DRAGON USER

The independent Dragon magazine

January 1988

Contents

STOP PRESS * STOP PRESS * STOP PRESS* **Pulser's** three-adventure compilation is £8.99 and not £3.99 as advertised last month. *Taped listings of *Music Extension* will be available from **John Penn**.

Editorial

LOOKING down the tunnel of another year, I am going to quietly forget that people keep asking 'what's going to happen to Dragon User?' and remember that people have been asking that question since 1984 and, like the Christmas turkey, we keep right on reappearing.

I shall start worrying about that Great Fricassee Pan in the Sky after Christmas. Right now I am staying put under the mistletoe. Happy Christmas!

Talking of mistletoe, I had the chance to meet old and new acquaintances at the 6809 Show on Saturday and talk things over. Many people agreed that there are opportunities for marketing Dragon software — for instance, producing legitimate, inexpensive conversions of American Tandy programs — which are not taken up because there is a better living to be made from other machines. On the contrary, dealers who have good contact with Dragon owners are gradually hiving off their less profitable games to specialist dealers who are more committed but are inevitably smaller.

More information on reliable, good value sources of software and hardware from overseas would be welcomed by everyone now.

Telephone number
(All departments)
437-4343

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 £14 for 12 issues
Overseas (surface) £20 for 12 issues
ISSN 0265-177. Telex: 296275
Dragon User, 12/13 Little Newport Street,
London WC2H 7PP
US address: c/o Business Press International,
205 East 42nd St, New York, NY 10017
Published by Scot Press Ltd.
© Scot Press 1987
Typesetting and Production by Artext Limited,
London NW1.
Printed by Headley Brothers Ltd. Ashford, Kent
Registered at the Post Office as a newspaper.
Dragon and its logo are trademarks of
Eurohard 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 discoveries that you can make with your Dragon. The Dragon computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

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

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

Letters

This is your chance to air your views — send your tips, compliments and complaints to Letters Page, *Dragon User*, 12-13 Little Newport Street, London WC2H 7PP.

Writers wanted

I am working on a new monthly magazine for the not-so-serious user called *Dragon Logic*, which will come out in January or February. I cannot be more specific about the dates because the magazine is run by myself only and therefore it will take longer to put together.

Dragon Logic will depend heavily on the readers. A vast percentage will be made up of things sent in by readers and suppliers. Therefore I would urge people to send their letters, top ten, software for review, news, problems, etc. to me.

If readers want to know more about the mag then please write to me.

Donald Morrison
72 Diriebught Road
Inverneww
IV2 3QT
Scotland

Why no '64 games?

I HAVE recently purchased a new Dragon 64 to add to my old '32, and am suprised that I have not seen any new Arcade and Adventure games advertised in *Dragon User* for the '64 using its full memory capabilities. Software houses like Microvision have produced programs for use with Flex Operating System, disc drive and Dragon 64 but have not produced any just for the Dragon 64 use. The same is with companies who produce OS-9 software, which needs the '64 to work.

If brilliant games like *Juxtaposition* can be made for the Dragon 32 just think of the possibilities for the '64.

I hope Software houses read this letter and take note of the points I have made as I do not think I am the only one with this opinion.

Richard Tyler
11 Madeira Park
Tunbridge Wells
Kent
TN2 5SX

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

Why oh Why The users cry

I would like to protest about the high prices of Dragon 64s and disc drives. A new disc drive costs £200 + with DOS cartidge and a new Dragon 64 costs £195. Why? Surely companies would sell more products if the prices were lowered. No wonder there is a booming trade in second-hand 64s and disc drives.

Remember one of the other computers — the Atari 800XL. You know the one where programs on cassette take a long time to load. Like *International Karate* in 20 minutes. A few years ago its popularity was dying and the makers decided to drop the price and now it has recovered and is selling well. Shouldn't this approach be adopted for the Dragon to get it back in the shops so that the user base increases to what it was back in 1984 or even treble that?

I have noticed some games for the Dragon and CoCo from America that are quite good. One is *Brew Master* by Tom Mix and others are *Zaxxon* and *Pooyar* by Datasoft. Why hasn't someone licensed them to sell in the UK as they are brilliant games up to *Shock Trooper* standard.

I would like to congratulate Quickbeam for *Superkid* and *Fire Force* (yes, I do like it) and Microdeal for *Airball* and also the other companies for still supporting the Dragon.

My final point is regarding Dragon shows. Why hasn't anyone (except Microdeal) put a Dragon show on in Birmingham? It does exist you know. I think it's ideal — central, easy for most people to get to and probably cheaper than London. Is it a case of anything north of Watford Gap (and more recently South of Osset) is ignored?

Mr. K. Hunt
11 Demarnham Close
West Bromwich
West Midlands
B70 6AJ

We strongly suspect that the answer to most of your questions is that Atari managed to develop into a very strong and successful company before it ran into rocky waters, whereas Dragon Data did not enjoy half so much success before trouble hit them. It helps to have friends at the bank if you want to relaunch your products.

In reply to your other question, what? You mean there really *is* something between Watford Gap and Osset? I thought it was all just Birmingham. Seriously, though — why doesn't someone put on a show at the NEC? It's central, purpose built, plenty of room to park . . . but have you *tasted* their Cornish pasties?.

Now we do the homework!

I would like to compliment you on an excellent magazine and I think October's issue was probably the best for ages. Not only packed with interesting information, it also helped me complete a maths GCSE problem!

The problem was to find the amount of triangles in a diagram and work out a solution. I was given the problem on 24th September, my Dad and I were stumped until lo and behold the *Dragon User* arrived the very next day! Looking at the competition I saw a little table accompanied by a large diagram. When I read the passage next to it there was my problem and equation neatly documented for me.

Keep it up Gordon, much appreciated.

Daniel Swan
118 Upland Drive
Derriford
Plymouth
PL6 6BG

Bulletin board bulletin

WHILE I was working an idea struck me. After a while I regained consciousness and it was still there so I thought I would tell you about it. How about a bi-monthly page devoted to bulletin boards. Readers could be encouraged to send in details of the boards that they use and a list of active boards published. Also special features of particular boards could be highlighted. I don't know how popular it would be but it may be worth a try.

K.G. Smith
33 Glack Road
Deal
Kent
CT14 9ND

IF anyone wants to send us details on their bulletin boards, we'll print them. We have something on BBs in the pipeline, in fact. As soon as I can find a bent coathanger, I'll try and fish it out.

Trans-atlantic Telegraph

I should like to react to a letter by Mr. Randy Longshore in the July issue of *Dragon User*. He comments on my CoCo/Dragon Basic converter published in the April issue.

Of course, Mr. Longshore is correct when he states that a Dragon cannot read CoCo discs, at least not directly. I take great interest in his procedure to modify CoCo discs to be loaded by a Dragon and will most definitely contact him in this respect, especially since I myself have just begun work on a program that would copy all files on a CoCo disc to a Dragon disc. However, though a Dragon user indeed needs to find a CoCo disc user who is prepared to copy files from disc onto cassette, Mr. Longshore's opinion that my program would then be more or less superfluous oversimplifies matters.

First of all I should like to emphasize that not all Dragon users own disc drives, and that they can use my program, even though it tries to deal with disc Basic tokens as well. Moreover, not all Basic software need necessarily come from a CoCo user. With my program the Dragon user also gets access to commercial software written in Basic, eg the compilation tapes published by, among others, the American *Rainbow* magazine. This gives the Dragon user the ability to tap into the immense software pool available for the CoCo, albeit in a very limited sense.

Far more important, though, is the fact that it just will not always do to ask a CoCo user to save Basic files in ASCII. It occurs quite frequently that Basic programs which have been 'packed' by removing superfluous spaces and by combining Basic lines get corrupted by saving them in ASCII: tokens are replaced with longer ASCII text, and long lines may overflow with the result that ends of lines may be chopped off while saving to cassette. This may even happen with 'normal' programs which have very long lines. This is not conjecture but sad experience and was one of the minor reasons for writing my converter in the first place.

I appreciate Mr. Longshore's interest, and feel that readers of *Dragon User* can only profit from this kind of exchange of information in the only large-scale medium available to Dragon users. I am of course interested in further comment or suggestions as regards my program.

Rudy Duyck
Abdijbekestraat 8-C2
B-8200 Brugge 2 (St. Andries)
Belgium

Screens x 4

SINCE light pens for the Dragon are common among us, I would like it that someone out there will write software to create graphics on four PMODE4 screens linked together. You will be able to see only the usual grid size, but can move around the screen that will be a grid of 256 + 256/192 + 192. There is a better way but more complex, by moving around with a kind of spying glass that will double every dot's height and width. But you will be able to look at the whole screen. Got it?

I'm writing for you! oh brave ones. Please send what you have done to DU

Ofir Rahar
Herev Laet 34
Emek Hefer 38860
Israel

Where is Tandy?

COULD you please include the address of the Tandy User Group in your next issue of Dragon User. I have tried every address I could find to try and track them down without luck, and so I wrote to you.

Craig Dillon

HAVING made enquiries, it seems that the Tandy Users Club no longer exists in this country. I am told that one of the Northern Dragon Clubs inherited some of the membership of the old group, but I am not certain which one or what their address is at present. I will let you know as soon as I can find out. Also, the National Dragon User Group deals with Tandy as well as Dragon interests. They are at 6 Navarino Road, Worthing, Sussex.

Range right

MAY I take this opportunity to give a poke to control the cursor link rate on a '64? Location &H FABA controls the rate, poke it with 1 to give a cursor that does not vanish when you hold down the space bar, and POKE 283,56 to update the checksum. POKE &H FABA,0:POKE 283,55 to turn off the cursor.

I also send a right justify routine which will take a line to A$ and justify it to a line length contained in variable LN, returning the justified line in B$. Has anyone got a better method.

```
100 RIGHT JUSTIFY
110 A$ = "This is just an
  example"
120 A$ = A$ + " "
130 LN = 32:F = 0:S = 0:N = 0:
  P = 1:L = LEN(A$):B$ = ""
140 GOSUB 280:GOSUB
  290:IF F = 0 THEN 170
150 S = S-P + F:P = F
160 IF P> = L THEN
  N = N + 1:GOTO 140
170 IF N = 1 THEN 310
180 P = 1:GOSUB 280
190 SP = LN-S-P-N + 2
200 B$ = B$ + STRING$
  (P-1,32)
210 DS = INT(SP/(N-1)):
  MS = SP-(N-1)*DS:
  ASS = 1
220 GOSUB 290:IF F = 0
  THEN 300
230 B$ = B$ + MID$
  A$,P,F-P): P = F
240 GOSUB 280
250 IF SP = 0 THEN
  B$ = B$ + " ":GOTO 220
260 IF PASS> = MS THEN
  B$ = B$—STRING$(DS +
  2,32):SP = SP-DS-1:
  PASS = PASS + 1:GOTO
  220
270 B$ = B$ + STRONG$
(DS + 1,32):SP = SP-DS:
  GOTO 280
280 IF MID$(A$,P,1) = " "
  THEN P = P + 1:GOTO 280
  ELSE RETURN
290 F = INSTR(P,A$," "):
  RETURN
300 B$ = LEFT$(B$,LN)
310 A$ = B$:PRINT A$
```

Fjord calling

THANKS for many excellent magazines. I wish to thank all software writers/publishers for all the supberb games/programs that have arrived. One year ago all my Commodore owning friends tease me because of the quality of the Dragon games. Now when they have seen *Shaolin Master Plus*, *Ice Castles* and others, they regularly come to me to play my Dragon games. I even have an Amiga-owning friend who likes my games.

And a last thanks to *Dragon User*. I read every page of it every month.

Continue like this!

Tor-Helge Skei
Ustaasen 87
7082 Kattem
Norway

Norway club

WE would like to tell you and your readers that there is a computer club for the Dragon and Tandy CoCo in Norway.

We have a monthly newsletter. We have members from all the countries in Scandinavia. We'd like to get in touch with other computer clubs all over the world.

If you want to know more about our computer club and what we are doing, please write to use for more information.

Tromso Dragon/Tandy Club
Sandneshamn
N-9105 Eidkjosen
Norway

Tyne and here

JUST to prove that the Dragon has still got a healthy following, could you please give a plug to my local club the NORTH EAST DRAGON USER'S CLUB. We have 14 members and meet every Wednesday at 8.00pm in the Etterick lounge of the Grindon Mill, Chester Rd., Sunderland. If anyone can make it we would be pleased to see them, or they can contact me.

Chris Jobson
23 Walsingham
Biddick Village
Washington
Tyne and Wear
NE38 7HF

SUPPORT you local Dragon club.

News desk

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

Harris front end KLIKs

HARRIS Micro Software's new complete front end and line editor, called the KLIK Utility, is now available.

The package features point and click operation of the entire system from the keyboard or a mouse, with pull-down menus, pointers, a dialogue box, control buttons and help messages, selective directories, file icons, repeating DOS commands and a set up module for easy control of the screen, windows, BREAK, etc.

The word-processor like line editor has trace and pause facilities, and desktop facilities include a disc based spooler, memo pad, snapshot and KLIK Basic, which allows you to design your own windows, icons and menus.

There is, however, no snapsbat function, as featured in one of our advertisements. Bob Harris says that he will give some thought to whether or not he should include it when he works out what it is.

The package costs £14.95 from Harris Micro Software, 49 Alexandra Road, Hounslow, Middx TW3 4HP. Bob Harris's Basic 42 was one of the best Dragon packages of 1986.

More from The Solver

SIMON Hargrave is pleased to announce the arriveal of his latest adventure for the Dragon 32 and 64, the fourth in the *Starcrash* series.

The Heir of Tyos is an unusually large, 200-location adventure in real time, with many assorted mobile characters and events, some of them interactive. The split-level parser will accept full English sentences.

You play the Prince of Tyos, places inside a mysterious pyramid by a rival monarch. To win peace for your kingdom, and the kingship, you must succeed in escaping. But your rival is devious and not to be trusted. To find the mysterious Amulet of Tyos is easy, but escaping alive is more difficult...

The game is randomised so that it will play differently on each occasion. Do you, asks Simon Hargrave, dare to accept the challenge?

This and all Simon's games are available for £5 each (includes p&p) from Crawley Hill Farm, Uley, Dursley, Glos. GL11 5BH.

New editor for Dragon Update – same old plea

THE November issue of *Dragon Update*, looking white and shiny, has just dropped out of the postman's bag and onto my foot. Pity it's bundled up with two cwt of other mail. Why am I standing so close to the postman? I'm not — I had to carry it up the stairs myself. Our postman' may be late, but he's not stupid.

This is the first issue by new, Irish, editor J. Barry Caruth, and Paul awards him eight out of ten for effort, which is fairly praiseful. This month's issue features Pascal for Beginners (part 4), Memory Management on the Large Dragon; Easy Machine Code part 14; Forth part 2; a report on disc controllers (DOSs), some notes on the Touchmaster Graphics pad, a couple of games reviews, a supportive assessment of Dragon User since the Great Change (it's nice to meet someone who can live without a full colour cover), and some thoughts from Paul Grade on copyright and circulating material. And yes, Paul, you're right, we professionals do have someone to make the coffee for us. He's called Russell Hobbs, and if anyone would like to organised a whip-round, we could buy him a new lid for Christmas.

Barry is, as usual, appealing for more material. Somewhere I have a box of fairly old submissions which just missed being published and which the owners did not ask to be returned. I wonder idly if DU would be interested in looking at some of that . . .

Contact NDUG at 6 Navarino Road, Worthing, Sussex, or send material to Barry Caruth, 132 Donaghadee Road, Bangor, Co. Down, N. Ireland BT20 4NH.

Harrry Whitehouse upgrades his power supply for the same price

THE famous A1 Supasmooth Heavy Duty Power Supply has gained a couple of extra features at no extra cost, thanks to Harry Whitehouse.

The A1, which carries a two year guarantee and lifetime service warrantry, is suitable for the Dragon 32 and 64, and recommended for all Dragonplus owners.

It now features a rapid red button on/off switch as standard. This means that the Dragon's own fault-prone switch can be left permanently 'on', reducing the risk of failure

A new design of D-connector has also been introduced, with a reliable cable clamping arrangement instead of a grub screw.

The A1 Supasmooth still costs £18.95 and £2 flat rate post and packing. It is manufacturered by Peaksoft, and is available from Harry Whitehouse, 48 Queen Street, Balderton, Newark, NG24 3NS, tel. 0636 705230.

"Quite a few owners have told us that they are purchasing an A1 as a backup for their original Dragon unit, but we always suggest they use the A1 and put their old unit in the cupbaord. After all, if an A1 ever failed — and to our knowledge, that has never happened — the user will at least have the security of knowing that it is covered by the guarantee. On the other hand, if a Dragon Data unit fails, all you can do is throw it in the dustbin," said Harry.

69 Microcosm this month

68 MICROCOSM, the journal of the 68 Micro Group has also just turned up. Contents include An Extrinsic Flex Disk Format Program, a look at the Atari 1040ST, disk-Type reporting, letters, 68 Micro Group Library, a guide to the use of 6821 PIA control register bits; Coco 3 update review, and some news and personal system reviews. The group holds regular meetings in the London and Birmingham areas.

For information contract Keith Barnes, 174 Glen Albuyn Road, Wimbledon, London SW19 6HG.

Image scanner from Epsom

FOR the wealthy and ambitious in the desktop publishing world, Epson have come out with a low-cost, entry-level image-scanner as an option on four of the company's printers, the EX800/EX1000 and the LQ2500/LQ2500+.

The bundled software with the scanner is, surprise surprise, compatible with IBM PC compatible machines, and the device costs under £200 and can read areas of 8 x 11in with the EX1000 and LQ machines, and 6 x 8in on the EX800.

September stats for subscribers

AS many of you will recall, we had problems with September 1987 edition of *Dragon User*. We are aware that a few subscribers had not received their copies well after the date of posting, and that these copies may not turn up. We had a grand total of two returned to the office, which have been found good homes.

The contents list of that issue follows. Would any subscriber who did not get his or her copy and wants a photocopy of any article or articles listed below please drop us a line, quoting their subscriber number, and we will in due course get copies of the relevant articles to them. No sae required.

Please act quickly on this offer, as we only have one copy here, with its back cover missing, and if we lose that one, we are stuck.

Letters: Stuart Beardwood's music offer; hi score corner; News: minor correction to Mr. Walker's article in July 1987; Dragonsoft: Magbase by Pulsar, Data Retrieval System by Grosvenor (both utilities, both good reviews); Datasort: mammoth home grown database program in Basic; Mountain Building: program draws up landscape contours and slices; Dragon Answers: converting Tandy code for Dragon Basic; modifying Windows; LIST/LLIST protection; On Error Run facility; Winners and Losers: April comp; Dumps for Epson FX-80, Tandy Colour Graphics Printer; Tandy CGP-115; Write Adventure: descriptions and moving characters; Adventure Trail: Total Eclipse, Juxtaposition, Aquanaut 471; Competition page: factors and number sequences; The Answer: to the June comp; Communication; Adventure Contact.

Three more packages from Dragonfire

DRAGONFIRE Services have come up with three new releases: a games pack and two utilities.

Kids Pack 1 is two simple games on one cassette: *Dam Attack* is a shooting game 'for the under 6-year-olds.' *Graphical Hangman*, a version of the old favourite that used to waste so much chalk, is designed to appeal to older players as well as kids. Over 1000 words in the memory; £2 for two games.

Small Business Word Processor is designed for use in the home and by small businesses. Features include tabulation, user defined page configuration, and audio typist! It says here, although it does not say exactly what this means. Free typist with tape? Who knows. There are also the usual save, load, edit, print and view functions. The package costs £4.

Small Business Telephone Driectory is a database program for strong names, addresses and phone numbers. Functions include view, edit, sort, add, delete, save, load, search and print in list or label format. The package costs £4.

Dragonfire are also looking for fresh software to publish, and are offering royalties of between 15 and 33%. Send for details without obligation to Dragonfire Services, 13 Parry Jones Close, Blaina, Gwent NP3 3NH. Dragonfire Services also run *Dragon Magazine*.

Dragonsoft

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

Trapped in a castle with The Mad Cook

Title: *The King's Quest*
Supplier: Simon Hargrave, Crawley Hill Farm, Uley, Dursley, Glos. GL11 5BH
Price: £5.00

THIS is the second in a series of four adventures by the Solver himself. Knowing that he hasn't even solved the fourth one himself without cheating suggests that the others might not be exactly simple.

After a few loading problems at my end (young Hargrave's probably sending someone round with a large mallet to beat up my cassette player at this moment) the game will load and run automatically and then you're free to roam the castle mainly at your leisure.

In this program you play the role of Kurgan who is still trying to get home after being helped in *Starcrash* (the first title in the series) and who has now got himself trapped in a vast castle. Your job is obviously to free this hopeless character, although this is easier said than done.

Your first location is the courtyard from where you can go north, south or along a narrow side passage. Other rooms nearby incude stables, kitchen and library all with their sprinkling of potentially useful objects for you to get hold of.

Then its upwards via a 'magnificent staircase' to a floor of bedrooms and wardrobes. Also on this floor is the tower of Minos which has within it objects like a gold key and brass trumpet, but it is deceptively easy to get lost in the mists.

The tower of Minos is easily accessible on the Eastern side of the first floor, but your passage to the Western tower is blocked, an inscription on the wall reading 'a noisy flame in reverse' — to progress into the tower simply solve the riddle.

To succeed in this game the art of solving riddles is not the only skill required, but also strategical tactics. For instance there are various violent characters wandering about the castle such as the Mad Cook and the Prince who will frequently come lurching at you. Tactics are needed here to decide whether your strength quota is enough for you to defeat the bounder or whether to flee as quickly as possible — although you also lose your possessions. One person who is slightly helpful is the Philosopher, although he also disappears too quickly.

As well as your strength other *Ring of Darkness*-style features are the fact that you need a minimum score and experience level to win, although it has to be stressed that these minimums are not the maximums. Weight is another problem — not the size of your stomach, but how much you are allowed to carry. Some objects naturally weigh more than others, such as the anvil found in the first location which weighs 2000 of a possible 3000 units, not leaving your hands free to carry much else.

The only problems with the game are the short time you are given for your responses, the 'time passes' message often appearing while you're entering a command; the text has not been redesigned, but this can hardly be called a fault.

Otherwise there's not a lot wrong with this game. One particularly useful feature is the 'Vlist' command which displays on screen all available verbs, thus helping you find the right phase when you know what you want to do but can't find a way to do it. This feature is just one indication of how much time and thought the author has put into this well designed game which is well worth taking a look at.

Philip Stott

🐉🐉🐉🐉

Follow the road from The Solver's Arms to Purley

Title: *The Quest For The Meaning Of Life*
Supplier: Simon Hargrave, Crawley Hill Farm, Uley, Dursley, Glos. GL11 5BH.
Price: £5.00

NUMBER three from Hargrave esquire — he's getting as devoted to writing adventures as he is to solving them.

Starcrash wasn't too hard for about the first 74 locations; *The Kings Quest* wasn't too difficult until all and sundry started attacking you. This latest quest isn't too hard until about the first location (it's a good job the author also supplies solution sheets, although I haven't resorted to them yet — emphasis on yet).

What sets this apart from the previous two games is that here *Tanglewood*esquely you have not one but four characters to control all of which have their own skills and aptitudes. You can switch between these characters by use of the 'persona' command and each can do such things as eating and sleeping with problems that can require not just one but two of the characters to solve them. The quartet can also split up and go their separate ways by using the command 'wait'.

The game begins in the 'Solvers Arms' public house with a landlord called Thhyhr (poor chap). Unfortunately, your characters start getting philosophical and are told to disappear into the night. So off your characters trek with odd articles like a crowbar tucked into a handbag and with four controllable characters you have four times as many hands and pockets.

So out into the world you go onto the A47 (luxurious setting for an adventure) on the 'seemingly endless road to Purley' (*and a long and chilly road that one is too — Ed.*) with only a policebox in sight. Now that police box is bound to be useful, isn't it?

After ignoring the first rule of advertising — to examine everything — I eventually managed to find the meeting place by disappearing down a nearby manhole and reappearing in a park.

Lurking in this park is a wall of worms (thats what it says!) and a row of marching hammers (that's what it says again!). Indeed there is a mad sense of humour apparent through out the game slightly reminiscent of that old devil *The Cricklewood Incident* except that it is harder and more playable.

There are only fifty locations here, less than in the previous two games, although each seems to be unique rather than certain repeated ones as in *Starcrash*. As the game is more complex though, the reduction in locations is natural.

The Vlist command is here again revealing: 49 commands this time all of which can be entered in three letters, for example 'transfer' = Tra. As with *The King's Quest*, the game is written in real time and therefore if you ponder too long, 'life passes you by' will flash up on screen. If the pace is too quick, though, you can always 'hold' and mop your brow.

There is however no 'score' feature so you can see how little of the game you have managed to complete, especially when you're not sure what you have to discover to end your search.

So that's the third in the series, not as easy to get immersed in as *The King's Quest* but more complex and challenging, certainly not a game to knock off in a couple of evenings. As for a rating it's probably 4 like *King's Quest*, but as I only gave that four dragons, I'll give this five in the hope it will induce you the reader to keep buying software. The next in the series has randomly placed objects, like *Madness and the Mindtaur* — I can't even manage the ones where the objects are static.

Philip Stott

🐉🐉🐉🐉🐉

Expert's Arcade Arena

Write to 'The Expert' at Dragon User
12-13 Little Newport St
London WC2H 7PP.

GREETINGS, viewers. Let me start by saying that I hope you are all about to have a Merry Christmas and a Happy New Year, though I haven't had a single Christmas present from you (probably because it's only November, but that's not the issue, is it?)

Moving swiftly to an apology from me (yes, I do hand them out occasionally — make a note in your diaries), without my fingers crossed, I must say that I'm sorry that the Arena has been on the air somewhat irregularly recently, but fear not, for I'm here for another series (however long). Don't forget to tune in every month.

All this crawling is leading up to an appeal. I've finally got a 'round tuit' and decided to broadcast some requests for help on behalf of myself and (mostly) my devoted public. If you can offer any solutions, then please send them in. My postman has now recovered from his hernia, and gloats as he walks past the hospital without even delivering a single entry form for the Readers Digest Prize Draw. The warehouse I bought last year to store all your letters in is now so empty that I've rented it out to film Michael Jackson videos in!

So, below in glorious technicolour black and white is the list of essentials that I'd like you, my loyal servants, to help with if you can:

Expert pleads

1) An answer to the question: 'What does Eddie do, on completing level 5 of *Back Track*'?

2) Does Universe II of *Total Eclipse* really exist?

3) A program using the information of so many years ago (DU November 1986, to be exact) concerning the way screens are stored on *Chuckie Egg*, to make a screen designer.

4) What are the codes at the end of each screen on *Time Bandit*? (Sounds familiar!)

5) A map of *Jet Set Willy II*. I believe that this isn't an official game, so don't try to buy it, but there seem to be millions of copies in circulation.

6) A list of the information that you receive while playing the game section on *The Death Mines of Sirius*.

7) Any POKES at all for any of the following: *Mud Pies, Ugh!, Glaxxons, Cashman, Screaming Abdabs, Superkid, Electron, Miser's Dream*.

8) More maps. How about one for *Module Man*?

9) The official solution to *Escape*.

10) A way of upgrading the old *Boris the Bold* to the better version without power surges.

Not too hard

Now, if all or any of that lot's too hard for you, how about something you can *all* answer, regardless of IQ, HQ, nosiness, etc.? This is your chance to vote in the event of the year, **The Expert's Trivial Software Survey.**

All you need to do is copy out the questions below, along with your answers, of course, and send them to the usual address, where the editor will get them muddled up with the competition entries. They will then be forwarded to me and evaluated by my own fair hand, with a little help from Giveup', my computer.

You may send in as many entries as you like, so long as you use a different name, address and style of handwriting for each.

Here are the immortal categories for which you must nominate a winner:

1) Best arcade game for the Dragon — from the year dot to 1988. What has been your favourite game?

2) Worst Arcade game — yes, I want to know what is the most utterly atrocious game in the history of the Dragon.

3) Favourite Programmer — you could say it's a little unfair to single out someone as the best Dragon programmer, but there are a lot of good programmers out there, and it's about time those who've beenworking the hardest should get some credit. Here's a few suggestions: Kenneth Kalish is well known for games such as *Invaders' Revenge, Danger Ranger, Escape, Phantom Slayer, Devil Assault, Cuthbert in the Jungle*, and more.

Best Programmes

Roy Coates has produced some good games — *Manic Miner, Jet Set Willy, Bean Stalker* etc. More recently, Jason Falcus has made a name for himself with titles such as *Miser's Dream (which we are still trying to obtain copies of from Microvision — Ed.)* and *Eddie Steady Go*. John Martin is another faithful choice, famous for *Moon Cresta* and *Jet Boot Colin*.

There are many programmers to vote for, but why don't we see their names on adverts? Usually just the name of the game, and the softwre house appears, which brings me to the next category:

4) Best Software House — I considered splitting this into two parts, those surviving and those who have given up, but decided to make it a mixed bag. Here are a few ideas: Microdeal (*Airball, The King, Time Bandit, Speed Racer*, all the *Cuthberts*), Incentive (*Moon Cresta, Ket Trilogy, Back Track, Eddie Steady Go*), Quickbeam (*Superkid, Shaolin Master, Fire Force*, the *Dickie* series), Tom Mix (*Cuber, Buzzard Bait, Electron, Katerpillar II*), Blaby (*Olympia, Boulder Crash, Kama-Karzy, Kung Fu*). Don't forget of course Hewson, Salamander, Peaksoft, Design Design, Morrison Micros, Melbourne House, and Mastertronic (oh, sorry, misprint. Did I say Mastertronic? I meant to say Software Projects) who have all produced some great games.

5) What's the most annoying aspect of any game? For example, the fact that you can't play *Hungary Horace* without cheating, you can't use a joystick for *Olympia*, and you lose all your lives if you die at a monster's start position on *Jet Set Willy*.

6) Most underrated game — which game was under advertised, undersold, but is really good? Here are a few ideas: *Beam Rider, Pogo Jo, Hotel on Mayfair, Darts, Ice Castles, BC Bill*.

7) Most overrated game — which game turned out to be least like its adverts? My suggestions would be *Hunchback, Drone Datatank, Space Shuttle, Module Man*.

8) The game you would most like to see converted for the Dragon — there are a lot of good arcade games not yet in Dragon versions. What would you most oike to be zapping to by this time next year? For example, *Chuckie Egg II* (various computers), *Phoenix* (arcade).

9) Best music to zap to. Type AUDIO ON before you load/EXEC a game, then insert a music cassette during play, and you should be able to 'zap away' with a musical accompaniment. This is a long standing question and should be resolved once and for all

10) Program which you would most like to see hacked — write and tell me which game you would most like a cheat for and I'll challenge the many hackers out there to see who can come up with the goods first.

11) Best game printed in DU — there have been some great games such as *Flee Flea, Cupid, Atom Hunt* printed in DU. Which is your favourite? (By the way, how about some hacks for games printed in DU?)

Expert answers

So, there are the questions. Send off your answers to reach DU as soon as you can, and I will do something about them as soon as I can.

That's all for this month. Thanks for all your letters, keep them coming in. I'll be back next month with the usual hints and tips. Goodbye.

Introducing Modems

***Ken G. Smith** takes the computer to the phone*

NEVER in the short history of microcomputers has a peripheral created more interest than the modem. The word 'hacker' entered the language and headlines were made by people leaving naughty messages in VIPs' electronic mailboxes. A young hacker in America managed to get into the Pentagon mainframe and was arrested by the FBI, or so the story goes. Holywood took the fantasy one stage further in the film *War Games*, with visions of a nuclear war being accidentally started by a young man with a micro. So much desire has it created that for some people it is the first gadget they buy. For those of us that for one reason or another have only just joined the on-line set or more importantly those who are considering joining, I hope this article will help you to find out what a modem is, what it can do for you and what you can do with it.

How it works

When it comes to computers, the advertising people seem to fall into the trap of assuming that we are all experts, so the first job is to explain briefly how the modem works. The name is an abbreviation of 'modulator/ demodulator', and what it does is to transform the binary ones and zeros into sound frequencies in a similar way to that used by the cassette port. The system uses a higher pitch for a one and a lower pitch for a zero. Using this system the computer transmits its information down the phone line and reverses the operation, converting the sound waves into ones and zeros, to receive. The rate at which this data is exchanged is called the baud rate and is quoted in bits per second, one bit being a single one or zero. Baud rates are being quoted less and less these days and are being replaced by 'V' numbers, which at first sight seem even more confusing; a brief guide follows:

V21: baud rate 300 receive and transmit, very popular with modem users in America and early users here, now losing ground to V23.

V22: baud rate 1200 send and receive, little used at present but gaining popularity.

V22bis: this is a business users' system for people whose time is money, and sends and receives at 2400 bit per second.

V23: often quoted as 1200/75. This is the most popular system. Receiving at 1200 baud and tranmitting at 75, it is used by most view data systems and many bulletin boards.

Full duplex: This is a must for a truly interactive system and I do not know of any modems that do not have at least one full duplex mode. Some will operate full duplex in V23 and half duplex in V22. The difference is most easily explained by comparing the difference between the telephone and a two way radio. On the telephone, both users can talk at the same time, interrupt each other and even argue (full duplex), whereas on two way radio, one operater must give an end of message signal and cease tranmission before the other can speak (half duplex). In fact, with a half duplex system, if both parties try to transmit at the same time nobody gets anything, just with the radio. Advertisements sometimes use 'Full Duplex' as a means of indicating that the machine will send and receive at the same baud rate, at the same time.

Xmodem: this is a protocol governing the way information is transmitted and received. It makes data transmission more reliable (not infallible), and allows your Dragon to talk to any other machine. As long as they are both using the same baud rate and the same protocol any make of computer can converse with any other. Most bulletin boards operate on Xmodem system.

Auto-dial: speaks for itself really, as it enables you to keep a personal phone directory on your computer. Once programmed, the machine will call anyone you tell it to call, which also means that you do not need a phone to operate your modem.

Auto answer: also self explanatory, only necessary if you want to run your own bulletin board. It is important to remember that a modem is operated through the RS232 port so if you have a Dragon 32 you will need an input/output cartridge with this facility.

At this stage of the game you need to decide why you want to get 'on-line'. If your aim is world domination by Friday, forget it. The military have switched all their important data onto microwave transmissions, and if you can afford to intercept them then you probably didn't buy a Dragon. Erasing the balance on your credit card is not as easy as some might think; it takes a lot of time and knowhow to penetrate such a system, and if you are capable of doing that then you are probably not going to bother reading this article. Anyone searching for a reason to get online really need look no further than electronic mail. There are several systems available at the moment with more to come.

Viewdata

These usually operate on Viewdata, which in addition to text, also gives a limited graphics capability. To use, these sytems you will need to be in PMODE4, so no colour is available. However, if like me you use a black and white television and a single colour printer, then colour is of little consequence. Most systems will have a local number for you to call, the host computer directing your message to the recipients mailbox. When he next uses the sytem he will be told that there is a message waiting. He can read an reply immediately or save it to disc to be dealt with later.

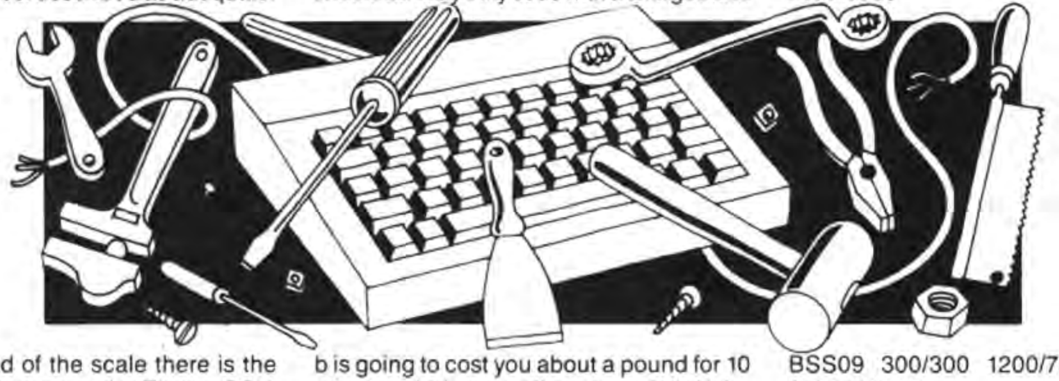
Bulletin boards

Another useful pursuit comes with the use of local bulletin boards. These normally operate on an Ascii scrolling screen principal, using the normal rext screen with file transfer under the XMODEM protocol. Here you will find sales and want ads, programs for you to download, a place to air your views to the world and above all a lot of people willing to help each other just for the pleasure of doing it. Some boards will expect some form of subscription but most are free. Many special interest groups set up their own boards. My local one was set up with dentists in mind but the systems operator or sysop doesn't mind people with other interests using it.

The adventurers among us will love another facility that is available to modem users, namely MUD. Short for *Multi-User Dungeon game* this is similar to *Dungeons and Dragons* or an adventure game but this time you are not just playing against the computer. Everyone that plays the game has their own identity within the game and the participants play against each other with the computer acting as referee. There is usually a subscription to pay first but to the hardened adventurer, the infinite variation will make that worthwhile. Armed with this knowledge and the fact that some of the long standing bulletin boards still use V21, you should be in a position to decide which device you want to buy.

Software

Once you have settled on which modem to buy, you will need the software to drive it. If you are lucky, it comes with the machine, but there are fewer people writing for the Dragon these days and if you want a particular modem, you might have to go elsewhere for the software to drive it. There are two different levels generally available. The first is somewhat basic, which is not to say that it isn't very useful. It would normally consist of two or more programs. Firstly there would be a Viewdata system; this would give you all the Viewdata graphics and text on a forty column screen. All messages would have to be entered via the keyboard while on line and to save messages you would have to save the whole screen to disc. The Ascii program for bulletin boards would operate on the standard 32x16 screen which can mean some very rapid scrolling since most boards operate on a 40 by 24. The result of this is that each line on the board will occupy two lines on the screen. All incoming data can be stored in a buffer and saved to disc or tape at regular intervals. This is sort of software you would expect to find bundled with your modem and is best described as adequate.



At the other end of the scale there is the system that only runs under Flex or OS-9 and supports auto dial, allows you to prepare messages before going on-line and allows incoming data to be diverted to the disc drive to be read later. With this system, you even get a 51 x 24 display on the Ascii program. However the price does tend to put you off somewhat. The real tragedy is there does not seem to be anything in between. A system that would run under *BASIC42* or *Edit+* for instance. The only advice I can give is get the best you can afford.

There is one tale that is certainly no myth, namely that a modem can get you in a whole lot of trouble, especially if you are not the one that pays the phone bill. The last thing that you want is to come home and find all your equipment has been sold to settle up with BT. So study your phone book, make note of all the codes marked L and use those in preference to others. Also, check all those with a charge code a or bl, these are your reserves. Keep the time down and you should be okay. Any code with a charge code b is going to cost you about a pound for 10 minutes during an off peak period. If the board you use has a cost so far facility, then use it frequently, and keep a rate of ther runnign totals over the weeks.

Well I hope this has helped any of you contemplating going on-line and so that once you have got your modem you will also have someone to call I have included some useful numbers. One last thing to mention before I go is that most of the people that operate bulletin boards have invested a lot of money and time on their system. They invariably get nothing out of it save the satisfaction of helping the rest of us. So before you log off leave a message for the sysop to tell him what you think of this board. It might make his day.

Below I have listed some numbers you might find useful:

6809 Board 1200/75 01-316-7402 24hrs
Walley 300/300 0909-773564 Sat-Thu 2100-0500
BSS09 300/300 1200/75 0705-736025 0705-736025
DBBS 300/300 0376-518818 24hrs
TUG 11 300/300 1200/75 021-444-1484

All these boards have Dragon sections. A list of boards can be found on two 'Distel' boards 01-679-1888 for 300/300 and 01-679-6183 for 1200/75.

You can contact me at MB207 on the 6809 board or leave a message on 0227-276162.

Crossword

The second great month of the Dragon Crossword. We have no reports of your verbal veracity yet, because, despite the legend (now you know why it's called a legend) on the front cover, it is still November here in Little Newport Street, none of you have seen the crossword, and only the practising telepaths have replied.

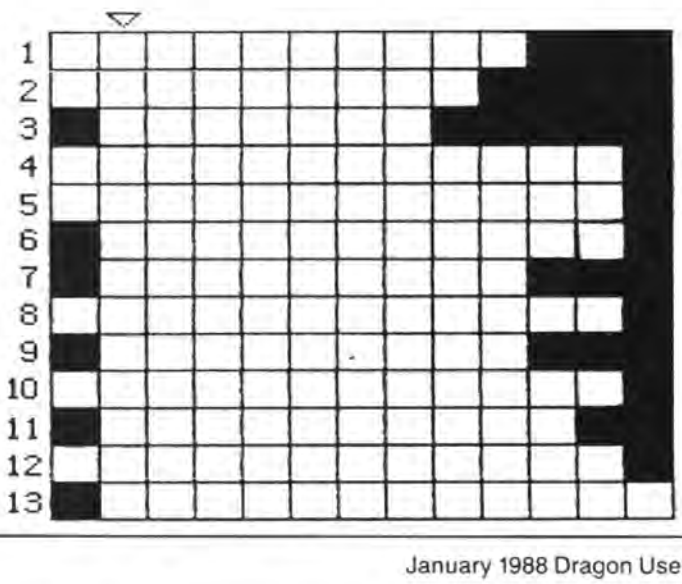
Don't forget that there will be a couple of free tapes from the Editor's Magic Bottomless Box for the first correct entries to reach us each month. You can even try telling us which tapes you'd like in an ideal world. No promises. It all depends on what we have in stock.

And you don't have to cut up your *Dragon User* either — heaven forbid! Entries can be written out on a photostat or a plain piece of paper, as long as we can read'em.

And no tie-breakers!

1. Put your jewellery into fortifications! (3,7)
2. Bird in Lincoln Green? (5,4)
3. Dirty things, ie dumps (7)
4. Is the Enterprise one? (5,7)
5. Her young char told you he's starving (6,6)
6. Does it fight its battles on the underground? (7,4)
7. A sign of two pawnbrokers plus two (9)
8. Did Van Gogh become one? (5,7)
9. Nigel and Alain are both in it (5,4)
10. Gender and rag ran a risky job in Yellowstone Park! (6,6)
11. Nose to cram into lunar rise (4,6)
12. Attack on Satan? (5,7)
13. Rock spot hero alarms a solder! (5,7)

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.



Pamcodes

Pam D'Arcy continues her introduction to machine codes

I trust that no difficulties were encountered in displaying your name in the top left position of the screen. My own program is shown in **listing one**.

I wonder if any readers thought 'why not poke the name direct to the text screen area (fixed at $400-$5FF on the Dragon) using LDA/STA instructions?' Using the 'output a character' ROM call $800C has many benefits in that it carries out some functions that would otherwise need to be performed within our programs. For instance, if you 'poked' a 'P' directly into the top left screen location ($400):

```
LDA #'P
STA $400
RTS
```

– in Basic, POKE&H400,ASC("P"), a capital P would appear. However, only the @ symbol, capital letters and a few other shift+ characters retain their original values (decimal ASC values, or character codes, 64-95) when being displayed. All other characters have adjustments made to their values to match the needs of the hardware display chips in the machine. I didn't use lower case characters in the display of my name for cleverness, but to make this point! Poking a lower case character into the top left of the screen does not result in the reversed screen (green on black) of character a . . .

```
CLS (else the result may be scrolled off the
  top by the OK display after the POKE
  command)
POKE &H400,ASC("a")
```

(quick memory jogger for those who need it — lower case — reverse screen — letters are obtained from the keyboard by holding down shift and pressing zero, after typing in required reversed characters, repeat shift+0 to return to normal display mode).

If that results in !, what does POKE&H401,ASC("!") give? The technical reasons for this are discussed in the book *Inside the Dragon*, page 159. As well as carrying out such character code conversions, $800C acts as appropiate when it receives an 'end of line' or carriage return ($0D) character — that is clears the rest of the line to spaces and sets the next print position to the start of the next line. It also takes care of scrolling the text screen up one line if the last screen position is being used ($5FF).

Keyboard input

Most programs will not be using fixed data as above but will be dealing with information held on a file or being input from the keyboard. We can then substitute printing the fixed characters of a name by printing name (and other) data as it is input from the keyboard. The program needs to be able to determine when input is complete, so that it can return to Basic, so you can choose what condition determines that for you. For example, I shall use a press of the ENTER key.

As with displaying a character on the text screen, there is a ROM routine that can be used to obtain keypresses. This is at address $8006. It puts the ASC (henceforth ascii) value of the keypress in register A. If there was no keypress, register A will be set to null ($00). **Listing two** clears screen and prints keypresses until ENTER is pressed. I shall be dealing with branch instructions (BEQ/BNE) in a moment.

Line 30 causes a jump back to the 'get a character' ROM call if there was no keypress. This forms a 'loop' of code that is repeated ad nauseum until the exit condition is met (in this instance, a key is pressed). Line 40 prints the keypress if it is a printable value (for example, left arrow still deletes a character; break is ignored etc). Line 50 checks for the terminating condition — CoMPare the contents of register A with the actual ascii value for the carriage instruction at label GETKEY (Branch Not Equal). This is a further loop of code that continues until its exit condition is met (enter key pressed).

If there is a program error preventing the exit condition being met, for example, above comparing the keypress with a value that can be input, the program simply sits there, looping away round the code — it never gets tired! The break key panic button of Basic programs is ineffective in machine code — we're on our own — so tight loops can only be broken by pressing the RESET button.

Unique labelling

In Basic programs, GOTOs and GOSUBs always jump to a unique location — the line number — which cannot, obviously, be duplicated in a program (the Basic editor overwrites the original line on any subsequent input of the same line number). Assembly source uses our own label names for moving around programs. Label names must be unique or the program will not know which location to use; duplicate names are usually detected at the assembly stage, but the error message may not be very clever — the likes of 'phasing error' or 'redefined symbol' (label names are usually referred to as symbols in assemblers) are often caused by accidentally duplicating a name in the label column.

The next screen position maintained by

Listing 1

```
EXPECTED                                       SOURCE  CODE
GENERATED      LINE        LABEL        MNEMONIC   OPERAND
OBJECT  CODE   NUMBER

BD BA 77         10        GO            JSR        $BA77
86  50           20                      LDA        #'P
BD 80  0C        30                      JSR        $800C
86  61           40                      LDA        #'a
BD 80  0C        50                      JSR        $800C
86  6D           60                      LDA        #'m
BD 80  0C        70                      JSR        $800C
39               80                      RTS
                 90
```

Listing 2

```
EXPECTED                                       SOURCE  CODE
GENERATED      LINE        LABEL        MNEMONIC   OPERAND
OBJECT  CODE   NUMBER

BD BA 77         10        GO            JSR        $BA77
BD 80  06        20       GETKEY         JSR        $8006
27  FB           30                      BEQ        GETKEY
BD 80  0C        40                      JSR        $800C
81  0D           50                      CMPA       #$0D
26  FA           60                      BNA        GETKEY
39               70                      RTS
                 80
```

the Rom routine $800C may also be known as a 'text screen pointer' or 'current cursor address' and is held in the two bytes at address $88,$89. The content of this pointer is another address — the next text screen address to be used for display. When the screen is cleared (CLS), the content of this pointer is reset to the first address of the text screen ($400). When the text screen has been scrolled up one line, the content of the pointer is reset to contain the address of the start of the last line of the screen, $5E0. Receipt of carriage return characters causes it to be set to the address of the start of the next line, subject to a scroll condition being encountered. Unless any of these special cases are met, display of a character causes the address in the text screen pointer to be incremented by 1. Unlike the graphics display, even in machine code, the position of the text screen is fixed at $400-$5FF.

To display text at a specific point on the text screen (Basic PRINT@ statement), the appropriate address can be calculated and stored in the text pointer prior to calling the ROM routine $800C. There is one print position per byte, so the @ value should be added to the text screen start address,, e.g. PRINT@256 is $400+256, or $400+$100, so to print a name as input @256 could appear as in **listing three**.

What $800C does not do is check that the content of the pointer is between $400-$5FF before it displays the next character (why sould it cross check something that it is in charge of?). As in all machine code (or Basic POKE) manipulation of Basic's workspace locations, the ROM interpreter is expecting everything to be valid so weird results may occur if we step out of line — it cannot distinguish between accidental or deliberate sabotage! As ZI have said, we are on our own in machine code. This also allows us to cheat the system when it suits us!

Branching out

The condition code register (CCR) was mentioned in the last issue as being one of the single byte registers on the Dragon. Each of the eight bits has a specific task and the execution or otherwise of a branch instruction depends upon current setting of one or more of these bits, or flags, at that moment. Some instructions affect some of these flags; others none. In general, we should be able to manage without gory detail.

As with the instructions in general, some are used much more frequently than others — and others almost never. It really does depend upon the nature of the program. A common problem with branch instructions is using a signed branch when an unsigned one should have been used — discussed in a moment. One of my greatest programming aids is having copied out **table one** onto a postcard for instant access.

There are also BEQ (Branch on EQual) and BNE (Branch Not Equal) that act independently of signed/unsigned conditions. BEQ is also a 'branch if zero' instruction with BNE its complement, 'branch if not zero'.

Listing 3

```
EXPECTED                                  SOURCE  CODE
GENERATED       LINE      LABEL      MNEMONIC       OPERAND
OBJECT  CODE NUMBER

BD  BA  77        10        GO          JSR         $BA77
CC  05  00        14                    LDD         #$500
DD  88            17                    STD         $88
BD  80  06        20      GETKEY        JSR         $8006
27  FB            30                    BEQ         GETKEY
BD  80  0C        40                    JSR         $800C
81  0D            50                    CMPA        #$0D
26  FA            60                    BNE         GETKEY
39                70                    RTS
                  80
```

Listing 4

```
OBJECT    EXPECTED                          SOURCE  CODE
ADDRESS   GENERATED        LINE      LABEL     MNEMONIC OPERAND
  $       OBJECT  CODE NUMBER

5001      BD  BA 77         10        GO          JSR       $BA77
5004      CC  05  00        14                    LDD       #$500
5007      DD0 88            17                    STD       $88
5009      BD  80  06        20       GETKEY       JSR       $8006
500C      27  FB            30                    BEQ       GETKEY
500E      BD  80  0C        40                    JSR       $800C
5011      81  0D            50                    CMPA      #$0D
5013      27  03            60                    BEQ       STOP
5015      7E  50  09        65                    JMP       GETKEY
5018      39                70       STOP         RTS
5019                        80
```

TABLE OF BRANCH INSTRUCTIONS

SIGNED	UNSIGNED EQUIVALENT
BGE	BHS/BCC/(BPL)
BGT	BHI
BLE	BLS
BLT	BLO/BCS/(BMI)
BVC	
BVS	

You can probably self determine the conditions of many of the above branch instructions — Greater/Equal; Greater Than; Less/Equal; Less Than; the V instructions are the less frequent oVerflow Clear/oVerflow Set — they probably won't be dealt with except in passing at the end of the series. On the unsigned side are High/Same; Carry Clear; PLus; HIgh; Lower/Same; LOwer; Carry Set; MInus. The BPL and BMI instructions are shown in brackets as there are certain conditions when they will give a false condition because of OVERFLOW intervening. Generally, after arithmetic on signed numbers, use their equivalent signed comparison instructions.

Also in the branch range are BRA (BRanch Always); BRN (BRanch Never — of little practical use — it 'maintains the symmetry of the instruction set' — all the branch instructions have complementary pairings — BLT/BGE etc. BSR (Branch to SubRoutine) completes the branch instructions.

The branch instructions, including BSR, generate position independent code — that is if you take the source code that we have been using so far in this series, apart from the likes of ROM calls being to a fixed position, the same, saved object code will work whether reloaded at, say, $5001 or $6600 or $EABC (the latter for Dragon 64 users who have mapped ROM to RAM while retaining 32K mode). This is because our 'goto' instructions have all been of the position independent nature of the branch instructions, as opposed to JuMP. Unlike some machines, there are no conditional JuMP instructions on the Dragon (but we do have goodies that other machines haven't got), so, for illustrative purposes, I shall adjust the second loop exit code of the above as in **listing four**.

If the object code is saved (CSAVEM "NAME" &H5001,&H5001) and loaded some time in the future at its 'default' address (address the code was saved from) and is executed, it will run correctly. However, if it is loaded elsewhere in the machine, unless the first keypress is <enter>, the instruction in line 65, JMP GETKET, is of the extended mode (=actual address) type discussed in last month's article, and the address generated within the object code, $5009, will be branched to REGARDLESS OF THE PROGRAM'S CURRENT LOAD ADDRESS, leading to potential disaster.

The branch instructions are so useful as they generate code that says 'branch to the instruction at + or — a number of bytes from this instruction'. Some assemblers permit

Continued on page 24

Music extender

David Makin presents an upgrade on his program Music Maker

FIRST of all a couple of tips for using *Music Maker*:

1. When using the 'Alter a particular note' option, middle C has a numerical value of 52, C an octave higher 64 and C an octave lower 40.

2. For those of you with a DOS, you can run *Music Maker* with the DOS attached by loading it using POKE&HBC,6:CLOADM instead of just CLOADM.

Of course, now that *Music Maker* has been out for a while, I have found several ways of improving the machine code tunes it produces, which are:

1. Storing the tune data in an improved form, resulting in a large memory saving (typically more than 60%).

2. Allowing several tunes to be put in the same program.

3. Using different sample waveforms on each channel, allowing a variety of different tones which may be altered for different tunes.

The listings give a total of six programs, incorporated respectively in listings 1, 2 and 3, 4, 5 and 6, 7 , 8 and 9. First type in and save the three machine code 'source' programs, listings 2 or 3, 5 or 6 and 8 or 9. Note: those of you using DASM replace ⟩ with and vice-versa in the assembler source code.

Next type in and save 1 'CONVRTER' and then load and run 3 or assemble 2 and use CSAVEM"SHRINKIT",&H4000,&H4333, &H4000 to save the resultant m/c after "CONVERTER". Then type in and save 4 "BUILDBOX" either after "SHRINKIT" or on another tape. Then load and run 6 or assemble 5 and use:

FORI=OTO255:V=32+31.5*SIN(3.14159 *I/128):FORJ=&H7C00 TO&H7F00 STEP256:POKEI+J,V:NEXTJ,I then CSAVEM"JUKEBOX",&H7800,&H7FFF,& H7A17 after "BUILDBOX"

Then type in and save 7 "MIXER" either after "JUKEBOX" or on another tape. Then load and run 9 or assemble 8 and use:

CSAVEM"WAVEMIX",&H2600,&H2755,& H2642

after "MIXER". The programs are written to work with or without a DOS attached, but do not use the disc. If you wish to run them from disc you must alter the relevant load and save routines and PEEK addresses in the Basic programs.

Use the programs as follows:

1. CONVRTER/SHRINKIT: load and run "CONVRTER" then, as instructed, position the tape to load the *Music Maker* tune file for convertion (EG "IMNTSMGD") then use the same play-shifts as you would to play the tune on *Music Maker* itself. Then go and have a cup of coffee while the tune is coverted (or in the case of longer tunes grow the beans!) as the convertion may take up to two hours. (1.5 in the case of "HGRNMRCH", a few minutes for 'IMNTSMGD"). When the conversion is done always make a note of the four values given, then save the converted tune to tape.

2.BUILDBOX/JUKEBOX: load and run "BUILDBOX" then for each tune you wish to add to the jukebox type in the four values, position the tape and load the tune. When finished loading tunes position the tape and save the finished jukebox — you can put up to 32 tunes on each! Note: to add to a jukebox that already has some tunes on it, position the tape at its start before RUNning "BUILDBOX".

3. MIXER.WAVEMIX: load and run "MIXER" then, after "WAVEMIX" has loaded, position the tape to load the copy of "JUKEBOX" which you wish to alter (it must have at least 1 tune on it). Then use the menu to build new waves using different porportions of the base waves and try using the built waves on different channels for different tunes. Note: pressing the space bar to return from viewing waves using option 2 does a PCLS before returning, whereas any other key doesn't, allowing you to compare different waveforms. Also using square, sawtooth or the higher harmonics for higher notes isn't recommended (see *Rainbow* Dec 1983).

Remember to CLEAR memory to below any "JUKEBOX" before loading it. "JUKEBOX" programs make no ROM calls and are relocatable in units of 256 bytes (subject to corrupting the stack, etc.), eg CLOADM" ",65536-1536 (the m/c tunes saved from *Music Maker* are also relocatable by units of 256).

For those of you who find the task of typing in the programs yourself too daunting and can't find a sufficient mug to do it for you, the programs are available from John Penn for $2.50, including a version for disc, a ready built jukebox of 30 tunes (each once only: it plays for one hours and nine minutes) and a copy of the m/c screen dump which I wrote for my latest release from John Penn, *Picture Maker*.

Listing 1 — The Converter Program

```
0 PCLEAR6:CLEAR200,&H3FFF:DIMS(3),E(3),NS(3),NE(3):CLOADM"SHRINKIT"
1 CLS:PRINT"POSITION THE TAPE FOR LOADING   THE NEXT TUNE.":MOTORON:AUDIOON
2 INPUT"FILENAME ";F$:MOTOROFF:AUDIOOFF:IFLEN(F$)>8THEN2ELSECLOSE:OPEN"I",#-1,F$
:INPUT#-1,F$:PRINT"LOADING '"F$"'":FORI=1TO5:INPUT#-1,T:NEXT:FORI=1TO13:INPUT#-1
,J:NEXT:CLOSE
3 FORI-0TO3:A$="CH"+STR$(I):CLOADMA$:S(I)=PEEK(487)*256+PEEK(488):E(I)=PEEK(126)
*256+PEEK(127)-1:PRINT"PLAYSHIFT FOR CHANNEL"I+1;:INPUTP:IFP THENFORJ-S(I)+1TOE(
I)STEP2:W=PEEK(J):POKEJ,W-P*(W>0):NEXTJ
4 NEXTI:SKIPFF$:PRINT"CONVERTING THE DATA ...":A=PEEK(&HBC)*256:FORJ=0TO3:NS(J)=
A:L=0:FORI=S(J)+1TOE(J)STEP2:IFPEEK(I)=0THENNEXTI ELSEN=PEEK(I):POKEA,&H81:POKEA
+1,86-N:POKEA+2,T:A=A+3
5 FORI=S(J)TOE(J)STEP2:V=&H80:IFPEEK(I+1)=0THENV=V+&H7C:GOTO8ELSEO=PEEK(I+1)-N:N
=N+O:IFO<-15ORO>14THENPOKEA,&H81:POKEA+1,86-N:O=0:POKEA+2,T:A=A+3
6 IFO>0THENV=V+4ELSEO=ABS(O)
7 V=V+O*8
8 IFL=PEEK(I)THENPOKEA,V:A=A+1ELSEL=PEEK(I):V=V+2:POKEA,V:POKEA+1,L:A=A+2
9 NEXTI:NE(J)=A:NEXTJ:PRINT"CONVERSION DONE - REPEATS NOW   BEING FOUND AND INSE
RTED.":Z=&H4334:FORI=0TO3:H=INT(NS(I)/256):L=NS(I)-256*H:POKEZ,H:Z=Z+1:POKEZ,L:Z
=Z+1
10 H=INT(NE(I)/256):L=NE(I)-256*H:POKEZ,H:Z=Z+1:POKEZ,L:Z=Z+1:NEXT:EXEC&H4000:CL
S:FORI=0TO3:S(I)=PEEK(Z)*256+PEEK(Z+1):E(I)=PEEK(Z+2)*256+PEEK(Z+3):Z=Z+4:NEXT:F
ORI=0TO3:PRINT"VALUE"I+1"="E(I):NEXT
11 PRINT"MAKE A NOTE OF THE ABOVE AND":PRINT"POSITION THE TAPE FOR SAVING THECON
VERTED TUNE THEN PRESS ENTER":MOTORON:AUDIOON:INPUTA$:PRINT"PRESS 'S' TO SAVE TH
E TUNE":MOTOROFF:AUDIOOFF
12 IFINKEY$<>"S"THEN12ELSEPRINT"SAVING '"F$"'":CSAVEMF$,S(0),E(3)-1,41194:PRINT"
DO YOU WISH TO CONVERT ANY MORE TUNES (Y/N) ?"
13 A$=INKEY$:IFA$="Y"THEN1ELSEIFA$="N"THENENDELSE13
```

Listing 2

```
4000                     OPG    $4000
4000   4F       START    CLRA
4001   B7 405A           STA    A
4004   B7 4270           STA    I
4007   8E 0000  ILOOP    LDX    #0
400A   BF 4103           STX    B
400D   C6 04             LDB    #4
400F   3D                MUL
4010   8E 4334           LDX    #SCNTBL
4013   30 8B             LEAX   D,X
4015 10AE 81             LDY    ,X++
4018 10BF 406B           STY    J
401C   EC 84             LDD    ,X
401E   FD 4073           STD    ENDSC
4021   83 0007           SUBD   #7
4024   FD 4269           STD    JCMP
4027   CE 0006  JLOOP    LDU    #6
402A   FF 4141           STU    C
402D   BE 4334           LDX    SCNTBL
4030   BF 40CB           STX    K
4033 10BF 4243           STY    KCMP
4037   7F 424A           CLR    PF
403A   B6 405A  KLOOP    LDA    A
403D   27 1E             BEQ    RPT1
403F   4F                CLRA
4040   CE 4354           LDU    #RPTBL
4043   AC C4    LLOOP    CMPX   ,U
4045   23 16             BLS    RPT1
4047   AC 42             CMPX   2,U
4049   24 0B             BHS    NEXTL
404B   EC 42             LDD    2,U
404D   83 0001           SUBD   #1
4050   FD 40CB           STD    K
4053   16 01E4           LBRA   NEXTK
4056   33 49    NEXTL    LEAU   9,U
4058   4C                INCA
4059   81 00             CMPA   #0
405A            A        EQU    *-1
405B   25 E6             BLO    LLOOP
405D   CC 0000  RPT1     LDD    #0
4060   FD 413D           STD    TL
4063   FD 4082           STD    D
4066 10BE 406B           LDY    J
406A   8C 0000  LOOP1    CMPX   #0
406B            J        EQU    *-2
406D 1024 00CB           LBHS   END1
4071 108C 0000           CMPY   #0
4073            ENDSC    EQU    *-2
4075 1024 00C3           LBHS   END1
4079   A6 80             LDA    ,X+
407B   A1 A0             CMPA   ,Y+
407D 1026 00BB           LBNE   END1
4081   CC 0000           LDD    #0
4082            D        EQU    *-2
4084   C3 0001           ADDD   #1
4087   FD 4082           STD    D
408A   B6 405A           LDA    A
408D   27 14             BEQ    RPT2
408F   4F                CLRA
4090   CE 4354           LDU    #RPTBL
4093   AC C4    MLOOP    CMPX   ,U
4095   23 0C             BLS    PPT2
4097   AC 42             CMPX   2,U
4099   25 CF             BLO    LOOP1
409B   33 49    NEXTM    LEAU   9,U
409D   4C                INCA
409E   B1 405A           CMPA   A
40A1   25 F0             BLO    MLOOF
40A3   86 01    RPT2     LDA    #1
40A5   B7 40E3           STA    E
40A8   FC 406B           LDD    J
40AB   FD 40C4           STD    F
40AE   FC 40C4  LOOP2    LDD    F
40B1   F3 4082           ADDD   D
40B4   FD 40C4           STD    F
40B7   F3 4082           ADDD   D
40BA 10B3 4073           CMPD   ENDSC
40BE   22 2E             BHI    END2
40C0   CC 0000           LDD    #0
40C3   CE 0000  LOOP3    LDU    #0
40C4            F        EQU    *-2
40C6   33 CB             LEAU   D,U
40C8   34 16             PSHS   D,X
40CA   8E 0000           LDX    #0
40CB            K        EQU    *-2
40CD   30 8B             LEAX   D,X
40CF   A6 C4             LDA    ,U
40D1   A1 84             CMPA   ,X
40D3   35 16             PULS   D,X
40D5   26 17             BNE    END2
40D7   C3 0001           ADDD   #1
40DA 10B3 4082           CMPD   D
40DE   25 E3             BLO    LOOP3
40E0   34 02             PSHS   A
40E2   86 00             LDA    #0
40E3            E        EQU    *-1
40E4   4C                INCA
40E5   B7 40E3           STA    E
40E8   81 FF             CMPA   #255
40EA   35 02             PULS   A
40EC   25 C0             BLO    LOOP2
40EE   B6 4082  END2     LDA    D
40F1   F6 40E3           LDB    E
40F4   3D                MUL
40F5   1E 89             EXG    A,B
40F7   34 06             PSHS   D
40F9   B6 4083           LDA    D+1
40FC   F6 40E3           LDB    E
40FF   3D                MUL
4100   E3 E1             ADDD   ,S++
4102   CE 0000           LDU    #0
4103            B        EQU    *-2
4105   27 1B             BEQ    CHKTL
4107   34 06             PSHS   D
4109   CE 4354           LDU    #RPTBL
410C   B6 405A           LDA    A
410F   4A                DECA
4110   C6 09             LDB    #9
4112   3D                MUL
4113   33 CB             LEAU   D,U
4115   EC 42             LDD    2,U
4117   A3 C4             SUBD   ,U
4119 10A3 E4             CMPD   ,S
411C   35 06             PULS   D
411E 1024 FF48           LBHS   LOOP1
4122 10B3 413D  CHKTL    CMPD   TL
4126 1023 FF40           LBLS   LOOP1
412A   FD 413D           STD    TL
412D   B6 40E3           LDA    E
4130   B7 4229           STA    E1
4133   FC 4082           LDD    D
4136   FD 4224           STD    D1
4139   16 FF2E           LBRA   LOOP1
413C   CC 0000  END1     LDD    #0
413D            TL       EQU    *-2
413F 1083 0000           CMPD   #0
4141            C        EQU    *-2
4143 1023 00F3           LBLS   NEXTK
4147   BE 4141           LDX    C
414A   8C 0007           CMPX   #7
414D   25 03             BLO    CHKB
414F   7A 405A           DEC    A
4152   BE 4103  CHKB     LDX    B
4155 1027 00A8           LBEQ   REPEAT
4159   8E 0000           LDX    #0
415C   BF 4103           STX    B
415F   8E 4354           LDX    #RPTBL
4162   B6 405A           LDA    A
4165   4A                DECA
4166   C6 09             LDB    #9
4168   3D                MUL
4169   30 8B             LEAX   D,X
416B   A6 08    LINE6    LDA    8,X
416D   4A                DECA
416E   27 0A             BEQ    SINGLE
4170   A7 08             STA    8,X
4172   EC 02             LDD    2,X
4174   A3 06             SUBD   6,X
4176   E3 04             ADDD   4,X
4178   20 0C             BRA    LINE7
417A   EC 06    SINGLE   LDD    6,X
417C   A3 02             SUBD   2,X
417E   F3 406B           ADDD   J
4181   ED 06             STD    6,X
4183   FC 406B           LDD    J
4186   ED 02    LINE7    STD    2,X
4188 10B3 406B           CMPD   J
418C   22 DD             BHI    LINE6
418E   A3 84             SUBD   ,X
4190 1083 0007           CMPD   #7
4194   25 68             BLO    DELETE
4196   BF 41F8           STX    RPTPS2
4199   CE 4354           LDU    #RPTBL
419C   EC 06    RPT8     LDD    6,X
419E 10A3 C4             CMPD   ,U
41A1   23 51             BLS    RPT9
41A3 10A3 42             CMPD   2,U
41A6   24 4C             BHS    RPT9
41A8   86 01             LDA    #1
41AA   A7 08             STA    8,X
41AC   EC C4             LDD    ,U
41AE   A3 04             SUBD   4,X
41B0   34 06             PSHS   D
41B2   EC 06             LDD    6,X
41B4   A3 C4             SUBD   ,U
41B6 10A3 E1             CMPD   ,S++
41B9   22 12             BHI    RPT10
41BB   EC C4             LDD    ,U
41BD   ED 06             STD    6,X
41BF   A3 04             SUBD   4,X
41C1 1083 0007  RPT11    CMPD   #7
41C5   25 37             BLO    DELETE
41C7   E3 84             ADDD   ,X
41C9   ED 02             STD    2,X
41CB   20 34             BRA    PEPEAT
41CD   34 06    RPT10    PSHS   D
```

Listing 2 continued

```
41CF   EC C4              LDD    ,U
41D1   A3 04              SUBD   4,X
41D3   E3 84              ADDD   ,X
41D5   ED 84              STD    ,X
41D7   EC 46              LDD    6,U
41D9   A3 44              SUBD   4,U
41DB 10A3 E4              CMPD   ,S
41DE   24 08              BHS    RPT15
41E0   32 62              LEAS   2,S
41E2   EC 46              LDD    6,U
41E4   A3 44              SUBD   4,U
41E6   34 06              PSHS   D
41E8   EC 44     RPT15    LDD    4,U
41EA   ED 04              STD    4,X
41EC   E3 E4              ADDD   ,S
41EE   ED 06              STD    6,X
41F0   35 06              PULS   D
41F2   20 CD              BRA    RPT11
41F4   33 49     RPT9     LEAU   9,U
41F6 1183 0000            CMPU   #0
41F8             RPTPS2   EQU    *-2
41FA   25 A0              BLO    RPT8
41FC   20 03              BRA    REPEAT
41FE   7A 405A   DELETE   DEC    A
4201   FE 413D   REPEAT   LDU    TL
4204   FF 4141            STU    C
4207   8E 4354            LDX    #RPTBL
420A   F6 405A            LDB    A
420D   86 09              LDA    #9
420F   3D                 MUL
4210   30 8B              LEAX   D,X
4212   FC 406B            LDD    J
4215   ED 81              STD    ,X++
4217   F3 4141            ADDD   C
421A   ED 81              STD    ,X++
421C   34 06              PSHS   D
421E   FC 40CB            LDD    K
4221   ED 81              STD    ,X++
4223   C3 0000            ADDD   #0
4224             D1       EQU    *-2
4226   ED 81              STD    ,X++
4228   86 00              LDA    #0
4229             E1       EQU    *-1
422A   A7 84              STA    ,X
422C   B7 424A            STA    RF
422F   7C 405A            INC    A
4232   35 06              PULS   D
4234 10B3 4269            CMPD   JCMP
4238   22 35              BHI    NEXTI
423A   BE 40CB   NEXTK    LDX    K
423D   30 01              LEAX   1,X
423F   BF 40CB            STX    K
4242   8C 0000            CMPX   #0
4243             KCMP     EQU    *-2
4245 1025 FDF1            LBLO   KLOOP
4249   86 00              LDA    #0
424A             RF       EQU    *-1
424B   27 06              BEQ    SMB
424D   BE 4141            LDX    C
4250   BF 4103            STX    B
4253   BE 4103   SMB      LDX    B
4256   27 05              BEQ    NEXTJ
4258   30 1F              LEAX   -1,X
425A   BF 4103            STX    B
425D 10BE 406B   NEXTJ    LDY    J
4261   31 21              LEAY   1,Y
4263 10BF 406B            STY    J
4267 108C 0000            CMPY   #0
4269             JCMP     EQU    *-2
426B 1025 FDB8            LBLO   JLOOP
426F   86 00     NEXTI    LDA    #0
4270             I        EQU    *-1
4271   4C                 INCA
4272   B7 4270            STA    I
4275   81 04              CMPA   #4
4277 1025 FD8C            LBLO   ILOOP
427B   8E 4334            LDX    #SCNTBL
427E   EE 84              LDU    ,X
4280   FF 429A            STU    O
```

```
4283   EC 0E              LDD    14,X
4285   FD 4324            STD    DONE
4288   5F                 CLRB
4289   FF 4316            STU    N
428C   F7 42C0            STB    X
428F   8E 4354            LDX    #RPTBL
4292   BF 42C7            STX    RPTPOS
4295   4F        L1       CLRA
4296   8E 4334            LDX    #SCNTBL
4299   CE 0000            LDU    #0
429A             O        EQU    *-2
429C 11A3 84     L2       CMPU   ,X
429F   26 09              BNE    L3
42A1   FE 4316            LDU    N
42A4   EF 8810            STU    +16,X
42A7   FE 429A            LDU    O
42AA 11A3 02     L3       CMPU   2,X
42AD   26 09              BNE    L4
42AF   FE 4316            LDU    N
42B2   EF 8812            STU    +18,X
42B5   FE 429A            LDU    O
42B8   4C        L4       INCA
42B9   30 04              LEAX   4,X
42BB   81 04              CMPA   #4
42BD   25 DD              BLO    L2
42BF   86 00              LDA    #0
42C0             X        EQU    *-1
42C1   B1 405A            CMPA   A
42C4   24 4F              BHS    L9
42C6   8E 0000            LDX    #0
42C7             RPTPOS   EQU    *-2
42C9 11A3 04     L5       CMPU   4,X
42CC   26 07              BNE    L6
42CE   FE 4316            LDU    N
42D1   EF 04              STU    4,X
42D3   20 0A              BRA    L7
42D5 11A3 06     L6       CMPU   6,X
42D8   26 08              BNE    L8
42DA   FE 4316            LDU    N
42DD   EF 06              STU    6,X
42DF   FE 429A   L7       LDU    O
42E2   4C        L8       INCA
42E3   30 09              LEAX   9,X
42E5   B1 405A            CMPA   A
42E8   25 DF              BLO    L5
42EA   BE 42C7            LDX    RPTPOS
42ED 11A3 84              CMPU   ,X
42F0   26 23              BNE    L9
42F2   FE 4316            LDU    N
42F5   4F                 CLRA
42F6   E6 08              LDB    8,X
42F8   ED C1              STD    ,U++
42FA   1F 30              TFR    U,D
42FC   A3 04              SUBD   4,X
42FE   ED C1              STD    ,U++
4300   1F 30              TFR    U,D
4302   A3 06              SUBD   6,X
4304   ED C1              STD    ,U++
4306   FF 4316            STU    N
4309   EE 02              LDU    2,X
430B   30 09              LEAX   9,X
430D   BF 42C7            STX    RPTPOS
4310   7C 42C0            INC    X
4313   20 0A              BRA    L10
4315   8E 0000   L9       LDX    #0
4316             N        EQU    *-2
4318   A6 C0              LDA    ,U+
431A   A7 80              STA    ,X+
431C   BF 4316            STX    N
431F   FF 429A   L10      STU    O
4322 1183 0000            CMPU   #0
4324             DONE     EQU    *-2
4326 1025 FF6B            LBLO   L1
432A   FC 4316            LDD    N
432D   8E 4334            LDX    #SCNTBL
4330   ED 881E            STD    30,X
4333   39                 RTS
4334             SCNTBL   RMB    32
4354             RPTBL    EQU    *
```

```
A      =405A   B      =4103   C      =4141   CHKB   =4152   CHKTL  =4122
D      =4082   D1     =4224   DELETE=41FE    DONE   =4324   E      =40E3
E1     =4229   END1   =413C   END2   =40EE   ENDSC  =4073   F      =40C4
I      =4270   ILOOP  =4007   J      =406B   JCMP   =4269   JLOOP  =4027
K      =40CB   KCMP   =4243   KLOOP  =403A   L1     =4295   L10    =431F
L2     =429C   L3     =42AA   L4     =42B8   L5     =42C9   L6     =42D5
L7     =42DF   L8     =42E2   L9     =4315   LINE6  =416B   LINE7  =4186
LLOOP  =4043   LOOP1  =406A   LOOP2  =40AE   LOOP3  =40C3   MLOOP  =4093
N      =4316   NEXTI  =426F   NEXTJ  =425D   NEXTK  =423A   NEXTL  =4056
NEXTM  =409B   O      =429A   REPEAT=4201    RF     =424A   RPT1   =405D
RPT10  =41CD   RPT11  =41C1   RPT15  =41E8   RPT2   =40A3   RPT8   =419C
RPT9   =41F4   RPTBL  =4354   RPTPOS=42C7    RPTPS2=41F8    SCNTBL=4334
SINGLE=417A    SMB    =4253   START  =4000   TL     =413D   X      =42C0
```

Listing 3 — Basic Source Update

```
0 CLEAR200,&H3FFF:L=2:FORI=&H4000 TO&H4333 STEP20:C=0:FORJ=I TOI+19:READA$:A=VAL
("&H"+A$):C=C+A:POKEJ,A:NEXTJ:READA$:IFC<>VAL("&H"+A$)THENPRINT"ERROR IN LINE"L:
ENDELSEL=L+1:NEXTI:PRINT"ALL DONE !!"
1 PRINT"NOW YOU CAN SAVE THE PROGRAM TO DISK OR TAPE AS YOU WISH.":PRINT"START A
DDRESS = &H4000":PRINT"END ADDRESS = &H4333":PRINT"EXEC ADDRESS = &H4000":PRINT"
NB. CALL IT 'SHRINKIT'":END
2 DATA 4F,B7,40,5A,B7,42,70,8E,00,00,BF,41,03,C6,04,3D,8E,43,34,30,06D6
3 DATA 8B,10,AE,81,10,BF,40,6B,EC,84,FD,40,73,83,00,07,FD,42,69,CE,0964
4 DATA 00,06,FF,41,41,BE,43,34,BF,40,CB,10,BF,42,43,7F,42,4A,B6,40,07DB
5 DATA 5A,27,1E,4F,CE,43,54,AC,C4,23,16,AC,42,24,0B,EC,42,83,00,01,06CB
6 DATA FD,40,CB,16,01,E4,33,49,4C,81,00,25,E6,CC,00,00,FD,41,3D,FD,089B
7 DATA 40,82,10,BE,40,6B,8C,00,00,10,24,00,CB,10,8C,00,00,10,24,00,0496
8 DATA C3,A6,80,A1,A0,10,26,00,BB,CC,00,00,C3,00,01,FD,40,82,B6,40,0860
9 DATA 5A,27,14,4F,CE,43,54,AC,C4,23,0C,AC,42,25,CF,33,49,4C,81,40,0783
10 DATA 5A,25,F0,86,01,B7,40,E3,FC,40,6B,FD,40,C4,FC,40,C4,F3,40,82,0B2D
11 DATA FD,40,C4,F3,40,82,10,B3,40,73,22,2E,CC,00,00,CE,00,00,33,CB,0814
12 DATA 34,16,8E,00,00,30,8B,A6,C4,A1,84,35,16,26,17,C3,00,01,10,B3,0631
13 DATA 40,82,25,E3,34,02,86,00,4C,B7,40,E3,81,FF,35,02,25,C0,B6,40,083E
14 DATA 82,F6,40,E3,3D,1E,89,34,06,B6,40,83,F6,40,E3,3D,E3,E1,CE,00,0A1A
15 DATA 00,27,1B,34,06,CE,43,54,B6,40,5A,4A,C6,09,3D,33,CB,EC,42,A3,0756
16 DATA C4,10,A3,E4,35,06,10,24,FF,48,10,B3,41,3D,10,23,FF,40,FD,41,0802
17 DATA 3D,B6,40,E3,B7,42,29,FC,40,82,FD,42,24,16,FF,2E,CC,00,00,10,0878
18 DATA 83,00,00,10,23,00,F3,BE,41,41,8C,00,07,25,03,7A,40,5A,BE,41,05B7
19 DATA 03,10,27,00,A8,8E,00,00,BF,41,03,8E,43,54,B6,40,5A,4A,C6,09,0601
20 DATA 3D,30,8B,A6,08,4A,27,0A,A7,08,EC,02,A3,06,E3,04,20,0C,EC,06,066C
21 DATA A3,02,F3,40,6B,ED,06,FC,40,6B,ED,02,10,B3,40,6B,22,DD,A3,84,0960
22 DATA 10,83,00,07,25,68,BF,41,F8,CE,43,54,EC,06,10,A3,C4,23,51,10,0771
23 DATA A3,42,24,4C,86,01,A7,08,EC,C4,A3,04,34,06,EC,06,A3,C4,10,A3,0828
24 DATA E1,22,12,EC,C4,ED,06,A3,04,10,83,00,07,25,37,E3,84,ED,02,20,07CB
25 DATA 34,34,06,EC,C4,A3,04,E3,84,ED,84,EC,46,A3,44,10,A3,E4,24,08,0979
26 DATA 32,62,EC,46,A3,44,34,06,EC,44,ED,04,E3,E4,ED,06,35,06,20,CD,08EA
27 DATA 33,49,11,83,00,00,25,A0,20,03,7A,40,5A,FE,41,3D,FF,41,41,8E,0697
28 DATA 43,54,F6,40,5A,86,09,3D,30,8B,FC,40,6B,ED,81,F3,41,41,ED,81,09A6
29 DATA 34,06,FC,40,CB,ED,81,C3,00,00,ED,81,86,00,A7,84,B7,42,4A,7C,0950
30 DATA 40,5A,35,06,10,B3,42,69,22,35,BE,40,CB,30,01,BF,40,CB,8C,00,06EA
31 DATA 00,10,25,FD,F1,86,00,27,06,BE,41,41,BF,41,03,BE,41,03,27,05,0647
32 DATA 30,1F,BF,41,03,10,BE,40,6B,31,21,10,BF,40,6B,10,8C,00,00,10,0543
33 DATA 25,FD,B8,86,00,4C,B7,42,70,81,04,10,25,FD,8C,8E,43,34,EE,84,08CF
34 DATA FF,42,9A,EC,0E,FD,43,24,5F,FF,43,16,F7,42,C0,8E,43,54,BF,42,0A0F
35 DATA C7,4F,8E,43,34,CE,00,00,11,A3,84,26,09,FE,43,16,EF,88,10,FE,082C
36 DATA 42,9A,11,A3,02,26,09,FE,43,16,EF,88,12,FE,42,9A,4C,30,04,81,077C
37 DATA 04,25,DD,86,00,B1,40,5A,24,4F,8E,00,00,11,A3,04,26,07,FE,43,05FE
38 DATA 16,EF,04,20,0A,11,A3,06,26,08,FE,43,16,EF,06,FE,42,9A,4C,30,06BD
39 DATA 09,B1,40,5A,25,DF,BE,42,C7,11,A3,84,26,23,FE,43,16,4F,E6,08,0834
40 DATA ED,C1,1F,30,A3,04,ED,C1,1F,30,A3,06,ED,C1,FF,43,16,EE,02,30,0970
41 DATA 09,BF,42,C7,7C,42,C0,20,0A,8E,00,00,A6,C0,A7,80,BF,43,16,FF,08AB
42 DATA 42,9A,11,83,00,00,10,25,FF,6B,FC,43,16,8E,43,34,ED,88,1E,39,0735
```

Listing 4

```
0 POKE&HBA,4:POKE&HB7,PEEK(&HBC):EXEC&HAA23:CLEAR200,(PEEK(&HBC)+8)*256-1:DIMP(4
):CLOADM"JUKEBOX":E=PEEK(487)*256+PEEK(488):IFE=&H7800 THENP=&H7AC0:N=0ELSEFORN=
1TO31:IFE=PEEK(&H7AB6+N*10)*256+PEEK(&H7AB7+N*10)THENP=&H7AC0+N*10ELSENEXT
1 CLS:P(0)=PEEK(&HBC)*256:FORI=1TO4:PRINT"VALUE"I;:INPUTP(I):NEXT:O=E-P(4):IFP(0
)+O<256*(PEEK(&HBC)+8)THENPRINT"SORRY - THAT TUNE IS TOO LONG.  TRY ANOTHER (Y/N
) ? ":GOTO4ELSEFORI=0TO4:P(I)=P(I)+O:H=INT(P(I)/256):POKEP,H:POKEP+1,P(I)-256*H:
P=P+2:NEXT
2 E=P(0):PRINT"POSITION THE TAPE TO LOAD THE    NEXT TUNE.":MOTORON:AUDIOON
3 INPUT"FILENAME";F$:MOTOROFF:AUDIOOFF:IFLEN(F$)>8THEN3ELSEPRINT"LOADING TUNE"N"
- "F$:CLOADMF$,O:N=N+1:IFN<31THENPRINT"ANY MORE TUNES (Y/N) ?"ELSEPRINT"THAT'S A
S MANY AS YOU CAN PUT ONONE JUKEBOX !!":GOTO5
4 F$=INKEY$:IFF$="Y"THEN1ELSEIFF$<>"N"THEN4
5 PRINT"POSITION THE TAPE TO SAVE THE   JUKEBOX AND PRESS A KEY.":MOTORON:AUDIOO
N
6 IFINKEY$=""THEN6ELSEAUDIOOFF:PRINT"SAVING THE JUKEBOX":CSAVEM"JUKEBOX",E,&H7FF
F,&H7A17:CLS:PRINT"REMEMBER TO POKE THE NUMBER OF  THE TUNE YOU WISH TO HEAR AT"
:PRINT"118 AND THE NUMBER OF TIMES YOU WISH TO HEAR IT AT 119 BEFORE   YOU EXEC
JUKEBOX !"
7 PRINT"NB. THE FIRST TUNE YOU LOADED ISTUNE 0 NOT TUNE 1 AND POKEING 0 AT 119 M
EANS PLAY THE TUNE AD   INFINATUM. PRESSING SHIFT OR":PRINT"CLEAR AND @ OR W STO
PS THE TUNE PLAYING. THIS COPY OF JUKEBOX   STARTS AT "E", FOR ALL"
8 PRINT"COPIES THE END ADDRESS IS 32767 AND THE EXEC ADDRESS IS 31255.":END
```

Listing 5

```
7800              0               ORG    $7800
7800  8C 0000     1     PLAY      CMPX   #0       4
7801              2     XCMP1     EQU    *-2
7803  26 03       3               BNE    SM1      3
7805  17 0137     4               LBSR   NEW1
7808  CC 0000     5     SM1       LDD    #0       3
780B  C3 0000     6               ADDD   #0       4
780E  DD 09       7               STD    <$9      5
7810  97 4A       8               STA    <$4A     4
7812  8C 0000     9               CMPX   #0       4
7813              10    XCMP2     EQU    *-2
7815  26 03       11              BNE    SM2      3
7817  17 012B     12              LBSR   NEW2
781A  CC 0000     13    SM2       LDD    #0       3
781D  C3 0000     14              ADDD   #0       4
7820  DD 1B       15              STD    <$1B     5
7822  97 4D       16              STA    <$4D     4
7824  8C 0000     17              CMPX   #0       4
7825              18    XCMP3     EQU    *-2
7827  26 03       19              BNE    SM3      3
7829  17 011F     20              LBSR   NEW3
782C  CC 0000     21    SM3       LDD    #0       3
782F  C3 0000     22              ADDD   #0       4
7832  DD 2D       23              STD    <$2D     5
7834  97 50       24              STA    <$50     4
7836  8C 0000     25              CMPX   #0       4
7837              26    XCMP4     EQU    *-2
7839  26 03       27              BNE    SM4      3
783B  17 0113     28              LBSR   NEW4
783E  CC 0000     29    SM4       LDD    #0       3
```

Listing 5 continued

```
7841   C3 0000              ADDD   #0      4
7844   DD 3F                STD    <$3F    5
7846   97 53                STA    <$53    4
7848   B6 7C00              LDA    $7C00   5
7849                WAVE1   EQU    *-2
784B   BB 7D00              ADDA   $7D00   5
784C                WAVE2   EQU    *-2
784E   BB 7E00              ADDA   $7E00   5
784F                WAVE3   EQU    *-2
7851   BB 7F00              ADDA   $7F00   5
7852                WAVE4   EQU    *-2
7854   B7 FF20              STA    $FF20   5
7857   30 01                LEAX   1,X     5
7859   0E 00                JMP    <0      3
785B                TBLE    FDB    $864C
$7EC2,$77A5,$70EE,$6A97,$649C,$5E, F6,$59A2,$549A,
$4FDA,$4B5F,$4724
785B   86 4C
785D   7E C2
785F   77 A5
7861   70 EE
7863   6A 97
7865   64 9C
7867   5E F6
7869   59 A2
786B   54 9A
786D   4F DA
786F   4B 5F
7871   47 24
7873                TBLE1   FDB    RPTBL1
7873   78 97
7875                        RMB    3
7878                        FDB    XCMP1,RPTBL1
7878   78 01
787A   78 97
787C                TBLE2   FDB    RPTBL2
787C   78 C1
787E                        RMB    3
7881                        FDB    XCMP2,RPTBL2
7881   78 13
7883   78 C1
7885                TBLE3   FDB    RPTBL3
7885   78 EB
7887                        RMB    3
788A                        FDB    XCMP3,RPTBL3
788A   78 25
788C   78 EB
788E                TBLE4   FDB    RPTBL4
788E   79 15
7890                        RMB    3
7893                        FDB    XCMP4,RPTBL4
7893   78 37
7895   79 15
7897                RPTBL1  RMB    42
78C1                RPTBL2  RMB    42
78EB                RPTBL3  RMB    42
7915                RPTBL4  RMB    42
793F   31 8DFF30    NEW1    LEAY   TBLE1,PCR
7943   20 10                BRA    NEXT
7945   31 8DFF33    NEW2    LEAY   TBLE2,PCR
7949   20 0A                BRA    NEXT
794B   31 8DFF36    NEW3    LEAY   TBLE3,PCR
794F   20 04                BRA    NEXT
7951   31 8DFF39    NEW4    LEAY   TBLE4,PCR
7955   86 7D        NEXT    LDA    #$7D
7957   B7 FF02              STA    $FF02
795A   B6 FF00              LDA    $FF00
795D   84 50                ANDA   #$50
795F   26 02                BNE    NEXTOK
7961   35 90                PULS   X,PC
7963   34 10        NEXTOK  PSHS   X
7965   8D 6D        NEXTVL  BSR    GTBYTE
7967   44                   LSRA
7968   24 0B                BCC    NOTE
796A   8D 68                BSR    GTBYTE
796C   A7 22                STA    2,Y
796E   8D 64                BSR    GTBYTE
7970   A7 8C1E              STA    $1E,PC
7973   20 F0                BRA    NEXTVL
7975   44           NOTE    LSRA
7976   24 21                BCC    SMLEN
7978   A7 8C1D              STA    $1D,PC
797B   4F                   CLRA
797C   5F                   CLRB
797D   ED 23                STD    3,Y
797F   8D 53        LINK    BSR    GTBYTE
7981   81 FF                CMPA   #255
7983   26 0B                BNE    LINKE
7985   4A                   DECA
7986   E6 8C08              LDB    8,PC
7989   3D                   MUL
798A   E3 23                ADDD   3,Y
798C   ED 23                STD    3,Y
798E   20 EF                BRA    LINK
7990   C6 50        LINKE   LDB    #80
7991                TEMPO   EQU    *-1
7992   3D                   MUL
7993   E3 23                ADDD   3,Y
7995   ED 23                STD    3,Y
7997   86 00                LDA    #0
7998                STO     EQU    *-1
7999   81 3F        SMLEN   CMPA   #63
799B   26 04                BNE    NOTE1
799D   4F                   CLRA
799E   5F                   CLRB
799F   20 27                BRA    REST
79A1   84 1F        NOTE1   ANDA   #$1F
79A3   44                   LSRA
79A4   24 01                BCC    INCNT
79A6   40                   NEGA
79A7   AB 22        INCNT   ADDA   2,Y
79A9   A7 22                STA    2,Y
79AB   5F                   CLRB
79AC   5C           GETOCT  INCB
79AD   81 0C                CMPA   #12
79AF   25 04                BLO    OCT
79B1   80 0C                SUBA   #12
79B3   20 F7                BRA    GETOCT
79B5   30 8DFEA2    OCT     LEAX   TBLE,PCR
79B9   34 04                PSHS   B
79BB   48                   LSLA
79BC   EC 86                LDD    A,X
79BE   6A E4        GETNTE  DEC    ,S
79C0   27 04                BEQ    GOTIT
79C2   44                   LSRA
79C3   56                   RORB
79C4   20 F8                BRA    GETNTE
79C6   32 61        GOTIT   LEAS   1,S
79C8   AE 25        REST    LDX    5,Y
79CA   ED 0B                STD    +$B,X
79CC   EC 23                LDD    3,Y
79CE   E3 E4                ADDD   ,S
79D0   ED 84                STD    ,X
79D2   35 90                PULS   X,PC
79D4   AE A4        GTBYTE  LDX    ,Y
79D6   EE 84        NOTEND  LDU    ,X
79D8 11A3 02                CMPU   2,X
79DB   25 19                BLO    NTRPTE
79DD   A6 06                LDA    6,X
79DF   27 0D                BEQ    CONT
79E1   4A                   DECA
79E2   26 0A                BNE    CONT
79E4   30 19                LEAX   -7,X
79E6   AF A4                STX    ,Y
79E8   AC 27                CMPX   7,Y
79EA   24 EA                BHS    NOTEND
79EC   35 F0                PULS   X,U,Y,PC
79EE   A7 06        CONT    STA    6,X
79F0   EE 04                LDU    4,X
79F2   EF 84                STU    ,X
79F4   20 E0                BRA    NOTEND
79F6   A6 C0        NTRPTE  LDA    ,U+
79F8   27 03                BEQ    REPEAT
79FA   EF 84                STU    ,X
79FC   39                   RTS
79FD   30 07        REPEAT  LEAX   7,X
79FF   AF A4                STX    ,Y
7A01   37 02                PULU   A
7A03   A7 06                STA    6,X
7A05   1F 30                TFR    U,D
7A07   A3 C1                SUBD   ,U++
7A09   ED 84                STD    ,X
```

Listing 5 continued

```
7A0B   ED 04                    STD     4,X
7A0D   1F 30                    TFR     U,D
7A0F   A3 C1                    SUBD    ,U++
7A11   EF 19                    STU     -7,X
7A13   ED 02                    STD     2,X
7A15   20 BF                    BRA     NOTEND
7A17   1F B8           START    TFR     DP,A
7A19   A7 8D009F                STA     DPSTO,PCR
7A1D   1F 50                    TFR     PC,D
7A1F   83 7A1F         COMP     SUBD    #COMP
7A22   5D                       TSTB
7A23   27 01                    BEQ     OK
7A25   39                       RTS
7A26   8B 78           OK       ADDA    #$78
7A28   1F 8B                    TFR     A,DP
7A2A   D6 7B                    LDB     <$7B
7A2C   DD 73                    STD     <$73
7A2E   97 78                    STA     <$78
7A30   97 7A                    STA     <$7A
7A32   D6 84                    LDB     <$84
7A34   DD 7C                    STD     <$7C
7A36   97 81                    STA     <$81
7A38   97 83                    STA     <$83
7A3A   D6 8D                    LDB     <$8D
7A3C   DD 85                    STD     <$85
7A3E   97 8A                    STA     <$8A
7A40   97 8C                    STA     <$8C
7A42   97 93                    STA     <$93
7A44   4C                       INCA
7A45   D6 96                    LDB     <$96
7A47   DD 8E                    STD     <$8E
7A49   97 95                    STA     <$95
7A4B   8B 03                    ADDA    #3
7A4D   97 49                    STA     <$49
7A4F   8B 01                    ADDA    #1
7A51   97 4C                    STA     <$4C
7A53   8B 01                    ADDA    #1
7A55   97 4F                    STA     <$4F
7A57   8B 01                    ADDA    #1
7A59   97 52                    STA     <$52
7A5B   DE 73                    LDU     <$73
7A5D   31 8D005F                LEAY    TUNES,PCR
7A61   80 7F                    SUBA    #$7F
7A63   34 02                    PSHS    A
7A65   B6 0076                  LDA     >$76
7A68   C6 0A                    LDB     #10
7A6A   3D                       MUL
7A6B   31 AB                    LEAY    D,Y
7A6D   C6 04                    LDB     #4
7A6F   AE A1           LOOP     LDX     ,Y++
7A71   AF C4                    STX     ,U
7A73   AF 44                    STX     4,U
7A75   A6 C4                    LDA     ,U
7A77   AB E4                    ADDA    ,S
7A79   A7 44                    STA     4,U
7A7B   A7 C1                    STA     ,U++
7A7D   AE A4                    LDX     ,Y
7A7F   AF C4                    STX     ,U
7A81   A6 C4                    LDA     ,U
7A83   AB E4                    ADDA    ,S
7A85   A7 C4                    STA     ,U
7A87   B6 0077                  LDA     >$77
7A8A   A7 44                    STA     4,U
7A8C   33 C828                  LEAU    40,U
7A8F   5A                       DECB
7A90   26 DD                    BNE     LOOP
7A92   32 61                    LEAS    1,S
7A94   1A 50                    ORCC    #$50
7A96   B6 FF01                  LDA     $FF01
7A99   84 F7                    ANDA    #$F7
7A9B   B7 FF01                  STA     $FF01
7A9E   B6 FF03                  LDA     $FF03
7AA1   84 F7                    ANDA    #$F7
7AA3   B7 FF03                  STA     $FF03
7AA6   B6 FF23                  LDA     $FF23
7AA9   8A 08                    ORA     #8
7AAB   B7 FF23                  STA     $FF23
7AAE   8E 0000                  LDX     #0
7AB1   9F 01                    STX     <$1
7AB3   9F 13                    STX     <$13
7AB5   9F 25                    STX     <$25
7AB7   9F 37                    STX     <$37
7AB9   9D 00                    JSR     <0
7ABB   86 00           RSTDP    LDA     #0
7ABC                   DPSTO    EQU     *-1
7ABD   1F 8B                    TFR     A,DP
7ABF   39                       RTS
7AC0                   TUNES    RMB     320
7C00                   ZZZ      END     START
```

Listing 6

```
0 CLEAP200,&H7FFF:L=4:FOPI=&H7800 TO&H7896 STEP20:C=0:FORJ=I TOI+19:READA$:A=VAL
("&H"+A$):C=C+A:POKEJ,A:NEXTJ:PEADA$:IFC<>VAL("&H"+A$)THENPRINT"EPPOP IN LINE"L:
ENDELSEL=L+1:NEXTI
1 FOPI=&H793F TO&H7ABF STEP20:C=0:FORJ=I TOI+19:READA$:A=VAL("&H"+A$):C=C+A:POKE
J,A:NEXTJ:PEADA$:IFC<>VAL("&H"+A$)THENPPINT"EPPOP IN LINE"L:ENDELSEL=L+1:NEXTI:P
PINT"ALL DATA CORPECT - INSEPTING    SOME WAVES PLEASE WAIT"
2 FOPI=0TO255:A=32+31.5*SIN(3.14159*I/128):FOPJ=&H7C00 TO&H7F00 STEP256:POKEI+J,
A:NEXTJ,I:PPINT"NOW YOU CAN SAVE THE PPOGPAM TO DISK OP TAPE AS YOU WISH.":PPINT
"START ADDPESS = &H7800":PPINT"END ADDPESS = &H7FFF":PPINT"EXEC ADDPESS = &H7A17
"
3 PPINT"NB. CALL IT 'JUKEBOX'":END
4 DATA 8C,00,00,26,03,17,01,37,CC,00,00,C3,00,00,DD,09,97,4A,8C,00,04F6
5 DATA 00,26,03,17,01,2B,CC,00,00,C3,00,00,DD,1B,97,4D,8C,00,00,26,0489
6 DATA 03,17,01,1F,CC,00,00,C3,00,00,DD,2D,97,50,8C,00,00,26,03,17,0486
7 DATA 01,13,CC,00,00,C3,00,00,DD,3F,97,53,B6,7C,00,BB,7D,00,BB,7E,074C
8 DATA 00,BB,7F,00,B7,FF,20,30,01,0E,00,86,4C,7E,C2,77,A5,70,EE,6A,0845
9 DATA 97,64,9C,5E,F6,59,A2,54,9A,4F,DA,4B,5F,47,24,78,97,00,00,00,0821
10 DATA 78,01,78,97,78,C1,00,00,00,78,13,78,C1,78,EB,00,00,00,78,25,0685
11 DATA 78,EB,79,15,00,00,00,78,37,79,15,00,00,00,00,00,00,00,00,00,032E
12 DATA 31,8D,FF,30,20,10,31,8D,FF,33,20,0A,31,8D,FF,36,20,04,31,8D,070C
13 DATA FF,39,86,7D,B7,FF,02,B6,FF,00,84,50,26,02,35,90,34,10,8D,6D,08A7
14 DATA 44,24,0B,8D,68,A7,22,8D,64,A7,8C,1E,20,F0,44,24,21,A7,8C,1D,075C
15 DATA 4F,5F,ED,23,8D,53,81,FF,26,0B,4A,E6,8C,08,3D,E3,23,ED,23,20,0886
16 DATA EF,C6,50,3D,E3,23,ED,23,86,00,81,3F,26,04,4F,5F,20,27,84,1F,0760
17 DATA 44,24,01,40,AB,22,A7,22,5F,5C,81,0C,25,04,80,0C,20,F7,30,8D,0610
18 DATA FE,A2,34,04,48,EC,86,6A,E4,27,04,44,56,20,F8,32,61,AE,25,ED,0910
19 DATA 0B,EC,23,E3,E4,ED,84,35,90,AE,A4,EE,84,11,A3,02,25,19,A6,06,097B
20 DATA 27,0D,4A,26,0A,30,19,AF,A4,AC,27,24,EA,35,F0,A7,06,EE,04,EF,07DE
21 DATA 84,20,E0,A6,C0,27,03,EF,84,39,30,07,AF,A4,37,02,A7,06,1F,30,077F
22 DATA A3,C1,ED,84,ED,04,1F,30,A3,C1,EF,19,ED,02,20,BF,1F,B8,A7,8D,0A5A
23 DATA 00,9F,1F,50,83,7A,1F,5D,27,01,39,8B,78,1F,8B,D6,7B,DD,73,97,07CD
24 DATA 78,97,7A,D6,84,DD,7C,97,81,97,83,D6,8D,DD,85,97,8A,97,8C,97,0C0E
25 DATA 93,4C,D6,96,DD,8E,97,95,8B,03,97,49,8B,01,97,4C,8B,01,97,4F,0931
26 DATA 8B,01,97,52,DE,73,31,8D,00,5F,80,7F,34,02,B6,00,76,C6,0A,3D,0751
27 DATA 31,AB,C6,04,AE,A1,AF,C4,AF,44,A6,C4,AB,E4,A7,44,A7,C1,AE,A4,0BF9
28 DATA AF,C4,A6,C4,AB,E4,A7,C4,B6,00,77,A7,44,33,C8,28,5A,26,DD,32,0AA1
29 DATA 61,1A,50,B6,FF,01,84,F7,B7,FF,01,B6,FF,03,84,F7,B7,FF,03,B6,0B55
30 DATA FF,23,8A,08,B7,FF,23,8E,00,00,9F,01,9F,13,9F,25,9F,37,9D,00,07A4
31 DATA 86,00,1F,8B,39,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0169
```

Listing 7

```
0 PCLEAR1:CLEAR400,&H25FF:CLOADM"WAVEMIX":CLS:PRINT"POSITION THE TAPE TO LOAD TH
E   JUKEBOX AND PRESS A KEY":MOTORON:AUDIOON:GOSUB6:AUDIOOFF:CLOADM"JUKEBOX":DIM
A$(23):PMODE0:CLS:PRINT@224,"PLEASE WAIT - SETTING UP THE    INITIAL 'BASE' WAVE
 FORMS."
1 JS=PEEK(487)*256+PEEK(488):PCLS:SCREEN,1:FORW=0TO15:READT,P:GOSUB2:NEXT:GOTO13
2 POKE&H2628,&H30+W:IFT=0THENA$(W)="SQUARE "+STR$(P):POKE&H2601,P:EXEC&H2600 ELSE
IFT=1THENA$(W)="SAWTOOTH ":POKE&H262B,1:EXEC&H2627 ELSEIFT=2THENA$(W)="TRIANGULA
R ":POKE&H262B,0:EXEC&H2627
3 IFT<3THENRETURNELSEA$(W)="SINE "+STR$(P):X=&H3000+W*256:FORI=0TO255:POKEI+X,128
+127.5*SIN(3.14159*P*I/128):NEXT:RETURN
4 CLS:PRINT"THE CURRENT BASE WAVES ARE :-":PRINT:FORI=0TO15:PRINTCHR$(8)I+1;A$(I
)," ";:NEXT:PRINT:RETURN
5 CLS:PRINT"THE CURRENT BUILT WAVES ARE :-  ":FORI=16TO23:PRINTCHR$(8)I-15;A$(I)
,"";:NEXT:PRINT:RETURN
6 POKE143,1:EXEC48053:A$=INKEY$:IFA$=""THEN6ELSERETURN
7 CLS:PRINT"POSITION THE TAPE":MOTORON:AUDIOON:INPUT"FILENAME ";A$:IFLEN(A$)>8TH
EN7ELSEMOTOROFF:AUDIOOFF:RETURN
8 PRINT"WOULD YOU LIKE TO "B$" (Y/N) ? ";
9 GOSUB6:IFA$="Y"ORA$="N"THENPRINTA$:RETURNELSE9
10 INPUT"WHICH ";A:IFA<1ORA>MX THEN10ELSEX=X+(A-1)*256:RETURN
11 PRINT"WHICH CHANNEL (1-4) ? ";
12 GOSUB6:IFA$<"1"ORA$>"4"THEN12ELSEPRINTA$:W=&H7B00+VAL(A$)*256:RETURN
13 CLS:PRINT@14,"MENU":PRINT@66,"0. ALTER A 'BASE' WAVE.":PRINT"  1. BUILD A NEW
 WAVE.":PRINT"  2. VIEW A WAVE.":PRINT"  3. VIEW THE 4 WAVES IN USE.":PRINT"  4.
 USE A WAVE ON A CHANNEL.":PRINT"  5. PLAY A TUNE.":PRINT"  6. SAVE THE BUILT WA
VES."
14 PRINT"  7. SAVE THE BASE WAVES.":PRINT"  8. SAVE THE JUKEBOX WITH IT'S      N
EW WAVE FORMS.":PRINT"  9. LOAD SOME WAVES.":PRINT:PRINT"WHAT NEXT ? ";
15 GOSUB6:IFA$<"0"ORA$>"9"THEN15ELSEONVAL(A$)+1GOSUB16,21,26,32,33,39,42,43,44,4
5:GOTO13
16 GOSUB4:INPUT"ALTER WHICH (1-16) ";A:IFA<1ORA>16THEN16ELSEW=A-1:PRINT"PLEASE S
ELECT THE NEW TYPE :-":PRINT"0. SQUARE","1. SAWTOOTH","2. TRIANGULAR","3. SINE",
"WHICH (0-3) ? ";
17 GOSUB6:IFA$<"0"ORA$>"3"THEN17ELSEPRINTA$:T=VAL(A$):IFT=1ORT=2THEN20ELSECLS
18 IFT THENINPUT"HARMONIC (1-16) ";P:IFP<1ORP>16THEN18ELSE20
19 INPUT"PULSE WIDTH (1-255) ";P:IFP<1ORP>255THEN19
20 GOSUB2:B$="ALTER ANY MORE'BASE' WAVES":GOSUB8:IFA$="Y"THEN16ELSERETURN
21 A=0:CLS:PRINT"NOW SELECT THE RELATIVE AMOUNTS OF THE BASE WAVES TO BE USED :-
":FORI=0TO15:PRINT"WAVE"I+1;A$(I):PRINT"AMOUNT ( 0 -";:IFA>1THENPRINT257-A") ";E
LSEPRINT255") ";
22 INPUTB:IFB<0ORA+B>257THEN22ELSEIFA=0ANDB>255THEN22ELSEPOKE&H2756+I,B:A=A+B:NE
XT:GOSUB5
23 INPUT"WHICH OF THESE DO YOU WISH TO   REPLACE (1-8) ";W:IFW<1ORW>8THEN23ELSEP
OKE&H26FB,&H27+W:PRINT"PLEASE NAME THE NEW WAVE IN LESSTHAN 13 LETTERS :-":W=W+1
5:A$(W)=""
24 FORI=1TO12:GOSUB6:PRINTA$;:IFA$<>CHR$(13)THENA$(W)=A$(W)+A$:NEXT
25 CLS:PRINT"THE NAME OF THE NEW WAVE IS":PRINT"    '"A$(W)"'":B$="TYPE IN A NEW
 ONE":GOSUB8:IFA$="Y"THENA$(W)="":GOTO24ELSEEXEC&H2642:B$="BUILD ANY MOREWAVES":
GOSUB8:IFA$="Y"THEN21ELSERETURN
26 CLS:PRINT"SELECT WHAT YOU WISH TO VIEW :- ":PRINT@98,"1. A 'BASE' WAVE.":PRIN
T"  2. A BUILT WAVE.":FORI=1TO4:PRINT" "I+2;CHR$(8)". THE WAVE FOR CHANNEL"I;CHR
$(8)".":NEXT:PRINT@320,"WHICH (1-6) ? ";
27 GOSUB6:IFA$<"1"ORA$>"6"THEN27ELSEPRINTA$:A=VAL(A$):IFA=1THENX=&H3000:GOSUB4:M
X=16ELSEIFA=2THENX=&H2800:GOSUB5:MX=8ELSE29
28 GOSUB10:SCREEN1:LINE-(0,PEEK(X)*3/4),PRESET:FORI=1TO255:LINE-(I,PEEK(X+I)*3/4
),PSET:GOTO30
29 W=&H7C00+256*(A-3):SCREEN1:LINE-(0,PEEK(W)*3),PRESET:FORI=1TO255:LINE-(I,PEEK
(I+W)*3),PSET
30 NEXT:GOSUB6:IFA$=" "THENPCLS
31 B$="VIEW ANY MORE WAVES":GOSUB8:IFA$="Y"THEN26ELSERETURN
32 SCREEN1:FORI=0TO3:LINE-(0,PEEK(&H7C00+I*256)*3/4+I*48),PRESET:FORJ=1TO255:LIN
E-(J,PEEK(&H7C00+I*256+J)*3/4+I*48),PSET:NEXTJ,I:GOSUB6:PCLS:RETURN
33 CLS:PRINT"WHICH DO YOU WISH TO USE :-":PRINT@98,"1. A BUILT WAVE":PRINT"  2.
A BASE WAVE":PRINT"  3. A WAVE ALREADY IN USE":PRINT:PRINT"PLEASE SELE
A BASE WAVE":PRINT"  3. A WAVE ALREADY IN USE":PRINT:PRINT"PLEASE SELECT (1-3) ?
 ";
34 GOSUB6:IFA$="1"THENX=&H2800:GOSUB5:MX=8ELSEIFA$="2"THENX=&H3000:GOSUB4:MX=16E
LSEPRINT:PRINT"FROM ";:GOTO37ELSE34
35 GOSUB10:PRINT"USE IT ON ";:GOSUB11
36 INPUT"VOLUME (1-63) ";MX:IFMX<1ORMX>63THEN36ELSEFORI=0TO255:POKEW+I,32+(PEEK(
X+I)-127.5)*MX/255:NEXT:GOTO38
37 GOSUB11:X=W:PRINT"USE IT ON ";:GOSUB11:FORI=0TO255:POKEW+I,PEEK(X+I):NEXT
38 B$="USE ANY MORE  WAVES":GOSUB8:IFA$="Y"THEN33ELSERETURN
39 CLS:INPUT"TUNE NUMBER (0-31) ";A:IFA<0ORA>31THEN39ELSEINPUT"REPEATS (1-255,0=
CONT.) ";B:IFB<0ORB>255THEN39ELSEPOKE118,A:POKE119,B:PRINT@224,"PRESS SHIFT OR C
LEAR AND 'Q' OR 'W' TO STOP THE TUNE":EXEC&H7A17
40 B$="HEAR ANOTHER  TUNE":GOSUB8:IFA$="Y"THEN39ELSERETURN
41 GOSUB7:PRINT"PRESS A KEY TO SAVE '"A$"'":B$=A$:GOSUB6:CLS:PRINT@266,"SAVING '
"B$"'":RETURN
42 GOSUB41:CSAVEMB$,&H2800,&H2FFF,41194:RETURN
43 GOSUB41:CSAVEMB$,&H3000,&H3FFF,41194:RETURN
44 GOSUB41:CSAVEMB$,JS,&H7FFF,&H7A17:RETURN
45 GOSUB7:PRINT"PRESS A KEY TO LOAD '"A$"'":B$=A$:GOSUB6:CLS:PRINT@266,"LOADING
'"B$"'":CLOADMB$:RETURN
46 DATA3,1,3,2,3,3,3,4,3,5,3,6,3,7,3,8,3,9,3,10,3,11,3,12,3,13,0,128,1,0,2,0
```

Listing 8

```
2600                      ORG    $2600
2600  86 40      SQUARE   LDA    #$40
2601             PLSWDT   EQU    *-1
2602  B7 260F             STA    STO1
2605  40                  NEGA
2606  B7 2613             STA    STO2
2609  BE 2628             LDX    WVPS
260C  1F 10               TFR    X,D
260E  C1 00      SQR1     CMPB   #0
260F             STO1     EQU    *-1
2610  25 0B               BLO    SQRMX
2612  C1 00               CMPB   #0
2613             STO2     EQU    *-1
2614  25 03               BLO    SQRHF
2616  5F                  CLRB
2617  20 06               BRA    SQRV
2619  C6 80      SQRHF    LDB    #$80
261B  20 02               BRA    SQRV
261D  C6 FF      SQRMX    LDB    #$FF
261F  E7 80      SQRV     STB    ,X+
2621  1F 10               TFR    X,D
2623  5D                  TSTB
2624  26 E8               BNE    SQR1
2626  39                  RTS
2627  8E 3000    TRISWT   LDX    #$3000
2628             WVPS     EQU    *-2
262A  86 00               LDA    #0
262B             TSF      EQU    *-1
262C  34 02               PSHS   A
262E  1F 10               TFR    X,D
```

Listing 8 continued

```
2630    6D E4            WV       TST    ,S
2632    26 05                     BNE    SWTWV
2634    5D                        TSTB
2635    2A 01                     BPL    TRI1
2637    53                        COMB
2638    58               TRI1     LSLB
2639    E7 80            SWTWV    STB    ,X+
263B    1F 10                     TFR    X,D
263D    5D                        TSTB
263E    26 F0                     BNE    WV
2640    35 82                     PULS   A,PC
2642    8E 3000          START    LDX    #$3000
2645    4F                        CLRA
2646    5F                        CLRB
2647    FD 2668                   STD    MAXVL
264A    4A                        DECA
264B    5A                        DECB
264C    FD 2671                   STD    MINVL
264F    CE 2756          L1       LDU    #VALS
2652    4F                        CLRA
2653    5F                        CLRB
2654    34 06            L2       PSHS   D
2656    A6 C0                     LDA    ,U+
2658    E6 84                     LDB    ,X
265A    3D                        MUL
265B    E3 E1                     ADDD   ,S++
265D    30 890100                 LEAX   256,X
2661    8C 4000                   CMPX   #$4000
2664    25 EE                     BLO    L2
2666 1083 0000                    CMPD   #0
2668                     MAXVL    EQU    *-2
266A    23 03                     BLS    MXOK
266C    FD 2668                   STD    MAXVL
266F 1083 FFFF           MXOK     CMPD   #$FFFF
2671                     MINVL    EQU    *-2
2673    24 03                     BHS    L3
2675    FD 2671                   STD    MINVL
2678    30 89F001        L3       LEAX   -$FFF,X
267C    8C 3100                   CMPX   #$3100
267F    25 CE                     BLO    L1
2681    FC 2668                   LDD    MAXVL
2684    B3 2671                   SUBD   MINVL
2687    B7 269A                   STA    DVSR1
268A    F7 26A9                   STB    DVSR2
268D    CC 8000                   LDD    #$8000
2690    FD 26DF                   STD    CURVL
2693    B7 269C                   STA    VL1
2696    F7 26A4                   STB    VL2
2699    86 00            DIV1     LDA    #0
269A                     DVSR1    EQU    *-1
269B    C6 00                     LDB    #0
269C                     VL1      EQU    *-1
269D    3D                        MUL
269E    34 06                     PSHS   D
26A0    B6 269A                   LDA    DVSR1
26A3    C6 00                     LDB    #0
26A4                     VL2      EQU    *-1
26A5    3D                        MUL
26A6    34 06                     PSHS   D
26A8    86 00                     LDA    #0
26A9                     DVSR2    EQU    *-1
26AA    F6 269C                   LDB    VL1
26AD    3D                        MUL
26AE    34 06                     PSHS   D
26B0    B6 26A9                   LDA    DVSR2
26B3    F6 26A4                   LDB    VL2
26B6    3D                        MUL
26B7    1F 89                     TFR    A,B
26B9    4F                        CLRA
26BA    E3 E1                     ADDD   ,S++
26BC    E3 E1                     ADDD   ,S++
26BE    5D                        TSTB
26BF    2A 01                     BPL    DIV2
26C1    4C                        INCA
26C2    1F 89            DIV2     TFR    A,B
26C4    4F                        CLRA
26C5    E3 E1                     ADDD   ,S++
26C7 1083 00FF                    CMPD   #$FF
26CB    23 11                     BLS    DIV3
26CD    FC 26DF                   LDD    CURVL
26D0    43                        COMA
26D1    53                        COMB
26D2    B4 269C                   ANDA   VL1
26D5    B7 269C                   STA    VL1
26D8    F4 26A4                   ANDB   VL2
26DB    F7 26A4                   STB    VL2
26DE    CC 0000          DIV3     LDD    #0
26DF                     CURVL    EQU    *-2
26E1    44                        LSRA
26E2    56                        RORB
26E3    25 11                     BCS    DIVDN
26E5    FD 26DF                   STD    CURVL
26E8    BA 269C                   ORA    VL1
26EB    B7 269C                   STA    VL1
26EE    FA 26A4                   ORB    VL2
26F1    F7 26A4                   STB    VL2
26F4    20 A3                     BRA    DIV1
26F6    8E 3000          DIVDN    LDX    #$3000
26F9 108E 2800                    LDY    #$2800
26FB                     NEWWVE   EQU    *-2
26FD    CE 2756          L4       LDU    #VALS
2700    4F                        CLRA
2701    5F                        CLRB
2702    34 06            L5       PSHS   D
2704    A6 C0                     LDA    ,U+
2706    E6 84                     LDB    ,X
2708    3D                        MUL
2709    E3 E1                     ADDD   ,S++
270B    30 890100                 LEAX   256,X
270F    8C 4000                   CMPX   #$4000
2712    25 EE                     BLO    L5
2714    B3 2671                   SUBD   MINVL
2717    B7 2724                   STA    H
271A    F7 272C                   STB    L
271D    F6 269C                   LDB    VL1
2720    3D                        MUL
2721    34 06                     PSHS   D
2723    86 00                     LDA    #0
2724                     H        EQU    *-1
2725    F6 26A4                   LDB    VL2
2728    3D                        MUL
2729    34 06                     PSHS   D
272B    86 00                     LDA    #0
272C                     L        EQU    *-1
272D    F6 269C                   LDB    VL1
2730    3D                        MUL
2731    34 06                     PSHS   D
2733    B6 272C                   LDA    L
2736    F6 26A4                   LDB    VL2
2739    3D                        MUL
273A    1F 89                     TFR    A,B
273C    4F                        CLRA
273D    E3 E1                     ADDD   ,S++
273F    E3 E1                     ADDD   ,S++
2741    5D                        TSTB
2742    2A 01                     BPL    OK
2744    4C                        INCA
2745    1F 89            OK       TFR    A,B
2747    4F                        CLRA
2748    E3 E1                     ADDD   ,S++
274A    E7 A0                     STB    ,Y+
274C    30 89F001                 LEAX   -$FFF,X
2750    8C 3100                   CMPX   #$3100
2753    25 A8                     BLO    L4
2755    39                        RTS
2756                     VALS     RMB    16
2766                     ZZZ      END    START
```

Listing 9

```
0 PCLEAR1:CLEAR200,&H25FF:L=2:FORI=&H2600 TO&H2755 STEP20:C=0:FORJ=I TOI+19:READ
A$:A=VAL("&H"+A$):C=C+A:POKEJ,A:NEXTJ:READA$:IFC<>VAL("&H"+A$)THENPRINT"ERROR IN LINE"L:ENDELSEL=L+1:
NEXTI:PRINT"ALL DONE !!":PRINT"NOW YOU CAN SAVE THE PROGRAM TO DISK OR TAPE AS YOU WISH.":PRINT"START
ADDRESS = &H2600":PRINT"END ADDRESS = &H2755":PRINT"EXEC ADDRESS =&H2642":PRINT"N.B. CALL IT 'WAVEMIX'":END
2 DATA 86,40,B7,26,0F,40,B7,26,13,BE,26,28,1F,10,C1,00,25,08,C1,00,05CF
3 DATA 25,03,5F,20,06,C6,80,20,02,C6,FF,E7,80,1F,10,5D,26,E8,39,8E,07A2
4 DATA 30,00,86,00,34,02,1F,10,6D,E4,26,05,5D,2A,01,53,58,E7,80,1F,0550
5 DATA 10,5D,26,F0,35,82,8E,30,00,4F,5F,FD,26,68,4A,5A,FD,26,71,CE,0837
6 DATA 27,56,4F,5F,34,06,A6,C0,E6,84,3D,E3,E1,30,89,01,00,8C,40,00,07BC
7 DATA 25,EE,10,83,00,00,23,03,FD,26,68,10,83,FF,FF,24,03,FD,26,71,07A3
8 DATA 30,89,F0,01,8C,31,00,25,CE,FC,26,68,B3,26,71,B7,26,9A,F7,26,08C2
9 DATA A9,CC,80,00,FD,26,DF,B7,26,9C,F7,26,A4,86,00,C6,00,3D,34,06,08F4
10 DATA B6,26,9A,C6,00,3D,34,06,86,00,F6,26,9C,3D,34,06,B6,26,A9,F6,07E3
11 DATA 26,A4,3D,1F,89,4F,E3,E1,E3,E1,5D,2A,01,4C,1F,89,4F,E3,E1,10,0925
12 DATA 83,00,FF,23,11,FC,26,DF,43,53,B4,26,9C,B7,26,9C,F4,26,A4,F7,09F1
13 DATA 26,A4,CC,00,00,44,56,25,11,FD,26,DF,BA,26,9C,B7,26,9C,FA,26,087D
14 DATA A4,F7,26,A4,20,A3,8E,30,00,10,8E,28,00,CE,27,56,4F,5F,34,06,06DF
15 DATA A6,C0,E6,84,3D,E3,E1,30,89,01,00,8C,40,00,25,EE,B3,26,71,B7,096B
16 DATA 27,24,F7,27,2C,F6,26,9C,3D,34,06,86,00,F6,26,A4,3D,34,06,86,0707
17 DATA 00,F6,26,9C,3D,34,06,B6,27,2C,F6,26,A4,3D,1F,89,4F,E3,E1,E3,08D3
18 DATA E1,5D,2A,01,4C,1F,89,4F,E3,E1,E7,A0,30,89,F0,01,8C,31,00,25,0883
19 DATA A8,39,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,FF,FF,02DF
```

Pete GERRARD'S ADVENTURE TRAIL

I'LL begin a hymn of praise to the hard working Sandra Sharkey, who will soon be producing as many magazines as she has children (five so far, but only two magazines....yet). *Adventure Probe* has been going for some time now, a general forum of hints, tips, solutions and letters and gossip all about adventures. This has now been joined by another magazine called *Soothsayer*, which is TOTALLY maps and solutions. Nothing else, just maps and solutions. Now I know that some of you may disapprove of this sort of thing, but if that's the case, then why do you keep writing to me for help? Personally I'm all for it, and if you want to see a sample issue of either *Soothsayer* or *Advebture Probe* then send a quid to Sandra at 78 Merton Road, Highfield, Wigan, Lancashire WN3 6AT and a magazine will be winging its way back to you before you can say *Colossal Cave*. Provided that she's not playing pool or down the pub. Where were you at the last pool delegates' meeting, eh, Sandra?

Mrs Webb of Liverpool appears to be having trouble locating hints for *Fishy Business*, so here we go, courtesey of Salamander. By the way, Mrs Webb, I saw what was crossed out on your envelope, just why is a person interested in *Fishy Business* when the company name that was crossed out is Shrimp Manufacturing? Fish salamanders and shrimps, oh well, this will put them in their place...

1) Everything around here starts underwater, which is all right if you don't mind looking like a prune.
2) Don't go too far from the island.
3) Laser pistols are really cutting weapons
4) Don't be afraid of the dark.
5) Shellfish are great music lovers, but octopussys prefer cake.
6) Those seahorses are too fat already.
7) Scrolls should be read in the right light.
8) The strangest things are said behind thrones.
9) Frog-kings aren't that dumb. Their harems are well hidden.
10) It's not easy being green, so be affectionate.

The hake's progress? Ought to be a dab hand by now. Sorry, I said that on porpoise.

Ahem. Back to the land of adventures again, and more hints asked for. This time the game is *Madness and Minotaur*, the person is J. Edgar of Windsor (posh! I wonder if he could introduce me to Lady Di), the company is Dragon Data, so here we go . . .

Ask the oracle

1) The oracle tells you what you need to kill creatures: ASK ORACLE to find out what you need, get the weapons and kill creatures.
2) You must TIE HYDRA with the ropes before you can kill it.
3) You can OPEN CRYPT only if you are at peak strength and holding nothing.
4) The drapes in the room with the narcissus plants have a 16% chance of opening — you get only one chance to OPEN DRAPES each game (P/G talking — this is the sort of stupid problem that should not exist in an adventure).
5) The sprite will cause any object on the ground in his room to be randomly relocated. The vial will prevent this, as will placing objects in the 'music' room on the first floor.
6) When the 'walls glow an eerie colour' turn off the lamp and LOOK.

7) You must PLAY FLUTE in the 'music' room and hold parchment to see the ledge. Then use rope to CLIMB LEDGE to get a treasure.
8) Many hard to find places may only be reached by jumping ie JUMP PIT, JUMP HOLE, JUMP POOL, JUMP MIST, JUMP UP, JUMP DOWN, or JUMP.
9) Use OKKAN to learn the secret of the glowing rocks.
10) You need the FOOD (1st floor) and the MUSHROOM (3rd floor) to learn the first spell.
11) Use VETAR to recover the lamp if it is blown out of your hands, then turn it on.
12) The URN contains oil to fill the lamp.
13) Once all the spells are learned, some pools of oil may be found at random.
14) AKHIROM will cause the scarab to glow. A glowing scarab will keep you from getting hurt (must take one to the pub with me on Friday nights).
15) Dimly lit rooms on the first floor may require a lamp to see everything.
16) LOOK POOL works in one room.

I don't know about you, but I didn't like that game at all, as it relies far too much on random events. I'm all in favour of random things happening, as they do in real life so they must in adventures, but this one was taking things a bit too far. What do you think?

Letter time

Pulls a letter from the pile, and finds the name Nick Welham staring at him. Nick lives at 27 Duke Drive, Clapham, Bedford. Bedfordshire MK41 6DE, and can offer help on the following thousand or so adventures: *Aquanaut 471, Ice Kingdom, Towers of Doom, Trekboer, Franklins Tomb, Caverns of Doom, Juxtaposition, El Diablero, Dragon Mountain, Lost in Space, Calixto Island and Return of the Ring*. SAE as usual.

Bartering time, folks, because Nick goes on to say "If anyone has any of the following adventures I would be willing to either swap or buy them: *Black Sanctum, Shenanigans, Wings of War, Mansions of Doom*, or any Scott Adams adventure". So there you go, you can barter away to your heart's content.

Two of the Scott Adams series of games were the second and third adventures that I ever played, and these were *Adventureland* and *Pirate Adventure! Colossal Cave* was the first, by the way. Anyway, if anyone can cast their minds back BDU (Before Dragon User) to 1978 they may remember an issue of an American magazine called *Byte* which was devoted to adventures. In it there was a complete listing, in Basic for *Pirate Adventure*! What would people give for a copy of that now, I wonder.

Don't all write in asking for a copy from me, because I haven't got one, it disappeared in a house move, much to my intense annoyance.

Scott Adams

I think there must be two categories of people as far as Scott Adams games are concerned: in those early days, with only *Colossal Cave* to comare him against, I marvelled at how he could possibly fit an entire adventure into the computer's memory all at once. Then, as adventures progressed, I began to wonder why he didn't progress as well: even the most ardent fan has to admit that his room descriptions are not particularly atmospheric. And look at things like *The Hulk*. 16 locations, and you spend most of your time digging over and over again to find umpteen games? Okay, he might have had to pay a lot of money for the rights to the character, but dear me, *Spiderman*, too, a popular character but a not very brilliant adventure. And yet, the man did a lot to popularise adventures so we can't be too hard on him. If it wasn't for him and the team behind Infocom we might never have had adventures on home computers at all.

Looks through rest of letters, requests for help and hint sheets, so I shall take the risky step of printing the rest of the *Syzygy* solution, since I still get millions of letters about it. Well, okay, I exaggerate slightly, thousands then. Without the benefit of a wonder memory I cannot recall whether it was the November or December issue which featured the first ten steps of the solution (and in case you're wondering why I don't just look it up it's because this is being written before I've seen either of those two issues, and our beloved editor has the power to print or not to print, as she sees fit) *(I just print these things, I don't remember them! — Ed.)*, up to htaerb ruoy gnidloh and entering the corridor. This is the rest of them, so don't read if you don't want to know...

11) Go to the other main computer and look at screen.
12) Go to the transporter and enter co-ordinates for planet (see below). Be sure to save game here so that it can be returned to later in the event of a disaster.
13) Go to the plant and fall off the cliff.
14) Listen, and if you can't hear anything then go back to step 12, reload, and repeat until you can hear something.
15) When there is a sound, GET COMMUNICATOR and USE COMMUNICATOR.
16) Enter co-ordinates for emerald.
17) Get emerald and return using communicator.
18) Go to Darth Vader by the transporter.
19) Kill him.
20) Return using the communicator again.
21) Go to the lift and press the button twice.
22) Go through the forcefield.
23) THE END

Note: don'forget to breathe at places other than the airless corridor from step 10 onwards.

Co-ordinates for transporters: Planet 0-4-1-5, Emerald 2-7-3-0, Vader 1-6-0-3.

For each place enter each number by pressing that number and at the end of the row you must PULL LEVER.

The problem of getting lost in the large forest with the strange light is solved quite simply by not going there!

End of solution, end of game, end of column! See you next month.

Write: ADVENTURE

***Pete Gerrard** gets a game gang together*

LAST month we had the minor problem of a dead Dragon, and hence no programming. The month before that, however, I'd said that we'd be talking about introducing characters into a game, and now, with a happy Dragon once more by my side, that's precisely what we're going to be doing.

In the October issue of *Dragon User* we touched on simple, individual characters who performed simple, individual tasks. They either attacked you, or stole things from you, or perhaps solved a riddle or two for you by carrying things, accepting things, performing some task that was beyond the talents of the main hero of the adventure — you! They might have been able to reach something you couldn't or possibly they had the ability to swim across a piranha-infested lake, but whatever it was it generally consisted of something that was pretty straightforward.

In short, they didn't really do very much, and programming that sort of character into your adventure is really quite an easy thing to do. More complicated are advanced characters or groups of characters, although I noticed that the thirteenth issue of *Adventure Contact* (once a local adventure magazine, now moved down to Bournemouth) did touch on this topic by giving us one way of having an individual adventurer join or, of course, leave a group of people in the game.

However, there we had the group behaving very much as one would expect an inanimate object to behave: it was shunted from one room to another, and didn't really play much part in the game, other than allowing someone to mingle and perhaps achieve something that would be impossible for a more conspicuous person roaming about by themselves.

So this month we'll be looking at more 'human' characters, or at least characters that have a touch of personality about them, with perhaps a brief nod in the direction of having lots of characters, time and space permitting of course.

Single Characters

I'm all for making adventures as realistic as possible, and just as in real life we wouldn't expect anyone to carry more than about six bulky objects at a time, that is the limit we would impose on the program. But, it doesn't seem to make sense to allow the person to carry anything else because he's holding too much. Bearing the immense load of six sheets of paper and failing to have the strength to pick up a key is clearly ridiculous, as is the equally absurd situation of being able to take the strain of, say, six hundredweight bags of coal. So, objects are given ratings which relate to their weight, the player is given a strength factor, and as his strength goes up and down during the course of the game he can carry more or less objects. All the time, though, the objects that he can carry have a practical limit imposed upon them: walk around with five hundred sheets of paper and if you want, just one bag of coal. All of this brings us to one vitally important factor that we can give our intelligent single character, and that is the factor of strength.

Making him stronger than the adventurer means that he can carry something that the adventurer can't, and thus he will have at least one thing to do in the game. Perhaps it might take two of you to carry a particular object from A to B: get our friend to help.

Help Offered

Intelligence, which is obviously something you'll need to solve the game, is something else that we can endow our characters with. It is, after all, one of the traditional six attributes that are imposed on players in the world of Dragons and Dragons and other role-playing games.

Suppose the player is faced with a conundrum. He might have to, for example, water a plant in order to make it grow, but the only thing he can find to carry the water in is a bucket with a hole in it, and however fast he runs he always finds that the water has run out before he gets there. Clearly the bucket has another purpose, since there is nothing to block the hole up with, or is there? Something else to think about.

Instead of using the more familiar HELP routine, where the program usually prints up some inane comment about not understanding you, or refusing to offer help unless you send off a postal order and take out a year's subscription to *Pigeon Fancier's Monthly* or something (just imagine all these blokes in bars, eyeing up pigeons, saying 'cor, I don't half fancy that one'), you could have the other character offering help and advice when asked. Always provided he's around, of course, and hasn't got tired and listless by being asked to run around with rapidly emptying bucket of water. If he sits down and starts singing about gold, well, that's your problem.

When I say 'he', by the way this is not me being sexist or anything. I just hate writing out 'he or she' all the time. Use whatever sex you feel like, they both have their uses! Perhaps you might like to leave it to the player to decide, and adjust everything accordingly: less strength, more intelligence, or however you see fit. Don't all write in and complain at once.

Another of the constraints usually imposed upon you in a role-playing game is that of dexterity, hand-to-eye co-ordination, that kind of thing. Thus you could easily give the player, and the character, a dexterity rating, so that, for example, a player with a low rating combined with a character who has a similar rating would not spot a trap until they fell into it, but two characters who are very dexterous could easily dodge life's slings and arrows of outrageous fortune and get on with things much more easily.

By giving the player, and any other characters who come into the game, these ratings of strength, intelligence and dexterity, you make the game much more realistic, however un-realistic the setting for your adventure might be. Perhaps, instead of setting the ratings for yourself, you let the player choose his own, giving him a maximum total of points to choose from. Someone might settle for lots of strength, and have little left over for intelligence and dexterity, thus ending up like Benny out of Crossroads: which is certainly a fate worse than death, as I'm sure you'll agree.

Other players might opt for lots of intelligence, and end up with the strength and dexterity of a banana skin. The game thus becomes far more than a simple 'solve this problem, collect that treasure' affair. The attributes for the characters in the game you would, of course, select for yourself. In adventures, as in life, you can determine those of other attributes to large extent but you can't determine those of people, which makes the adventure, like life, all that more interesting.

You might keep these three as three distinct variables, each one being affected by the various situations encountered during a game. Eating and drinking might increase strength but decrease dexterity if you overdo it. Finding a book in a library, or a map on the floor, might increase intelligence, and so on.

Character Building

But all of this is to no avail if you don't tell the player what the character actually looks like. Saying 'a dwarf leaps out from behind a rock, and tells you that his name is Dimli Gloing' (knew I'd get him in somehow!) at least makes you think of something typically dwarvish, but saying 'a man walks out from behind a boulder, and tells you that his name is John Smith' gives you absolutely nothing to go on apart from the fact htat he lurks behind boulders and pounces on unwary adventurers. I'm not implying that you launch yourself into screenful after screenful of Barbara Cartland style pose (heaven forbid), but you should give the player some kind of impression of his new acquaintance. He might be smiling or surly, bald or looking like Father Christmas, short or tall, it only takes a few words to convey the necessary information.

Having done all that you're in much more of a position to give your games that certain something that all good adventures have: lots of atmosphere. It doesn't take massive room descriptions and hologram graphics to do that. Just a little of common sense and some careful programming will do the job equally well.

Balanced Talents

A realistic adventure is always going to be better (and sell more) than an unrealistic one, and one way of doing that is to look after your characters. Give them the various attributes and they become alive, playing a real part in the game, and apart from increasing the quality of the game they also allow you to set much more ingenious puzzles for the player to solve. Never make it impossible of course, because even if someone for whatever bizarre reason selects an intelligence of zero, they should still be able to complete the game. You might like to compensate for the ratings that the player selects by choosing suitable ratings for your other character. If the player decides to have no dexterity whatsoever you would give the other character the ability to carve a scale model of a spider's web with a blunt axe and a match, thus ensuring that everything remains the same overall and that every problem is therefore solveable. One thing that we haven't really touched on, and I'm sure this will be a topic dear to our beloved editor's heart, is the subject of women in adventures. Why is just about every adventure devoted to the male persona, and then people moan because they can't think of any original problems anymore? Okay, two of the mighty Income adventures gave you the opportunity to choose between male and female, and the game adjusted itself accordingly. Or at least I hope so, playing as a female I never made it to the sultan's harem in one of the games!

But generally speaking our typical adventure casts the hero as a male of the Rambo (I can't spell Arnold whateverisnameis) variety, and I'm, sure there's mileage to be had in exploring the female role. And not just as the kind of female that adventure use when promoting Barbarian-type games either: Maria Whittaker might earn five times as much as the prime Minister, but then (as the *Mail on Sunday* put it) she's got five times as much on her front bench. Many people find that sort of image offensive, and just as not all male adventures look like Arnold so not all female ones look like Maria.

Conclusion

I saw it quoted somewhere that around 30 per cent of adventurers are women. Where they get these figures from I have no idea, but there you go. (*He's off about figures again — Ed.*) Thus there is obviously a lot of female interest out there, and as I said it would give you some new problems to ponder rather than the usual adventure scenarios. I'm not suggesting that you go on an out and out feminist campaign, but it would make a change. Oh well, having no doubt stirred up a hornet's nest I shall retreat quietly from the scene until another month rolls by. There we'll get back to adding a few verbs to our adventure, and give you another insight into how a game is put together. Plenty to think about, in terms of characters (male or female) (*What about Luggage? Is this man is an objectist?*) so until then, goodbye.

CLASSIFIED ADS

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

Name ....

Address ....

.... **Tel:** ....

Classified rate: 35p per word.

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

Continued from page 11

you to calculate the number of bytes yourself (BNE * −23, for example), but the easy way is to always give the destination instruction a label name and to branch to that. This way, the assembler calculates the number of bytes between the destination and the branch instruction for us — and if we change the number of instructions between the two locations, a re-assemble automatically generates the revised value whereas if we wish to be the computer and use the branch * + or * − form, we need to check and re-evaluate new values prior to each assemble. There is a limitation on the distance between the branch and destination of +127 and -128 bytes from the end of the (2 byte) branch instruction (+ start of the instruction following the branch). However, this does not limit our program capability as if greater distances are involved (and the assemble will usually flag an instruction as in error if the distance is exceeded), all the branch instructions may be prefixed with an L for Long (LBEQ/LBNE etc). These allow two bytes for the +/- value, allowing distance of +32767 and −32768 bytes to be reached — which, as the total memory of an unmodified Dragon is 65536 bytes enables every possible location to be reached.

We have mentioned + and − numbers here, which is where 'signed' and 'unsigned' values come in to focus. However, if I start going into detail at this point, the Editor's guillotine will come into greater focus, so, sadly, further detail will have to wait until next month.

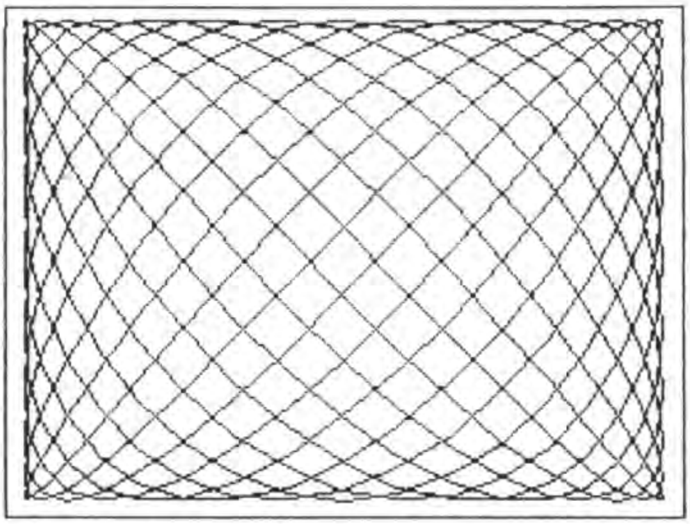
Workout

Meanwhile, as programmers, we need to protect our programs from operator misuse, either accidental or deliberate. The PRINT@ code above contains a weakness that can be exploited by a user — have you found it? There is a check in the $800C routine that prevents the display of the left arrow causing the text pointer to fall down below address $400. However, in the PRINT@256 version, have you tried left arrow beyond the amount of data you have already (if any) input then carried out further keypresses? To prevent this, we must decide either not to allow use of left arrow for erasure (unfriendly) or to insert our own check that if left arrow is received, only allow it to be displayed (+erase) if there is data to be erased (current pointer contains value > $500). The CoMPare instruction cannot directly compare the contents of a single/double byte with another memory location or actual value; one or other value must be in a register. We do not want to corrupt the contents of register A prior to knowing whether it is to be displayed or ignored, so register D should not be used for the compare; use register X. Try adding in instructions to the PRINT@256 version to both prevent use of left arrow for erasure and a further version to allow use of left arrow for erasure but to guard against backtracking beyond $500.

Next month, at last, we will make a start on tackling programs that won't run on our machines because of their position dependency.

Down in the dumps

***Dragon User** presents a screen dump for the Memotech DMX80*



BASIC is very good for screen dumps because it is so easy to alter to suit your own particular printer. As far as I know, the Memotech is almost the same as the Panasonic and some Epsons, so I hope this dump is going to be useful to quite a few Dragon owners.

Although the dump takes 5 minutes (providing your Dragon can handle lines 40 and 130) to do its job, the beauty of it is that it can be altered so easily.

```
10 'SCREEN DUMP - MEMOTECH DMX80
20 'MIKE TOWNSEND - (C) - 4/6/87
30 PMODE4,1:SCREEN1,0:Y=0
40 POKE65495,0
50 PRINT£-2,CHR$(27);"@";
60 PRINT£-2,CHR$(27);"P";CHR$(0)
;
70 PRINT£-2,CHR$(27);"K";CHR$(0)
;CHR$(1);
80 FORX=0TO255
90 A=PPOINT(X,Y)*128+PPOINT(X,Y+
1)*64+PPOINT(X,Y+2)*32+PPOINT(X,
Y+3)*16+PPOINT(X,Y+4)*8+PPOINT(X
,Y+5)*4+PPOINT(X,Y+6)*2+PPOINT(X
,Y+7)
100 PRINT£-2,CHR$(A);:NEXT
110 PRINT£-2,CHR$(27);"A";CHR$(8
);:PRINT£-2,CHR$(10);
120 Y=Y+8:IFY>191THEN 130 ELSE70
130 POKE65494,0
```

LINE 50 resets printer. Clears any commands you have already given it.

LINE 60 sets printer to "Elite" mode, otherwise you get oval circles.

LINE 70 Standard bit image designation command.

LINE 80 to 100 collects the data from each row accross the screen, 8 pixels high.

LINE 110 set line feed to 8/72 inch, to get no gaps between the rows, and then gives the line feed command.

LINE 120 instructs the next line to be scanned or, if the whole picture is complete, ends the program.

Winners and Losers

Every month
Gordon Lee will
look at some prize programming

HOW do you perform a calculation such as 2^{75} * 5 + 1 with absolute accuracy? This was the problem posed in August's competition, and one method of solving this and other similar problems is discussed, beginning on page 26 of this issue.

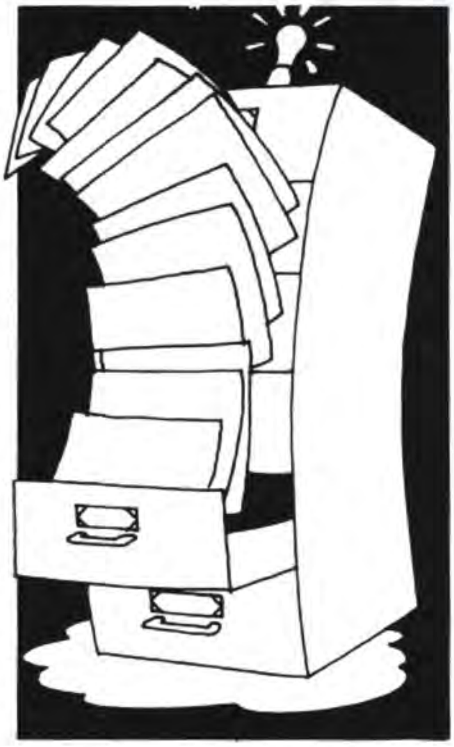
Having come up with a possible answer, by whatever means, this is one type of question in which it is not immediately apparent if the result obtained is the correct one. This proved to be the case in a number of incorrect answers which were received from readers who were clearly unaware that their solutions were wrong.

Fortunately, there are a number of checks which can be applied in order to detect any possible error which might have been made. Note that the use of these tests does not guarantee that the solution is correct, but they can often be used to eliminate an incorrect answer. If an answer posses all of these tests then the odds in favour of it being correct increase correspondingly.

So, let's take the calculation given and apply some of these tests. Our first test would be to ask the computer to print out an approximate answer: PRINT 2↑75*5.

As this is still inside the 'overflow' capability of the Dragon, the answer 1.88894663E+23 is given. From this we now know that the correct answer begins with the digits 18889... and the E+23 tells us that we need to move the decimal point 23 places to the right — ie to end with w 24 digit number. in calculations of this magnitude, don't assume that all of the digits displayed are the correct ones, but you can be certain of the first fiew digits. Even if the computation were greater than the Dragon could handle, we could still use ordinary common logarithms to give us an indication of the magnitude of the answer together with its first couple of digits.

Now we ought to turn our attention and look at the last few digits of the answer. The listing here performs the calculation given in the competition question, but it restricts its attention to just the final few digits. Each time the calculation exceeds a total of six digits the excess are removed by use of the RIGHT$ command at line 50. Consequently, the final six digits which are printed out are the precise six which will appear at the right-hand end of the final solution. We now know that the answer will be a 24 digit number in the form: 1888.............547841.



Having established the top and tail of the answer, what about the middle? Unfortunately, there is no easy way to determine this (except by solving the puzzle) — but there is a useful 'negative' test which can be applied which uses digital roots.

The digital root, or DR, of a number, is the digit which remains when the are added together until a single digit remains. The digital root of the current year, 1988, is 8, since 1+9+8+8 = 26, and 2 + 6 = 8.

Whenever an operation is performed with two numbers the result is reflected in the corresponding digital roots. For example, if 1988, which has a DR of 8, is squared, the answer will have a DR of 1, because 8 times 8 is 64, and 64 has a DR of 1. The correct answer, 3952144, does indeed have a DR of 1. Otherwise we should check for an error.

We can now apply this test to the competition. First, look at the first few powers of two: 2,4,8,16,32,64,128,256, and so on. These ahve digital roots which run 2,8,7,5,1,2,4, and so on. Not how they repeat in cycles of six numbers. This tells us that 2^6 has a DR of 1, as do 2^{12} and 2^{18}. This is true of any power of 2 in which the power is a multiple of six. This means that 2^{72} would have a DR of 1 and that consequently 2^{75} has a DR of 8. If 2^{75} is then multiplied by 5 (as in the next stage of the computation), the answer will have a DR equal to the DR obtained by multiplying our existing DR of 8, by 5 — ie a DR of 4. If 1 is added finally to this we get a DR of 5, which should be the root of the final answer (and indeed it is).

It must be stressed again that by passing all of these tests the solution is not definitely correct, as two errors could cancel each other out, but it would give a strong indication that the answer was OK. However, if any single test were failed, then the answer would have been definitely wrong. When this was done to all of the incorrect answers submitted, none passed all three of the tests.

```
10 N=1
20 FOR P=1 TO 75
30 N=N*2
40 N$=STR$(N)
50 IF LEN(N$)>6 THEN N$=RIGHT$(N$,6)
60 N=VAL(N$)
70 NEXT P
80 N=N*5+1
90 PRINT N
```

Communications

Write down your problem on the coupon below (make it as brief and legible as possible) together with your name and addres and send it to Communication, 12/13 Little Newport Street, London WC2H 7PP.

Problem ....

....

....

....

Name ....

Address ....

....

....

Adventure Contact

To help puzzled adventurers further, we are instituting an Adventure Helpline — simply fill in the coupon below, stating the name of the adventure, your problem and your name and address, and send it to Dragon User Adventure Helpine, 12/13 Little Newport Street, London WC2H 7PP. As soon as enough entries have arrived, we will start printing them in the magazine.

Don't worry — you'll still have Adventure Trial to write to as well!

Adventure ....

Problem ....

....

Name ....

Address ....

....

Happy New Year

***Gordon Lee** can't even get to 1988 without making a puzzle out of it . . .*

WHENEVER the competition involves using numbers greater than those which the computer can normally handle we inevitably receive lots of letters from readers reporting that their attempts end in an overflow error report on the computer. So here is a step-by-step introduction to the method that the Dragon can use to perform computations on numbers with hundreds (or even thousands) of digits!

First of all it is necessary to understand the limitations of any micro. Ask it to multiply 7654321 by seven and it will display the correct answer, but try to multiply this same number by *itself*, and the answer given is only the approximation 5.858863E+13. The E+13 is the computer's way of indicating that the decimal point of the displayed value must be moved thirteen places to the right giving a 'real' value of 58588630000000. This does not mean, of course, that the true answer ends in seven zeros, merely that the computer can only handle a relatively small number of digits with absolute accuracy. It is possible to coax a few more digits out of the computer as the internal registers contain more digits than are displayed on screen, but these cannot be relied upon to be accurate. The calculation, therefore, should only be regarded as correct to about nine significant figures. On practical level this would, for example, allow us to measure the length of the equator to an accuracy of a few feet — but in mathematical theory (and some of our competitions) this isn't enough!

One way of overcoming this problem is by using string variables to hold the digits being operated on. Clearly, it is not possible to multiply, divide, subtract, or, in the mathematical sense, add strings, but by converting each digit to a numeric variable as it is required, large computations can be made. There is nothing complex in the mathematics either — the method being to simulate that used in the 'pencil and paper' way of performing the calculation. Let's take a simple example of multiplying 987654321 by 4. Mathematically, we would take each digit in turn, starting at the right hand end, and multiply by 4. Any product greater than nine would result in a 'carry' which would be taken on and added to the next stage of the calculation. Now look at **listing one** which performs the calculation using this precise

Listing 1

```
10 A$="987654321":B=4:Z$="":CARRY=0
20 FOR F=LEN(A$) TO 1 STEP-1
30 V=VAL(MID$(A$,F,1))*B+CARRY:CARRY=0
40 IF V>9 THEN CARRY=INT(V/10):V=V-CARRY*10
50 Z$=RIGHT$(STR$(V),1)+Z$
60 NEXT F
70 IF CARRY<>0 THEN Z$=RIGHT$(STR$(CARRY),1)+Z$
80 PRINT Z$
```

```
          7129076834 .........A$
          2763128729 .........B$
         64161691506 .........L$(1)
        14258153668  .........L$(2)
       49903537838   .........L$(3)
      57032614672    .........L$(4)
     14258153668     .........L$(5)
     7129076834      .........L$(6)
   21387230502       .........L$(7)
  42774461004        .........L$(8)
 49903537838         .........L$(9)
14258153668          .........L$(10)
19698557011273763986 .........Z$
```

Prize

IT's a while since we had an adventure up for offer on the Puzzle Page. John Smallwood, author of the adventure *Larkspur Waldorf is Trapped!*, has offered us five Larkspurs, and a set of discount vouchers for another 15 lucky winners of our January Crossnumber.

Rules

AS soon as you have made it across the gap from 1987 to 1988 (in fact, if your *DU* gets there on time, you can do it between the turkey and the pudding, if that's how you see your mum and dad), rustle up a printout, tell us your answer (don't do what someone did last month and include the listing without an answer attached to it!), with any program notes you want Gordon to read, complete the tiebreaker, and whap it into an envelope marked JANUARY COMPETITION to the usual address.

For the tiebreaker, I want you to write another verse to Jingle Bells or, come one, be fair, any other piece of Yuletide music which has really been driving you up the wall recently. It only needs to be two lines, but it must be in keeping with the spirit of the festive season, whatever you perceive that to be.

October winners

THIS proved a popular competition and we saw a higher-than-usual number of new faces among the entries. We also had a high proportion of correct entries, but boy, it's a good thing we don't deduct marks for neatness (or whatever the opposite is. Neatless?).

The winners are:

Robin Telkman of Sale, John Smallwood of Preston (win again next month and save yourself a few bob, John!), C. R. Barber of Seville, Denis O'Mulloy of Comberton, N.J. Welham of Bedford, Fred Willers of Yarnfield, Olav Nielsen of Denmark, J. Smith of Twyford, Alan Thomas of Staplehurst, Phil Sapiro of Liverpool, Austan Henderson of Bromsgrove, D.J.Grey of Middlesbrough, Mark Towlson of Long Eaton, C. Hitchinson of Middlesbrough, S.A. Siddiqui of Chiswick, Nils Lindgren of Sweden, E.A. Newman of Addlestone, Don Robertson of Epsom, I.J. Huggins of Caerphilly and Randy Longshore (that's 'anfalas' in Sindarin, Randy, but we don't know the word for your first name yet) of California.

There were quite a few good tiebreakers, including a short poem from Randy, but my personal favourite was "I go to work on an egg because the spoon always gets stuck in a jam" from I. Huggins. Best Late Note from Olav Nielsen, who swears that the Pentagon hacked into his Dragon and deleted his original solution. Anything to keep them away from Star Wars, Olav.

The prizes for the October comp were a batch of *Chuckie Eggs* and *Screaming Abdabs* kindly donated by Harry at **Computape**, one of the Dragon's longest serving and most consistent supporters. See yez at the Show, Harry.

Solution

This month's solution should be opposite.

method but arranged into a program which the computer can handle. Line 10 stores the nine-digit number in A$, and the digit by which this is to be multiplied is in the numeric variable B. The final product will be stored in string Z$, initially defined as a null string, and any 'carry' in the appropriately named variable, again initially set at zero. The loop (lines 20 to 60) takes each digit in turn from the right hand end, multiplies by B and adds the 'carry' (if any). Note that once the carry is added it is reset to zero. The value of this individual calculation is checked to see if it is greater than 9 — ie to see if there is a carry to be taken on to the next stage. This is done at line 40 which also gives a revised value for V if a carry is made. Finally, the resulting digit is converted to string format and is appended to the left hand end of Z$, thus building up, digit by digit, the final product.

Dragon ghost

The only irregular command in this listing is the use of the RIGHT$ instruction in lines 50 and 70. Regular readers of this column will realise that this is because of the 'ghost' blank character which the Dragon places on the left hand end of any string variable created by use of the STR$ command. Once the left had digit in A$ has been reached it is then necessary to append any 'carry' (if there is one), so that the final result is now held in Z$. It is important that this listing, and the theory behind each of the instructions, is understood before considering the next stage — that of the multiplication of two large numbers.

Figure one is a long multiplication of two 10-digit numbers, much as it might appear if worked out longhand. By modifying **listing one** it would be an easy matter to store the second 10-digit number as string B$, and then take each digit in turn from this second string and multiply in the manner already outlined the value held in A$. The resulting sub-products could then be stored, for example in an array, before being finally totalled up to give the final result. The disadvantage of this is that memory needs to be used simply to store all of these sub-products until the final totalling.

1000 bytes

In the example given this is not a problem as there are only ten lines of digits each containing ten or eleven digits, but if the numbers being multiplied contained a hundred digits, it would be necessary to reserve 10,000 bytes of memory for this purpose. The way around this is to add each digit as it is computed immediately to the appropriate digit in the final product (Z$). In this way the whole need for the array can be eliminated and it is only necessary to reserve string memory for the two numbers being multiplied, and the final product. In the case of the two 100 digit numbers mentioned, this would only tie up 400 bytes — quite a saving on the 10K which would otherwise be necessary.

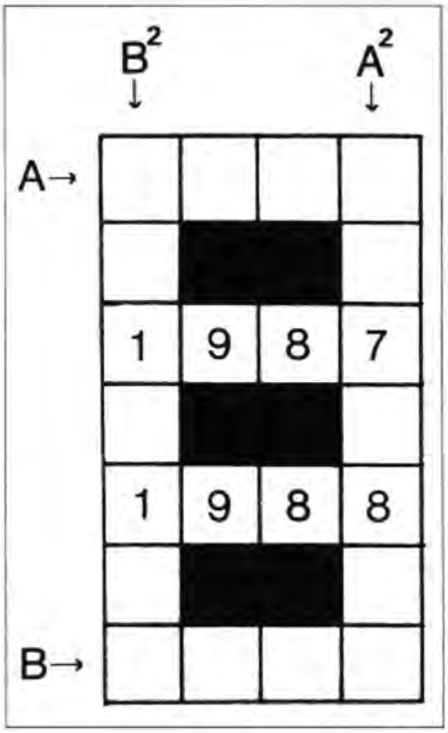
As an example of how this would work take the third column of digits from the right in the long multiplication shown. Here we have to add the digits 5, 6, and 8. The sum of 19 means that the 9 will appear as the third digit from the right in the final product, and a 1 will be carried on to fourth position. This in turn will be the total of 1, 6, 3, and 2, plus the 1 carried, giving a final digit 3, with, again, 1 to carry.

In order to do this we need to initially define Z$ as a string of zeros long enough to contain the final product, and to have a means of knowing the position within this string to which each individual digit must be added. Next month we will be taking a look at how this can be achieved.

Competition

To commemorate the start of the new year, this month's competition is a crossnumber puzzle based on the year which has just ended, 1987, and the one which has just begun.

In the grid below these two dates have already been entered. What you have to do is to find two values A and B such that they, and their squares, can be fitted into the grid.



The Answer

This is Gordon Lee's own solution to the November competition see page 25 for results

ANSWER: the boxes should be filled as follows:

(a) Triangular/square: 36, 1225, 41616
(b) Triangular/tetrahedral: 10, 120, 1540
(c) Triangular/pyramidal: 55,91, 208335
(d) Square/cubic 64, 729, 4096
(e) Square/tetrahedral: 4, 19600
(f) Square/pyramidal: 4900

SOLUTION: The competition was related to the relationships between the five main types of figurate numbers. As was stated certain of these have been proved to be impossible, and one — a tetrahedral number which is also pyramidal — although not disproven, is extremely unlikely. If such a number exists it will be very large.

The list above in sections a to d give, in each case, the three smallest numbers (not counting 1). Sections e and f show the only possible answers that exist — a fact which has been proved mathematically.

The listing given compares two types of number and examines the relationship between them. Numbers which correspond are printed out. In the example given the squares and triangular numbers are being compared. Starting with A=1 and B=1, each pair of figurate numbers is generated in turn. The logical method would be to generate one of these figures in turn and then test a series of the other type of number to see if the two values match. This would result in a lot of duplicate working by the computer so a much quicker method is used here. Note that TA is the value produced from A by formula 1, and TB is the value produces from B by formula 2. These two values are compared. If TA is the larger then B is increased before the next test. If TB is larger then A is increased. In this way the higher orders of figurate numbers can ve tested without any duplication as the values 'leapfrog' over each other in turn. If two values are found which match, the result is printed out as these values are what we are looking for. When this happens, one of the generating values is incremented and the search continues. In the program it is A which is increased but it could equally well have been B.

By changing the formulae in lines 20 and 30, the different relationships can be tested.

Listing A

```
10 A=1:B=1
20 TA=A*(A+1)/2:REM Put Formula 1 here
30 TB=B*B:REM Put Formula 2 here
40 IF TA=TB AND TA<>1 THEN PRINT A;"
   ";B;"   ";TA
50 IF TA>TB THEN B=B+1:GOTO 30
60 IF TB>TA THEN A=A+1:GOTO 20
70 A=A+1:GOTO 20
```

Dragon Answers

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

ROM can't be a RAM

I need a RAM pack for my Dragon 32 (only 8K is needed). Is it possible to remove the two eproms from my *Ghost Attack* cartridge and insert two 4K RAM chips to make $C000 to $DFFF RAM?

Are there any RAM packs for 8K available on cartridge?

Phil Callaghan
54 Pentivale Road
Moss Pit
Stafford

UNFORTUNATELY you cannot simply replace a ROM chip with a RAM chip, as the cartridge does not have all the necessary read/write circuits needed for RAM. It is certainly possible to attach extra RAM in cartridge form as all of the important 6809 pins are brought out at the cartridge port. The problem is that no one seems to make this (obvious) product for the '32 — unless one of our readers knows differently?

Binary coded decimal

I have recently started to write 6809 assembler and have come across 'binary coded decimal' in some of the manuals I have. Could you explain what this is and what it is used for?

Eric Fields

IN the space I have here it isn't really practical to give a full blown explanation of Binary Coded Decimal (BCD), but I'll try and give you the absolute basics.

Normally, one byte stores a number from 0 to 255 using eight bits. In BCD the byte is divided into two four bit 'nibbles' which can each hold a number between 0 and 9 (the values 10 to 15 are not used in BCD).

For example, the niumber 67 would be stored in the following way in binary and BCD:

Binary: 01000011 — 64+2+1 = 67
BCD: 01100111 — 6 (0110) and 7 (0111)

Binary Coded Decimal was originally intended for use where exact precision was required, but an obvious application for us is storing and updating a score in a machine code game. If you use standard binary, then converting this into Ascii screen characters is not very simple, however, a BCD byte can simply be converted as follows (assuming BCD score is in B register: (see listing one).

A special instruction 'DAA' exists to adjust the accumulator after a BCD addition. So to add 5 to the score (again assumed to be held in B), the following routine could be used: (see listing two).

Of course, only numbers between 00 and 99 can be stored in a BCD byte. I suggest you look at 'Programming the 6809' by Zaks for more information on BCD.

```
PSCOR TFR  B,A
      LSRA      * Get upper nibble
      LSRA
      LSRA
      LSRA
      ORA  #48  * Convert to ASCII Code
      JSR  PRT  * Print the char in A
      TFR  B,A
      ANDA #15  * Get the lower nibble
      ORA  #48  * Convert to ASCII Code
      JSR  PRT  * Print this digit
      RTS

INCSC TFR  B,A  * Get score in A
      ADDA #5   * Add 5 to it
      DAA       * Re-adjust back to BCD
      TFR  A,B  * Put back in B register
      RTS
```

Maplin muddle

COULD you help me with a problem I have interfacing the Maplin Dragon 32 I/O port to my Dragon 64? The instructions put the operation of this module at $C000, but I don't seem to be able to make the module work on my '64. I would appreciate your comments.

A. Atkinson
Knowle Cottage
Knowle Road
Kirkheaton
Huddersfield
W. Yorks

THIS is a problem which I have come across before (on an enhanced cartridge I reviewed a few years ago). On the Dragon 32, it is possible to interface an I/O port at address $C000 with minimal decoding. However, on the '64 the address select circuit is more complicated and I/O ports must be fully address selected using all 16 bits.

The problem seems to arise in circuits designed for the Dragon 32 before the '64 was released. Although valid for the '32, this 'quick and dirty' decoding will not work on the '64.

The sharps from the flats

I have a copy of the 'Composer' program for my Dragon which I bought secondhand. Alas, I have no manual for this. I think I have just about to to know the whole system, but althought I can enter sharps in the data statements ('#' seems to do the trick), I cannot find how to enter 'flat' or 'double dotted' notes. A single dot works for a dotted note, but '..' does not work for double dotted. Can you help?

Dave Lowe
Farm Cottage
Donisthorpe
N. Midlands

A flat note is denoted by an exclamation mark after it (eg A!5) in a similar way to a sharp note (eg C#4). Double dotted notes are marked with a colon (eg H:) at the end of the line.