TX, pin 583=CTS, pin 583=DSR, pin 584=TX, pin 585=CTS, pin 585=DSR, pin 586=TX, pin 587=CTS, pin 587=DSR, pin 588=TX, pin 589=CTS, pin 589=DSR, pin 590=TX, pin 591=CTS, pin 591=DSR, pin 592=TX, pin 593=CTS, pin 593=DSR, pin 594=TX, pin 595=CTS, pin 595=DSR, pin 596=TX, pin 597=CTS, pin 597=DSR, pin 598=TX, pin 599=CTS, pin 599=DSR, pin 600=TX, pin 601=CTS, pin 601=DSR, pin 602=TX, pin 603=CTS, pin 603=DSR, pin 604=TX, pin 605=CTS, pin 605=DSR, pin 606=TX, pin 607=CTS, pin 607=DSR, pin 608=TX, pin 609=CTS, pin 609=DSR, pin 610=TX, pin 611=CTS, pin 611=DSR, pin 612=TX, pin 613=CTS, pin 613=DSR, pin 614=TX, pin 615=CTS, pin 615=DSR, pin 616=TX, pin 617=CTS, pin 617=DSR, pin 618=TX, pin 619=CTS, pin 619=DSR, pin 620=TX, pin 621=CTS, pin 621=DSR, pin 622=TX, pin 623=CTS, pin 623=DSR, pin 624=TX, pin 625=CTS, pin 625=DSR, pin 626=TX, pin 627=CTS, pin 627=DSR, pin 628=TX, pin 629=CTS, pin 629=DSR, pin 630=TX, pin 631=CTS, pin 631=DSR, pin 632=TX, pin 633=CTS, pin 633=DSR, pin 634=TX, pin 635=CTS, pin 635=DSR, pin 636=TX, pin 637=CTS, pin 637=DSR, pin 638=TX, pin 639=CTS, pin 639=DSR, pin 640=TX, pin 641=CTS, pin 641=DSR, pin 642=TX, pin 643=CTS, pin 643=DSR, pin 644=TX, pin 645=CTS, pin 645=DSR, pin 646=TX, pin 647=CTS, pin 647=DSR, pin 648=TX, pin 649=CTS, pin 649=DSR, pin 650=TX, pin 651=CTS, pin 651=DSR, pin 652=TX, pin 653=CTS, pin 653=DSR, pin 654=TX, pin 655=CTS, pin 655=DSR, pin 656=TX, pin 657=CTS, pin 657=DSR, pin 658=TX, pin 659=CTS, pin 659=DSR, pin 660=TX, pin 661=CTS, pin 661=DSR, pin 662=TX, pin 663=CTS, pin 663=DSR, pin 664=TX, pin 665=CTS, pin 665=DSR, pin 666=TX, pin 667=CTS, pin 667=DSR, pin 668=TX, pin 669=CTS, pin 669=DSR, pin 670=TX, pin 671=CTS, pin 671=DSR, pin 672=TX, pin 673=CTS, pin 673=DSR, pin 674=TX, pin 675=CTS, pin 675=DSR, pin 676=TX, pin 677=CTS, pin 677=DSR, pin 678=TX, pin 679=CTS, pin 679=DSR, pin 680=TX, pin 681=CTS, pin 681=DSR, pin 682=TX, pin 683=CTS, pin 683=DSR, pin 684=TX, pin 685=CTS, pin 685=DSR, pin 686=TX, pin 687=CTS, pin 687=DSR, pin 688=TX, pin 689=CTS, pin 689=DSR, pin 690=TX, pin 691=CTS, pin 691=DSR, pin 692=TX, pin 693=CTS, pin 693=DSR, pin 694=TX, pin 695=CTS, pin 695=DSR, pin 696=TX, pin 697=CTS, pin 697=DSR, pin 698=TX, pin 699=CTS, pin 699=DSR, pin 700=TX, pin 701=CTS, pin 701=DSR, pin 702=TX, pin 703=CTS, pin 703=DSR, pin 704=TX, pin 705=CTS, pin 705=DSR, pin 706=TX, pin 707=CTS, pin 707=DSR, pin 708=TX, pin 709=CTS, pin 709=DSR, pin 710=TX, pin 711=CTS, pin 711=DSR, pin 712=TX, pin 713=CTS, pin 713=DSR, pin 714=TX, pin 715=CTS, pin 715=DSR, pin 716=TX, pin 717=CTS, pin 717=DSR, pin 718=TX, pin 719=CTS, pin 719=DSR, pin 720=TX, pin 721=CTS, pin 721=DSR, pin 722=TX, pin 723=CTS, pin 723=DSR, pin 724=TX, pin 725=CTS, pin 725=DSR, pin 726=TX, pin 727=CTS, pin 727=DSR, pin 728=TX, pin 729=CTS, pin 729=DSR, pin 730=TX, pin 731=CTS, pin 731=DSR, pin 732=TX, pin 733=CTS, pin 733=DSR, pin 734=TX, pin 735=CTS, pin 735=DSR, pin 736=TX, pin 737=CTS, pin 737=DSR, pin 738=TX, pin 739=CTS, pin 739=DSR, pin 740=TX, pin 741=CTS, pin 741=DSR, pin 742=TX, pin 743=CTS, pin 743=DSR, pin 744=TX, pin 745=CTS, pin 745=DSR, pin 746=TX, pin 747=CTS, pin 747=DSR, pin 748=TX, pin 749=CTS, pin 749=DSR, pin 750=TX, pin 751=CTS, pin 751=DSR, pin 752=TX, pin 753=CTS, pin 753=DSR, pin 754=TX, pin 755=CTS, pin 755=DSR, pin 756=TX, pin 757=CTS, pin 757=DSR, pin 758=TX, pin 759=CTS, pin 759=DSR, pin 760=TX, pin 761=CTS, pin 761=DSR, pin 762=TX, pin 763=CTS, pin 763=DSR, pin 764=TX, pin 765=CTS, pin 765=DSR, pin 766=TX, pin 767=CTS, pin 767=DSR, pin 768=TX, pin 769=CTS, pin 769=DSR, pin 770=TX, pin 771=CTS, pin 771=DSR, pin 772=TX, pin 773=CTS, pin 773=DSR, pin 774=TX, pin 775=CTS, pin 775=DSR, pin 776=TX, pin 777=CTS, pin 777=DSR, pin 778=TX, pin 779=CTS, pin 779=DSR, pin 780=TX, pin 781=CTS, pin 781=DSR, pin 782=TX, pin 783=CTS, pin 783=DSR, pin 784=TX, pin 785=CTS, pin 785=DSR, pin 786=TX, pin 787=CTS, pin 787=DSR, pin 788=TX, pin 789=CTS, pin 789=DSR, pin 790=TX, pin 791=CTS, pin 791=DSR, pin 792=TX, pin 793=CTS, pin 793=DSR, pin 794=TX, pin 795=CTS, pin 795=DSR, pin 796=TX, pin 797=CTS, pin 797=DSR, pin 798=TX, pin 799=CTS, pin 799=DSR, pin 800=TX, pin 801=CTS, pin 801=DSR, pin 802=TX, pin 803=CTS, pin 803=DSR, pin 804=TX, pin 805=CTS, pin 805=DSR, pin 806=TX, pin 807=CTS, pin 807=DSR, pin 808=TX, pin 809=CTS, pin 809=DSR, pin 810=TX, pin 811=CTS, pin 811=DSR, pin 812=TX, pin 813=CTS, pin 813=DSR, pin 814=TX, pin 815=CTS, pin 815=DSR, pin 816=TX, pin 817=CTS, pin 817=DSR, pin 818=TX, pin 819=CTS, pin 819=DSR, pin 820=TX, pin 821=CTS, pin 821=DSR, pin 822=TX, pin 823=CTS, pin 823=DSR, pin 824=TX, pin 825=CTS, pin 825=DSR, pin 826=TX, pin 827=CTS, pin 827=DSR, pin 828=TX, pin 829=CTS, pin 829=DSR, pin 830=TX, pin 831=CTS, pin 831=DSR, pin 832=TX, pin 833=CTS, pin 833=DSR, pin 834=TX, pin 835=CTS, pin 835=DSR, pin 836=TX, pin 837=CTS, pin 837=DSR, pin 838=TX, pin 839=CTS, pin 839=DSR, pin 840=TX, pin 841=CTS, pin 841=DSR, pin 842=TX, pin 843=CTS, pin 843=DSR, pin 844=TX, pin 845=CTS, pin 845=DSR, pin 846=TX, pin 847=CTS, pin 847=DSR, pin 848=TX, pin 849=CTS, pin 849=DSR, pin 850=TX, pin 851=CTS, pin 851=DSR, pin 852=TX, pin 853=CTS, pin 853=DSR, pin 854=TX, pin 855=CTS, pin 855=DSR, pin 856=TX, pin 857=CTS, pin 857=DSR, pin 858=TX, pin 859=CTS, pin 859=DSR, pin 860=TX, pin 861=CTS, pin 861=DSR, pin 862=TX, pin 863=CTS, pin 863=DSR, pin 864=TX, pin 865=CTS, pin 865=DSR, pin 866=TX, pin 867=CTS, pin 867=DSR, pin 868=TX, pin 869=CTS, pin 869=DSR, pin 870=TX, pin 871=CTS, pin 871=DSR, pin 872=TX, pin 873=CTS, pin 873=DSR, pin 874=TX, pin 875=CTS, pin 875=DSR, pin 876=TX, pin 877=CTS, pin 877=DSR, pin 878=TX, pin 879=CTS, pin 879=DSR, pin 880=TX, pin 881=CTS, pin 881=DSR, pin 882=TX, pin 883=CTS, pin 883=DSR, pin 884=TX, pin 885=CTS, pin 885=DSR, pin 886=TX, pin 887=CTS, pin 887=DSR, pin 888=TX, pin 889=CTS, pin 889=DSR, pin 890=TX, pin 891=CTS, pin 891=DSR, pin 892=TX, pin 893=CTS, pin 893=DSR, pin 894=TX, pin 895=CTS, pin 895=DSR, pin 896=TX, pin 897=CTS, pin 897=DSR, pin 898=TX, pin 899=CTS, pin 899=DSR, pin 900=TX, pin 901=CTS, pin 901=DSR, pin 902=TX, pin 903=CTS, pin 903=DSR, pin 904=TX, pin 905=CTS, pin 905=DSR, pin 906=TX, pin 907=CTS, pin 907=DSR, pin 908=TX, pin 909=CTS, pin 909=DSR, pin 910=TX, pin 911=CTS, pin 911=DSR, pin 912=TX, pin 913=CTS, pin 913=DSR, pin 914=TX, pin 915=CTS, pin 915=DSR, pin 916=TX, pin 917=CTS, pin 917=DSR, pin 918=TX, pin 919=CTS, pin 919=DSR, pin 920=TX, pin 921=CTS, pin 921=DSR, pin 922=TX, pin 923=CTS, pin 923=DSR, pin 924=TX, pin 925=CTS, pin 925=DSR, pin 926=TX, pin 927=CTS, pin 927=DSR, pin 928=TX, pin 929=CTS, pin 929=DSR, pin 930=TX, pin 931=CTS, pin 931=DSR, pin 932=TX, pin 933=CTS, pin 933=DSR, pin 934=TX, pin 935=CTS, pin 935=DSR, pin 936=TX, pin 937=CTS, pin 937=DSR, pin 938=TX, pin 939=CTS, pin 939=DSR, pin 940=TX, pin 941=CTS, pin 941=DSR, pin 942=TX, pin 943=CTS, pin 943=DSR, pin 944=TX, pin 945=CTS, pin 945=DSR, pin 946=TX, pin 947=CTS, pin 947=DSR, pin 948=TX, pin 949=CTS, pin 949=DSR, pin 950=TX, pin 951=CTS, pin 951=DSR, pin 952=TX, pin 953=CTS, pin 953=DSR, pin 954=TX, pin 955=CTS, pin 955=DSR, pin 956=TX, pin 957=CTS, pin 957=DSR, pin 958=TX, pin 959=CTS, pin 959=DSR, pin 960=TX, pin 961=CTS, pin 961=DSR, pin 962=TX, pin 963=CTS, pin 963=DSR, pin 964=TX, pin 965=CTS, pin 965=DSR, pin 966=TX, pin 967=CTS, pin 967=DSR, pin 968=TX, pin 969=CTS, pin 969=DSR, pin 970=TX, pin 971=CTS, pin 971=DSR, pin 972=TX, pin 973=CTS, pin 973=DSR, pin 974=TX, pin 975=CTS, pin 975=DSR, pin 976=TX, pin 977=CTS, pin 977=DSR, pin 978=TX, pin 979=CTS, pin 979=DSR, pin 980=TX, pin 981=CTS, pin 981=DSR, pin 982=TX, pin 983=CTS, pin 983=DSR, pin 984=TX, pin 985=CTS, pin 985=DSR, pin 986=TX, pin 987=CTS, pin 987=DSR, pin 988=TX, pin 989=CTS, pin 989=DSR, pin 990=TX, pin 991=CTS, pin 991=DSR, pin 992=TX, pin 993=CTS, pin 993=DSR, pin 994=TX, pin 995=CTS, pin 995=DSR, pin 996=TX, pin 997=CTS, pin 997=DSR, pin 998=TX, pin 999=CTS, pin 999=DSR, pin 1000=TX, pin 1001=CTS, pin 1001=DSR, pin 1002=TX, pin 1003=CTS, pin 1003=DSR, pin 1004=TX, pin 1005=CTS, pin 1005=DSR, pin 1006=TX, pin 1007=CTS, pin 1007=DSR, pin 1008=TX, pin 1009=CTS, pin 1009=DSR, pin 1010=TX, pin 1011=CTS, pin 1011=DSR, pin 1012=TX, pin 1013=CTS, pin 1013=DSR, pin 1014=TX, pin 1015=CTS, pin 1015=DSR, pin 1016=TX, pin 1017=CTS, pin 1017=DSR, pin 1018=TX, pin 1019=CTS, pin 1019=DSR, pin 1020=TX, pin 1021=CTS, pin 1021=DSR, pin 1022=TX, pin 1023=CTS, pin 1023=DSR, pin 1024=TX, pin 1025=CTS, pin 1025=DSR, pin 1026=TX, pin 1027=CTS, pin 1027=DSR, pin 1028=TX, pin 1029=CTS, pin 1029=DSR, pin 1030=TX, pin 1031=CTS, pin 1031=DSR, pin 1032=TX, pin 1033=CTS, pin 1033=DSR, pin 1034=TX, pin 1035=CTS, pin 1035=DSR, pin 1036=TX, pin 1037=CTS, pin 1037=DSR, pin 1038=TX, pin 1039=CTS, pin 1039=DSR, pin 1040=TX, pin 1041=CTS, pin 1041=DSR, pin 1042=TX, pin 1043=CTS, pin 1043=DSR, pin 1044=TX, pin 1045=CTS, pin 1045=DSR, pin 1046=TX, pin

MIKE GERARD'S ADVENTURE TRAIL

What happens when you take over your brother's adventure column? The first query I looked at was from one Eric Stockwell, asking Mike if he could possibly help to track down a tape written by his 'namesake' Peter Gerard. Namesake? Blood brothers, dear boy. The tape in question was that which accompanied an early book of mine entitled Exploring Adventures on the Oregon, and if anyone else is interested in the tape (three adventures on one tape, although I must stress that they were written a couple of years ago) then they must write to Duckworth, The Old Piano Factory, 121 Gloucester Crescent, London NW9. Mention this column if you want, I know you lot, try anything for a discount. (Duckworth's have a few copies of the book left at £10.95, and a tape/supply of the tapes. Call or write to the book dept. on 01 498 3484 for further details, postage etc. -Ed.) For calling me a 'namesake', Mr Stockwell's tape is already on its way to him.

Following on from that, a very interesting letter from Andrew McBride of 199 Main Street, 11 Harwooden, Wt. Wellington, Northants NN9 5BA, his apparently received a letter from Simon somebody-or-other who will probably get another mention before we're through, stating that at the test in the infamous *Di Diablos* could be read by entering the following command: FOR A=0 TO 32767 PRINT CHR\$(PEEKING);NEXT A.

To quote Andrew: "Well, I thought if you could do this with *Di Diablos* why not do it with any other adventure? It worked! All you have to do is reset if you're wary if a cold starts." A bit late on he continues "when you have listed it, it will look as if it has gone funny, but it's just the CHR\$(PEEKING) statement using an IF ... THEN statement."

Does he condone this sort of thing? Well, no, provided that it is only used in *DIPE EMERGENCY* and you have given up all hope of solving the game. With this little trick Andrew has managed to solve *Wings of War*, *Mountains of Air*, *Temple of Ulan* and *Spyguy*. Interesting stuff.

Interesting, because one Roy Williams has written to me with queries on *Mountains of Air* and *Temple of Ulan*. Spot the connection? Brother Mike has often mentioned the role he plays as a surrogate postman, so it looks as if that is another name that I will have to adopt. Roy also

wants to know how to avoid getting aimed in *Final Mission*. Any offers, anyone? He says that he's got through the hole with the crossbar, the soap, the garlic, he's found the sensor, which he used to get under the floor, but then can't open the further door due to aforementioned stink. All help gratefully received.

Help in exchange for help seems to be another common ploy used by you lots to get your names into this illustrious column. Richard Heath, residing at 86 Newcastle Road, Leek, Staffordshire ST13 5PL, has written in with a number of queries. He'd like to know what has to be done with the rock cakes in *Templest*. They can be (SLIMP HOGOUT RETRAJ KRAHCA YB EMTGS DEPMUT NEED SAN HOGHAW RETGARAHG YMA OF DEF. This backwards writing really slows down the typing speed, let me tell you. Flaming I was just thinking that too — Ed.) Richard also wants to know how to stay alive in the radiation room past the second set of security guards in *Aquasaur 479*. Well, presumably you've found the seaweed and the mushrooms. Anyone who's anyone and who is profoundly into vegetarian food will know that combining seaweed and mushrooms produces a radiation vaccine pill which you can then eat.

And taking of *Aquasaur 479*, young Michael Edwards of Lincoln, who has been mentioned in this column already as often as Simon (that's a name I must admit has name rings a bell from somewhere else as well ...) tells me how to cheat so that you don't have to play that ***** security robot game. Fancy I don't remember reading that on the wrapper ... but anyway, beat the game, and when you've started press RESET. Type:

POKE \$A28H,\$A7E
POKE \$A28H,\$A69
POKE \$A28H,\$A80
END:0344

Type QUIT to restore the position of any objects that you might have left lurking around in the game somewhere. That should also help Gerny Stephens, who in that gawdlike style writes *GREAT COLOSSUM* in enormous letters. Paper your ears (turn, big brother) and then sendin please for aid. Help sheets should have reached you by now, and your last query on 'helpful snigs' should by now be solved.

But before we leave Richard Heath entirely, having helped him out, he is now willing to help you out with his solutions to *Templest*, *Sea Quest*, *Shenango* and *Justaposition*. As with all people who have their addresses published here, please send an A4 to *you@wattani.co.uk*. Every letter I've encountered so far that's been sent to this column has remembered their SAE, and although it took some time you should all have had replies by now. Is it my fault that Megan got dumped out of the PA Cup and the world is broken into (dis)unity? It is not. On the other hand, however, I've received one or two teeny weeny complaints from people who have written (enclosing SAEs) to people mentioned in this illustrious column and have not had a reply. Shame on you, if I cut miss Catherine to write a few letters, I'm sure you can.

Peter Neale is going to be a thorn in the Gerard side. I can tell. Queries in mega-quantity, one of which is worthy of mention here because it's cropped up a couple of times. How do you get rid of the statue in *Callisto Island*? Having found some nice you-can then EGM PCRD and SBL DQAP SURE! I must admit the lad's handwriting caused a bit of a headache at one point, since he seems to be *Demon Knight* is asking how to get exceedingly drunk in the portfolio. (Shame mislate, shunt?) Nice, on taking over I received a stack of letters full of queries, along with brother's extensive file of solutions and adventures. Two adventures that you have asked about were conspicuous by their absence from this file, the earlier mentioned *Final Mission* that Roy Williams was having trouble with, and now *Demon Knight*. No solutions, no maps, not even a hint, so once more I send out a plea for help.

On a large number of queries, the program that received the most mention was *Spyguy*, and easily the most troublesome feature of that game concerned entering the transporter co-ordinates. If you've solved this technology for re-stating what you write obvious, and from the number of letters I've read this many, many people are having problems here. My technique writing this time, either, so read at your peril. The co-ordinates for the transporter are as follows:

Planet 0-4-1-5
Emerald 2-7-3-8
Vader 1-8-0-0

For each place you must enter each number by pressing that number and at the end of the row you must RECALL LUR and if it doesn't work, blame James Bonfield, not me. Anyone who lives in a place called Barclay-Cannerton really is out of anything.

Nicholas Hart wants to know about good graphics adventures, so let's start up a tips portfolio. Write and let me know what your favourite graphics adventure has been to date, and as well as telling Nicholas what it is, or two it might also make one or two companies sit up and take note and try and produce something better.

As an update to Mike's last column, in which he mentioned the Dragon Magazine (Michael Edwards, Andrew Hill, the latter contactable at 13 Perry Jones Close, Blaine, Great MP3 9H9). A small update from the purveyors of the rag, stating that a year's subscription is priced at £650 for twelve issues, and that apparently barney (sic) costs extra. A sample issue is available for £10 from Andrew. It would appear that the magazine is now over one year old, a mere update in comparison to D4, but then we've all got to start somewhere!

Do you know, I nearly made it. Almost at the end of the column, no mention of Simon Thurgumsey, and what happens? Letters (or emails) in from all sides, ranting about Dragon writers who use Pental and play Shades, hints for Tanglefoot and Acquest 4? (and the aforementioned

Shades: does this mean his future's so bright he's gotta wear shades?), and desperate pleas for me to mention yet another Dragon magazine. Okay, Hargrave, have it your way.

I mention these two magazines as a lion might talk about a hobby — appropriately, the one called Dragon's Tail, and a year's subscription costs £750. Shouldn't have? I mean a copy yet I've refer to this as The Hind you will have to send off your pennings on spec, but the person to write to is Jason Cogdell, Dragon's Tail, 34 Shooters Drive, Nazeing, Essex EN9 2GD. Ask 'em if they sell sample copies if you just want one to look at.

Perhaps someone will send me a copy someday.

Hope that you enjoyed my first flight into Dragonland. It's always difficult taking over an established column, but we'll see what can be done. Next month I'll be looking at (among other things) half a dozen adventures that people have sent in for review, plus if course any other games that might crop up between now and then. Your letters are of course always welcome. I can already spot the troublemakers and will deal with them accordingly (you know your names, you guys!) and the rest of you too — just keep 'em coming.

Adventure Contact

To help puzzled adventurers further, we are initiating an Adventure magazine — 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 Helpdesk.

12/13 Little Newport Street, London WC2H 7PP. As soon as enough entries have arrived, we will start printing them in the magazine.

Don't worry — you'll still have Adventure! That's right to write to us well!

Adventure
 Problem

 Name
 Address

CLASSIFIED ADS

ADVENTURE SYSTEMS: Having trouble fitting your adventures into memory? For details also available, tailor made loading screens and graphics. Write to J Foster, 94 The Oval, Fish Park, Sheffield, S5 6SP.

SOFTWARE FORTY FOUR: original games including some latest in Tanglefoot and Wizard Quest only £35. Telephone 0438 262294.

DRAGON 32: (good condition, original) — £300. Balfour, 45 Southwicks, Yarnell, Somerset BA20 2QG.

FOOLCASTER II: Predicts any future. Compares results with predictions and permits algorithm adjustment. £35. DCL, 6 Hoptone Terrace, Shoreham ME12 1JH.

DRAGON 850C SYSTEMS: (Drive & Data Interface) for £300. For a 3" 500KB system. (Plus 1MB & Post). Also 1" and

4800 track drives. Send a.s.k. to: Datacan Ltd, 18 Pine Drive, Haslem, Camberley, Surrey. (Phone 0276-35657).

DRAGON 44 (1600): Dragon 32 £30.00, books £16.00, magazines £1000, cassette £8.00, management system £8.00, forth language £8.00, joystick £9.00, in use (Bargain). Save £48.00. Together £166.00. Tel. (0303) 798475.

DRAGON 32 with joystick and various software for sale £75.00. Tel: 0781-68621.

DRAGON DOS: disc controller. 80 track double sided drive, number Dragon User Magazines. Offer £725-£1550.

DRAGON 32: Joystick, sound extension model, Philips cassette recorder, manuals, games and magazines. £80. Tel: 01-242 0821 (joystick) or 01-981 8102 (after hours).

PLUS ORBICON: system, anylograph, and H.M.R. Database. All discs £10 each. Also I have a Delta (disc) controller £10, serial cartridge for 85, Dragon data joystick £5, Ticon RM100043 modem plus all leads cost £200 will accept £100 o.n.o. It is Hayes Compat. and auto-everything. Dragon 64 £98 s.u.s. Finally broken down Dragon 32 plus power supply which is working. Phone Moseley 552441.

FLEX OPERATING SYSTEM: disc plus crunch cost £1 basic disc plus FlexDatabase up-edit chip plus comm, graphics controller cave adventure CD-R operating system disc.

GGP115 printer/poter/lite pen speech synthesizer joystick software books £75 s.u.s. Plus £6.226H.

HERE'S MY CLASSIFIED AD.
 (Please write your copy in capitals on the lines below)

Name
 Address

 Title

Please cut out and send this form to: Classified Department, Dragon User, 12-13 Little Newport St, London WC2H 7PP.

Three letter cards

Gordon Lee shuffles uneasily while he waits to see what the cards say

WOULD-be competitors often write to say that they would like to enter the competition but feel that they lack the necessary experience to tackle the problems. In fact, apart from the possibility of winning one of the prizes, one of the spin-offs of attempting the competition is the opportunity to develop original programs, starting with the interpretation of the problem, finishing with the final program, which should, you hope, provide the correct answer. Many readers possibly use their *Designs* either only for loading pre-recorded programs, or for copying listings from the magazine. In fact, most of the competitors require only a logical approach plus a knowledge of some simple basic commands. Even the inevitable mistakes can help to pinpoint a particular pitfall, and the debugging of a program can prove particularly instructive. Also the experience gained can have an application in other areas of programming.

This month, this column is devoted to all you beginners, so experienced programmers may like to jump to the competition at this point.

At the heart of all programming is the management of variables. Whether it is in the creation, examination, comparison or alteration of variables, a knowledge of them is essential for successful programming. Take a look at a basic listing, and you will find that most programs have a variable in some form or another. Note that these variables come in two flavours — numeric (used to store numbers) and string (used to store text or other characters). The string variables are identifiable by the dollar sign.

Quite often variables can be used to 'mimic' actual objects or procedures. For example, if we wished to create a pack of cards, we could use listing 1. This is typed in and run, the command PRINT PACK\$

will result in a 104 character string containing each of the playing cards, in order. Note that each card is represented by two characters in the string, for example, 3H, JS, or AD, for the three of hearts, jack of spades and ace of diamonds respectively. From this it is easy to select a card at random, for example by typing in `X=INT(52/(2-1)*RND(MID$PACK$,X,2))`

The value X denotes the position in the string at which the random card is to be found, and must obviously be an odd number — hence the routine of multiplying by two and subtracting one. Thus, if the character and the text in the string is protected with a dollar sign which indicates a playing card value. An extension of this technique can be used to 'shuffle' the cards. The computer does this by starting with an empty string (`SHUFF$=""`), and adding to this a series of 52 cards selected at random from the `PACK$` string. To

Listing 1

```
10 PACK$="":SUITS="CDHS":RANK$="A23456789TJQK"
20 FOR S=1 TO 4:FOR R=1 TO 13
30 PACK$=PACK$+MID$(RANK$,R,1)+MID$(SUITS,S,1)
40 NEXT:NEXT
```

Listing 2

```
50 SHUFF$="":FOR N=52 TO 1 STEP -1:X=INT(52/(2-1)*RND(MID$PACK$,X,2))
60 SHUFF$=SHUFF$+MID$(PACK$,X,2)
70 PACK$=LEFT$(PACK$,X-1)+MID$(PACK$,X+2)
80 NEXT
```

Listing 3

```
5 DIM HAND$(4)
10 C=1
100 FOR N=1 TO 4:HAND$(N)=HAND$(N)+MID$(SHUFF$,C,2):C=C+2:NEXT
110 IF C<104 THEN 100
```

Listing 4

```
50 N$=PACK$:T=1
60 SHUFF$="":FOR C=1 TO 52
70 SHUFF$=SHUFF$+MID$(N$,52+(C*2-1),2)+MID$(N$,C*2-1,2)
80 NEXT
90 PRINT:PRINT SHUFF$;" (";T;")"
100 IF SHUFF$=PACK$ THEN 120
110 T=T+1:N$=SHUFF$:GOTO 60
120 END
```

prevent the same card being selected twice, as a card isolated if it is removed from the HANDS string at line 70. Using 5 lines 90 to 80 perform this shuffling operation.

Having done the shuffling, we might wish to deal four hands of cards. If you add listing 3 (line 4), and lines 80 to 110, the cards are dealt into the dimensioned array HANDS(). Of course, it is perfectly possible to combine listings 2 and 3 and feed the randomly selected cards directly into the array without the need for a separate 'shuffle', but the listings as given more closely imitate the actual procedure using actual cards.

Another type of shuffle is the riffle shuffle. Here a pack is divided into two equal halves and the cards are riffled so that they fall alternately (at least, that is how it works in theory). In practice, the cards are not likely to oblige and interleave to exactly. Curiously, a series of perfect riffle shuffles

will restore the pack to its original order if you replace lines 80 to 110 with listing 4; the computer will now perform a series of riffles until the order of the pack returns to its original state. Note that in performing the shuffles, when the pack has been divided the card which is allowed to fall first is the one from the top half of the pack. You will be surprised to find that, by an odd coincidence, fifty-two shuffles are required to restore the order: three cards are removed, then only eight shuffles suffice!

Competition

For this month's competition we need to simulate some packs of cards, not with the conventional suits, and values, but bearing the letters of the alphabet from A to Z. I have three piles of cards, each comprising a complete alphabet. They are placed in a row, face up, with the cards in alphabetical

order: A on top and Z at the bottom. I cut each of the packs several times, completely re-order each occasion. Note that this does not alter the order of the cards, just their relative positions from the top of the pack. For example, the top card might read 'J', if I then remove one card from each pile I will get a new sequence, and by repeating the procedure twenty-five more times, will work my way through the packs. In so doing, I will arrive at a series of three-letter combinations, for example, 'VUB', 'JXC', 'ALD', 'BME', 'CNP', 'DOG', 'EPH' etc. Occasionally, I may produce a word, such as 'GOO', and in some cases it is possible to find two or more words in the same 'cut' of cards. 'ATE', for instance, will produce 'PAT' and 'NOT' will produce 'STY'. Some 'trips' are 'TWO', 'LOO' and 'LUA', or 'BUS', 'HAF' and 'PIG'.

Can you find any sequences of more than three words?

Prize

Peter Gerrard is taking over *Adventure* this month from his occasional brother-titles. Peter is uniquely suitable for this demanding post since he is not content simply to carry on the family name in his adventure business, he has to go writing books about it as well.

Pete's book, published some time ago by Duckworth's and still — just about — in print, comes out alongside a cassette of the same subject, *Exploring Adventures on the Dragon* on the (you guessed it) Duckworth label.

Details are here to track down the book and the tape are given by Peter in the *Adventure* file on page 26, but Duckworth's have kindly awarded us an unspecified number of copies for prizes for this month's competition — so let's tighten 'em.

Rules

So there as your hand. You know the rules, save out the strategy plan the program, send us a listing of your solution (no cassette please) and any notes you want to include. Mark your envelope with the words 'MAY COMPETITION', as well as Dragon User's name and address (don't forget to include your own name and address) and send it to us.

Come to enter for the May competition is to answer the question 'What is a Duck Worth?', using your wit, skill and judgement. No packet tape required.

February winners

In contrast to the March competition (also popularly known as the 'January' competition, thanks to the efforts of a sub-editor sitting out a thousand miles from this installation), where entries are pouring in in big packets (OK, opened out big packets) and the backs of blank cheques, the entries for the 'FEBRUARY' competition have tumbled in, slow and stately, reading, few in the water under their

massive weight of program analysis, wit and footnotes. We're offering a set to the *Dragon* people's Distance as a follow-on.

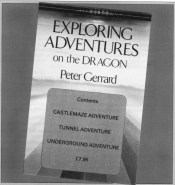
This month's winners, building somewhat off, are R. H. Wilson of Gillingham, G. R. Barber of Sutton Coldfield, S. A. Baskin of Chislewick, T. Rawson of Heston, Phil Sapwood of Liverpool, H. Gandy of Sheffield, Mark Dawson of Nottingham, M. Phillips of Bristol, Mike Darrel of Leicester, Alan Thomas of Stapleford, Greg Cole of Cusbury and Paul Wheaton of Wootton Under-Edge. Copies of Baby's Ruby Rabbit are in the post, and we have a few spare tickets. No begging letters please! They will be going to deserving causes

(such as the Editor's Magic Bottomless Box for special services and precious prizes). Baby's list of games is now available from R. B. AJ Preston Software, Call (0595) 880085 for more information.

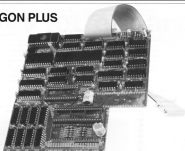
And an emergency apology: Siskind G. J. Gray of Middleborough didn't get his name check in last month's winners. Gordon wants to know why, and says his entry was first class and one of the most original. Sorry, G. J. Your game is on its way.

Solution

Up to 100 primes were counted.



DRAGON PLUS



The best upgrade for your DRAGON: fast 80 by 24 screen, extra 64K memory, fits inside your DRAGON. Hundreds of satisfied users. Installation service available. Requires 64K DRAGON upgrade service for DRAGON 32K — GALL US\$1. Especially recommended for OS-9 users — get the 80 " 24 screen AND save the 6K overhead of the 81 column screen. Use 80k and C without having to reboot!

DRAGONPLUS

— £100.00 EDIT + Cartridge/Disk for DRAGONPLUS

FLEX DRIVERS

— £5.75

OS-9 Drivers

£14.95

£5.75

FLEX SOFTWARE

Get the best out of your DRAGON with FLEX operating system. 51 by 24 screen standard. 48K free memory for your programs. Includes the best 8088 assembler available for the DRAGON. DBASIC lets you run DRAGON BASIC programs — includes cheque book program. Requires 64K and disc drive.

FLEX Editor/Assembler/DBASIC

— £99.99

DYNACALC

— £49.95

SPE word processor/spell check

— £85.00

RMS Database

£49.95

**** SPECIAL OFFER — complete package — £199.00 ****

FORTH 80 for FLEX — complete implementation on two disks with source £25.00

The Best Modern software for the DRAGON 64

CHAT COMBO (chat plus 2, IMODEM, CHAT/Viewdata £45.00)

CHAT/Viewdata access PRESTEL (cassette/Dragon/Dos/FLEX) £14.95

DRAGON HARDWARE — SPECIAL OFFERS

EPROM PROGRAMMER Mark 1 (KIT) for DRAGON 32 reads/programs up to 2764

£25.00

EPROM PROGRAMMER Mark 2 (Assembled) reads/programs up to 27256

£55.00

MODERN/RS-232C I/O interface for Dragon 32/64 (uses 6551 A/D and 6522 V/I)

£5.95

(case board with instructions)

CASE for P2 - 232

£1.95

FLOPPY DISKS (Certified double density double sided)

10 for £9.95

FREE DISKETTES WITH DRAGON DISK DRIVE SYSTEMS

DRAGON/PC Convert

£24.95

Now you can convert your Dragon 64/32 programs and data to run on the IBM or compatible (eg Amstrad) Personal Computer PC, XT or AT.

Dragon Basic is very similar to the standard BASICs used on PCs. The hard part is transferring your programs and data to the PC's disk. DRAGON/PC Convert is a program for PCs which reads DRAGON-DOS format data and copies data and programs to standard MS-DOS format.

Buy an AMSTRAD PC — get DRAGON/PC Convert FREE

From £49.95

(£39.95 inc VAT)

plus £20.00 carriage

FAST MAIL ORDER — PRICES INCLUDE VAT — PLEASE ADD £2.00 POST & PACKING

COMPUSENSE

66A WILLOUGHBY LANE, LONDON N17 6SP

Telephone: 01-885 3300 (2 lines)



FAST MAIL ORDER

COMPUSENSE

