

DRAGON USER



Yabud
magazines

The independent Dragon magazine

March 1988

Contents

Letters 2

What became of the Dragon Professional? ... Rights arranged ... Here's Surety ... where is the HDCT202? ... Hi Score center ... Dec and Jan crossword solutions.

News 4

Adventure from Dragonfire ... update on Dragon shows ... two computer clubs: north-East England, and Norfolk

Crystal chandeliers 5

Alan C. Smith goes to the 8th 6800 Show, samples the wares and the carpets.

Dragonsoft 6

Robert Postall from Quickbeam, Her of Teotihuacan Smeemargravs, Zetoks and/or game selections, join us on Risk & All Proton.

Pamecodes 8

Pam D'Arcy continues with the fourth part of her introduction to machine code.

Crossword 10

Last month's winners, this month's contest.

Total profits 11

Some advice on total profits for space capitalists from Julian Bowen.

READ-PC 12

A program by Maria Vermeir to read MSDOS PC disks to either DragonDOS or OC-8 for the Dragon.

Green on black 16

A small hardware modification by Paul Field to give the Dragon a green on black display.

Expert's Arcade Arena 17

Peter's, Warren's Quest and some advice on maps from his *Business the Expert*.

Down in the Dumps 18

A dump for the British GPT00A.

Dragon comms 20

An introduction to the Dragon's communications by Martin Armitage.

Winners and losers 22

Gordon Lee finds that some October contestants went to unnecessary lengths.

Adventure Trail 23

Pete Gerard dispenses more handy adventure hints.

Write: ADVENTURE 24

Pete Gerard turns his back on ambition and learns about DROPPing it ALL.

Competition 26

Gordon Lee takes something simple like Gray Code (what?) and challenges you to turn it into binary (how?).

The Answer 27

Gordon Lee's own solution to the December competition.

Communications 27

You ask the questions, you answer.

Editorial

If this Dragon User is late, then I'm an holiday; if it's on time, then I missed my plane, and if it's very, very late, then I'm hospitalized somewhere in the Alps, hopefully suffering nothing worse than an overdose of spaghetti.

But seriously, there is a degree of controlled lateness here in an attempt to avoid the uncontrolled lateness which decelerated last time the staff took a week off. Don't blame the Postie Bill Marsh.

The Postie may be to blame for the fact that we haven't got our *Issue's Dreams* yet. Roy is looking into it.

The Cardiff Airport Dragon show has been bogged by confusion, but if today is earlier than 27th Feb, there's still time to pack your bags and go. The enquiries number is on page 25.

This month we have a program to read PC (MS/DOS) disks onto Dragon drives, a report from the 8th 6800 Show, a hardware inventory, and all the regulars.

Even the Classified ads.

Dragon Answers 28

You ask the questions, we answer: DO write-protect, GET insert error, adding a RAM extension.

STOP PRESS - STOP PRESS - STOP PRESS

Just in case you're late ... the Cardiff Airport Dragon Show really is on February 27th.

Telephone number
(All departments)
0262-1343

Editor
HELEN ARMSTRONG

Production Editor
DRAGON EDITORIAL

Administration
CAROL FRITH

Advertisement Manager
DRAGON EDITORIAL

Marketing Manager
HELEN PERRY

Managing Editor
PETER KANE
Publishing Director
JENNY IRELAND

Subscriptions
UK £4 for 12 issues
Overseas (air/air) £28 for 12 issues
0262 1255-57 Telex: 296275
Dragon User, 1075 Little Newport Street,
Larkhall G62 4 5YF

US address: via Business Press International,
205 East 42nd St, New York, NY 10017
Published by Yabud Publications 1988
© Yabud Publications 1988
Typesetting and Production by Asian Limited,
London W8 7JF
Printed by Habbey Brothers Ltd, Ashford, Kent
Registered at the Post Office as a newspaper.
Dragon and Yabud are trademarks of
Yabud Ltd.

How to submit articles

The quality of the material we can publish in Dragon User each month will, to a very great extent, depend on the quality of the submissions that you can make with your Dragon. The Dragon computer was launched onto 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 8000 words long. All submissions should be typed (which leaves 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 every submitted article or program, so please keep a copy if you want to have your program returned; you must include a stamped addressed envelope.

I have seen the High Score corner in two issues of DU, so here are some of mine:

Shaqolin Master — 108,200
Chuckie Egg — 183,043 — level 19
Speed Racer — 10 rules, 64,720
Shacktopper — 52,260

I've been reading DU for two years, keep up the good work.
Rick Carr
88 Main Lane
Mt Hill London SE67 3HT

MH oh. We just opened the can of worms again...

I was reading September's high scores for Chuckie Egg and none of them are as high as the score my brother reached a couple of years ago. I don't remember the exact score, but it was just short of 750,000 (I think level 80).

Jeff Boor
48 Cannon Street,
Dover,
West Kent CT11 1TE

Hi-score corner again . . .

I am fed up with looking at those pathetic high scores for Chuckie Egg. I am the champion, and that is final. I enclose a photograph of my second highest score of 919,476. I once topped my million, but had to fire in my caters at the time.

I am a 38 year old mother of 4. Dragon's Lair etc has now sponsored Chuckie Egg from me, so that I have more time for cooking, cleaning etc.

Mrs. Sheila Stobbs (Basso)
Hill Cottage
30 Oakley Lane
Wilmslow
Dorset
SK9 1JL

Don't let them turn Basso back into Basso. Buy the best a cookbook and yourself a copy of Real Escape. And now, for the entertainment of champions everywhere...

I think I can claim to be Chuckie Egg master! I have had many times well over a million points — my record being just under 1,480,000 (level 79). As anyone else who may have got this far will know, this score is really irrelevant as I could easily beat it if I had enough time. After level 32, the game goes back to what was called level 25 (only it is now level 33). So it is possible to play for hours. My longest game lasted about 32 hours. But boredom takes over after the first couple of hours!

Andrew Laird
2 Marston Road
Rosedale Grove
Stirling
Clackmannanshire KY7 5JF

PC Are there any 6800 stores planned for around Manchester in the future?

THE last 6800 software heard of is being arranged in Cardiff for April 1985.

NEW HIGH SCORE ON CHUCKIE EGG: 2,216,260 LEVEL 118 (17th AUGUST 1985) BY MARK AND JACOB HINDSON (GAME LASTED ABOUT 2.48 HOURS).

THIS has been a public service announcement...

FOR fear of strangulation from the McCullough (September letter) I put forward my claim for the highest score ever on Chuckie Egg. I scored 4,081,700 on 24-10-84 and have rarely played it since (so wonder). After level 86, the numbers change to eggs and various other parts of the graphics of the game. I reckon I reached around level 233.

Ever though talk may not believe me, this was definitely not pain aided and was on my truly old Dragon 32. The game stops getting harder after about level 32 and stays at the same speed for infinity, so I could have scored an awful lot more, but after 8 hours almost continuous play I simply got fed up and went to bed! If anyone has beaten the score I'd very much like to see it.

W J MacArthur
Middie-PC
Jole of Tree
Argyll
Scotland
PA7 5BN

Robesse oblige...

I was very humiliated at the high scores people sent into high score corner. Paul's sister will have to struggle me because I got 6,750,040 and I had to kill all my lives because it was getting late, the level numbers changed into funny shapes and everything went really faster. The levels which generally hard were level 7 and level 8.

Lee Kalyan
60 Angelhill Road
Bedford
MK43 5LP

DOES this mean there's even less to do in Bedford of an evening than in the late of Time? Perhaps you need a foreign break. But...

TO end all speculations about the highest score on Chuckie Egg, I do have it! It's 15,204,000! This seems unbelievable, but is easy to explain: from about 450,000 the speed of the ducks stays constant, so you only have to aim to get more bullets than you lose. I had enough after 32 hours, and interrupted the game in level 819.

With best wishes to all British Dragon users.

Andreas Meyerow
Rathenaustr. 2
D-8530 Garmisch
West Germany

WHAT we need is a totally new approach to this game...

REFERENCE to score corner, you lot make me so envious with your multi-800s scores.

I have never got past the level which has three ladders at the top of the screen, each with an egg at the top.

How do you get the two eggs at the left hand ladders?

Anybody help me to get on and beat those massive scores!

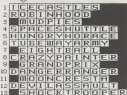
H. Jones
84 Allison Street
Gardiner
Northgate
Canter
GL11 2LL

No, No, H, I think you have the game of a new concept here! All it needs now is somebody to work out a way of playing Chuckie Egg for 48 hours at a stretch without getting beyond level one.

We haven't printed the solutions to the Dragon crosswords yet, so here...



The December 1984 crossword solution.



The January 1985 crossword solution.

New adventure from Dragonfire

DRAGONFIRE Services have announced their first adventure for the Dragon, *Pyzadventure*. *Pyzadventure* you seem to enter the lost tomb of King Tutankhamun's father, you might find the ancient Amethysts, the gold death mask and his fabulous treasures. But will you come forth alive?

Pyzadventure features real speech, traces and save game facilities. "A real humdinger" say Dragonfire, Theprison.co.uk.

plus 50p post and packing in the UK, £1.50 post and packing overseas. Overseas payment in pounds sterling by international money order please.

Dragonfire have several other titles in the pipeline, including *Deathfire*, *Magrace* and *Spellbox*, *Dragon Music* (series) and *Underlings of Crook*. Send an SAE for a full price list. Dragonfire are also interested in looking at software for prospective publication.

6809 show — Dates mixed up

IMPORTANT UPDATE: John and Helen Parrin apologise for getting the dates mixed up in their Show ad, in last month's issue, and place a local Telex epidemic and Demin extract in mitigation.

The REAL dates of the ap-

pearing shows are Saturday 30th April at Ouseley Town Hall, one of the Dragon world's favourite venues, and Saturday 23rd FEBRUARY at Cardiff Wales Airport. For further information, contact John Parrin Software on 04-203-5076.

WP waits for words

RIP Software of Surrey has produced a new word processor to meet his requirements, and would like to know if other readers are interested in his program.

The vital criteria are a 64-character WYSIWYG MS (what you see is exactly what you get) fit on screen display, all necessary instructions for use displayed on the screen during run-time, easy-to-use facilities for retaining printer codes for any connected printer and amend the display of printer codes for easy comprehension, as well as user-definable auto-repeat keys, user-definable line and fill, auto word wrap, alternate line feed print, keyboard-locked, on-screen repositioning of windows or windows, display with or without fontlines on screen, and, with the Epson FX80 printer, 32 fontstyles and sizes, fully mixable.

Anybody wanting further details can contact Mr. Siltborough at High Green, The Drive, Baitmore, Surrey GU2 7DH. It is too clear whether or whether offering the package for sale — we got the impression that he's waiting to see if there is any interest.

North East User's club

THE North East Dragon Users Club has 14 members and meets every Wednesday evening in the Etchells lounge of the Grindon Mill, Chester Road, Sunderland. Anyone who is interested in visiting along, or contact the organisers, Chris Johnson, 22, Washington, Biddock Village, Washington, Tyne and Wear NE28 7H.



Club in Norway

The Dragon Computerklub of Norway has written to say that they will send a copy of 1 issue 20 page monthly magazine (A4, duplicated both sides, in Norwegian) DRG-Mens to anybody who enquires.

Dragon Computerklubb has been running for three years, and are organisers of software from England for Scandinavian members. Enquiries to Ole Eddy Stokken, N-5410 Sagvåg, Norway.

68 Microcosm this month

THE December 1989 issue of the 68000cosm, the journal of the 68000 users group, includes *CGEMIX* on the Macintosh, the *Scanputer*, a Flex disc formatter program, an introduction to the Data Protection Act, look at topics from Tom Goodfellow's Media bulletin board, small ads, lots of letters and the offer of a Cardfile program. Also a review of the 68k 6809 Colour Show. According to the membership sec., the last

address of the printed version of the journal was a typesetting mistake in the title word in the middle. As this is no more ambiguous than the average person's handwriting, it magazine copies will still reach their destination. The next issue's title address is Mr. G. Barnes, 176 Green-Ards Road, Wimbledon, London SW20 6ED.

Computa — Text call

WOULD anyone who purchased a copy of *Computa* — Text from Dragonfire Services at the 6809 Colour Show in December please contact Dragonfire at the address

below, sending their names and addresses, so the instructions are now ready.

Dragonfire Services, 13 Perry Jones Close, Watna, Great NP3 2PH.

In the land of the crystal chandaliers

Ken G. Smith returns from the 8th 6800 Show with his impressions

WHEN I saw there was going to be another 6800 Show in London, I didn't believe it. When I got to the Connaught Rooms, I believed it even less. Crystal chandaliers? GARBETOS? And casual seating — this was some venue for a Dragon Show. The Grand Hall seemed much larger than the Old Horticulture Hall. The exhibitors, who I'd never felt safer with their backs to the wall, were spread around the perimeter. The new venue made this a very comfortable place to attend. The attendance figures seemed to me to be more in the hundreds than thousands and the size of the hall seemed to emphasize this.

The Dragon market is now so small that it is a pleasant surprise when new software is released. Well, just to prove that they are still surprise us, two suppliers chose to unveil new games at the show. The first of these, Quickbeam, placed their spotlight on all their current large on sale players with an art, video football. Now a new and fascinating program with crowd noise and action replays, it is almost as much fun to watch someone play and I kept the younger showgoers amused for hours. The second came from Permacom, but previously noted for bargains, Paris D'Arcy's come up with a real winner for his exciting Global Prix. It is a tight screen motor race game featuring several different tracks and the ability to race against the computer or another player. Available on disc or cassette, it provides real competition for Speed Racer. Well done, Paris! It is still a can't miss.

Bus Extension

Harry Whitehouse, who you will remember used to trade as Peasewell, was there with his usual supply of goodies, these wonderful power supplies and some new keyboards. This time Harry, who jointly promoted the show with John Parn, was also bringing his New Era PE232 interface which seems to be bringing a lot of Dragon 32 users on line. Offering more sockets and about extension, it can be used in conjunction with a disc drive. However, it must be said, you will end up with a rather long arrangement. Although he is one of the few that will have to be told when it stops producing guides for us, Harry is getting involved with other machines and sites, in fact, keeping up Spectra's daily five pounds each.

Another corner show was Bill McCowan demonstrating how to use his Printer Control as a desk top publishing system. Also

*It is well worth
it to meet old
friends, make new
ones, and talk about
the machine to
which we are all
committed.*

on show were the results of Bill's Colour-print program which converts the four colours of PMode 3 into different densities of print. The end result is a black and white picture of amazing quality. McGowan's software has reached the stage where it outstrips anything of a similar nature and should advise you accordingly anyone who has just bought a printer is that to Bill McGowan! Bob Harris, who also markets McGowan software, was demonstrating his own PDK utility. This gives the humble Dragon 64 something of the environment of the Apple Macintosh and runs under the BASIC/2 operating system. Bob's is one of the few software houses that is actually increasing their range of Dragon products. They even offer an upgrade service where for thirty five pounds they will board your old 68 to a full 68K, making it capable of running BASIC, FLEX and GEM.

H C Anderson's arrival proved that stories of his fairly late existence were nothing more than a Grimm's version. The big blunder Datto was cooking up got games at a pound each and 200 is an order eighty pounds, complete with both manuals. He certainly seemed to be enjoying himself, I only hope he found it a rewarding experience.

With mountains of supply software slowly arriving away John Parn has shifted his emphasis away from selling off surplus stock at discount prices and is now distributing more current software. For those of you who like the word 'enterprising' what has happened to the final big free version of Total Eclipse I can tell you that John

Parn is also distributing David Mather's excellent Pictur Maker. David was at the show demonstrating his wares and I must say the results were remarkable.

All the usual user groups attended plus one new one dedicated to Delta Dos users (DUBB). They had one of the new Tandy 1284 CoCos with enhanced graphics on show. Rumours were rife about the machine, mainly as to whether production was on or off. The price many asked quite good but whether it could stand against competition from the likes of Am is another matter. DUBB were up on the balcony along with the boys from 'Dragon's Room' magazine. Their supply of free copies soon ran out and Simon Jones was taking orders. I have now received my copy and have to say that it is very professionally presented and catered for a wide variety of readers. Also up on the balcony, I had a very interesting chat to Neil Buchanan of 88 Microcom. If any of you wants to run hard discs on a Dragon then feel in your mats. He can probably even supply the parts.

Well, as usual, I enjoyed my time at this year's London show. However, the falling attendance makes me wonder how much longer such an event can continue in its present form. As long as it does I, for one, will gladly pay for my ticket. It is worth it to meet old friends, make new ones, and talk about the machine to which we are all committed (or for which we should all be committed.)

"Stirring" Smith

Now, to get things up to 11, I have decided to list the following recommendations for an imaginary show event called the Golden Digi. 1. Best new hardware: Harry Whitehouse for his New Era PE232/232toModern driver.

2. Best new software: I have to judge this one between Permacom's for Global Prix and Quickbeam for Indoor Football.
3. Best show idea: Dragon's Fear for their live Microcal link up.
4. Best bargain: National Dragon User Group for their thirty five pound Modern package.
5. Man of the show: John Parn, Harris, Anderson for his multi-users and the effort he put in just to get there.
6. Show:en: Raspberry Computers for being the only people offering less than one 'Whispermore' turn up when they have sold all their 'Flur' boards!

Dragonsoft

Getting into this game is just for kicks

Title: Indoor Football
Supplier: Quickbeam, 508
Avenue, 20 Salisbury Road, Hod-
deston, Herts EN11 9JX
Price: £9.95

I seem to have become Dragon-
soft's football correspondent as
my dear Editor has given me
the task of reviewing Quick-
beam's latest offering...*Indoor
Football*.

There seems to be no short-
age of quality Dragon football
games around, what with
Champions (and the new
Champions), *Football Manager*
and *Crucy Foote*. But now
Quickbeam have come up with
what is quite simply the best
full perspective arcade style
football game on the Dragon.
These words come from
Quickbeam's advert for the
game, but what they mean, are
true. Sure the ad is good,
but *Indoor Football* is better.



Indoor Football follows the
same method of play as *Crucy
Foote* (where the player con-
trols his team and battles it
out with an opponent). I prefer
this type of football game to the
less of *Football Manager*
where there is absolutely no
skill involved, and the result of
each game depends really on
luck. However, I don't think it's
quite right to drop all the
features of that type of game,
one of the better features of
Football Manager and *Cham-
pions* was that each team
played in a league and
throughout the season battled
for a place in the Cup and
Europe. In *Indoor Football*, one
simply plays either the computer
or an opponent for a certain
time and once the time is
up, that's it.

In the game you control the
kicking character (you'll
notice there are two, you control
the one in your colours).
You have direct control over
this player, and with him it is
possible to tackle another
player, simply run up the field



with the ball, pass the ball to
another player, or take a shot at
goal.

There are other players on
the field which move independ-
ently, and they are capable of
running towards a loose ball.
Once you've got the ball it's a
hard slog to get to the other
side because there is always
an opponent hot on your heels
and you'll get the ball, you're
slower than the other player.
What it does mean is that you
frequently have to run round in
circles trying to dodge your op-
ponent's player. However if you
manage to beat this player and
the rest of his team you'll just
have goalkeeper to contend
with, and assuming you score
you will be awarded with a loud
cheer and another motion action
replay.

The game is split into two
halves, the ball being replaced
in its centre spot at the begin-
ning of each half.

Before the game begins you
are given a number of choices.
You can choose the number of
players in each team, you can
choose whether to play a friend
or the computer and you can
select the length of each game.

One thing which struck me
after loading was the lack of
graphics. As most of you will be
aware, Quickbeam have pro-
duced some great sounds on
their games with their DAME
program (must get it some
day). However there's no
music on this game and in-
deed there is very little sound
on the game, the noise of the
ball bouncing and the cheers



from the invisible crowd being
the only sound.

The graphics are in 3D and
are good (a little simple). The

players seem very familiar to
me — they're very much like
the soldiers on *Fire Force* and
they too are drawn well. As
usual Quickbeam have used
larger than normal sprites to
draw the men.

There seems to me to be a
slight fault in the direction, as
all too often I've had goals
scored against me that are
quite obviously not valid. Even
in slow motion it's obvious that
a goal should not have given.
The design of the goalposts
don't help the matter very
much either as they're in 3D,
making it even more difficult to
distinguish just what is a goal.

Another problem which hap-
pens quite often was that al-
though I had kicked T or B
player, after a while most of
them disappear — in fact I've
seen many disappear before
my very eyes. (Transferred to a
large file, very likely — Dr.)
Quickbeam have come up with
some great cassette inlays but
this time their presentation is
amateurish to say the least.
Even *Crucy Foote* seems to be
more professional.

Now I know many of you will
say it's the game that counts,
not the cassette inlays, but if
computer enthusiasts who sell
their games go to the bother of
making their product look de-
cent enough then it's safe
to assume Quickbeam could do the
same.

If these few comments lead
you to the impression that I
loathe this game, you're being
foolish. I absolutely love this
game, the good graphics
coupled with the extreme real-
ism of this game combine to
give possibly one of the games
of '88. OK, it may not be
graphically stunning, it may
not have hot sound, but after
a playing *Indoor Football* you'll
know all that to one side, and get
on with playing what is a great
game.

For me it was a joy to play
Indoor Football, it was certainly
addictive and very competi-
tive. I'll give it five Dragons, but
if you don't agree you know
where to find us.

David Morrison



Living on

Title: Zetola
Price: £2.99
Supplier: Paces

CIVIL war, hurricane and in-
vading armies, all part of daily
life and misery under Zetola.
Zetola, having accidentally
discovered the life you get
the job of running it, and you might
guess that it's not one of the
most secure jobs in the world.

Zetola is almost two pro-
grams: on side A of the
cassette are instructions with
what seems like several dozen
pages of text. These instruc-
tions delve to great depths to
help you understand just what
is going to happen — but after
so many games where in-
structions have consisted of
something the Type Command
and the game will start auto-
matically. Use right joystick,
these attempts have to be
understood.

The instructions even go to
the extent of giving you a little
text to see how much you've

understood. If you answer cor-
rectly you move to further in-
structions. If you are nearly
right you are told of the error of
your ways and then move on.
However, if you type 'Z' when
the answer is 3 you're told not
to be silly and answer again.

The length of time I've spent
describing the instructions
shows that the size but the game is
not as complicated as it may
momentarily seem. You are
given a town of office ranging
from 5 to 20 years starting in
1987. (Survival level 1 is easy,
Survival level 2 will have a bit
of, including I suppose. You're
also asked if you really
Dragon will permit the 65495, 0
Zetola to help speed up things
slightly.

The little Southern paradise
consists of 15 blocks of sea and
38 of land making up a 9 by 6
grid. Given a starting popu-
lation of 500 and food for that
number, the task with a limited
supply of money is to meet the
growing population.

Heir raising tasks

Title: The heir of Igo
Supplier: Simon Hargrave
Price: £5.00

SIMON Hargrave's adventures, starting with *Demons*, a game with a good plot, which I rated as four Dragons in the Dec '87 issue, have by the fourth in the series been enhanced tremendously so that the end product is even more of a professional all-round package.

In *The Heir of Igo* in the tradition of the hero having an odd name, you are called as Krantz Ilanobu and live on planet Igo in the Xenopod system. The task is to find the smelter of Igo to stop a neighbouring King's siege.

You start when you awake at the top of a pyramid stage class by a teleporter and the number of locations increase as you progress down the four levels of the pyramid. In total there are over 200 locations, although despite my best and drastic attempts I have absolutely no idea how many.

Other locations include medical laboratories, where a nurse (obviously) lures you with a syringe, an apartment, covered balconies and a cafeteria. Plus there's a deserted 'city' where you can no longer exit and you are promptly sucked to your death.

The reason for my scepticism over this task being simple in this adventure is that instead of merely finding an object and next game going back there you have to search other areas for the item. Not being able to bypass previously unfruitful locations is due to the items are randomly placed.

Complex could be used to describe the commands. Give the syllings to the nurse after removing the needle being ministrations of desperation.

This is easily the hardest of the series, so give it a five.

Philip Scott



an island

Each year you can buy items like traps or fowling, one unit of each being sufficient for 500, don't buy enough and the natives will get restless. Also on the shopping list are various priced items like factories which provide work and earn extra money (each cut for black in Zooka labels), schools, hospitals and others also help keep the masses quiet for a while.

A small island is easily attacked. Though and to a defence is needed. For 75 units of money 50 soldiers are added to the starting number of 150. These men are certainly needed when foreign armies start attacking because if you allow 3 of their bases to be set up permission reluctantly forms and the gullies will be whetted out for you.

Clad wars are perhaps even worse because due to the spread, the massive rampage through the streets and fields destroying most of the hard work and effort.

Hurricanes have a similar effect, there's one at least every seven years (occurs like the bus service) which starts and travels randomly destroying all in its path (perhaps it is the bus service).

And that's about it. The games displayed on what has become almost the norm — that is the black/tefl screen, the island displayed graphically with red/white text messages underneath. If you survive your town you win if you don't you lose. There is a score feature but this is almost irrelevant in a live or die game.

For a brilliant game, or for that matter original but it is excellent value and is well programmed and displayed. Definitely one for the trained regimentation.

Philip Scott



Put another disc on the juke box, baby

Title: various
Supplier: Preston
Price: £4.99 each (Dragons disc)

A nice range of budget priced disc games has been launched by H. & J.J. Preston. The games are not new ones but this is the first time they have been available on disc at the low price of £4.99. Two game discs are reviewed here, the first is a compilation of three games, *Ruby Robe*, *Perilous Pit* and *Desperado Dan*.

An initial menu is displayed after BOOTING the Dragons disc. From this you can choose to play any one of the three games. It's not possible to go back to this menu, to change games requires powering off.

Ruby Robe is a curious mixture of the Connect 3 and Sliding Square puzzles. Using either keyboard or joystick you control a pointer which moves around the edge of a 10 by 10 grid of blue blocks. These blocks are either blank, or contain snakes or guards or a mystery bonus of between 100 and 300 points. The ruby is shown as a red block. Within the blank squares are mines which can kill the snakes and guards for extra points.

The blocks can be moved directionally using the fire (button or space bar). The object of the game is to push the square containing the ruby into a 'trap' at the top left of the screen. This is made more difficult than it sounds by the snakes and guards, which will return the ruby to the bottom right of the screen if they get near it.

Although this initially seems quite an original game, the challenge isn't high — there isn't really enough variety in the total of 18 difficulty levels. The object of the game is simple to grasp, but to capture the ruby requires some logical thinking — hence this game is ideal for the younger player.

Desperado Dan is a familiar platform and ladder type game — guide the little figure down the screen avoiding the flying bits, dodging the falling bits and jumping the stationary bits. *Desperado Dan* is not one

of the classics of its genre.

Perilous Pit is a simplified version of *Donkey Kong*. The slight difference is that diamonds must be collected one at a time and taken to the track at the top of the screen. Three diamonds are needed to move onto the next screen and there are four screens in all. An 'Orbit' appears on each screen (it looks like an old-style radio to me) which slowly rotates in on you once you start jumping. Moving platforms and other enemies must be negotiated.

The mode 3 graphics are fairly well defined, but do tend to get rather flickery at times (is this game machine coded or just compiled Basic I wonder?). If you specifically want a version of *Donkey Kong* then stick to Microcass's offering. As a third on this budget disc *Perilous Pit* is a nice choice.

None of the games on this disc are extraordinary in the world-of-100. However, as a set offering, and at a price of £4.99 they can be recommended to the gamemaster who can't wait for tapes to load.

The second disc I looked at contains just one game, *Kung Fu* — The Master. Again this is loaded by BOOTING the Dragons disc. The object of the game is to rescue your girlfriend (isn't it always?) who is, for some unexplained reason, being held captive. To do this you must get past screen after screen of 'troubles' (who look just like you), knives, snakes, bombs and fire-breathing dragons. The joystick is used to do all the usual things to start like run, jump, duck, fall, swing and various types of kick — all nice and varied! Points seem to be scored for hitting just about anything in sight.

The mode 3 graphics are quite well animated and flicker free with some nice horizontal scrolling included. The battles are also nicely defined.

This is probably one of the better budget games around for the Dragon at the moment.

Brian Gedge



Pamcodes

Pam D'Arcy continues her introduction to machine code

HOW far did you get with yellow blob? With my experience (grey hairs and all), I can't see a number of problems far-out. However, I shall try and put myself in your shoes — or at least, try and give you footage to follow it. My first move on a step-by-step basis, was to type in the source as it stood, and attempt to assemble. This is now where fail of this article may not be relevant to you, as your assembler may have accepted and assembled it all quite correctly (and I apologise for a just spotted correction in the comment column of line 00023 — it should have read .DT PB). For those with similar assemblers (mine, four lines of code were rejected — where three were values in the operand column that contained either alpha-numeric decimal letters, A-F, representing decimal values, 0-9).

This is because my assembler assumes a decimal value with digits 0-9 only unless it is told otherwise by preceding the value with a \$. Listing seven was derived at by assembling having inserted \$ signs after the \$ in the operand columns of instruction lines with addresses 00021, 00023, 00025, and 00027. It is, Pam obvious that the generated code (left hand side of listing seven) is quite different from the expected code in the back of 'comment' column on the right hand side of last month's listing. Line at 00021, for instance, is 000190 rather than 000400; the get keypress ZSR is 0001F0 rather than our familiar 000200. Generated object code is always shown in hexadecimal. As the required generated values are identical to the operands of the original source listing, the source must be

given in hexadecimal but not preceded by a \$ as required by my (Drew's) assembler. DECIMAL values have therefore been assumed excepting those rather mysterious instances where the sign and contained the digits A-F. If you experienced the same problem the next stage is to insert \$ as necessary throughout the source code where the value given in the operand column is identical to the required-generated object code. The resulting assembly is still likely to contain discrepancies because the calls to the UR, INDRW, MLPT and MRGHT subroutines use the extended mode-actual address type of instruction, .JSH and unless you assemble at the address the original is intended to run from, these addresses will be appropriate for the current assemble, as in listing eight. Apart from making the code position dependent — that is, it must be run from the location assembled for — but it not been for other extended mode instructions such as

STC 0400
CLP 0400

the code would execute correctly despite being run from a different position. However, the above makes it an entirely different proposition that can be tackled in a

There is no listing nine. We do not know why — so merely observe this fact.

0001					* LISTING 7
0002					
0003					* TRUSET 'P' (LAWARD)
0004					
0005					* THE YELLOW BLOB - PAGE 24
0006					* FROM "PAMON RECOVER CODE"
0007					* BY JAMES S. CORRELL, 19-07-81
0008					
0009					* TYPED IN AS PER TABLE LISTING
0010					* EXCEPT FOR A PREVIOUS COMMENT
0011					* DATA AND \$ INSERTS TO DEFINE
0012					* AN ERROR FREE ASSEMBLY
0013					
0014					* LISTED OPERAND ASSEMBLY
0015					* AFTER CLEARING SUBRONS
0016					
0017	0001F0	4404	LEI	040000	00 04 00
0018	001F00	0713	LDI	0400	00 04 00
0019	000001FF	0007	LEI	0401FF	04 00 01 FF
0020	043C	0009	LDI	0400	00 04 00
0021	001F00	0114	LDI	0400	00 04 00
0022	001F00	0114	CLR	0400	00 04 00
0023	0100	0114	STI	0400	00 04 00
0024	013F	0001	LDI	0400	00 04 00
0025	0274	0001	LDI	0400	00 04 00
0026	0400	0104	LDI	0400	00 04 00
0027	001F00	0114	LDI	0400	00 04 00
0028	0104	0114	STI	0400	00 04 00
0029	001F00	0114	LDI	0400	00 04 00
0030	0104	0114	STI	0400	00 04 00
0031	0104	0114	STI	0400	00 04 00
0032	001F00	0114	LDI	0400	00 04 00
0033	0104	0114	STI	0400	00 04 00
0034	0104	0114	STI	0400	00 04 00
0035	0104	0114	STI	0400	00 04 00
0036	0104	0114	STI	0400	00 04 00
0037	0104	0114	STI	0400	00 04 00
0038	0104	0114	STI	0400	00 04 00
0039	0104	0114	STI	0400	00 04 00
0040	0104	0114	STI	0400	00 04 00
0041	0104	0114	STI	0400	00 04 00
0042	0104	0114	STI	0400	00 04 00
0043	0104	0114	STI	0400	00 04 00
0044	0104	0114	STI	0400	00 04 00
0045	0104	0114	STI	0400	00 04 00
0046	0104	0114	STI	0400	00 04 00
0047	0104	0114	STI	0400	00 04 00
0048	0104	0114	STI	0400	00 04 00
0049	0104	0114	STI	0400	00 04 00
0050	0104	0114	STI	0400	00 04 00
0051	0104	0114	STI	0400	00 04 00
0052	0104	0114	STI	0400	00 04 00
0053	0104	0114	STI	0400	00 04 00
0054	0104	0114	STI	0400	00 04 00
0055	0104	0114	STI	0400	00 04 00

Listing 8					
0001	000400	4404	LEI	040000	00 04 00
0002	001F00	0713	LDI	0400	00 04 00
0003	000001FF	0007	LEI	0401FF	04 00 01 FF
0004	043C	0009	LDI	0400	00 04 00
0005	001F00	0114	LDI	0400	00 04 00
0006	001F00	0114	CLR	0400	00 04 00
0007	0100	0114	STI	0400	00 04 00
0008	013F	0001	LDI	0400	00 04 00
0009	0274	0001	LDI	0400	00 04 00
0010	0400	0104	LDI	0400	00 04 00
0011	001F00	0114	LDI	0400	00 04 00
0012	0104	0114	STI	0400	00 04 00
0013	001F00	0114	LDI	0400	00 04 00
0014	0104	0114	STI	0400	00 04 00
0015	0104	0114	STI	0400	00 04 00
0016	0104	0114	STI	0400	00 04 00
0017	0104	0114	STI	0400	00 04 00
0018	0104	0114	STI	0400	00 04 00
0019	0104	0114	STI	0400	00 04 00
0020	0104	0114	STI	0400	00 04 00
0021	0104	0114	STI	0400	00 04 00
0022	0104	0114	STI	0400	00 04 00
0023	0104	0114	STI	0400	00 04 00
0024	0104	0114	STI	0400	00 04 00
0025	0104	0114	STI	0400	00 04 00
0026	0104	0114	STI	0400	00 04 00
0027	0104	0114	STI	0400	00 04 00
0028	0104	0114	STI	0400	00 04 00
0029	0104	0114	STI	0400	00 04 00
0030	0104	0114	STI	0400	00 04 00
0031	0104	0114	STI	0400	00 04 00
0032	0104	0114	STI	0400	00 04 00
0033	0104	0114	STI	0400	00 04 00
0034	0104	0114	STI	0400	00 04 00
0035	0104	0114	STI	0400	00 04 00
0036	0104	0114	STI	0400	00 04 00
0037	0104	0114	STI	0400	00 04 00
0038	0104	0114	STI	0400	00 04 00
0039	0104	0114	STI	0400	00 04 00
0040	0104	0114	STI	0400	00 04 00
0041	0104	0114	STI	0400	00 04 00
0042	0104	0114	STI	0400	00 04 00
0043	0104	0114	STI	0400	00 04 00
0044	0104	0114	STI	0400	00 04 00
0045	0104	0114	STI	0400	00 04 00
0046	0104	0114	STI	0400	00 04 00
0047	0104	0114	STI	0400	00 04 00
0048	0104	0114	STI	0400	00 04 00
0049	0104	0114	STI	0400	00 04 00
0050	0104	0114	STI	0400	00 04 00
0051	0104	0114	STI	0400	00 04 00
0052	0104	0114	STI	0400	00 04 00
0053	0104	0114	STI	0400	00 04 00
0054	0104	0114	STI	0400	00 04 00
0055	0104	0114	STI	0400	00 04 00

number of different ways — see if you can understand at least one of them!

Addressed by \$D000 is used directly in operands and the value \$B04 is appears in the label returned the first instruction, one can assume that the code is intended to reside from \$B400 in memory, with \$B404 being the address of the first program instruction. Using Dream and assemblers with similar "redirection" facilities, listing 8 can be instantly adapted to match the original generated code by including the lines

```
ORG $B404
PUT $B001
```

before the first program line. The listing of this is not included as I am continually expanding my monthly space allowance! The generated code is then identical to the book and occupies \$B404-\$B458 inclusive. This can be saved using

```
CSAVE$F$BLO$B001$H$B001,$H$B051,
&H$001
```

and relocated for use with

```
CLCAD$F$BLO$B001,$H$400
```

Received from there with

```
CSAVE$F$BLO$B404$H$B404,$H$458,
&H$404
```

if existed. Remember that the program also uses locations \$B400-\$B403 as variable space, so a clear of at least CLEAR:200,\$H\$B400 is required before running the code.

A second option with the code should you not be able to generate it to execute it from \$B404 would be to leave it position dependent but to change the addresses of the variables space from \$B400-\$B403 to be within our chunk of code. This involves creating variables space using label names and an assembler directive that may be slightly different to your assembler (ie. RMB (Reserve Memory Bytes) in Dream. One can usually reserve any number of bytes using an RMB directive. To avoid having to totally understand the program logic when trying to adapt similar published listings to run on your systems, I suggest that you adopt the method of allocating a label name per possible variable address and substituting actual addresses in source code operands with its respective label name. That is, in the double byte units, sometimes as double byte units. Rather than have to sort out each case individually, give each byte a label then substitute each address with its label, which should result in a position dependent but working program as in listing 10. The code is saved using

```
CSAVE$F$BLO$B010$H$B001,$H$B051,
H$000
```

and is located in the data file that addresses only R R. The use of the third parameter in the CSAVE\$M — the default CXXG address which is not the first byte saved if including the variables data area \$B001-\$B004 in the saved chunk of code in any preference so that you don't forget about such areas when fitting several chunks of machine code close together in memory!

Another option is to make the entire chunk of code relocatable sorted if without then whenever we wish to load it at any one time. To achieve this is a relatively simple task if any actual memory addresses have been substituted with label names as in listing 10.

Extended address instructions need to be avoided if relocatable types. JSPs, as we have met previously, should be replaced with JSPs (Branch to Subroutine). To make references to variables relocatable, if they have been given label names as in listing 10, simply append PCRR (Program Counter Relative) to the operand — or your assembler's equivalent as this, again, may be an area that differs slightly and perhaps, PC rather than, PCRR is needed (Dream actually sorts with either). As with branch instructions that generate code to branch "Relative to the end of the current program instruction". Program counter relative access of variables causes code to be generated that refers to

Listing 10

```

$B001 MOVW R0B 1      ; F0R $400
$B002 MOVW R0B 1      ; $401
$B003 MOVW R0B 1      ; $402
$B004 MOVW R0B 1      ; $403
$B005
$B006 MOVW R0B 1,004  LDX  #B400
$B007 MOVW R0B 1,004  STX  #B400
$B008 MOVW R0B 1,004  MOVW #B400
$B009 MOVW R0B 1,004  MOVW #B400
$B010 MOVW R0B 1,004  CLR  #B00C
$B011 MOVW R0B 1,004  CLR  #B00C
$B012 MOVW R0B 1,004  STX  #B400
$B013 MOVW R0B 1,004  STX  #B400
$B014 MOVW R0B 1,004  STX  #B400
$B015 MOVW R0B 1,004  STX  #B400
$B016 MOVW R0B 1,004  STX  #B400
$B017 MOVW R0B 1,004  STX  #B400
$B018 MOVW R0B 1,004  STX  #B400
$B019 MOVW R0B 1,004  STX  #B400
$B020 MOVW R0B 1,004  STX  #B400
$B021 MOVW R0B 1,004  STX  #B400
$B022 MOVW R0B 1,004  STX  #B400
$B023 MOVW R0B 1,004  STX  #B400
$B024 MOVW R0B 1,004  STX  #B400
$B025 MOVW R0B 1,004  STX  #B400
$B026 MOVW R0B 1,004  STX  #B400
$B027 MOVW R0B 1,004  STX  #B400
$B028 MOVW R0B 1,004  STX  #B400
$B029 MOVW R0B 1,004  STX  #B400
$B030 MOVW R0B 1,004  STX  #B400
$B031 MOVW R0B 1,004  STX  #B400
$B032 MOVW R0B 1,004  STX  #B400
$B033 MOVW R0B 1,004  STX  #B400
$B034 MOVW R0B 1,004  STX  #B400
$B035 MOVW R0B 1,004  STX  #B400
$B036 MOVW R0B 1,004  STX  #B400
$B037 MOVW R0B 1,004  STX  #B400
$B038 MOVW R0B 1,004  STX  #B400
$B039 MOVW R0B 1,004  STX  #B400
$B040 MOVW R0B 1,004  STX  #B400
$B041 MOVW R0B 1,004  STX  #B400
$B042 MOVW R0B 1,004  STX  #B400
$B043 MOVW R0B 1,004  STX  #B400
$B044 MOVW R0B 1,004  STX  #B400
$B045 MOVW R0B 1,004  STX  #B400
$B046 MOVW R0B 1,004  STX  #B400
$B047 MOVW R0B 1,004  STX  #B400
$B048 MOVW R0B 1,004  STX  #B400
$B049 MOVW R0B 1,004  STX  #B400
$B050 MOVW R0B 1,004  STX  #B400
$B051 MOVW R0B 1,004  STX  #B400
$B052 MOVW R0B 1,004  STX  #B400
$B053 MOVW R0B 1,004  STX  #B400
$B054 MOVW R0B 1,004  STX  #B400
$B055 MOVW R0B 1,004  STX  #B400
$B056 MOVW R0B 1,004  STX  #B400
$B057 MOVW R0B 1,004  STX  #B400
$B058 MOVW R0B 1,004  STX  #B400
$B059 MOVW R0B 1,004  STX  #B400
$B060 MOVW R0B 1,004  STX  #B400
$B061 MOVW R0B 1,004  STX  #B400
$B062 MOVW R0B 1,004  STX  #B400
$B063 MOVW R0B 1,004  STX  #B400
$B064 MOVW R0B 1,004  STX  #B400
$B065 MOVW R0B 1,004  STX  #B400
$B066 MOVW R0B 1,004  STX  #B400
$B067 MOVW R0B 1,004  STX  #B400
$B068 MOVW R0B 1,004  STX  #B400
$B069 MOVW R0B 1,004  STX  #B400
$B070 MOVW R0B 1,004  STX  #B400
$B071 MOVW R0B 1,004  STX  #B400
$B072 MOVW R0B 1,004  STX  #B400
$B073 MOVW R0B 1,004  STX  #B400
$B074 MOVW R0B 1,004  STX  #B400
$B075 MOVW R0B 1,004  STX  #B400
$B076 MOVW R0B 1,004  STX  #B400
$B077 MOVW R0B 1,004  STX  #B400
$B078 MOVW R0B 1,004  STX  #B400
$B079 MOVW R0B 1,004  STX  #B400
$B080 MOVW R0B 1,004  STX  #B400
$B081 MOVW R0B 1,004  STX  #B400
$B082 MOVW R0B 1,004  STX  #B400
$B083 MOVW R0B 1,004  STX  #B400
$B084 MOVW R0B 1,004  STX  #B400
$B085 MOVW R0B 1,004  STX  #B400
$B086 MOVW R0B 1,004  STX  #B400
$B087 MOVW R0B 1,004  STX  #B400
$B088 MOVW R0B 1,004  STX  #B400
$B089 MOVW R0B 1,004  STX  #B400
$B090 MOVW R0B 1,004  STX  #B400
$B091 MOVW R0B 1,004  STX  #B400
$B092 MOVW R0B 1,004  STX  #B400
$B093 MOVW R0B 1,004  STX  #B400
$B094 MOVW R0B 1,004  STX  #B400
$B095 MOVW R0B 1,004  STX  #B400
$B096 MOVW R0B 1,004  STX  #B400
$B097 MOVW R0B 1,004  STX  #B400
$B098 MOVW R0B 1,004  STX  #B400
$B099 MOVW R0B 1,004  STX  #B400
$B100 MOVW R0B 1,004  STX  #B400

```

Listing 11

```

$B001 MOVW R0B 1      ; F0R $400
$B002 MOVW R0B 1      ; $401
$B003 MOVW R0B 1      ; $402
$B004 MOVW R0B 1      ; $403
$B005
$B006 MOVW R0B 1,004  LDX  #B400
$B007 MOVW R0B 1,004  STX  #B400
$B008 MOVW R0B 1,004  MOVW #B400
$B009 MOVW R0B 1,004  MOVW #B400
$B010 MOVW R0B 1,004  CLR  #B00C
$B011 MOVW R0B 1,004  CLR  #B00C
$B012 MOVW R0B 1,004  STX  #B400
$B013 MOVW R0B 1,004  STX  #B400
$B014 MOVW R0B 1,004  STX  #B400
$B015 MOVW R0B 1,004  STX  #B400
$B016 MOVW R0B 1,004  STX  #B400
$B017 MOVW R0B 1,004  STX  #B400
$B018 MOVW R0B 1,004  STX  #B400
$B019 MOVW R0B 1,004  STX  #B400
$B020 MOVW R0B 1,004  STX  #B400
$B021 MOVW R0B 1,004  STX  #B400
$B022 MOVW R0B 1,004  STX  #B400
$B023 MOVW R0B 1,004  STX  #B400
$B024 MOVW R0B 1,004  STX  #B400
$B025 MOVW R0B 1,004  STX  #B400
$B026 MOVW R0B 1,004  STX  #B400
$B027 MOVW R0B 1,004  STX  #B400
$B028 MOVW R0B 1,004  STX  #B400
$B029 MOVW R0B 1,004  STX  #B400
$B030 MOVW R0B 1,004  STX  #B400
$B031 MOVW R0B 1,004  STX  #B400
$B032 MOVW R0B 1,004  STX  #B400
$B033 MOVW R0B 1,004  STX  #B400
$B034 MOVW R0B 1,004  STX  #B400
$B035 MOVW R0B 1,004  STX  #B400
$B036 MOVW R0B 1,004  STX  #B400
$B037 MOVW R0B 1,004  STX  #B400
$B038 MOVW R0B 1,004  STX  #B400
$B039 MOVW R0B 1,004  STX  #B400
$B040 MOVW R0B 1,004  STX  #B400
$B041 MOVW R0B 1,004  STX  #B400
$B042 MOVW R0B 1,004  STX  #B400
$B043 MOVW R0B 1,004  STX  #B400
$B044 MOVW R0B 1,004  STX  #B400
$B045 MOVW R0B 1,004  STX  #B400
$B046 MOVW R0B 1,004  STX  #B400
$B047 MOVW R0B 1,004  STX  #B400
$B048 MOVW R0B 1,004  STX  #B400
$B049 MOVW R0B 1,004  STX  #B400
$B050 MOVW R0B 1,004  STX  #B400
$B051 MOVW R0B 1,004  STX  #B400
$B052 MOVW R0B 1,004  STX  #B400
$B053 MOVW R0B 1,004  STX  #B400
$B054 MOVW R0B 1,004  STX  #B400
$B055 MOVW R0B 1,004  STX  #B400
$B056 MOVW R0B 1,004  STX  #B400
$B057 MOVW R0B 1,004  STX  #B400
$B058 MOVW R0B 1,004  STX  #B400
$B059 MOVW R0B 1,004  STX  #B400
$B060 MOVW R0B 1,004  STX  #B400
$B061 MOVW R0B 1,004  STX  #B400
$B062 MOVW R0B 1,004  STX  #B400
$B063 MOVW R0B 1,004  STX  #B400
$B064 MOVW R0B 1,004  STX  #B400
$B065 MOVW R0B 1,004  STX  #B400
$B066 MOVW R0B 1,004  STX  #B400
$B067 MOVW R0B 1,004  STX  #B400
$B068 MOVW R0B 1,004  STX  #B400
$B069 MOVW R0B 1,004  STX  #B400
$B070 MOVW R0B 1,004  STX  #B400
$B071 MOVW R0B 1,004  STX  #B400
$B072 MOVW R0B 1,004  STX  #B400
$B073 MOVW R0B 1,004  STX  #B400
$B074 MOVW R0B 1,004  STX  #B400
$B075 MOVW R0B 1,004  STX  #B400
$B076 MOVW R0B 1,004  STX  #B400
$B077 MOVW R0B 1,004  STX  #B400
$B078 MOVW R0B 1,004  STX  #B400
$B079 MOVW R0B 1,004  STX  #B400
$B080 MOVW R0B 1,004  STX  #B400
$B081 MOVW R0B 1,004  STX  #B400
$B082 MOVW R0B 1,004  STX  #B400
$B083 MOVW R0B 1,004  STX  #B400
$B084 MOVW R0B 1,004  STX  #B400
$B085 MOVW R0B 1,004  STX  #B400
$B086 MOVW R0B 1,004  STX  #B400
$B087 MOVW R0B 1,004  STX  #B400
$B088 MOVW R0B 1,004  STX  #B400
$B089 MOVW R0B 1,004  STX  #B400
$B090 MOVW R0B 1,004  STX  #B400
$B091 MOVW R0B 1,004  STX  #B400
$B092 MOVW R0B 1,004  STX  #B400
$B093 MOVW R0B 1,004  STX  #B400
$B094 MOVW R0B 1,004  STX  #B400
$B095 MOVW R0B 1,004  STX  #B400
$B096 MOVW R0B 1,004  STX  #B400
$B097 MOVW R0B 1,004  STX  #B400
$B098 MOVW R0B 1,004  STX  #B400
$B099 MOVW R0B 1,004  STX  #B400
$B100 MOVW R0B 1,004  STX  #B400

```

Listing 10				000	040000	127 00	LEAD	-1, 0	120 07
LP	LEA	#00	080 00	LEA	#00	080 00	LEA	#00	080 07
	ORCA	#400	081 04 07	STL	, 1	087 04	STL	, 0	087 04
	000	00000	027 00	1AC	0400	070 04 07	ENGLPT	RTS	120
	LEA	#00	080 00	LEA	22, 1	070 00 20			
	STL	, 1	087 04	LEA	#00	080 00			
	000	00000	027 00	STL	, 0	087 04	FRONT	LEA	#17
	LEAD	-02, 0	080 00 00	000000	RTS	120	ORCA	0400	081 04 07
	LEA	#00	080 00	PLPT	LEA	#00	000	000000	027 00 07
	STL	, 1	087 04	ORCA	#400	081 04 07	LEA	#00	080 00
	00000	RTS	120	000	00000PT	027 00	LEA	1, 0	080 01
				LEA	#00	080 00	STL	, 0	087 04
	LEA	#10	080 00	000	0400	070 04 07	ENGLPT	RTS	120
	ORCA	#400	081 04 07						

the memory location (relative to) — that is, a fixed distance from. "The end of the current program instructions". The distance value is calculated for the program as it stands when assembled. If instructions are added or deleted between the variable's definition (the source line that gives it a label) and program instructions that refer to the label, *.FOR*, the re-assembler automatically calculates and generates the new distance value as part of the generated object code for that instruction. The length of code generated for instructions varies according to the instruction (numeric and operand). Unlike the standard (short) and long branch instructions, one doesn't need to differentiate regarding the distance in *.FOR* type instructions. The assembler generates a distance value in one byte for a variable within a range of +127-128 bytes from the start address of the next program byte, as within the examples, or two bytes for the

+127-128 range.

One final technique that I have included in the program is that I prefer the variables area to be able to be part of the program and for the BASIC address to be the same as the load address to overcome memory tapes — a condition often encountered when working under pressure. If using the new machine code routines from Basic programs, I also like to have a variables area at a fixed place near the beginning of a machine code program so that values (parameters) can be *PEEK/POKE*d easily between the Basic and machine code. I therefore like to start off machine code programs with *LEA* (long branch) to the next beginning of the program followed by a variables area. I use a long branch even where a short one would do so that I always know, regardless of the length of the variables area, that my Basic/machine code parameter areas will always start at a fixed point. — program, too/over

address +3 rather than have some program parameter areas commencing at +2 (the length of a short branch instruction) and others at +3, three being the length of a long branch instruction. Although Dream has not thrown it out (probably because nothing changes to use it), Dream's label names should commence with an alphanumeric character rather than the 0404 appearing at the beginning of the original program. I have therefore replaced this with the label name *00* for listing 11.

Monthly workout

I suspect that, with the number of listings this month, it is tough and close to feather the final listing makes it into print — but I'll chance it. Listing 10 is the cursor movement routines as they appear in the book. By incorporating them, in relocatable form, in listing 11 (or your equivalent),

Crossword

The fourth Dragon Crossword is here to perplex you with perplexing puzzles from the Dragon's publishing faculty. And we have the results from crossword two: the fortunate few, picked out from the rest, were Paul Harrison of Bristol, whose entry arrived on the back of what looks like an electronic circuit for a single LED, and Jack Lund, who wants a surprise. Surprises, we are good at.

There will be a couple of free tapes from the Editor's Magic Bottomless Box for the first correct entries to reach us each month. You can win by telling us which tapes you'd like in an e-mail world. It all depends on what we can find.

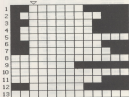
And you don't have to cut up your Dragon User, either — entries can be written out on a photostat or a plain piece of paper, as long as we can read them. And as long as there are no official secrets on the tape!

1. It's very hard to say — almost sounds like a party mask! (6)
2. Mix mud on mud to build a person up (8,3)
3. Cook, usually goes with a leap! (7)
4. A learner they tax for sport (8)
5. Yusuf brews an exotic mixture to save the Jolly Plumber! (12)
6. Birdman lived here (8)
7. Sounds like he only steals red stones! (4,5)
8. Three stories of glory it took to (3,1,7)
9. Head your horses — the Mexican robber has nothing on! (7)
10. Does he kill ghosts? (7,8)
11. Team up with this steel fender! (4,8)
12. Get away to do this (8)
13. Nothing is circus-tricks to take you to the stars! (6,7)



by Terry and Derek Probyn

All this month's answers are names of Dragon software. When the crossword is complete, the column marked with an arrow will spell out a phrase.



Total Profits

Alpha Centauri School of Cost Effective Plundering graduate Julian Brown has some tips for traders.

EVEN after all this time many of you out there are still struggling, trying to make a living out of the much simplified trading, while people like me have made multiple visits to the second universe. So out of the kindness of my empty bank account I have decided that all you sufferers out there need my help badly. (Anyone who needs a living writing for Dragon User must help me). — Ed.)

I have been limited on space, so this will have to be brief, so listen good!

Here are the basic tips:

1. Avoid asteroids by orbits, unless you want to be used for target practice.
2. Keep your ship in a good orbit of repair at all times.
3. Never pay the pirates.
4. Know where the repair centres are (see table one).
5. Make sure you have a good trading route.

you wished to narcotics, ethics aside, this is more efficient than it sounds. Things to get narcotics is nearly impossible. Tailoring the amount you collect is easy and best illustrated with by table two.

Table two

Number of holds	Maximum no. of cargo units
1	2
2	4
3	6
4	8
5	9

When you have the maximum number for your hold, regardless of whether it is food, textiles, etc., you can't collect any more. Better adding other cargoes to the total in the

which increases the cash you gain per trip if you use a full cargo hold. Follow this up with fuel tanks, then lasers, then computer expansions.

Once you have all of these buy the drilling kit, eight antennas and a hyperspace. With all this done, you are on your way.

Drilling

Find an asteroid with a reserve of at least 10 and fire its weakest point in the crust. Using a HEH drill, start drilling, stopping every time the drill bit reaches about 40 degrees, and allow it to cool. Once the asteroid is full then stop the well.

Go to a space station you won't find one in the same galaxy and sell your load. Once this is done, it's worth waiting a few minutes while the value of asteroid goes back up again. And get yourself a new drill.

The Prophet

Once you have collected cash in the region of a few billion credits you should start to receive telegrams from the prophet. It is Phelan and you will receive a shard. All that remains is to fly down to the nearest black hole.

Universes two is a really place. Everything costs far more so much, there are dead ships to be salvaged, and scores of alien megaprograms don't work because of a bug in the program. Last but not least, even though the names of the cargo items don't change, you'll have to find new initials for them, except the fumes. The names are supposed to change, but another bug in the program prevents it.

Finally, if you'd get stuck and need continual help I can supply a completed version of my editor program that runs on ALL Dragons for 25 plus 75p for postage and packing. Just send your request to: Julian Brown, 2 Lavender Cottages, Galvanic Lane, Freshwood, Bucks HP16 6PY. Don't forget to include your own address, and don't forget the money, or you may not receive anything!

Table one

PLANET	COORDINATES	PLANET	COORDINATES
JUPITER	1:1:1:5	ZEUS10	1:1:9:3
ISABELA	3:1:4:2	ZEUS5	2:1:1:3
MAJITTA	2:1:4:4	NAKIBY	3:1:3:6
THOR1	4:1:1:1	KORR1	4:1:1:3
KORR2	4:1:1:5	FLACRA	5:1:2:2
GAFFA (asteroid)	5:1:1:2	WIKES	5:1:1:4
MAJITTA	5:1:1:5	LARDE	5:1:1:5
POJIB	6:1:1:3	ZOVICH	6:1:2:6
ZOVIEA	7:1:4:3	LIARBA	6:1:1:1
WHICE	6:1:1:4	ATTOB1	6:1:2:6
ZOVICO	6:1:1:6	TTAEP1	1:1:1:2
TTARBE	10:1:1:5		

The format for the co-ordinates is sector by arc. All of the above have repair facilities but some are cheaper than others. Scarfa is the most expensive planet and is best avoided at all times.

The recommended trade route is between Tavee (7:1:5) and Sraol (8:2:5). Buy uranium at Tavee and sell it at Sraol (but don't buy anything for the return journey).

Technicalities

Basic planets that have landing taxes are best avoided unless you know what you are doing. If you do visit them make sure it is well paid.

At the end of each combat you automatically collect cargo left by the pirates. This cargo is always a collection of three items: plants, narcotics and narcotics. With no load and one cargo hold you collect two items; one cargo hold you buy increases this number to a maximum of six; after this the number reverts to zero and then two again.

Until you start to collect six or more items

hold is more than the maximum listed above, you will receive extra.

For example, if you had one hold with a single unit of food, you would collect 1 unit of booty, but if you had instead 3 units of food you would collect 7 units of booty.

It is best to tailor the results so that you receive 8 units of booty, which is the maximum safe number.

For example, if you have 1 hold, carry 5 units of cargo, if you have two holds carry either 16 or 9 units of cargo, with three holds carry 1, 11, 16 or 25 units of cargo.

Don't forget always pay import taxes and avoid narcotics.

Upgrades

If you take my advice and start trading between Tavee and Sraol then upgrades will be readily available to you. Don't be tempted to buy extras for your ship as soon as you have enough money, wait till you have about twice the amount of money you need, in case of disaster.

Always expand your cargo hold first,



READPC

Martin Wornear devises a program to read PC discs on a Dragon

A program to read DragonDOS diskettes on a PC has been available from CompuShare for some time now. One of my friends who came with a Dragon and PC is very happy with it: it is well made and easy to use. Personally however I have been much more interested in software to do the reverse: transfer PC software in Basic, Pascal and C to run on my little Dragon.

The problem

More than a year ago I was faced with a professional need to get a number of large files from a PC diskette into my Dragon. The solution I chose was to purchase the PC-READ software from D.P. Johnson — an American professional OS-9 programmer — which is made for the Color Computer and also runs on the Canadian-made Sanyo SC-2900. The latter machine is the one used by a friend of mine — he built it himself from the circuit boards — and asked him to transfer my files to Color Computer OS-9 format diskettes, which the Dragon can read.

A little complicated, though, I suppose there must be many Dragonists who would like to use some of the supposedly abundant PC software on their home machines, or who use a PC at work and have made programs and texts for it which they would like to use at home also. There might be even frustrated PC owners fascinated by the idea of upgrading to the Dragon's superior operating system, but held back by the difficulty of transferring their materials. Anyway, those days are over now, as the following programs allow for the easy and quick transfer of even large ASCII files — text, software source codes, whatever — from MS-DOS diskettes to either DragonDOS or OS-9.

Using READPC

Accompanying this text is a Dragon Basic listing of a program which will read an MS-DOS diskette, present you with a directory and transfer the file chosen by you by number to ... a cassette tape, as a Dragon Basic Ascii file. This unconventional solution was chosen (i) to allow single-diskette access to use it and (ii) not to place any limit on transferable file sizes.

LOAD the program and RUN it; wait until the MS-DOS directory appears on the screen. Choose the file to be transfer exactly number and press ENTER. Make sure the tape recorder is connected and ready and an empty cassette lined up and ready to record, and both Play and Record buttons down. Especially with large files, the transfer may take some time, so be patient.

If case file directory is larger than the screen, you can stop the listing by pressing the shift/F10 key. You can repeat the directory listing by just pressing ENTER, or by specifying the number 8. A limitation of this program is that only basic files in the MS-DOS root directory.

Be careful if you want to transfer text files in a non-British European language: the special characters which occur in Spanish, German and Swedish/Finnish are treated by MS-DOS in a way which violates the Ascii standard. Where Ascii has those characters in the same places as the British or US Ascii set has square brackets, braces and/or the like, IBM has been creative in using 8-bit codes instead. All 8-bit codes are replaced by READPC with question marks. So are braces and other characters that the Dragon's cassette I/O system does not recognise. In case you want those transferred — important in for example C source code — you should either replace them on your PC, or be prepared to correct them afterwards on your Dragon.

The above program puts your file to cassette as an Ascii file, which can be read by word processors like Microsoft Alternatively, a Basic program can access the file as a data file by using the well known LINE INPUT \$: statement, which reads exactly one line (direct from carriage return to carriage return) from the file into a single text string.

If your text happens to be a Microsoft Basic program, you can try to load it straightaway into the memory and edit it by the Dragon's own line editor. It is likely however that it needs some cleaning up before this works.

Getting to OS-9

To transfer the files to tape further to OS-9 requires additional software written to transfer Ascii-type files in general to OS-9. In the building block approach the OS-9 philosophy is additive! What I did was simply write an OS-9 device driver for the Dragon's cassette recorder; it treats the device as a sequential character file which is only capable of input. With the driver goes a descriptor, a little table of values defining the settings of the peripheral device. I obtained both by disassembling and suitably patching the ACSM1 serial port driver and its I/O descriptor.

Perhaps these programs could be streamlined a little more, but they have been tested and work as they are in order to use them, assemble these source codes and let them reside in your tapeless memory. Ready your cassette, BOOT OS-9 (always before every transfer) and load the

modules CASS, CR, and COPY or LIST. To get your file on the screen, you would write:

```
OS9:tel for (ENTER),
```

but presumably you are more interested in file transfer, the work with COPY is as follows:

```
OS9:copy for pathname (ENTER).
```

Cleaning up

When listing the file received, you will generally notice that it contains besides carriage returns (Ascii 10) also line feeds (Ascii 13). This is typical for MS-DOS Ascii files, but the Dragon wants only carriage returns. A second problem is that the cassette I/O system has generated spurious carriage returns in the file.

Both problems are corrected by the following little program T, which is written in BasicOS. Write it in, SAVE it, PACK it and use it together with RUN it, as follows:

```
OS9: (code) >write (ENTER),
```

which will produce a new file with the proper amount of carriage returns. At the end of the file there will still be some junk however, but that is easily edited out. READPC does not check for the precise file length of an MS-DOS file.

Nonstandard formats

I have succeeded in reading at least the 700K 3.5in MS-DOS diskettes used in many portable machines by the following modifications: 1) change the parameter DISKPART inlined to 73) change line 120 to:

```
100 FOR L=0N+P0 TO P0+2
```

For other formats, other parameters should be changed (number of sides, number of sectors per track), but I have no experience with this. There are plenty of good books on MS-DOS disc structure (in public libraries, for example) to help you, as well as the excellent (and extremely boring) Horton's Utilities.

You could also generalise a general-purpose disc browsing utility from the READPC sector read routine. When doing so, be aware that this routine is (nearly) general-purpose, and not limited to sector sizes of 256 bytes, but reads any sector size. Unfortunately, the sector read is capable density only, but even this limitation can be removed: the Dragon hardware supports single density disc I/O and only waits for the programmer to use it.

Listing 1

LISTING 1: SEARCH FOR DEVICES.

```

10 PRINT "Exit..." : GOTO 100 : END : REM *****
20 DIM name$(255),type$(255),start$(255),end$(255)
30 FOR device=0 TO 255 : GOTO 40 : NEXT device
40 DIM start$(255),type$(255),end$(255)
50 name$="*****" : type$="" : start$="" : end$=""
60 FOR i=0 TO 255 : GOTO 70 : NEXT i
70 name$(i)=CHR$(i) : type$(i)="*****" : start$(i)=0 : end$(i)=255
80 FOR i=0 TO 255 : GOTO 90 : NEXT i
90 name$(i)=CHR$(i) : type$(i)="*****" : start$(i)=0 : end$(i)=255
100 PRINT "*****" : GOTO 100
110 PRINT "*****" : GOTO 100
120 PRINT "*****" : GOTO 100
130 PRINT "*****" : GOTO 100
140 PRINT "*****" : GOTO 100
150 PRINT "*****" : GOTO 100
160 PRINT "*****" : GOTO 100
170 PRINT "*****" : GOTO 100
180 PRINT "*****" : GOTO 100
190 PRINT "*****" : GOTO 100
200 PRINT "*****" : GOTO 100
210 PRINT "*****" : GOTO 100
220 PRINT "*****" : GOTO 100
230 PRINT "*****" : GOTO 100
240 PRINT "*****" : GOTO 100
250 PRINT "*****" : GOTO 100
260 PRINT "*****" : GOTO 100
270 PRINT "*****" : GOTO 100
280 PRINT "*****" : GOTO 100
290 PRINT "*****" : GOTO 100
300 PRINT "*****" : GOTO 100
310 PRINT "*****" : GOTO 100
320 PRINT "*****" : GOTO 100
330 PRINT "*****" : GOTO 100
340 PRINT "*****" : GOTO 100
350 PRINT "*****" : GOTO 100
360 PRINT "*****" : GOTO 100
370 PRINT "*****" : GOTO 100
380 PRINT "*****" : GOTO 100
390 PRINT "*****" : GOTO 100
400 PRINT "*****" : GOTO 100
410 PRINT "*****" : GOTO 100
420 PRINT "*****" : GOTO 100
430 PRINT "*****" : GOTO 100
440 PRINT "*****" : GOTO 100
450 PRINT "*****" : GOTO 100
460 PRINT "*****" : GOTO 100
470 PRINT "*****" : GOTO 100
480 PRINT "*****" : GOTO 100
490 PRINT "*****" : GOTO 100
500 PRINT "*****" : GOTO 100
510 PRINT "*****" : GOTO 100
520 PRINT "*****" : GOTO 100
530 PRINT "*****" : GOTO 100
540 PRINT "*****" : GOTO 100
550 PRINT "*****" : GOTO 100
560 PRINT "*****" : GOTO 100
570 PRINT "*****" : GOTO 100
580 PRINT "*****" : GOTO 100
590 PRINT "*****" : GOTO 100
600 PRINT "*****" : GOTO 100
610 PRINT "*****" : GOTO 100
620 PRINT "*****" : GOTO 100
630 PRINT "*****" : GOTO 100
640 PRINT "*****" : GOTO 100
650 PRINT "*****" : GOTO 100
660 PRINT "*****" : GOTO 100
670 PRINT "*****" : GOTO 100
680 PRINT "*****" : GOTO 100
690 PRINT "*****" : GOTO 100
700 PRINT "*****" : GOTO 100
710 PRINT "*****" : GOTO 100
720 PRINT "*****" : GOTO 100
730 PRINT "*****" : GOTO 100
740 PRINT "*****" : GOTO 100
750 PRINT "*****" : GOTO 100
760 PRINT "*****" : GOTO 100
770 PRINT "*****" : GOTO 100
780 PRINT "*****" : GOTO 100
790 PRINT "*****" : GOTO 100
800 PRINT "*****" : GOTO 100
810 PRINT "*****" : GOTO 100
820 PRINT "*****" : GOTO 100
830 PRINT "*****" : GOTO 100
840 PRINT "*****" : GOTO 100
850 PRINT "*****" : GOTO 100
860 PRINT "*****" : GOTO 100
870 PRINT "*****" : GOTO 100
880 PRINT "*****" : GOTO 100
890 PRINT "*****" : GOTO 100
900 PRINT "*****" : GOTO 100
910 PRINT "*****" : GOTO 100
920 PRINT "*****" : GOTO 100
930 PRINT "*****" : GOTO 100
940 PRINT "*****" : GOTO 100
950 PRINT "*****" : GOTO 100
960 PRINT "*****" : GOTO 100
970 PRINT "*****" : GOTO 100
980 PRINT "*****" : GOTO 100
990 PRINT "*****" : GOTO 100

```

Listing 2

```

1
2
3
4 LISTING 2: Cass Dragon cassette read driver
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

THE MOST

POWERFUL

SOFTWARE

AVAILABLE

FOR THE

IBM PC

AND

XT

COMPUTERS

FOR

YOUR

NEEDS

CALL

1-800-

428-2222

IBM MICRO SOFTWARE

10000 W. Higgins Road, Roseville, IL 60018

How green is my black

Paul Reid gets bored with black on green, and invents an inversion.

The video display generator (vdg) chip in the Dragon, the one that dictates that we are stuck with a really rather bland black-on-green text display and plain screen border, is the Motorola MC68477, located on the main board near (on the Dragon) the cassette and joystick sockets.

Consulting the specification sheet for this particular chip, I discovered that pin 32 can be used to invert the screen colours in Alpha (text) mode. This is done by grounding pin 32 "high" at +5 volts. As with most mods however, there is a trade-off: the normal inverse video characters, the lower case set, are not generated in this way and are unaffected by the mod. This has both advantages and disadvantages. On the plus side, a BASIC program listing on the screen which includes lower case text intended for output to a printer, will appear as "normal" text - but inverted along with everything else, rather than as blocks of inverted characters which if they have upper case characters mixed in with them, makes the whole thing very difficult to read. On the minus side, it is difficult at times when inputting lower case text in the inverted state, to know whether you are in upper or lower case mode!

Not Ascii

A general trade-off is, that because the cursor is not an Alpha ascii character, it does not invert. Instead, it becomes a flashing black graphics block on a black background - somewhat difficult to see!

Nonetheless, these apparent problems can be overcome. The essence of this modification is that it is switchable - even during the running of a program, so that if confusion exists, simply by switching back to normal video, the display instantly returns to the original format. Moreover, the problem of the "missing" cursor can be overcome by replacing the graphics block with a hyphen or other Alpha character using the routine published in OJ a while ago (OJ - how about reprinting the item referred to?) (at the time of writing I still haven't managed to identify it - Ed).

The advantages of using this method to achieve inverted display are considerable when it is considered how much BASIC programming is required to achieve inverted spaces and punctuation marks in print statements!

Here then, are the instructions for carrying out the mod which requires only the careful use of a soldering iron, preferably low wattage with a miniature tip, a drill for making the hole in the case in which to fit

the switch and the switch itself, a single pole double throw (spdt) type along with three lengths of insulated wire each about 40cm long. Switch off and disconnect the power supply.

I discovered that pin 32 can be used to invert the screen colours in Alpha (text) mode

(see Figure 1). On some boards, the vdg is fixed in a socket, if this is so with yours, remove the chip taking care to bend the legs. An IC fitting/removing tool should be

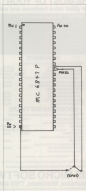
After separating the two halves of the Dragon's case (by removing the four screws located one in each corner of the base (well, there may be some Dragon owners who haven't been inside theirs yet!), drill a suitable size hole in the most convenient, accessible place for the switch, remember that you will want to be able to operate it while using the computer. Identify pin number 32 of the vdg chip and also pin 30 used if possible. Bend pin 32 outwards so that it is at 90 degrees to the other pins and solder one end of the three lengths of wire onto it. The other end of the wire should be soldered to the centre (common) pin of the switch. Next, solder one end of the second piece of wire to the upper wide part of pin 30, taking care not to touch the pin or to get solder on the thin lower part. The other end of the wire should be soldered to either of the two remaining pins of the switch. The third length of wire must be soldered between the remaining pin on the switch and the socket which pin 32 normally fits into. If on your Dragon, like mine, the vdg chip is soldered directly to the board, pin 32 should either be de-soldered (using bread or a pump/heat the underside of the board) and carefully pulled back through the board, or cut with a pair of side-cutters so close to the board as possible.

If it was removed, the vdg chip can now be refitted, taking care that pin 32 does not make contact with the wire manipulated in its place. Fit the switch to the case in the drilled hole and reassemble the case. Reconnect the power supply, monitorily lead and switch-on. Don't worry if the display is normal, operate the new switch and see what happens!

Ten minutes

The next invention will in no way affect the computer's operation, or necessitate any program changes, but if like me you are tired of that black-on-green display, it makes a refreshing change! Even if you are not a wizard with the soldering iron, provided you are careful not to use too much solder, you should have no problems and the results you will see after 10 minutes or so to do the modification and the cool, which is only a few pence.

Although this mod has been tried and tested, as usual, the author and Dragon User can accept no responsibility for any problems arising from the modification. We also suggest that you don't attempt a modification, even a simple one, inside your computer unless you are experienced at construction.)



Expert's Arcade Arena

Write to "The Expert" at Dragon User
12, Little Newport St
London WC2H 7PP.

HELLD and welcome. Folks to start this month's column with friendly POKE which makes all games load four times as fast, operate with infinite lives and play that national anthem. Unfortunately I can't. But I've got lots of other goodies for you instead, including the first of the answers to the questions which I set in January. Please keep your letters coming in, especially those entries in the software survey. At present, *Steelrunner* and *James's Revenge* are the hottest contenders for the all-time best arcade game, and Ed Scio has received over 75% of the votes and looks like winning the title of best programmer. Thanks, as dear father would say, unless you know better!

And so to business. As promised, here's some more from the one and only James Bonfield. Mr Goodwoodson provides us with a new POKE for Pitfall from Pocket Money

Pitfall cheat

Load the game using program C (like Dragon User September 1987) compiled with 30:CHANGE maze.

30:POKE792L:POKE792A
where A is a number 0-255
(about 3)

Unfortunately, I'm having a lot of trouble with this one. As

James himself said, "It's a buggy experiment". And now for another game arriving from James, the first messages in *Copie Smash* (but a good question, "did he play the game to get them?"). Over to you, James.

"In *Copie Smash*, when you land on the last platform, the screen displays:

RETURN TO HQ TO
COMPLETE MISSION

On returning, the field message is:

MISSION COMPLETED

It also says that while this is displayed, the computer plays a tune and 'springs' things fly around the screen, but if you ask me it's only a little recognition for such a mammoth effort needed to complete the game!

James sent me a few maps, the *Mapage* map being particularly interesting, but unfortunately we weren't able to print it. I've listed below a few points to remember if you draw a map and would like us to consider printing it. No more *Copie Smash* maps please!

1. Always use plain paper. If you need squares then use a sheet of graph paper or squared paper underneath your plan and it will show through.
2. Always use a black pen, and

preferably not something which bleeds, smudges, or goes (sorry *Colours* won't show up).

3. Try to ensure that lettering and details are really clear.

4. Maps should be large enough to be clear but preferably no larger than A4, and lettering should be of a readable size.

These technicalities should enable us to bring your efforts to the world if the opportunity arises.

And now for *Total Eclipse* fans (like up-Joe Briscoe, are you receiving me?), two loyal servants, Peter Macmillan and David Linsley, have replied to question 2 of January's pleix, and I am pleased to announce that *Univers* it does in fact exist. Peter informs me that it's a crazy world in which you trade in *Chalk Dust*, *Blue Socks*, *Fog* and *Kangeroos*. (What? I can't read.)

He also offers a helping hand to those who have *Univers* it and want to know how to play it — he thinks people are in such a situation, but I don't know why!

Thank you Peter and David. But the way David, I'm not any of the people you suggested, especially Kenny Everett!

AND NOW THIS ... I must give you a Wizard's Quest POKE, so no-one else sends it in. I've had so many people send it to me that I've forgotten both their names (years). So

give credit where credit is due, thank you Tom Wilkinson, James Bonfield, Michael Dunn and Sean Hoare.

Wizard's Quest cheat

Load using program C, compiled with:

20:POKE2649A:0

This gives infinite lives.

And now, with most of the official business out of the way, you and I must have a serious talk. We shall and pull your Dragon User up closer because I want a private word.

Now, I don't ask much of you folks, so you could at least grant me this one, my request. NO MORE CHEVY PDS NAMES PLEASE! Yes, you, I refer to you, of course, Godfrey the Incredible Games Playing Godfist. I don't want to be hearing any more from you under this name as I get very angry and disinvite your carp pen name, so on your grief for those interested, this filthy fellow, who sent me a most unusual letter, is in fact Simon Harrison, so now that you know his real name you'll never take him seriously again!

Not again! I've reached the end of another page, so I must sign off. Keep your letters coming in, and lend your eyes next month for some help of *Catecomb-Crabs*.

Communication

Problem: I am finding difficulty in obtaining floating joysticks.
Name: Scott Hadden
Address: 336, Clancy Place, Glenrothes, Fife KY7 4DN

Problem: Need a printer cable for MCP-80 printer/paper.
Name: Duncan Chambers
Address: 30 Park Avenue, South Shields, Tyne & Wear

Problem: I need an Edi Plus cartridge for Dragon 3264.
Name: W A Slater
Address: 44 Hopes St., Grantham, Cheshire, Derbyshire S40 1DG

Problem: Wanted: one Dragon computer for an unemployed person. Accompanied for a youth club — joystick in

any condition for the Dragon. Can repair if broken.
Name: Dennis Gates
Address: 184, Bee Road, Newton Hall Estate, Durham

Problem: I am looking for an original copy of *Arise* as Mayfair by Datacom. I will pay £4.00 for it.
Name: P MacMillan
Address: Rose Cottage, Oakwood Road, Sling, Nr. Oxford, Oxon. OX18 8JG

Problem: Wanted: one based word processor, *Writer* or *Sandy* (except, suitable for Dragon 32. Write to me.
Name: Andy Fell
Address: 17 Gorse Farm Road, Thornhill Park, Mutton-Croft BH.

Communication

Write down your problem on the coupon below, make it as brief and legible as possible together with your name and address and send it to Communication, 1293 Little Newport Street, London WC2H 7PP

Problem

.....

.....

.....

Name

.....

Address

.....

.....

Down in the dumps

Dragon User presents another screen dump

Seikosha GP100A

This basic program can dump the contents of the Dragon's basic screens 0, 2 and 4 directly to the Seikosha GP100A printer in one of two sizes (about 4in by 3in) and large (just under A4 size).

After the program has been typed in correctly and run, it is simple to use. Full instructions are in the program, but simply enter the PRMODE which the screen in question was saved in, the size of the

dump, and various other minor details, and the screen file, saved in the normal way (created by an art program, or from a game screen, etc.), is then loaded and dumped. It is worth pointing out that some utilities save screens by different methods. By one method, each a screen has been loaded it "reases" any program in memory. The program allows for this simply load in your screen using LOADM beforehand, then the dump program, and then choose option 2 from the menu.

This program is also available on tape from myself, priced £2.00 (UK). Cheapest PC saveletters Andrew Hill, 10 Perry Jones Close, Balmie, Gwent NP23 5RH.

Note: the Seikosha GP100A uses an unusual method of graphics printing, it needs 7 bits at a time instead of the more usual 8.

PRMODE 1 and 3 colour screens can only be printed if you change the PRMODE to 2 or 4 so that they use monochrome.

Andrew Hill

Dump C

```
100 REM *****
200 REM * - SCREEN DUMPER - *
300 REM *****
400 REM * DO COPYRIGHT BY *
500 REM *****
600 REM * WRITTEN BY A.HILL *
700 REM *****
800 CLEAR 255
900 CLR
1000 PRINT "-----")
1100 PRINT "XXXXXXXXXXXXXXXXXXXX")
1200 PRINT:PRINT "XXXXXXXXXXXXXXXXXXXX")
1300 PRINT "-----")
1400 PRINT "WELCOME TO 'seikosha screen dumper', A PROGRAM WHICH ALLOWS YOU T
1500 "O DUMP THE CONTENTS OF YOUR HI-RES SCREEN (0/2/4) TO THE PRINTER."
1600 PRINT:PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
1700 PRINT " - PRESS THE (SPACEBAR) NOW -"
1700 IS=INKEY
1800 IF IS=CHR$(32) THEN GOTO 170 ELSE PLAY FOR
1900 FOR HOLD=1 TO 1000:NEXT HOLD
2000 CLR
2100 PRINT:PRINT " 111 = YOU REQUIRE INSTRUCTIONS."
2200 PRINT:PRINT " 221 = YOUR HI-RES SCREEN HAS ALREADY BEEN LOADED IN."
2300 PRINT:PRINT " 331 = YOU DO NOT REQUIRE THE INSTRUCTIONS."
2400 PRINT
2500 INPUT " WHICH (1/2/3) - ";I$
2600 IF INSTR("1") AND INSTR("2") AND INSTR("3") THEN GOTO 290
2700 IF INSTR("1") THEN FOR HOLD=0 TO 1000:NEXT HOLD:GOTO 310
2800 IF INSTR("2") OR INSTR("3") THEN FOR HOLD=0 TO 1000:NEXT HOLD:GOTO 310
2900 REM INSTRUCTIONS FOR
3000 REM SCREEN DUMP
3100 CLR
3200 PRINT:PRINT "-----")
3300 PRINT:PRINT "WELCOME SEIKOSHA DUMPER":PRINT:PRINT "-----")
3400 PRINT:PRINT "331" - instructions -"
3500 PRINT:PRINT "331" -"
3600 PRINT:PRINT " TO USE S.L.S., ALL YOU HAVE TO DO IS TO LOAD IN YOUR HI-RES
3700 " SCREEN FROM TAPE WHEN PROMPTED, "
3800 PRINT " SET UP (WITH PAPER READY) AND SWITCH ON YOUR PRINTER AND GAB LET "
3900 " PROGRAM DO THE REST!"
4000 PRINT:PRINT " - PRESS THE (SPACEBAR) NOW -"
4100 IS=INKEY
4200 IF IS=CHR$(32) THEN GOTO 290
4300 PLAY FOR
4400 FOR HOLD=0 TO 1000:NEXT HOLD:CLR
4500 PRINT "XXXXXXXXXXXX")
4600 PRINT "seikosha screen dumper"
4700 PRINT "XXXXXXXXXXXX")
4800 PRINT:PRINT " IF THIS PROGRAM IS NEEDED AGAIN, THE HI-RES SCREEN HAS LOADED,
4900 " FOLLOW THE INSTRUCTIONS (1/2/3) ON THE INSTRUCTIONS."
5000 PRINT:PRINT " - PRESS THE (SPACEBAR) NOW -"
5100 IS=INKEY
5200 IF IS=CHR$(32) THEN GOTO 290
5300 PLAY FOR HOLD=0 TO 1000:NEXT HOLD
5400 REM ** MAIN PROGRAM **
```


Dragon Comms

Martyn Armitage introduces communication for the Dragon

SO, you've got a Dragon 64 and the only socket that you've probably never used is the one marked up Serial 1/5, the RS232C interface to give it its real title. There are two main uses for the interface. 1) for connecting a serial printer, and 2) for connecting a modem. Most of what is said in this article will be regarding the use of the port as a communications interface, along with a modem, but most of what is written will also hold true for the connection of a printer. To start off with I shall attempt to put into layman's English some of the most commonly used jargon that is associated with the RS232C interface, such as parity, baud, etc. First we'll start off with protocols, which itself can, and will, be split into sections.

Setting standards

Protocol can best be described as communications ' etiquette', and is simply a set of standards that one should adhere to if you wish to get anything other than garbage on the screen (paper if you're using a printer). As I have already said, I am going to break this heading down into the various parts that it consists of.

PARITY is a very simple method of error detection on received data. When in the communications world you will hear the phrases 'odd parity' and 'even parity' quite regularly, and just as often you will hear 'toggle parity'. Just what is parity and how does it work? As you will know, a byte is made up of eight bits, each bit being either on(?) or clear(?). Parity works by counting the number of set bits (1s) in a byte of data, and either setting or clearing the 'parity bit' in the transmitted data. Odd parity sets the parity bit to make the number of set bits 'odd'. For example, if we were to transmit the character A, its 8-bit pattern is 01000001, and as you can see there are an even number of set bits, (3). So with odd parity the parity bit would be set to make an odd number of set bits (that the parity bit is left clear.

The same thing still happens under even parity, except that the parity bit is used to make an 'even' number of set bits. The error detection works in this way: when data is received, the set bits are counted. If it is found that there is an even number of set bits, and odd parity is in use, then obviously there is an error somewhere in the byte. There is no indication as to which bit or bits have been affected, only that there is an error. It can be seen from this, that parity checking adds on its own is of very little use, and in fact the use of no parity, where no parity checking is done, is quite widely used.

Mark/space

MARK and SPACE when we talk of bits within a byte we usually say that they are either set or clear. When talking in terms of communications a set bit is termed a mark, and a clear bit is given an term space.

START/STOP BITS: there are two types of transmission of data, synchronous and asynchronous. As the interface for the Dragon 64 is the RS232C Asynchronous Communication Interface Adapter, ACIA, for short, and is not capable of true synchronous transmission, I shall simply say that with synchronous methods, the two communicating computers are synchronised with each other and so once synchronised with each other there is no need to indicate where each byte of data starts or stops. A series of synchronisation bytes being transmitted at intervals, very much similar to the methods used for timing programs etc on tape. The ACIA, being asynchronous, has to use other methods of determining the start and end of a data byte, this is done by the use of start and stop bits. A start is simply a bit that indicates the start of data. The start bit is transmitted as a mark, (then come the data bytes, which are obviously mark or space depending on the bit being transmitted. They are then followed by the stop bits), which like the start bit is transmitted by the mark. As you can see an eight bit byte, when transmitted by the RS232C asynchronous interface adapter is at least ten bits long. Depending upon the protocol being used either one or two stop bits are used.

BAUD RATE: the baud rate is a measure of speed of transmission, and can be translated to 'bits per second'. For example 300 baud can be regarded as 300 bits per second, or 375 bytes per second (300/8 = 37.5). This figure of 37.5 bytes is of course assuming the transmission of 8 bits per byte, but as I have said above we are transmitting at the least 10 bits per byte, and so 300 baud becomes 30 bytes/characters per second (300/10 = 30).

SPLIT BAUD RATE: a split baud rate is a way of saving transmitting and receiving data at different speeds. The standard split baud rate is 1200/75 and 75/1200. The first number is the speed that the receiver is working at and the second is the speed that the transmitter is working at. In the first case the data is received at 1200 baud and transmitted at 75 baud, the second case is the opposite, receive at 75 baud and transmit at 1200 baud.

FULL/HALF DUPLEX: full duplex describes the ability of the ACIA to communicate in both directions at the same time, whereas half duplex describes the ability to transmit and receive but at the same time. When using the half duplex method it is necessary for each machine to let the other know when it has finished sending and is ready to receive. This is very much like two people communicating over a two way radio, and having to say 'OVER' etc. Full duplex is like a telephone conversation, each can talk at the same time.

Internal checks

That just about covers most of the most commonly used phrases and terms. The user of the 6551 should realise that all the parity checking, addition and subtraction of the start and stop bits is carried out internally by the chip and any errors are indicated by the flags in the status register of the chip. The status is presented by the user in the same condition as was transmitted, unless of course an error has occurred, in which case the data could be just about anything.

The RS232C interface does not operate on TTL levels (0V to 5V) for signalling but instead uses voltage levels between -25 volt and +25 volt. The Dargos interface uses what appears to be the norm, namely -20V to +20V. One other thing with the RS232C interface is that on the data lines a voltage level of between +5V and +25V represents a 1 (0), and -5V to -25V represents a 0 (1), with the opposite being true for the control lines.

The computer is given the term Data Terminal Equipment (DTE) and the modem is given the name Data Communication Equipment, or DCE for short.

Now onto the Dragon's RS232C port, you probably won't be surprised when I say this but the Dragon's serial port looks as though it was lodged into the machine, the user being provided with two of the five possible control lines available. However, the control lines provided do allow the interface to be used in a satisfactory manner. There are seven connectors available, they are: 1) TX, 2) GND, 3) DTR, 4) RX, 5) CTS, 6) +12V, 7) -12V. Lets go into a little bit of detail.

TX is the transmitter for transmit, a little data being sent by the Dragon leaves on this pin.

GND is the ground pin and its uses as a return for all of the signal lines. **RX**, is the receive data pin, here all the data comes into the Dragon. **DTR**, which is short for Data Terminal Ready, is the line used by the computer (DTE) to indicate to the modem

(DCE) that it is Ready and in a state to receive data. Most all cards use the CTS (Clear To Send) pin. The modem uses the modem to indicate to the computer that it is in a condition to receive data, and then send it outside to the telephone line. It is essential that this line is connected on the Dragon otherwise you will find transmission impossible although you will be able to receive data.

Connection

The last two pins, +12V and -12V, are provided in order to drive the control lines fit or fit, and are not capable of providing power sources for any other use.

CONNECTING TO A MODEM: of the five lines that the serial port provides the least number that we can get away with connecting is three: They are TX, RX, and GND, if you wonder then you should be saying: But what about CTS? It's impossible to transfer without that connected! Well noticed, yes we do have to connect to CTS, but we can fool the Dragon. There are two methods open to us, 1, we can connect the -12V pin to CTS, which will then inform the 6551 that it is always clear to send, or we can connect DTR to CTS, which will inform the 6551 that it is clear to send when ever the DTR line is low, ie whenever the Dragon is on line. The links between the pins should

be made inside the 7 pin plug that is on the end of your modem cable, and not inside the Dragon itself. When connecting to the modem it should be realised that the TX and RX connections should be reversed, that is the TX of the Dragon should be connected to the RX of the modem, and the RX of the modem should be connected to the TX of the Dragon. The GND pins should be connected to each other. The above is the minimum connections that we can make, if your modem has extra control lines that it makes sense to make use of them. I shall illustrate all of the connections possible from the Dragon's port.

Dragon	Modem	Modem	Modem
GND	GND		
TX	RX		
RX	TX		
DTR	DTR	RTS	
CTS	CTS	DSR	DCD

The connections in the first two columns are the recommended ones but some modems, like the Dragon, do not possess all of the connections possible under the RS232C interface standards, and so the connections in the third and fourth columns are alternatives if the previous option isn't available.

GETTING ON LINE is the final stage. We've connected the Dragon to the

modem, the modem connected to the phone line, what next? Well before doing anything we must make sure that we are set up using the same protocol as the computer that we are going to call. Most Bulletin Boards (BBS) use 300/300 baud transmit and receive at 300 baud, 1 start bit, seven data bits, even parity, 1 stopbit, using this protocol should give you some measure of success in most cases. On the increase is the use of 1200/1200 baud rate, this is receiving data at 1200 baud and transmitting data at 75 baud. The 6551 as used on the Dragon is not capable of working at duplex at split baud rates, but it is possible all the same, to access computers running at split baud rates, for example Proseal.

Conclusion

The above is a very basic introduction to what is a very interesting subject, many books have been written about communications, although none of them, to my knowledge, specific to the Dragon, if there is enough demand and the editor is in agreement, then I shall attempt to write further articles which will go into more detail, and possibly include some programming the 6551 (in machine code) so that you could write your own personal communications package.

Adventure Contact

To help passed adventurers to find us we are publishing an Adventure Helpings — simply fill in the coupon below, stating the name of the adventure, your problem and your name and address, and send it to Adventure User Adventure Helpings, 1219 Little Newport Street, London WC2E 9PP. As soon as enough copies have arrived, we will start printing them in the magazine.

Don't worry — you'll still have Adventure Helpings to go with!

Name

Problem

.....

Name

Address

.....

If any reader does have serious delivery/non delivery problems with any supplier, whether or not they advertise in DU, we would like to know. Only rarely can we do anything to improve a genuinely dodgy situation (which fortunately are rare) but it helps us to build up a profile and identify any long-term problems. Come to think of it, that includes Dragon Users, as, naturally, we want to know about any bottle necks as early as possible. Apologies in advance for not acknowledging every letter, but where we can be of practical help, we will.

COME ALONG TO THE DRAGON COMPUTER SHOW

at Caselt Town Hall,
near Wakefield, Wiltshire

on Saturday 30 April 1988
10.00 — 3.00

Adults £1.00 Children under 16 50p

Easy Access from M62 and M1

If you are interested in taking part
as a retailer or a demonstrator,
or would like further details
of the show, please contact
JOHN PENN on 04203 5970

- ★ DON'T FORGET THE CARDIFF DRAGON ★
COMPUTER SHOW
ON SATURDAY 27 FEBRUARY
★ AT CARDIFF — WALES AIRPORT ★

Winners and Losers

Every month
Gordon Lee will
look at some prize programming

In January's *Dragon User* I gave my solution to the October competition problem. This was to find certain integers which belong to more than one class of the figurate numbers. To keep the solution as simple as possible, the listing that was given compares just two of the types of figurate numbers. In the example given, the comparison was between the square and triangular numbers. When this program is run, each matching value is printed in turn and the program can be stopped as soon as the required number of values has been found. The formulae used in lines 20 and 30 of last month's program are those which give the tables in October's issue. By amending one or both of these formulae between each run, all of the required pairings can be compared. The effect is that the program uses what is called a "looping" technique to prevent unnecessary duplication of calculation, as you will see last month. This speeds up the execution time of the program.

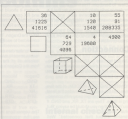
Listings

I was therefore surprised at the length of some of the entries received — quite a number having a listing with

over a hundred program lines — too long to be reproduced here. In considering some of the shorter listings, I was surprised to find that some, when tested, failed to produce the solutions submitted with them!

Function

However, one which did come up with the right answer is given here. It was from E A Newman of Addlestone, Surrey and is interesting in that it makes use of the function (FN) command of the *Dragon*. The five formulae which produce the triangular, square, cubic, tetrahedral and pyramidal numbers are initially defined in lines 10 and 20. To perform any of these calculations subsequently in the program it is simply necessary to use a line such as $X = FN(10)$, which, if N is at that moment, say 5, will compute X as 15 — the 5th triangular number. The advantage of using the FN command is that it can be used repeatedly in different parts of the program without the need to type in the generating formulae there but once. In the case of particularly complex formulae this can prove quite an advantage, and will also help to minimise errors in E A Newman's listing (the pairs of figurate numbers required at each



Austin Henderson's solution, using a Tandy 600-110 printer.

stage of the computation are calculated, using the FN command, in the subroutine at lines 30 to 80. In lines 90 and 95 is another selection of IFs, THENs and ELSEs. Normally I should not recommend the use of more than one set of these commands in a single program line unless you fully understand the logic which the computer uses in interpreting these lines (in the same way that a too liberal use of ANDs and ORs can also produce confusing results).

A number of other readers also included a printer routine in their programs to reproduce the grid arrangement for the answers. Thanks to Rob Liddgen of Walsley, Swindon, and Randy Longshore of Chesterfield, USA, who were among the winners who presented their solutions graphically. Special mention that also be made of Austin Henderson of Broomsgrove for a very neat piece of graphics produced using an inexpensive Tandy 600-110 printer.

```

10 CLS:DIR FCB,31:DEF FNTR(X)=X*(X+1)/2:DEF FNSQ(X)=X*X:DEF FNCU(X)=
  X*X*X:PL=1
20 DEF FNTH(X)=3*(X+1)*(X+2)/6:DEF FNPY(X)=3*(X+1)*(2*X+1)/6:GOTO 90
30 X=FNTR(N):Y=FNSQ(N):GOTO 40
40 X=FNTR(N):Y=FNTH(N):GOTO 50
50 X=FNTR(N):Y=FNPY(N):GOTO 60
60 X=FNSQ(N):Y=FNTH(N):GOTO 70
70 X=FNSQ(N):Y=FNTR(N):GOTO 80
80 X=FNSQ(N):Y=FNPY(N):RETURN
90 T=2:N=2:N=2:PL=PL+1:IF PL=6 THEN GOTO 150 ELSE IF PL=4 THEN F=1 EL
  SE IF PL=5 THEN F=2 ELSE F=0
100 ON PL+1 GOSUB 30,40,50,60,70,80
110 IF X=Y THEN R(PL,T)=X:GOTO 140
120 IF X<Y THEN N=N+1 ELSE N=N+1
130 GOTO 100
140 IF T=0 THEN GOTO 90 ELSE T=T-1:N=N+1:GOTO 100
150 FOR J=0 TO 2:Y=2-J:IF R(J,M)<0 THEN PRINTR(J+1,K)+7*H,J,R(J,M)
170 NEXT J,1:PRINT#400

```

E A Newman's program uses the function command.



GLORIFY game at chess. Sometimes he loses his master file complete with numerous Dragon adventure solutions, but panic over as it is retrieved from beneath two copies of Dragon User, two programming utilities with manuals in the size of telephone directories, and a Spanish edition of one of my books. I must sort out this filing system of mine some day.

A belated thank you to the people who sent me Christmas cards via Dragon User — ah, I didn't know you yourself! A special thank you to Joe Binns, our noted responder from Malta, who sent me not only a Christmas card but a post-order for one you said so I could buy myself a drink. What a star! I must point out, by the way, that when I gave Joe's address last time I got things slightly wrong. The full address is 75 Ammiration Street, Hamms, Malta. What I printed as the post code last time was, in fact, the date! Foolsy say, stupid writ, apparently all the people who write to Joe (I pass on his thanks) included that as the post code, much to the amusement of Joe and his postman.

It seems that many of you write to him, exchanging theories and giving details of many games and utilities. Some, it seems, had well over 200 files on offer. Well done chess and chessers, keep up the good work.

Keeping briefly on the foreign side of things, Ole Ekoy (who can be found at Stokken, N-5480 Sagrev, Norway — postorder@ccw.no) contacted with a series of little data bits to tell you that he runs a Norwegian Dragon magazine, and that he can get help for the following adventures: *Black Sanctuary*, *Castle Island*, *Castle of Doom*, *Caverns of Doom*, *Cirrus Adventure*, *Don't Panic*, *Dragon Mountain*, *Franklin's Tomb*, *Goblin Baron* (easy-peasy, solved it myself in one afternoon!), *Joustaction*, *Mention Adventure*, *Master of Doom* (lord of Doom), these adventure writers, *Reignier's Diary*, *Prince Adventure*, *Return of the King*, *King of Darkness* (part 1 only), *Sea Quest*, *Shenangoes*, *The Art of Magic*, *Time Machine*, *Wizard*, and finally the *Stone Factor*: if you fancy a few exotic stunts, drop him a line.

Plus letter from gills, finds out that it's from Sean Neale in Wakefield, who says that I write a superb column. Well is this

man, get his query answered immediately. It's an old friend, it's Spuggy and I've got yet another letter here from one Steven Wood in Essex — we shall be returning to this chap later) about the same problem in the same game. It's all to do with entering these co-ordinates in wrong you about all over the place once you've got to the transporter. Well, as we all should know by now, the co-ordinates for the various places that you want to get to are:

Planet 0-4-1-5
 Emerald 2-2-0-0
 Island 1-8-5-5

Read this carefully. Sean and Steven, for each place, enter each number by pressing that number, and at the end of the row (after the fourth number has been entered in other words) you must **PULL LEVER**. And every you go. I thought nobody would ever have problems with Spuggy ever again, but it just goes to prove that this extremely popular adventure for the Dragon still manages to confuse people. The sign of a good game, I suppose.

Problems with *Reignier's Diary* for Andrew Lamb in Hazel Grove, Stockport, Cheshire. An address that should be familiar to a few of you... but I shall say no more, stereo must be observed would probably be this column if I mentioned another publishing company. Any road up, as they say, he's having problems getting the anachronal, despite (and I quote) many, many hours of trying. Well, actions at the magic first sheet, the part that deals with the collecting of the anachronal, reads as follows:

"Goed Bruce into Dead? Dive and press the button to reset the life. Bring him out via the northern exitance. Send Foghorn in through the northern entrance and down on the life to get the whiskers. Bring him out via either the northern or southern exit and transfer the whiskers to Goliath. Get Goliath to wear the whiskers. To complete Dead? Dive, send Goliath in, down and into life and ladder and get the anachronal from the lowest machine and give the anachronal to Reardon. This is hard to complete without getting caught, so use the HOLD and QUIT commands."

There, let's hope that's sorted that par-

toiler problem. A little bit further on in the game, Andrew tells us that he can't get past the walled garden. I shall quote step number 11 in the solution:

"Send Goliath into the walled garden carrying the wand and Cast Spell then get the rod. He returns from the ride with the wand at daybreak. Use the wand to fish for the spaces and give them to Foghorn to wear."

That's all we need to know about *Reignier's Diary*, so back to Stephen Wood and his many problems. All of them, I happen to add, are connected with adventure games. You'd not want anyone to get the wrong idea...

Where should we start, Stephen? "See the things that are going on the tube and see the route as they go?" Well? And for printing that, I win the bet. Anyway, let's take a look at *Reignier's Diary*. Where's the *Reignier's Diary*, for which I know. On the garden planet, as last month's column will tell you. Where's the acid? North from the ship on 8200. Where's the acid? The thing to help you breathe on planet? So technical, a suit is the object in question I believe. Have we unlocked the cabinet? Have we opened the manual? Have we pressed the red button in the teleport room? Are we, in fact, wasting our time attempting to play adventures and should we do something marginally easier instead, like unrolling paper clips or sharpening pencils? Hee hee, sorry Stephen, could I not be a wee bit at someone. Where are other useful objects, he would like to know. Well, I can't tell you everything, how can I? Read last month's column in minute detail, it's all in there somewhere.

Antonio Juxta position, where the question is able to something out of the Restaurant at the end of the Universe, where's the space with which to dig in the base where you paid the dice past the nightys and so on to get the Red? *Castle One*? Your hero will be full of wells, and all that. Any last, what the must do is to blind the nightys proof with your camera and take the space from him. Where's the camera? In the photo gallery. Where's the photo gallery? Go to the mosaical pattern (don't enter endless red spaces), wait for the car and board it when the doors open. And if you can't find the mosaical platform then

you're going to have a hard job finishing the rest of it! No, once you've got past the nightmare dread dragons a kobolds, and you should have little to worry about. I shall await your next letter...

Now this, my friends, is where we talk about something else, thanks to a letter from Paul Harrison in Singapore. Anyone who's managed to evade their way through Tolkien's (Tolkien's...!) The Silmarillion without getting bored is being deserving a mention. Our local pastiche guide is more interesting. (Rumour has it that a game based on The Silmarillion is under development; it will be played on a MUD, either have no-solution, and slide into the sea at the end; Muck as I loved The Hobbit and Lord of the Rings I could never get on with The Silmarillion (I found a gem in it once and then realised it wasn't meant to be there). Sounds like a spaced out heavy metal band to me, but there you go. If it is, I was up till 2am last night watching them, which may explain my present state of mind. What if you called Lord of the Rings and arranged to make up one's mind there? SR, of course.)

Paul's first comment is a novel one: have a competition where someone sets up a title and anyone with nothing better to do tries to make an adventure out of it. (This month's title is: David Plot of 37.

Secondly, he goes on to say that he would like to write an adventure based on The Silmarillion and other such books and would like to know how to start going about it. Well, first of all there's a story little thing called The Law of Copyright. (I think a little bit)

To take a specific but non-real (I hope) example, suppose that you'd just finished reading The Price of Eggs by Denis Dising, published by Trollslayer Press, and

wanted to turn that into an adventure game. Where to start?

The obvious place to start, and this is where a great many ideas fall straight away into getting in touch with the author or the publishers, and then you'll have to find out who holds copyright on the book. Generally speaking anything that a writer by someone who's been dead for over fifty years is going to be out of copyright and languishing limply in the public domain, but it's as well to find out first. In many instances you'll find that copyright is passed on either author's estate or someone else may have picked up on the copyright, and any attempt to publish a game that infringes upon that copyright will land you in very severe trouble. This can happen whether you sell a thousand copies or just hand a couple out to a few friends: the same rules apply.

To begin with, then, you must write to the publisher. Don't bother with the author at this stage. For one thing they might be dead, and for another they might have sold all copyright to the publishers when it was released. So in our case a letter to Trollslayer Press is the place to start. Several things can then happen.

1) They have only published the British version of an American book, in which case you're in for a long correspondence with the original publishers, and the can also happen if it's a paperback edition of a book originally published in hardback, as quite often the different companies take charge of the same book. When you eventually reach the first publisher, we can go on to the next stage.

2) Somebody else has already done it, or is

in the process of doing it, so forget about it.

3) You'll have to get in touch with the author, so the publisher would pass on your letter to Denis Dising. I would like a dim view of any publishing house that released the home address of an author, and you'll just have to be prepared for a wait. In either case we reach the next stage.

4) Writing is out of copyright, so go ahead and write your game without infringing anything and, more importantly from your point of view, without having to pay any royalties to anyone. The ideal situation.

5) Yes it's okay but we want a royalty or some money up front. We bank manager then determines whether or not you can write the game, and if this happens to be the case then I think you'd be well advised to give up on the idea and find something else to write about.

6) No it is not okay, it is completely out of the question, and don't even think about it. This quite possibly means that the publisher are doing themselves, or having received your idea think it's a wonderful one but want a massive software house to do it. Oh well, nice try, but back to the library.

If you actually survive all these hassles and receive permission to write the game (whether you're paying anything or not, and I'd strongly advise knowing that I'm repeating myself, not to take on a game that requires you to buy money if someone else you're better off out of it), then what do you do?

Well I've run out of space, you'll just have to buy next month's Dragon User!

Write: ADVENTURE

Pete Gerrard discusses literary decorum and the joys of spring

ONE word which has entered the adventure player's vocabulary lately, and which certainly wasn't seen in any of the early adventures, is the word 'fall'. People now frequently try and GET ALL or DROP ALL, whereas before they were content to get or drop things individually. This is part of the increasing power of the parser and the demand for more sophisticated games.

Anyone who has played, for example, The Pleas, on another computer, will know what I mean. From my own point of view I cannot see the point in having to type in the cumbersome 'fall the plant pot, however successful it may make the parser appear. In this case I'm sure it was only ever included so as to impress novice players, for the blessed pot point is a 10-point pot to begin with! You have to take it out, using a towel if it remains rightly, and then put it back in again, all this to score about 5 points. A complete and utter

waste of time, but Magneto Sorrels like to boast about their parser, so there you go.

Again from my own point of view, however sophisticated the parser might be it still tend to play an adventure in verb-noun format, only occasionally using more words where a situation clearly requires it. Why on earth type 'Climb the slippery rope wearing the stout gloves when you can just so easily have 'Wear Gloves, Climb Rope instead? Less typing, and less time for the program to have to work out what you mean. Perhaps I'm in a minority, but personally I think that a parser that understands four words is sufficient, and I absolutely dislike and detest adventures that insist that you (for example) go north quietly, or walk rope carefully. If it was a real life situation rather than an adventure,

if you had to go north quietly you WOULD go north quietly, you wouldn't stumble in like a wounded elephant, and I think that

programmers should accept that these things are implicit in the instruction given to the program. An adventure should be about solving problems, not working your way through Magneto Sorrels in a vain attempt to find the correct word.

What's brought all this on, you might be wondering? Hingwong? No. Sound breaking in last night's pool league match? No. It was a quote written by a friend of mine, which reads that a definition of frustration was spending three months typing "Say 'give me with three my tudy blade'" near the end of an adventure only to find (quite by accident) that the correct inputs "Shout 'give me with three my tudy blade!'" That sort of thing should never happen, so keep it out of your own adventures, all?

However, I can appreciate the usefulness of the word ALL. Not only does it save time, it can also be a useful aid to

olving the adventure, if you go around typing GET ALL everywhere and seeing what the responses are. The first time I tried to implement such a routine in an adventure I gave up, thinking that it was impossibly difficult and nobody would ever want to use it anyway. However, I tried again recently, and discovered that it wasn't quite as difficult as I thought. We'll look at GET next month, perhaps, but this month we'll concentrate on DROP. This is mainly because it's the easier one of the two to understand and program!

We won't be going any further than GET or DROP though. Even the mighty intocan don't have **GO DOWN ALL**, for instance, in their mighty **LOOK 10-10** adventures. Mind you, they do have **Play the role()**, probably the most powerful character ever created in any adventure anywhere. Beg, borrow or steal the chance to play **Plentiful** and the even better follow-up **Darwin!** The creation of Floyd was a work of genius, and whoever thought of him first deserves an adventure medal. **Quake** is a laugh in six minutes, then silent tears at the end of it, too! A truly memorable character, and quite, quite real and believable.

If you look at the listing headed **Figure one** you will see a fairly conventional DROP routine from one of my adventures, and in order to understand it I'd better explain what a few of the variables are.

Lines 2550 and 2551 concern themselves with object number 81, a dog. They use a different sort of conventional idea, because the dog is in fact a guide dog, and without his dog you cannot get through the darkness of the cave section of the game. It makes a change from wandering around trying to find a lamp and some batteries, a situation I am somewhat tired of seeing. If you're dropping the dog then tell the player so, and if he's not carrying object number 11 in the correct manner (a variable in a box, to stop the wind blowing it out) then the variable 'c' is set to zero, indicating light off.

Line 2552 sorts out forwards and backwards by going to line 2558 if it is allowed has been typed in. The player might have typed **DROP BOX ON FLOOR**, which we just interpret as **DROP BOX** after lines 2568 and 2569 have had their say. Line 2554 (and

2566 for that matter) concerns objects numbers 24 and 26, which are the same thing. If they're in a pair of tennis rackets, and you've entered **TENNIS** and noun 26 is the word **RACKETS**, just in case the player decides to **DROP TENNIS** instead of **DROP RACKETS**, a case of convenience for the player rather than the programmer.

Line 2556 checks to see whether the player is carrying the object in question, and if not (and their message number 193 is printed out courtesy of a routine at line 2585). This is just something to the effect that you can't drop something that you haven't got.

The rackets are meant to be worn, like sweaters, and if the player is dropping the rackets then the 'rackets worn' variable 'w' is set to zero by line 2558.

Assuming that the object to be dropped isn't object number 42 (the number is mere coincidence, I assure you, and nothing to do with the making of life!) then line 2560 sorts everything out by placing the object in the current location and informing the player that he has, in fact, dropped it.

Lines 2562, 2564 and 2570 are all designed with object number 42 in mind, and object 42 is in fact a tightly coiled spring. Reasonably enough, dropping such an object causes it to bounce away from you, so we use the current position variable 'cp' to determine the new position ('np') of the spring after it's finished its bouncing. Then we put the spring in its new position and, as with the 'drop' routine, decrement the number that keeps track of how many objects the player is carrying. This, for some obscure historical reason that now escapes me, is the variable.

That's what the conventional drop routine, so how might we go about testing it around so that we can understand **DROP ALL**, as well as **DROP OBJECTY** and **drop** in the listing in **Figure two** you'll see (and the way of doing it. If you think about it, we're only really concerned with those objects, namely objects numbers 26, 42, and 81. The rest can be dropped with wild abandon, but dropping those three causes something to happen, so as long as we keep that in mind we should be all right.

Assuming a healthy bit of time remaining,

we could have a new line 2556 that checks for **throw** and if it's present then bounces the noun routine starting at line 2568. Then, with 'w' being the variable that tells us how many objects there are in total, we set up a loop to scan for every object in turn. First of all, in line 2560, we look for the dog being present, and if it is then proceed as before and tell the player that it's been left behind. Again, if he's not carrying the candle in the box then the logical variable 'fo' must be set to zero.

In line 2564 we look for the tennis rackets, and if they're being carried then we have to set the tennis worn variable 'w' to zero.

Assumes, we need three lines (code for the spring, and these are now 2555, 2556 and 2567) Same sort of thing, found out the current position from 'cp' and determine the new position 'np' of the spring from that, and print out the 'Bounceyyyyyy!' message to inform the player that something's happened to the spring. Incidentally, one of the more annoying features when testing the program that these listings came from was the spring, for I had included a routine that made it automatically hop away from you every few moves, regardless of whether you had dropped it or not. I spent ages chasing the thing all over the game, and just when I thought I'd caught up with it after solving a problem or two it would hop away again. It might only be moving two locations at a time, but those two locations are not necessarily anywhere near each other!

Line 2567 checks to see whether an 'ordinary' object is in the player's possession, and if it is then set it to the floor if the player what's been dropped before continuing around the loop with the NEXT statement in line 2568. Line 2569 then sends us back to the main part of the program, and you might care to expand on this section of the program slightly so that it would cater for an input of **DROP ALL** when the player wasn't actually carrying anything. Perhaps I think too highly of players, and either of my testers found it, but someone, somewhere, would no doubt try it!

So, **DROP ALL** wasn't too difficult, and we'll take a look at a GET ALL routine next month. Bye for now.

Figure one

```

2550 IF nm=81 AND ob(81)=1 THEN PRINT "
You leave the dog." : ob(81)=cp : ss=ss-1 : IF
ob(11)=3-3 THEN nm=GO TO 14
2551 IF nm=81 AND ob(81)=cp THEN 26
2552 IF 14<0 THEN 2566
2553 IF nm=24 THEN nm=26
2554 IF ob(14)=1 THEN nm=193 : GO TO 50
95
2555 IF nm=26 THEN nm=8
2556 IF 11=8 AND nm=8 AND ss>42 THEN PR
INT "Okay, " : ob(8a) : dropped : " : ob(8a)=
cp : nm=1 : GO TO 14
2557 IF nm=42 AND cp=18 THEN sp=sp+3 : GO
TO 2574
2558 IF nm=42 THEN sp=sp-3 : GO TO 2574
2559 nm=11-12 : IF nm=24 THEN nm=26
2560 11=8 : nm=8 : GO TO 2568
2574 PR INT " : Bounceyyyyyy! " : ob(8a)=cp : nm=24
p=1 : GO TO 14

```

Figure two

```

2550 IF nm="all" THEN 2564
2551 FOR i=1 TO nm
2552 IF i=81 AND ob(81)=1 THEN PRINT "B
OX LEAVE THE DOG." : ob(81)=cp : nm=nm-1 : IF
ob(11)=3-3 THEN 11=8 : GO TO 2568
2553 IF i=26 AND ob(26)=1 THEN nm=nm-GO
TO 2567
2554 IF i=42 AND ob(42)=1 THEN PRINT "B
ounceyyyyyy!" : IF cp=18 THEN sp=cp+3 : GO TO
2574
2555 IF i=42 AND ob(42)=1 AND cp=17 THE
N sp=sp-3 : GO TO 2566
2556 IF ob(11)=1 THEN PRINT "Okay, " : ob(
8a) : dropped : " : ob(8a)=cp : nm=nm-1
2557 NEXT
2558 GO TO 14
2559 ob(42)=sp : nm=nm-1 : GO TO 2568

```

From Gray to black and white

Gordon Lee and Graham Barber try to harness Gray codes

A bonus for completing the competition takes the form of letters (mostly complimentary) which readers include with their entries. One such comes from Graham Barber of Sutton Coldfield, who writes:

"Enclosed is an idea for the Competition Page. During the past 20 months or so I've had so much pleasure and interest from BU (especially the competition) that I thought it time to put something back, and then being called a 'stakeout' in the June issue almost led me into asking so you could say this is a 'Stakeout's revenge!'

Figure one



Graham's competition problem is to devise a program which will convert an input Gray code into its binary equivalent. But, I hear you ask, what is a 'Gray' code? (You can hear me in Birmingham — GJ.)

This, at its most elementary, is a system of counting in which consecutive numbers are obtained by changing one digit and one digit only at each stage, the difference in that digit being 1. They were first developed by Frank Gray, an American research physicist, and the codes now bear his name. Gray codes can be formulated for number systems in any base — for example, in the familiar decimal system an adjacent run of codes might be 475, 476, 477, 486, 485, ... etc. Note that there is no logical key to this sequence except

fewer digits changing between successive counts will reduce the possibility of error. A typical link between a mechanical device and an electronic counter is by means of an optical disc using on/off switching and hence a binary system, so we will confine our attention to the binary Gray code. This was first developed by Frank Gray in the 1930s to reduce errors in the transmission of signals by pulse code modulation.

At first sight it might appear difficult to rationalise a logical order for binary Gray codes. If we limit ourselves to just two bits we get the cyclic series 00, 01, 11, and 10. This can be represented diagrammatically by placing these four codes at the corners of a square (figure one). In this case each step in the sequence is forced, producing a single path (or its reverse) going around

the square, increasing the number of bits to three is equivalent to using the corners of a cube. Here the number of possible paths increases, as at each corner there is a choice of two possible routes. The only requirement is that the route chosen should take us to every corner once only. Starting at corner '000', it is quite easy to find a number of possible routes, all different from each other, but nevertheless all producing a series of codes which obey the basic requirements. Some of the routes are cyclic, ie ending on an adjacent corner to '000'; others are not. For most applications, a cyclic route is preferable. For each additional digit used in a sequence, the number of possible paths increases at an alarming rate. Using just four digits, there are 5712 different sets of codes of which 2688 are cyclic. However, without getting a bit off the track!

To be of practical use it is therefore necessary to formulate some additional rules whereby each binary Gray code will bear a 1:1 relationship with a conventional binary number in this way conversion from one code to the other (in either direction) is



Figure three: using 4 bit codes to mark nine sectors.

Prize

If the Dragon is the Computer that Time Forgot, then Zetoka is the land the Dragon remembered. Released this month by Donald Morrison, Zetoka is the island paradise discovered by R & AJ Preston (currently popping off to Mahe to recharge their island-spotting batteries), which is invaded by hurricanes and volcanoes while you are in the presidential chair (Zetoka, not Mahe), to provide islands for 10 lucky winners. Hurricanes optional.

Rules

WHY bother about answers to spin and why your mind ceases to whirl, you may discover that your Dragon has produced an answer. Print it out, quickly add any notes

or words of advice you deem necessary, send it to us in an envelope marked **SEARCH COMPETITION** at the usual address, of back and exit.

Oh yes, and don't forget the telephone. We want to know why you think that you deserve to be sent to a paradise island. We'll be trying the lines out on the boss, who's heard it all, so make it good!

December winners

It seems that being into black hole is good to the brain, because we had stacks of entries. The winners, who will get either a Zetoka or a discount voucher from Jones Peril Discount Software, are Ian Huggins of Caerphilly, Clive Scott of Ashfield, R H Wilson of Borehampton, Dave Lambson of Rutherford, D R Sharples of Mersynside,

Lee Simpson of Littleport, Felicia Hill of Carverston Beaches, John Blatch of Addlestone, Paul Priestland of Lechlade, James Bonfield of Sully, J Smallwood of Preston, Gail Fries Nelson of Donmar, Denis O'Malley of Cambridge, M. Buxfield of Llanymyne, Neil Davidson of Axminster, Christopher James of Mahe, Don Robertson of Epsom, E A Newman of Addlestone, J. Smith of Telford and D R Greenwood of Inglestone.

Favourite rebreather from Paul Priestland writes "I fell into a black hole, and when I came out ... my Dragon User had actually arrived." This is an unreal universe we live in.

Solution

This month's solution is opposite.

a simple matter. The table shows the first five Gray codes together with their binary and decimal equivalents. Figure two shows a diagrammatic representation of an optical disc based on these numbers, each sector of the disc differing by just one

bit from each of its neighbours. This is equivalent to the four-bit series given in the table. Reading from the centre of the disc outwards a hole represents a 1 and a non-hole, a zero. A similar disc constructed using conventional binary code would not

work in the same way as these would be a difference in two of the bits when going from (decimals) 1 and 2, three bits between 3 and 4, and all four bits between 7 and 8.

This then is the basis of Graham Barber's competition question. Can you discover an easy method of converting from a binary Gray sequence to its equivalent standard (or BCD) binary number? Having done this, then adapt the algorithm into a computer program so that an input Gray code is converted and displayed. The method should be capable of converting any code of any reasonable length.

Hint: a binary and its Gray code equivalent will contain the same number of bits.

As a test of your technique, Graham also asks you to give the binary equivalents of Gray-codes 111111 and 1010101.

DEC	BINARY	GRAY	DEC	BINARY	GRAY
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000
			0	0000	0000

The Answer

ANSWER: The new cubed in dimensions will have a central inner core of $1 \times 1 \times 1$ units in length. This produces 4,320,584 individual unit cubes of which 4,073,128 form the central unpaired core. These paired blocks can be arranged into a perfect cube 54 units along each edge.

SOLUTION: Any cubed $(n \times n \times n)$ in dimensions will have a central inner core of $1 \times 1 \times 1$ $(1 \times 1 \times 1)$. The difference between these two volumes will give the number of cubes in the single layer outer shell of the cubed. These will be the cubes which bear paint on at least one of their faces.

In the listing (1) is the smallest edge of the cube (which must have a minimum size of 3 if there is to be any inner cubes). From this dimension (n) and (1) are found. The corresponding dimensions of the inner cube are then

calculated in line 40. The two respective volumes $V1$ and $V2$ are then found, and their difference (line 60) results in the number of unit cubes in the outer shell. As we need to do now is to print out these series of values in which this difference (P) is a perfect cube.

If you are a regular reader of this column you will at once realise that this is not a straightforward task as the cube-root calculations on most microcs is a bit of a tin and miss affair! Unfortunately, due to the logarithmic method of calculation used by the micro, cube roots (and other powers) can be a tiny bit off. This normally does not show on displayed results as these are correct to the nine digit value which is printed on screen. However, the computer's internal registers hold the value to several more significant figures and there may be a slight inaccuracy.

This is Gordon Lee's own solution
see page 26 for results

```

10 N1=3
20 V1=N1*1:V2=N1*1
30 P1=N1*P1*1
40 N2=N1-3:V3=N2-3:V4=N2-3
50 V2=V2*V2*V2
60 P=N2-V2
70 C=P*(1/3)
80 D=INT(V3):D=C+1
90 IF C^3=C*P OR D^3=D*P THEN 1000
100 N1=N1+1:GOTO 20
1000 PRINT N1;" "
1010 GOTO 100

```

For example, if a value n is cubed and the cube root is calculated by the computer, the value that we would expect in a perfect world would be n . Unfortunately the number that we do get is usually a minute fraction less/higher/low. The program overcomes this by first finding the integral value of the number being tested. This integral of the test value was too high or +1 if it was too low. Lines 80 and 90 will program

take these two values and find the cube of them by direct multiplication. If either of these values equals the number being tested then the result is printed out, as a perfect cube has been found. In the case of non-perfect cubes, neither value would be equal to the number under test.

This technique forms a much quicker alternative to merely testing by a succession of perfect cubes.

Classified

SONY CCP-200 high speed cassette to cassette (audio) computer duplicator. Duplicates three tapes (both sides) of a time. Four channels stereo, nearly new. £100.00. Includes delivery. John Penn, Dean Farm Cottage, Kingsley, Bordon BA20 5BR.

DRAGON 32 leads, books, database, spreadsheets, wordprocessor, utilities, HD port, 120 DM2. Tel 0483 20078.

DRAGON 84 120 Double disk drive £100. Touchmaster tablet

£25. CD-6 games, business software. Tel (0649) 2173 (Devon). **DRAGON 32** (plus DemoDOS, no. 6) includes OS, Dragon Users, tapes, cartridges, games, books, all leads and peripherals. £150-ONW. 0081 243 580.

DISC DRIVE systems. At the lowest cost. Cartridges, OS, Drives (new), from only £48. Send address on envelope and two (penn) 1p stamps to: Pro-Soft Systems (S&P), 28, Pelican Rd., Pamber Heath, Basingstoke.

HERE'S MY CLASSIFIED AD.

(please write your copy in capitals on the lines below)

Name

Address

Tel

Classified rate: 10p per word.

Please cut out and send this form to: Classified Department, Dragon User, 12-13 Little Newport St, London N2 2JH 199.

Dragon Answers

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

Black to green again

SWI back in July 1983, you published a program in *Popular Computing Weekly* to detect test cards to green or black. I've used files in its program since then without any problems until now. The program I am writing uses the GET and PUT commands. No matter what I do the GET command causes a SN ERROR when used with the test program.

K. J. Hinchey
21 Doctor Avenue
Newcastle-Upon-Tyne
NE4 6JD

THE 'test' used to interrupt the CPU command at location 418-419 is shared by the GET command. However, my program did not take account of this and this is what causes the SN ERROR. The simple solution is to surround the GET command with the POKE \$...

POKE 418.57: GET(X);:POKE 419:POKE 418.56

RAM and ROM

I had quite a response to the letter headed 'ROM can't be RAM' in the January 88 edition. As there seems to be quite a number in a cartridge based ROM connector for the Dragon 32 here is some information on how to build your own.

The easiest solution was sent in by Mr A N Martin at 143 Black Haynes Road, Selby Oak, Birmingham B29 8BE. It is a bit circuit that I have copied the following article. Thanks to other readers who also sent in suggestions.

For an SRAM pack you'll need

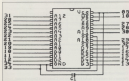


the following parts, all available from Molex Distributors. (1x BY 1803 324155 for anoprotect; a 3294 DR16C-16M chip, a 4.7mf capacitor electrolytic, and a 250 ohm IC socket for the RAM chip. You'll also need a prototyping board with 40-way edge connector and plastic case (available from Compucon). The costs about £78.

The diagram shows the RAM chip and its pin connections. The numbers at the end of the lines are in the Dragon edge connector pin. The smoothing capacitor is connected between +5V and Ground and the CS* line is fed high. All other connections are direct to the edge connector.

The Dragon's edge connector is numbered as follows — taking it 2 end on the left numbered pins 1,23 to 38 are (above) the (mean) ryming right:left. The pins numbered 39,40 are 40pins below the board on right to left.

This is perhaps not a project for those who don't know one end of a socketing iron from another. But providing you take reasonable care and test your circuits with an oscil-



6284 cmos ram chip, Dragon 16 M cartridge ram.

I/O as input

Could you tell me the I/O address of the 8255 ppi card (or) how to program it as input, rather than Output?

Richard Evans
St Asaph
Conwy

IOB are correct in assuming that the printer port is connected to a PPI. Its address is 1F982 for the data register and 1F983 for the control register. However, the 8 data lines are shared with the Dragon's keyboard and so it is not possible to use the port programmed as input. Instead you'll have to obtain one of the IO peripherals which plug into the cartridge socket.

Any comments on the circuit are welcome. Mr Martin had got to fully build and test a version of his RAM pack and has tentatively offered to build one for a Dragon User reader for a letter — but please contact him first.

Pin #	Signal	Description
1	+12V	+12 Volt
2	+5V	+5 Volt
3	DATA*	Data Input to the CPU (unconnected)
4	DATA*	Microprocessor In. CPU (unconnected)
5	MEM*	Mem. Input (Signal) (Unconnected)
6	CS*	Read/Store (Chip Select)
7	W*	Write (CPU (Unconnected)
8	R*	Chip (Unconnected)
10	+5V	CPU Data Bus 0
11	10	CPU Data Bus 1
12	11	CPU Data Bus 2
13	12	CPU Address Bus 0
14	13	CPU Address Bus 1
15	14	CPU Data Bus 3
16	15	CPU Data Bus 4
17	16	CPU Data Bus 5
18	17	CPU Data Bus 6
19	18	CPU Data Bus 7
20	19	CPU Data Bus 8
21	20	CPU Address Bus 9
22	21	CPU Address Bus 0
23	22	CPU Address Bus 1
24	23	CPU Address Bus 2
25	24	CPU Address Bus 3
26	25	CPU Address Bus 4
27	26	CPU Address Bus 5
28	27	CPU Address Bus 6
29	28	CPU Address Bus 7
30	29	CPU Address Bus 8
31	30	CPU Address Bus 9
32	31	CPU Address Bus 10
33	32	CPU Address Bus 11
34	33	CPU Address Bus 12
35	34	Parallel (Data) (Unconnected)
36	35	Signal (Unconnected)
37	36	Signal (Unconnected)
38	37	Signal (Unconnected)
39	38	Signal (Unconnected)
40	39	Signal (Unconnected)
41	40	Signal (Unconnected)
42	41	Signal (Unconnected)
43	42	Signal (Unconnected)
44	43	Signal (Unconnected)
45	44	Signal (Unconnected)
46	45	Signal (Unconnected)
47	46	Signal (Unconnected)
48	47	Signal (Unconnected)