WACCI

CPC

WACCI celebrates it's 12th birthday this month

Contents

Thanx & Stuff

Welcome everyone to another issue of your monthly er! Well nearly monthly magazine. Why was it late this time you may ask. To be honest if I was to give you all to sordid details there wouldn't be any room for any of the excellent articles that are in this issue. So just accept my apologies and lets get down to what you all want to read about. Next months issue (NOV) will be for December also...

Fair Comment

No letters were received to go into this issue and so the convention report has taken it's place. The report was once again written by Angela Cook with a little help from Carol.

Carol also took all the photo's again this year, so thanks Angela and Carol for all your hard work.

Thanks

Dave Simpson would like to thank Christine and Chris Raisin for buying a load of stuff from him at the convention. His stock is now low but his pockets are bulging with all the cash you gave him.

A very big thank you goes once again to Angie for all the hard work she put in. We all had another wonderfully magical day that will most surely be repeated again next year. So 'THANK YOU' Angie.

Articles

I have received articles from lots of members over the last few months but unfortunately I have had (once again) a major set back with my computer.

Unfortunately I tried the freebie internet server supplied from Dixons or PC World. It's called Freeserve. I loaded it in and it crashed my entire system and consequently I lost all the articles sent to me.

I know I should have made back-up copies to disc, Carol had a right go at me because I am the first one to remind others to do just that. Anyway I didn't so I lost them all.

So if you have sent me an article over the last few months, would you *please* send it to me again. Thanks.

Comments

Recently an article was submitted to me and the author asked me to give him my thoughts on it. He wanted me to say whether I liked the article or not and any criticisms that I might have about the article. I told the author that I would prefer not to make these type of comments about an authors work as some articles may not be my cup of tea and so for me to say that perhaps I did not like the article would be wrong. It would not be fair for me to make a judgement on someone's work and perhaps by being negative about it they may be put off writing anything again.

PCW Owners

Should you have any problems with your PCW then Kathleen Thorpe is just the person you might wish to speak to. She has a helpline of people who can sort out your troubles so make a note of these very important details:

Kathleen Thorpe
Tel: 0115 9720114

Questionnaires

If any of you at the convention took a questionnaire away with you, could you please now send it back to Thomas Fournerie ASAP

And finally

A mention goes to Bill Ford for his kind donation of the CPC items he sent for the club. Thanks Bill and yes, Carol still has that outfit you mentioned!

-Happy reading-

Members Spotlight on Dave Caleno

Current editor of Bonzo's Scrapyard, the occasional Amstrad CPC Tape to Disc Newsletter
Bonzo's Scrapyard, 29 Chapter Road, Strood, Rochester, Kent ME2 3PX
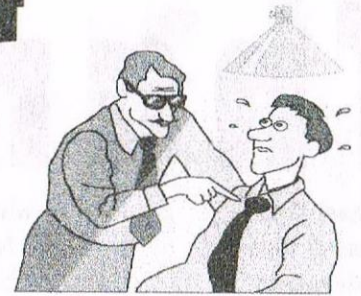
I am 48 years young, married to Dot, and have three grown up children, two girls and a boy, plus we now have three grandsons, one of eight, one of three and one of two weeks. I am a printer by trade, which is the occupation I have been in since I left school at 16. I obtained my first CPC 6128 in 1987, owing to it being on special offer at *Dixons*. At that time I had quite a few local *Amstrad* contacts at my place of work. A year later Dot and I bought a CPC 464 for the kids to use mainly to keep them away from mine.

Goldmark Systems

Although I did gradually obtain quite a few games for the CPC, my interests started to concentrate on the tape to disc transfer side of the machine. I obtained all of the routines that were on sale from the late Brian Barton of *Goldmark Systems*. Over a number of years Brian became a very close friend, and I paid many happy visit to the man, at his various homes. It was with great regret that I heard of his tragic demise. Before that happened Brian had passed the copyright of the majority of his programs over to me and they are still currently for sale. During this period I also had the pleasure of contacting Colin Harris when I purchased copies of the *Bonzo* suite of tape to disc programs, they being to augment the *Goldmark* ones.

Hubble Bubble

I had sent off for a copy of *WACCI*, issue 42 I think it was, but upon receipt I was not overly impressed with it as there were duplicate pages

and some of them were blank. I had already taken out a subscription to the *UAUG's CPC User* and for a while I carried on with that as the main reading material to supplement my subscription to *Amstrad Action* and *ACU*. As my contacts grew, I was cajoled into partaking of another issue of *WACCI* by three ladies! They being Jacqui Owen, Angie Hardwick and Dot Williams, dire consequences were foretold if I did not take out a subscription, Still looking over my shoulder I have been a subscriber ever since.

Hello Playmates

The tape to disc programs were adding many useful contacts to myself, one being Nigel Mells, who kept the database up to date for the *Bonzo* programs, we had many hours on the phone to each other and swapped bits and pieces of information between us. I started a subscription to a newsletter by Carl Surry called *Playmates*, which had taken over the tape to disc transfer news from Colin Harris, who had edited a previous one called *The Bonzo Bulletins* as Colin had emigrated to Canada. When Carl decided to change his computer, the newsletter was taken over by Martin Cossins under the title of *Bonzo's Scrap Yard* which after a couple of issues became *Bonzo's Scrapyard*. After ten issues the tape to disc information began to dry up, and Martin said he was going to call it a day and close the newsletter. I then stepped in and am still currently managing to get an issue out approximately every three months.

Having obtained the information from the *Goldmark* tape to disc transfers I have been able to supplement the newsletter with information from those as well, so hopefully I can keep going for a while longer yet.

Have CPC will Travel

I have travelled to the *WACCI* Conventions for three of the last four years and have thoroughly enjoyed meeting old and new friends at all of them, and would hope that they will carry on in the future. I have also made a point to travel to Plymouth for the past three years to attend the *CPC SouthWest* meetings, as I feel that if someone spends the time and effort organizing something like these then we should make an effort to show our support to our fellow CPC users.

My current set-up

My current computer set-ups are as follows CPC 6128, two ROMBO ROM Boxes various ROMS, Brunword Mk II ROM, second 3 inch drive, 3rd 3.5 inch drive, Canon BJ30 Inkjet printer; CPC 464 converted to 6128 spec, DDI 1 disc drive, Star LC 24-10 printer, mainly for tape to disc transfers; a PC (given to me by my old firm) was a 486 66 MHz, but recently upgraded to a AMD 266 MHZ chip, using Corel Word Perfect 6.1 for the printing of my newsletter.

If anyone would like a copy of the current issue then please send an SSAE A4 envelope and an extra first class stamp to me at the above address. *Dave*

WACCI Convention 98

It all started with a very interesting conversation about CP/M with Clive Bellaby at a B&B in Birmingham. Thankfully the weekend did get better, with the not-so-small exception of the dreadful breakfast the next morning.

The 20th of September and we were all at Bescot Football Stadium once more. Angie Hardwick had managed to organise the convention (many thanks to her), although the stadium let us down at the last moment with regards to using the lifts. But we all survived another year. And unlike previous conventions, we haven't all collapsed afterwards and said "never again".

ROAMING AROUND

The main room this year was slightly more inhabited than the previous two conventions. As you walked through the door immediately on the left was Merline-Serve. Dad (Arthur) was largely in charge of technical questions, while I pocketed all the cash...
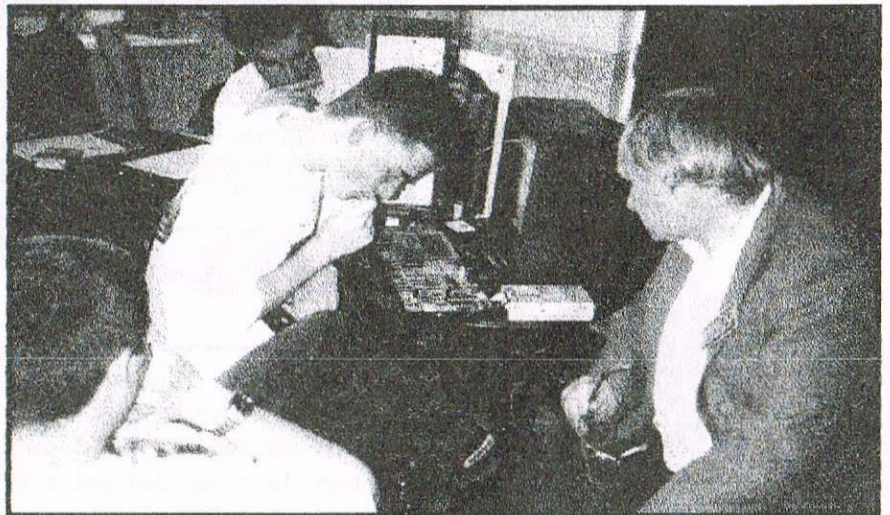
Doh! "I think It's a computer" said Richard

Brian Watson

The next table was Merline-Serve overflow, some PC items I had, along with issue three of my PC magazine Wibble. Brian Watson had brought up for me that day, fresh from the printers. As at least three people mentioned, it's a good publication to bring to the convention as it eases the pressure from WACCI to go over to PCs.

Following on from that was the STS Computers/BTL/Systeme D table. It was here that Rob Scott, Paul Fairman, Richard Fairhurst, Richard Wildey (Widdles), Simon Matthews and Simon Green were to be found for most of the day.

Following the tradition of previous years, most BTL members wore specially designed t-shirts, based on the new late-night cartoon South Park. Among the other activities at this table, Rob and Richard compiled

and were selling the Robot PD CD, which contained 800mb of data for CPC emulators on the PC. You'll read more about this another time, possibly at the News Desk. Simon Matthews also had a 6128 Plus on this table, which was used and abused at various points of the day . It's obviously adjusted to this as Simon confessed he'd lost the screws for the casing years ago....

A GRAVE DAY

At the next table was Roger Graves selling some of his CPC items. I didn't realise Roger was a WACCI member and was glad to meet him. A long time ago he was mad enough to catalogue all the clip art for the KAD PD library. Obvious why he's a "WACCI" member I guess.

Next to that was Jacqui Owen selling WACCI gear via the Market Stall. Her son Chris was also at the ready to call me Dragon Lady and help with heavy boxes. Chris designed the posters for the stalls and those pointing the way to the WACCI convention.

Chris Murray

The three tables on the far side of the room were taken up by Dave Simpson and all his computer items. Dave took even more stuff than Merline-Serve did this year, but we had the excuse that we had to fit BTL into the van.

Dave had so much stuff to sell it was amazing. Ranging from whole machines, to magazines (including old WACCIs) and even some PC items. I think he still has a lot of this available, so if he drops a line to the Market Stall or Fair Comment with his address people can drop him a line.
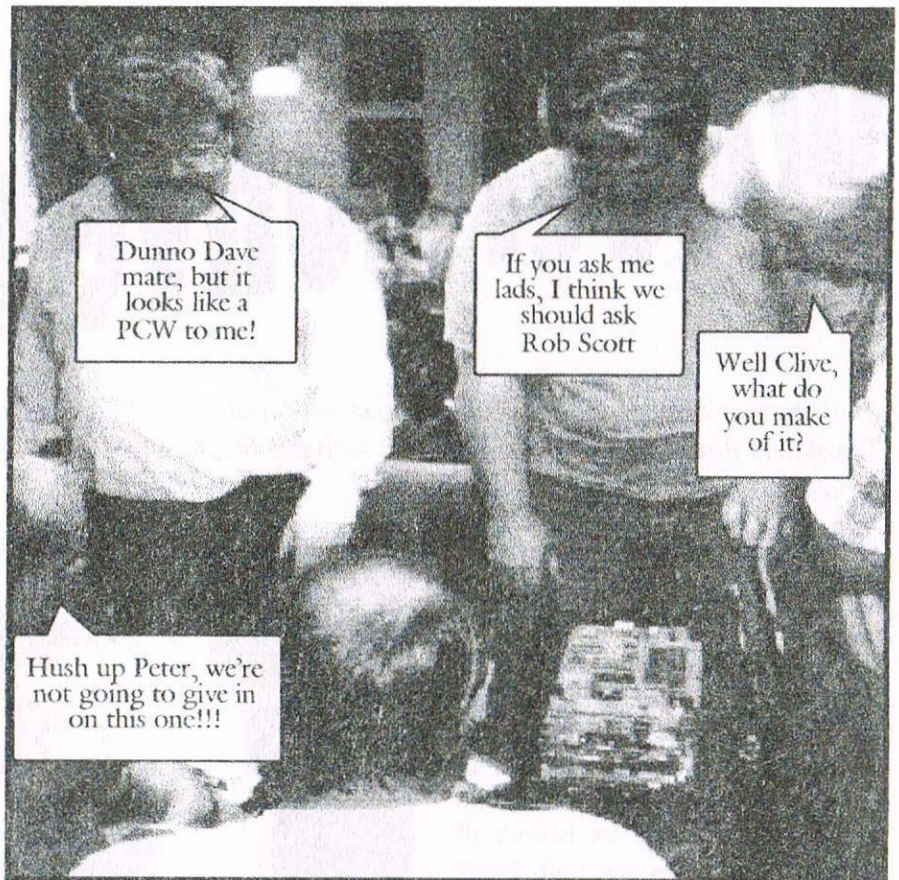
In front of the (closed) bar was another trader. They had lots of PCW stuff for sale, which pleased some members. Even my dad got a bit excited over some mice.

EVERYBODY PEEPS

In the other room, where the open bar was situated, lots of people gathered and chatted. There was also a lot of fuss over a somewhat strange CPC. A prototype PCW was the decision in the end, as chips were taken out and transplanted into Simon Matthews' ever-suffering Plus machine.

There was also the smoking room, which didn't seem quite so inhabited this year - perhaps that was just the absence of Philip DiRichleau. Clive Bellaby (and wife Josie) was the only other ex-WACCI Editor to attend the convention this year.

I saw Chairman Doug Webb briefly. John and Carol Bowley turned up around lunchtime with greetings of "Bowely! Bowley! BOWLEY!" from the BTL mobsters. John and Carol's middle son Andrew also made it this year and managed to convince his dad

to buy him some clothes. Boy these Editors are hard people. Carol, as always, went around snapping people with her camera, so expect some nice (read embarrassing) photos dotted around these here pages.

It's difficult to remember everyone you meet at the convention. Christine
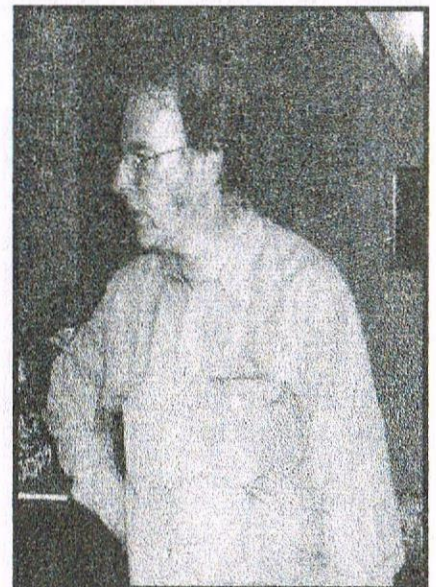
Dave Caleno

Raisin braved us all again this year and was playing on Rob Scott's PC at one point. Bob Houston was wondering around, Matthew and

Christine and Chris Raisin

one point. Bob Houston was wondering around, Matthew and Hilary Phillips (who hosts the WACCI website for us) I also saw. I had a chat with Dave Caleno, who was wearing a shirt almost as bright as John's.

Frank Frost and his wife Margaret were in attendance. I saw Lee Bragginton wander around in the morning, Bob Miller grabbed a Wibble from my table, Dave Teague was talking technicalities with Richard Fairhurst and Rob, Tony Jordan made it from Slough and was tickling Carol <<*What? –John*>>, Steve Potts pottered around and managed to make my day.

James Coupe arrived and had a fun and stimulating conversation with Simon Green, Brian Watson was lively as ever, even with a hefty suitcase full of goodies ("Two ROMs for a tenner, you're robbing me...."), Angie's foster daughter Kelly was bounding around and kept attacking Clive, not to mention stealing the sweets given to Angie as a thank you (which everybody else managed to steal from her later).

Clive was also presented with a gift from BTL, a cassette version of Dominoes.... And we all saw lots of other people who seem to have blurred into one long day. It was nice that there seemed to be an equal amount of people I recognised and those that I didn't have the foggiest about – proving we are still having a fresh influx of people. But the most important people of all this year were also in attendance – the Donut Stand in the market.

BACK TO THE FUTURE

This will probably be mentioned elsewhere in the magazine and also over the coming few months, but we now have to decide about next year.

As I mentioned above, new people were around this year, as well as those that have been around year in and year out. To keep things fresh it has been suggested to open up the convention as an all formats gathering. Facilities suggested to be included: bring and buy, demonstrations, emulators running, chat area, etc..

Everything is still very open to suggestion. Early thoughts are that it wouldn't be an "Amstrads and others" event, but rather a completely open event that WACCI just happen to be sponsoring. This was started this year, with PCs, PCWs and some hybrid computers creeping out of the woodwork.

The location is also being rethought for next year, but that's Angie's department. It's open to you for suggestions. We know most of you have more than just your CPC and we want to interest you and make sure you get the most out of the day, especially as it only happens once a year.

Michelle and Paul Wass

Bob Houston

AND AS THE SUN SETS

I seem to enjoy each convention more than the last. This year was especially good as we didn't have so many boxes to lug around. It was a very relaxed atmosphere and the day seemed longer than it has in the past.

People sit and chat leisurely, wander around the market and prop up the bar (at least, we think that's where Widdles was). It gets easier as you get to know people, seeing them once a year and getting snap shots of their lives. It's a weird concept, as is WACCI, but it works somehow.

Kelly

Eileen Large

And then the convention was over. It took us nine hours to do a four hour journey in an overheating van, eventually relying on the RAC as a taxi service. At one o'clock in the morning we were all squashed into a large rescue truck. Dad and Simon Green were in the front nodding off.

Simon Matthews and Paul Fairman were asleep next to us. Rob and I were talking and joking. As Rob said, even though we had stopped on every stretch of the M40's hard shoulder, we managed to keep our sense of humour throughout, which I think was the point of the whole weekend.

Angela

THE CAMERA NEVER LIES

Another good convention year for the Bowleys. I felt it to be as successful as previous years although not as busy. It is nice to meet friends again and I mean friends because over the few years that I have been involved with WACCI through John, I have always been greeted warmly by people that I only otherwise speak to on the telephone.

These same people always treat me as if I have been a personal friend of theirs for years. This year I decided to concentrate on the personal portrait style of photography instead of the
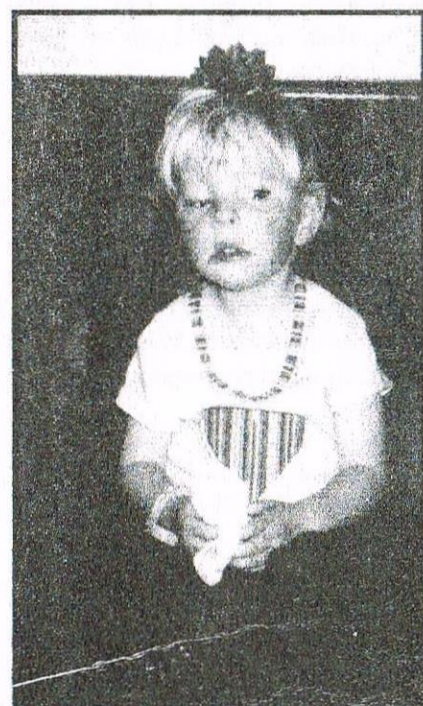

Peter Rogerson

usual group shots. I thought you would be interested in seeing more of the faces behind the names.

For the first time I was pleased to meet Eileen Large who graced me with a beautiful smile that lit up the room. A very pleasant lady to meet who told both John and myself that she was pleased to get her monthly WACCI and how much she liked it. I left her chatting to John as I was lured away by the sight of another photo in the making.

Chris Murray cut a dash in his smart maroon jacket. He didn't think the photo would come out well but as you have seen he was so wrong.

My only complaint about the photography side of things at a WACCI convention is the lack of suitable places to site your subject. It either has too much outside glare or the inside lights get in your way also the walls have a lot of pictures on which take the eye from the subject. But hey, never mind as it just adds to the fun.

I was particularly pleased with the photo of Christine and Chris Raisin as I am reliably told it is a rarity to get them both in the same photo so a pat on the back for me I think! Looking at the photo of Paul Wass and Bob Mohan forgetting themselves and

Paul Wass and Bob Mohan

showing off their skills at line dancing. Or at least thats what it looks like to me.

The photo of Dave Caleno I was wondering, with the position that his arm is in was he too taking the same line dancing class that Paul Wass and Bob Moham were.

I just had to get the photo of Bob Houston. Many of you may remember when the Compo Page was going, he nearly always came out of the winners hat. Well not Bob personally but his name did.

It was for this reason I took a good shot of him as I though that maybe a few of you would like to cut the photo out and put it up on a wall so you could throw darts at it! Only joking Bob....*honestly.*

Stephen Potts

Angela writes that Tony Jordan was tickling me....yes he was. But what tickled me most and I am sure that Tony was tickled too, was our conversation on massage. I will leave it to your imagination as to what the content of the conversation was, I'll just say that it was both naughty and nice.

I had the pleasure of meeting Paul Fairman for the first time. He was introduced to me as 'the guy who lives with Rob Scott'. Hmm, that certainly leaves a thought in your head. How could anyone put up with Rob for a start. Anyway he seemed a nice bloke but very quiet, probably because he was sat next to Angela for most of the time and maybe he couldn't get a word in.

I hope you will agree that I took some pretty decent photos this year but the one that I think is the best is the one of Stephen Potts, here on the left. I had a very quick look around at the lighting before I snapped him and I an extremely pleased with the result. I hope you are too Stephen.

The group shot that John has used was the one I took of a collection of enthusiasts musing over a CPC thing. Clive Bellaby, Peter Curgenven, Dave Stitson and a few others looked at this prototype PCW/CPC at every angle they could.
Even the knowing talents of Richard Fairhurst failed to throw any extra light on the subject but in any case they all seemed to be in awe of it.

I understand that it was later pulled to bits. Mind you, as John and I looked at the photo we both had an evil thought. What if we put speech bubbles in the photo and made them all say something funny. Well we put the speech bubbles in alright but as for what to write in them well that was something else. We giggled at a lot of each others suggestions and some were a bit too risque but we settled on the ones you see in the photo. Yes I know they are not very funny but then I am only a poor simple downtrodden woman who knows no better.........

Brian Watson was flashing...his t-shirt that is, so I took a photo to remind everyone that it is Brians 50th this month. Mind you I think he looks much younger than that. (*Will I get that small donation now Brian?*)

The Franks' Smart and Frost both smiled their usual brilliant smiles when I took photos of them with their wives Janet and Margaret. Frank Smart slipped me an intriging envelope that I was to give to John on our return home. I was to find out that it contained Franks Members Spotlight.

As usual Frank (*The Vindaloo Kid*) writes with much wit and interests the reader greatly. It is my belief that he has missed his calling because he writes so well.
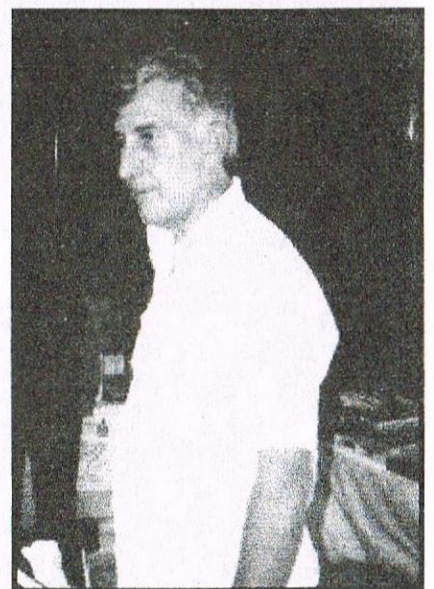
Tony Jordan

Frank and Janet Smart

Paul and Michelle Wass journeyed to the convention with the Raisins. They are a lovely couple and I chatted with Michelle as if I had known her for ages. Their journey home was a little on the cramped side though.

Chris Raisin had a gleam in his eye as he loaded up the car with his end of convention bargains. The poor cars boot looked more like an Alladins cave than a motor vehicles rear end. Fortunately it only took them about an hour to get home.

I couldn't begin to describe the conditions that Christine, Michelle and Paul had to travel in, I can only say that had they been stopped on a routine police check, they would not have been allowed to continue their journey!

Once again Angies little foster daughter Kelly stole the show. She not only pinched Angies choccies but adhered herself to Uncle Clive for most of the day. Clive took the most wonderful photo of her on his digital camera that makes my camera look like a cheapy. As soon as I can afford one myself I am going to indulge. Probably in the year 2010 when they replace everything else.

Roger Graves

Roger Graves is a lovely chap and I had the pleasure of chatting to him for a short while. He has given me 14 pages of compos for you all to try and I will see if there is room to start them in the next issue of WACCI.

We left the convention tired but happy. I was pleased with everyones opinion that there will be another convention next year.

We took Jacqui Owen, and her Daughter home before setting off on our journey home, after a supper at the McDonalds with Angie, Helen and Kelly. We got held up in the traffic on the M6 but didn't mind too much as it was at least moving.

All in all it was another successful year and we all look forward to the next one. Thanks Angie!

Ta-ra for now

Carol

Margaret and Frank Frost

Arnolds Basic   by Paul Fairman

Did you type in the telephone directory last month? Okay go and get last months 124 and come back..... Right, it was stated in the last part of Arnolds Basic that all numbers in brackets means line numbers such as [20] If you want to obtain this excellent series PD disc 115 is the disc that you need. Lets begin..... Over to you Paul. John

[20]

Next the program skips to line 4000 which is the routine to load the files (names and numbers) from disk. This routine is jumped to immediately to load in the files automatically without the user having to load them in. The loading routine is detailed later on.

[25-40]

A window is set (see towards the end of part six) that makes a nice box in the middle of the screen on which using the highly complex (joke) PRINT command to print the main menu of the screen. Just before this an ON BREAK CONT is set so that the ON BREAKs later on in the program are made ineffective.

The next WINDOW command sets the normal size screen window, uses a CLEAR INPUT command (6128 only) to make sure any keys you have pressed before are wiped. KEY DEF 66,0,13 sets the ESC key to be the same as pressing RETURN.

[40-50]

The option that the user selects from the menu is stored into the variable "z". Now, if you pressed RETURN (or ESC 'cos RETURN=ESC from the KEY DEF) then z=0. So IF z=0 THEN set the normal KEY DEF and END the program. ESC is given as the exit option from the menu.

[60-70]

A superbly good command for a menu system is ON. The value of "z" depends of the entry from the menu.

The possible routines that can be GOSUBed to are 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, and 9000 which contain the vital routines for the program that

were written in separately. So if the value of "z" was 1 then the ON command makes the program jump to line 1000 etc etc. You will have to use your imagination for the rest.

The MODE 1:CLS:GOTO 25 bit in line 70 is so that when the various routines are RETURNed they will RETURN to line 70, the line after to which they were GOSUBed. CLS CLears the Screen and GOTO jumps to the specified line number, remember.

The first routine is the one that allows INPUT of the names and numbers. All routines have REM commands above them to identify them during the de-bugging process. A little message informing the user of the possible key presses are given at the top of the screen. All such messages appear at the top, so the user gets used to it very quickly to look here for help. User friendliness is important, especially in a program that has no documentation with it.

A WINDOW is set just below this message, to ensure that if the screen scrolls upwards then the message remains there; cunning programming. Also a few years ago when I owned an Atari 65XE computer a bloke called Ian Finlayson, a columnist of Atari User said that it was bad practise to GOTO or GOSUB to REM commands. Therefore I have acted on this advice and the REM commands are just before the routine.

The FOR NEXT loop is something that has been introduced, but is very important in BASIC. Many of them are used in this program. Take a look at this short program, but first look at around page five of part two

for details on what variables are. You will find this section under the heading VARIABLES (cor).

```
10 FOR a=1 TO 12
20 PRINT a
30 NEXT
```

If you RUN this program then on the screen, line by line, the numbers 1 to 12 inclusive will be printed. The FOR NEXT loop makes the value of a variable increase or decrease throughout the loop. The NEXT command makes the program jump back to the FOR, increasing the value of "a" by one. So the program takes the variable "a" and it starts at one and goes up TO 12. The value of "a" is then printed by line 20, and the loop continues after the FOR instruction by line 30. The two numeric values in line 10 can be anything, and an additional command can be used if you wish for the increment or decrement (by how much the value of "a" goes up or down) using STEP.

```
FOR a=56 to 43 STEP -3
```

Will make "a" start at 56 and go down to 43 in STEPs of 3.

[1000]

Many simple FOR NEXT loop are created in the telephone directory. It makes the variable "x" start at 1 and possibly go all the way up to 500 (the maximum number of entries in the directory), although it rarely goes as far as 500 because it can be terminated before this. There are many more instructions that are carried out within the FOR NEXT loop in the telephone directory.

[1010-1030]

The LEN command is used to detect how many characters are in a string

variable. Our string is name$(x), and this command looks at the string and checks to see if it has more than zero characters in it. In other words it checks to see if name$(x) contains something. name$(x) to begin with means name$(1) because remember the value of "x" depends on the FOR NEXT loop.

IF LEN(name$(x))>0 The ">" sign means "greater than" and the "<" means is "less than" if you recall primary school.

The value of "x" is printed which is the datafile number and then the INPUT command is re-introduced from part two in a slightly different guise. Instead of using:

    PRINT"Name ";
    INPUT name$(x)

you can use: INPUT"Name ";name$(x). These things are given by BASIC as short cuts. The next line should be easy to work out. The CALL &BB18 bit simply waits for a key to be pressed, a very useful CALL.

WARNING: Don't use the CALL command if you do not know what you are doing. The computer can easily reset using values after the CALL that you don't know. Also don't forget the "&" (ampersand) before the number.

[1040]
The next line uses UPPER$, NOT a variable. This converts a string from whatever into capital letters. Next, if you have pressed RETURN (or ESC) then the program jumps to line 5000 to save the latest entries; automatic saving as well as loading.

Again INPUT is required for the phone number and then the NEXT command is encountered to go back to the command straight after the corresponding FOR. The loop can be broken at line 1090. Fine. Moving along to the next routine which is designed for searching through the datafiles. As the REM command says, this routine is "shared" by the searcher

and the printer. Just remember that the variable "dump" equals zero when the search routine goes into action. At this point it is advised to skip to the description of WINDOW in part 6 which you should have done before. This section gives details on streams.

Lines 2010 onwards for a while prints a sub-menu offering to search for names telephone numbers or a specific datafile. Pressing ESC exits this menu. KEY DEF defines the ESC key as an "*" for use in the following line. The INKEY$ command was detailed in the previous part. This routine on accepting the option selected from the user jumps to a sub-routine which offers the features. The first feature is the NAME search.

You should be able to follow this and the other sub-routines through yourself as it is much the same programming commands used and detailed previously. A new command not discussed yet is INSTR. A frightfully useful command if I may say is the INSTR command. Take a look at the format:

    IF INSTR(name$(x),s$)=0
    THEN 12060

[12030]
(the program skips to this sub-routine from line 2060. Yet again we are in a FOR NEXT loop hence the big explanation earlier on.

s$ contains the "name" that you entered a few lines ago and a search for this is commenced and compares it with name$(x). If the search fails then the equals zero comes into action and jumps to the NEXT command.

However, if the value should not equal 0 ie a corresponding string has been found, then the next line is executed. Having said that a corresponding string must be found, name$(x) does not have to be identical to s$, as long as s$ is included in name$(x) then that is

accepted. This vague search is because of the INSTR. If you press RETURN or ESC then the sub-routine RETURNs to the main routine, and then if the user desires RETURNs to the main menu. Notice the constant GOSUB and RETURNing system to flip between parts of the program. Good programming technique.

[3000-3090]
The listing of files routine is very simple and is a stand alone routine, not jumping to other parts. It checks to see if name$(x) is empty and if so skips the printing of the next datafile on the screen using the NEXT command, therefore increasing the value of "x".

If x>500 the command after the NEXT is executed, which prints that the list is complete, and GOSUBs to line 62000 which is a routine that is detailed below. When a file has been placed on the screen, different colours are used and after the detection of whether key 66 (ESC) has been pressed to quit, or the [ENTER] key (6) to continue listing.

[62000]
Line 62000 is a routine that asks for the user to press SPACE BAR. A CALL &bb18 is used to wait for a key to be pressed and the routine is terminated using a RETURN. Great inputting, searching, listing these files I am sure but when you turn the computer off, all files are lost, unless of course you save them to disk. The routine at line 5000 will be dealt with first, as saving must been done before loading, which is at line 4000.

The existing file called DATA.DAT on the disk is erased, if saving of files has been done when using the program before. DATA.DAT is an ASCII file containing the names and numbers and as such the ASCII file must be opened for input, and the command that does the trick is OPENIN as in OPEN file for INput. Before this, a pleasant centred prompt tells you that the file is loading. *More next month.*

[5000-5040]

Forget about RUN and LOAD with ASCII files. If you want more info on what an ASCII file is take a look at the last but one page of part ten.

Such files must be opened with OPENIN and OPENOUT and closed with CLOSEIN and CLOSEOUT. If you want to save a file to disk use OPENOUT. You should close the file once you have finished taking INPUT from the disk, and once finished writing a file to disk close it with CLOSEOUT.

You still have to INPUT and PRINT information (data) from and to disk and the much discussed streams are used. You have to direct the PRINT command to the disk, stream nine.

So a FOR NEXT loop is introduced and so the value of "x" goes up and up and up. The PRINT#9,name$(x) bit writes name$(x) to disk. The same for the tel$(x). IF name$(x)=nothing ("") THEN 5030. This means that the files have come to an end. The OUTput file is then CLOSEd.

You may notice that the programming commands are being used as part of an English sentence to try and make it easier to understand. The routine is RETURNed to the menu.

[4000-6010]

Reversing of the process is done at line 4000 to load in the files. Option six justs CATs the disk which I thought I'd bung into the program, and option eight makes a back up of the entire program. Routine number eight was written in during the programming stage to quickly and easily SAVE the latest version of the program without having to do it directly from BASIC.

[7000]

Two left, two to go. Editing of the files is routine number seven and what a comprehensive routine it is to. You can search for names, numbers, specify the datafile number (determined by "df"), or the program produces a list of the files for you to wade through and edit at will.

[52000]

For the name and number editing the program uses the INSTR command detailed above to find the necessary datafiles, it then produces a list of the files that are accepted and asks for the datafile number to edit. Let's say that you entered the number 45 for the variable "df" (datafile number), name$(aa), that's name$(45) is the datafile that is printed. The variable is then given to be edited. Typing in a new string for a variable always overwrites previous data. The datafile number editor is the same as the previous two excluding the searching for names and numbers. You will find this sub-routine faster than all the others because the program does not have to go searching through the array. The list editor prints the files as with the lister routine earlier on, but gives the extra option to press "E" to edit the file.

[9000]

It has been said before that the searching and printing routines are shared. When option nine is selected from the menu, the program jumps to line 9000. This sets a variable called "dump". The word dump in computers means putting something somewhere. If we dump the data (names and numbers) to disk we do so using PRINT #9 as described before, and the PRINT #8 is used to dump them to the printer. However you will find no PRINT #8 within the program; the variable "dump" controls this. If "dump"=0 which it does normally when the program is RUN (when a program is RUN all variables from BASIC are lost, so don't use RUN within a program when variables are vital), then data gets printed on the screen, but if option nine is selected from the menu then line 9000 makes "dump"=8, the printer. Then when PRINT #dump ... is used data gets printed either on the screen or on the printer, because if dump equals 8 then it is the same as #8, or if dump equals 0 then it is the same as #0.

Another point to mention is that in line 2090, if dump=8, ie option 9 has been selected, then the searcher/printer gives an extra option, option 4 from the sub menu. This goes wizzing off to line 15000 where a routine has been written that prints all of the datafiles to the printer.

The ON BREAK GOTO command means the program jumps to line 16000. At line 16000, the variable "brek " is set to 1. Note brek is a variable, that is set to one when ESC is pressed. This is what is known as setting a "flag". The flag is set and the information can be used later on in the program. It can prove that the lines in a program have been already run. In this example, brek is set to one so that when the program RETURNs to line 15052 and brek is detected as equalling one, the program jumps to the main menu with another RETURN.

[63000]

Part five of this text details error messages churned out by the CPC, and the below program uses an error detection system from BASIC so that if a disk is missing or other such errors you are not returned to BASIC. The DERR command meaning Disk ERRor is used with the corresponding code. RESUME resumes normal execution.

I don't think there are any bugs in the program, but if there is then don't think "Oo er, a program bug. I can't use this program anymore". Look around the program lines where you think the problem is and "de-bug" it yourself. The program was written from BASIC as normal but not SAVEd in the normal way. SAVE"DIR",A was used and the listing was saved as an ASCII text file. The file could then be imported into my word processor using the MERGE feature of VDE and mixed with the main text. *More next month.*

EQUATION GRAPHER

by James Hoskisson

In case you haven't guessed from the title, the program on yonder page is designed to draw a graph from any given equation. This is useful for mathematical applications but I can't think of anything else that it could possibly be used for.

```
10 MODE 2:POKE &AE67,&1C:POKE &AE69,&1C:POKE &AE6B,&1C:POKE
&AE6D,&1C:ADDR=6716
20 INPUT "ENTER EQUATION (USE x FOR X-AXIS)";C$
30 A=0
40 A=INSTR(A+1,C$,"{"):IF A>0 THEN GOSUB 500:D=0:C$=A$:A$="":GOTO 40
ELSE A$=C$
50 FOR I=1 TO LEN(A$):B$=UPPER$(MID$(A$,I,1)):IF B$="Y" OR B$="X" THEN
POKE ADDR,&D:POKE ADDR+1,0:POKE ADDR+2,0:POKE
ADDR+3,&D8:ADDR=ADDR+4:CQR$="":MN=0:NUM$=""
60 IF B$="=" THEN POKE ADDR,&EF:ADDR=ADDR+1:CQR$="":CQR=0:MN=0:NUM$=""
70 IF B$="^" THEN POKE ADDR,&F8:ADDR=ADDR+1:CQR$="":CQR=0:MN=0:NUM$=""
80 IF B$="+" THEN POKE ADDR,&F4:ADDR=ADDR+1:CQR$="":CQR=0:MN=0:NUM$=""
90 IF B$="-" THEN POKE ADDR,&F5:ADDR=ADDR+1:CQR$="":CQR=0:MN=0:NUM$=""
100 IF B$="(" THEN POKE
ADDR,&28:ADDR=ADDR+1:CQR$="":BRACKET=BRACKET+1:CQR=0:MN=0:NUM$=""
110 IF B$="/" THEN POKE
ADDR,&F7:ADDR=ADDR+1:CQR$="":CQR=0:MN=0:NUM$=""
120 IF B$=")" THEN POKE ADDR,&29:ADDR=ADDR+1:CQR$="":BRACKET=BRACKET-
1:CQR=0:MN=0:NUM$=""
130 F$=UPPER$(MID$(A$,I+1,1)):IF ASC(B$)>47 AND ASC(B$)<58 THEN
NUM$=NUM$+B$:IF F$<>"" THEN IF ASC(F$)>47 AND ASC(F$)<58 OR F$="."
THEN 160
140 IF B$="." AND DP=0 THEN NUM$=NUM$+B$:DP=1:IF F$<>"" THEN IF
ASC(F$)>47 AND ASC(F$)<58 THEN 160
150 IF ASC(B$)>47 AND ASC(B$)<58 THEN CQR=VAL(NUM$):D=0:GOSUB 370:DP=0
160 IF F$="X" AND INSTR("{^=+-/*(",B$)=0 THEN POKE
ADDR,&F6:ADDR=ADDR+1 ELSE IF F$="(" AND INSTR("-+*/{^(=",B$)=0 THEN
POKE ADDR,&F6:ADDR=ADDR+1
170 IF BRACKET=1 AND INSTR("+-*/^()",F$)>0 AND NQR=1 THEN POKE
ADDR,&29:ADDR=ADDR+1:BRACKET=0:NQR=0 ELSE IF BRACKET=1 AND F$="" AND
NQR=1 THEN POKE ADDR,&29:ADDR=ADDR+1:BRACKET=0:NQR=0
180 IF B$="*" THEN POKE
ADDR,&F6:ADDR=ADDR+1:MN=0:CQR=0:CQR$="":NUM$=""
190 IF B$="}" THEN POKE ADDR,&F8:POKE ADDR+1,&28:POKE ADDR+2,&F:POKE
ADDR+3,&F7:ADDR=ADDR+4:BRACKET=1:NUM$="":NQR=1
200 NEXT
210 POKE ADDR,1:POKE ADDR+1,&C9:POKE ADDR+2,0:POKE ADDR+3,0:POKE
ADDR+4,0:ADDR=ADDR+5:POKE 6712,(ADDR-6714) MOD 256:POKE 6713,(ADDR-
6714)\256
220 INPUT"What Range is the X-Axis (Separate Numbers with a
Comma)";A,B
230 STRL=0:STRH=0:FOR I=A TO B:X=I:GOSUB 530:IF X>STRH THEN STRH=X
240 IF X<STRL THEN STRL=X
250 NEXT:CLS:WINDOW 1,80,25,25:IF SGN(STRL)<0 THEN YOFF-
SET=ABS(STRL*(318/(STRH-STRL))) ELSE YOFFSET=0
260 IF SGN(A)<0 THEN XOFFSET=ABS(A*(586/(B-A))) ELSE XOFFSET=0
270 PRINT CHR$(23);"3"
280 MOVE 44,72+YOFFSET:DRAWR 586,0:FOR I=44 TO 630 STEP 39:PLOT
I,70+YOFFSET:PLOT I,68+YOFFSET:PLOT I+1,68+YOFFSET:PLOT I+1,70+YOFFSET
290 IF ABS((I-44)*((B-A)/586))<1 THEN DP=2 ELSE IF ABS((I-44)*((B-A)/
586))<10 THEN DP=1 ELSE DP=0
300 STRIG$=STR$(ROUND((I-44)*((B-A)/586)+A,DP)):TAG:MOVE I-
(LEN(STRIG$)*6),58+YOFFSET:PRINT STRIG$;:TAGOFF:NEXT
310 MOVE 44+XOFFSET,72:DRAWR 0,390:FOR I=72 TO 390 STEP 27:PLOT
43+XOFFSET,I:IF ABS((I-72)*((STRH-STRL)/318))<1 THEN DP=3 ELSE IF
ABS((I-72)*((STRH-STRL)/318))<10 THEN DP=2 ELSE IF ABS((I-72)*((STRH-
STRL)/318))<100 THEN DP=1 ELSE DP=0
320 STRIG$=STR$(ROUND((I-72)*((STRH-STRL)/318)+STRL,DP)):TAG:MOVE
```

If you are one of those people, like me, who dives in and types any listings they find that look interesting, you'd better read this article first, otherwise you could waste yourself a lot of time.

Who Needs a Reason?

Ok, I know that there isn't really any point in writing a program, and then saying that it isn't that useful, but I started to write it for my own purposes i.e. maths coursework. I wasn't really successful with that idea since it took me about four months to write the program, and this was well past the deadline. (I ended up drawing all the graphs by hand anyway!)

Spurned on by reading a few letters in WACCI about people who were interested in mathematical problems, I decided to add a few extras and make it into a program which could grace the pages of WACCI. I finished the program, eventually, and I'm now sitting writing this article about it.

What Does it do?

For those of us who aren't really au fait with graph drawing the program takes an equation and plots each point on the y-axis from the result of the value on the x-axis. The program allows the user to enter any range for the x-axis, including negative numbers, although the bigger the range the longer the graph will take

to draw. The program requires the equation of the graph to be entered first and should be entered in the form y= 12x, where y is the y-axis and x is the x-axis. To achieve any power of a number you must enter a '^' character, like in BASIC, e.g. 4^2, would be 4 squared.

Since there isn't a root symbol on the keyboard I decided to use '{' for root, e.g. 2{ would be square root, 3{ would be cubed root etc. For square root the root symbol must have a '2' before it otherwise the program would probably come up with 'operand missing'.

The only restriction is that mathematical functions like sin, cos, and tan can't be used. All numbers including floating point numbers can be used within the equation, though.

The program only plots from the x-axis onto the y-axis so don't try it the other way around. (You could try it, but the program would probably get confused and come up with the same graph anyway, I think!)

What's so Good About it?

This program is unusual in the fact that it is written in BASIC and it actually alters the BASIC listing from within the program, hence the length of the listing. For the program to graph the equation the program needs the equation as part of the program.

This is sort of a contradiction in terms when you say that it can graph any equation but needs the equation permanently as part of the program, so I had to design it so that the start of the program alters the end of the program before the end of the program is executed.

The alternative to doing this would be to edit line 530 of the listing and add the equation to the program before it is executed but this isn't exactly user friendly and BASIC doesn't like equations that are in mathematical form, anyway. (e.g. 12x

```
43+XOFFSET-(LEN(STRIG$)*8),I+4:PRINT STRIG$;:TAGOFF:NEXT
330 TAG:MOVE 631,40+YOFFSET:PRINT CHR$(188);:MOVE 10,398:PRINT
"y";:X=A:GOSUB 530:TAGOFF:MOVE 44,72+(X-STRL)*(318/(STRH-
STRL)):FOR I=A TO B:X=I:GOSUB 530:DRAW 44+(I-A)*(586/(B-
A)),72+(X-STRL)*(318/(STRH-STRL)):NEXT
340 CLS:PRINT CHR$(23);"0";:PRINT"ENTER X VALUE       ";:NUM$=""
350 A$=INKEY$:IF A$="" THEN 350 ELSE IF ASC(A$)>47 AND
ASC(A$)<58 OR A$="." THEN NUM$=NUM$+A$:PRINT A$;:GOTO 350 ELSE
IF A$=CHR$(127) AND LEN(NUM$)>0 THEN
NUM$=MID$(NUM$,1,LEN(NUM$)-1):PRINT CHR$(8);" ";CHR$(8);:GOTO
350 ELSE IF A$<>CHR$(13) THEN 350
360 IF LEN(NUM$)=0 THEN 350 ELSE PRINT "  ";:PRINT CHR$(243)"
";:X=VAL(NUM$):GOSUB 530:PRINT X;:CALL &BB18:GOTO 340
370 IF DP=1 THEN 390
380 IF CQR<10 THEN POKE ADDR+D,CQR+&E:GOTO 440 ELSE IF CQR<256
THEN POKE ADDR+D,&19:ADDR=ADDR+1:POKE ADDR+D,CQR:GOTO 440 ELSE
IF CQR<32768 THEN POKE ADDR+D,&1A:ADDR=ADDR+1:POKE ADDR+D,CQR
MOD 256:ADDR=ADDR+1:POKE ADDR+D,CQR\256:GOTO 440
390 POKE ADDR+D,&1F:ADDR=ADDR+1:LP=0:S=CQR:IF CQR<32768 THEN
450
400 LP=LP+1:S=S/2:IF S>=32768 THEN 400
410 ADDR=ADDR+4:POKE ADDR+D,LP+&8F:CQR=CQR-(32768*(2^(LP-
1))):ADDR=ADDR-1:TMP=INT(CQR/((32768*(2^LP))/256)):POKE
ADDR+D,TMP:CQR=CQR-TMP*((32768*(2^LP))/256):ADDR=ADDR-1
420 TMP=INT(CQR/((32768*(2^LP))/65536)):POKE
ADDR+D,TMP:CQR=CQR-TMP*((32768*(2^LP))/65536):ADDR=ADDR-
1:TMP=INT(CQR/((32768*(2^LP))/16777216)):POKE
ADDR+D,TMP:CQR=CQR-TMP*((32768*(2^LP))/16777216):ADDR=ADDR-1
430 TMP=INT(CQR/((32768*(2^LP))/4.2949673E+09)):POKE
ADDR+D,TMP:CQR=CQR-TMP*((32768*(2^LP))/
4.2949673E+09):ADDR=ADDR+4
440 ADDR=ADDR+1:RETURN
450 S=S*2:IF S<=32768 THEN LP=LP+1:GOTO 450
460 IF CQR>=0.5 AND CQR<=1 THEN LP=LP-1
470 ADDR=ADDR+4:POKE ADDR+D,&8F-LP:CQR=CQR-(32768*(2^(-LP-
1))):ADDR=ADDR-1:TMP=INT(CQR/((32768*(2^-LP))/256)):POKE
ADDR+D,TMP:CQR=CQR-TMP*((32768*(2^-LP))/256):ADDR=ADDR-1
480 TMP=INT(CQR/((32768*(2^-LP))/65536)):POKE
ADDR+D,TMP:CQR=CQR-TMP*((32768*(2^-LP))/65536):ADDR=ADDR-
1:TMP=INT(CQR/((32768*(2^-LP))/16777216)):POKE
ADDR+D,TMP:CQR=CQR-TMP*((32768*(2^-LP))/16777216)
490 ADDR=ADDR-1:TMP=INT(CQR/((32768*(2^-LP))/
4.2949673E+09)):POKE ADDR+D,TMP:CQR=CQR-TMP*((32768*(2^-LP))/
4.2949673E+09):ADDR=ADDR+5:RETURN
500 F=LEN(C$)-A:IF A>0 THEN FOR I=A+1 TO LEN(C$):IF
MID$(C$,I,1)="(" THEN D=D+1 ELSE IF MID$(C$,I,1)=")" AND D=0
AND F=LEN(C$)-A THEN F=I-A-1 ELSE IF MID$(C$,I,1)=")" THEN
D=D-1 ELSE IF INSTR("+-/*^{",MID$(C$,I,1))>0 AND D=0 AND
F=LEN(C$)-A THEN F=I-A-1
510 NEXT:L=A-1:D=0:FOR I=A-1 TO 1 STEP -1:IF MID$(C$,I,1)="("
AND D=0 AND L=A-1 THEN L=A-I-1 ELSE IF MID$(C$,I,1)=")" THEN
D=D+1 ELSE IF MID$(C$,I,1)="(" THEN D=D-1 ELSE IF INSTR("+-/
*^{=",MID$(C$,I,1))>0 AND D=0 AND L=A-1 THEN L=A-I-1
520 NEXT:A$=A$+MID$(C$,1,A-L-
1):A$=A$+MID$(C$,A+1,F):A$=A$+"}":A$=A$+MID$(C$,A-
L,L):A$=A$+MID$(C$,A+F+1,LEN(C$)-F-A):RETURN
530 :
```

would have to be 12*x in BASIC) So I decided to go with option number one, this is where our helpful Firmware Guide comes into its own. The Firmware Guide has all of the BASIC tokens listed on pages 72 and 73, so the next step was to find a way of converting the equation into the BASIC tokens for it, so the interpreter could understand it.

The problem with this method is that the address of the line number, where the equation will be, changes as more is added before that line.

This means that if any of the characters in the following listing are missed out, or more added in, the address of the last line will change and could lead to the program crashing and writing over the listing. To this

end it would be a good idea to save the listing before running it.

The Listing - Line by Line

10 This line sets the variables area 256 bytes higher in memory so that the new line (containing the equation) won't overwrite the variables area and cause the program to crash.

30 Sets A to 0. A is used to point to the '{' symbols in the equation because they have to be reversed.

40 This line looks for the '{' symbols and makes a detour to line 500 if any are found so that it can be reversed.

50-200 This part of the program searches through the CS string and converts it into BASIC form at line 530.

130-150 These lines find any digits and store them in the NUM$ string until all of the digits in the number have been found and then converts them into a number and stores it in CQR.

210 This line changes the length of line 530 so that the BASIC interpreter knows how long the line is.

230-330 This part of the program draws the axes and the graph of the equation.

230-240 Checks the range that the Y-axis will have to be.

250-260 Calculates whether the axes will need to be offset and by how much so that they go through the point 0,0.

270 Instructs all the lines to be transparent so that the text will be visible if it is drawn over.

280-300 Calculates where the Y-axis will be drawn and the values that will be at each point along it.

310-320 Does the same with the X-axis.

330 Labels the axes (Very Important!), calculates where the line/curve will start, and draws it.

340-360 Asks for a value on the x-axis and then calculates the value on the y-axis.

370-490 Converts the numbers into a form the interpreter can understand. The numbers come in four flavours, single digit numbers i.e. 0-9, single byte numbers i.e. 10-255, integer numbers 256-32767, and finally, large numbers and floating point numbers i.e. over 32767 Š(or less than -32767) and decimals.

380 Deals with the first three types of number

400-430 Deals with the numbers over 32767. As you can see things start to get complicated around this point.

450-490 Deals with floating point numbers less than 32768.

500-520 Is a subroutine to swap around the expressions before and after any '{' symbols because the roots have to be in the form $n^{(1/x)}$ in BASIC rather than the more familiar $x\{n$.

530 This line seems pointless but is necessary otherwise the new line that is created for the equation won't have a line number and lots of problems will arise.

That's It

Well have fun typing, and remember to save the program before you run it. If you want to know more about the inner workings of the BASIC interpreter read my next article, coming to a WACCI soon. (I hope!)

James

WACCI PC

Part 1 : Where To Begin. The real problem with PC's (Personal Computers) is that there are so many problems, and they are of such a diverse nature, that I could spent the rest of my life answering questions, and find that the list of unanswered questions was growing longer as I grew older. So where do I start? I suppose that some background information, followed by an idiots guide to buying a PC might be the best place to start.

ITS CHANGED

The one constant with PC's is that the rate of change is increasing, and products have a very short shelf life.

The Epson Stylus 800 inkjet printer had a three year guarantee, but was replaced by the Epson Stylus 820 with 18 months of its launch, mind you the Epson Stylus 820 was replaced by the Epson Stylus Colour 200 within a year, and the Epson Stylus Colour 200 was replace within a year by the Epson Stylus Colour 300 and so on.....

It was new and expensive, and then it was old and out of date within a year constant change and a short shelf life is what's happening.

The other main problem is that constant decrease in prices, allied to a constant improvement in specification. Because I like a laugh (or cry) I've kept a copy of PC Directs' August 1995 price guide.

I'll give you the average price in August 1995, you guess the current price, and I'll give you the real price at the end of the page. All these prices include the dreaded VAT.

    16 mb Simm : £447
    Pentium Motherboard : £387
    Soundcard : £105
    14" Colour Monitor : £186
    2gb Hard Drive : £878

You'll need to be up to date on your PC prices if you want (or expect) to be anywhere near to the correct answers!

WHICH MACHINE TO BUY?

The real answer is that you need to buy a computer that matches your current needs, and your immediate future needs. Don't bother planning any further than that, because by the time you get there, whatever you thought you needed will be old hat, out of date, and will have dropped rapidly in price.

I well remember a discussion at work some ten years ago as to how long the recently purchased Amstrad 2286 would remain as a current computer as it had the brand new Intel 286 processors at the heart of its system. And I well remember that the tutors were talking of between 5 and 10 years. Here we are ten years later and the 286 was replaced by the 386, the 386 was replaced the 486, the 486 was replaced by the 3.3volt 486, the 486 3.3volt was replaced by the 586, the 586 was replaced by the 686
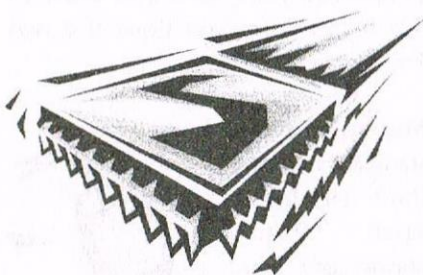
(Pentium) and the 686 (pentium) has been replaced by the Pentium II. And there are least two more brand new generations of processor under development.

If we measured the computers processing power (or speed) processing speed, a 286-20 mhz from

1988 would have a speed rating of 1, and on the same scale a Pentium II 350mhz would have a speed rating in excess of 800 (and it's cheaper today than the 286 was back then.)

If you want to use a lot of Graphics or do a lot of scanning then you do need a fast chip (Pentium 166 or better) and 32mb of memory. You should also have a graphics card with a minimum of 2mb of memory on board, and you'll need a good monitor. If you decided to use Windows 95 instead of Windows 3.11 you'll need to increase the specification to a minimum of, Pentium 200 or better, and 64mb of memory. You should also have a graphics card with a minimum of 4mb of memory on board.
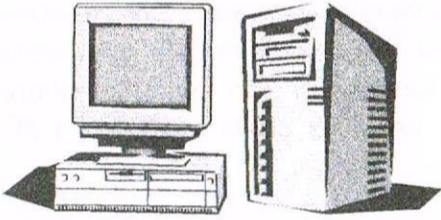
If you are going to write the odd letter or two and place the odd graphic in your homework (or the letter to the boss telling him you've won the lottery) then you'll be able to get away with a 486 Dx2-66, with 8mb of memory and a 256k memory graphics card. It'll be slower, a lot cheaper, and it'll do the job adequately. Unless you want to use Windows 95, in which case you'll need to Pentium 200 specification outlined above.

SELECT-A-CASE

PC's come in three types of case, and two styles of case. The styles are "Desktop" or "Tower". Desktop cases are designed to sit on a desktop with the monitor sitting on top of the case, and Tower cases were designed to fit under the desk (leaving a larger work

surface.) Desktop or Tower, just chose the one that suits you.

The three type of case are "Half-Height or Slimline", "Standard" and "ATX". ATX cases have been specifically designed for Pentium II use only. Slimline cases look nice but will not take the newer, faster PCI graphics cards, and have little room from simple additions such as CDRom drives - so they should be avoided.

Pick a standard case for 486, 586 or 686 (pentium) machines, and an ATX case for a Pentium II machine.

Graphics cards are either the older slower ISA 16 bit cards, or the newer faster PCI cards. The PCI cards are the current standard. However, unless you intend to do a lot of graphics it doesn't matter which you use. PCI graphics cards will not fit in the "Half-Height or Slimline" cases.

The older motherboards which accept the 468-Sx33, 486 Dx33 and 486 Dx2-66 processors (chips) all work on 5 volts, the later motherboards which accept the 486 Dx4-100 and above and 586 range of chips are all 3.3 volts.

If you intend the upgrade the chip from the 486 Sx33 range of motherboards you must have a motherboard which will accept both voltages. If it doesn't - you cannot upgrade.

BUYING A CDROM
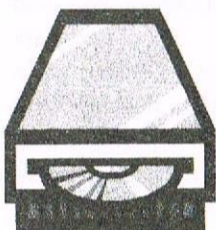
If you are going to use a CDRom to play games then you'll need a 16x speed or better. You can buy a 32x speed CDRom for £60. If you are only going to use the CDRom to load software it doesn't matter how fast or slow it is, because it'll be used so little.

However, as a warning, you'll almost certainly need a CDRom for a new computer as more and more manufactures are putting their software on CDRom (it used to be on 3.5 inch disks) as it saves them money.

THANKS FOR THE MEMORY

Memory comes in 30 pin Simms, 72 pin Simms and Dimms. Most new motherboards accept Dimm Memory which is much faster than Simm memory. Pentium II's only use Dimm memory. 30 pin Simms are the old standard, and have little or no use for computers built in the last 5 years or so. 72 pin Simm memory was the standard until recently, when it was replaced by Dimm Memory.

Simm memory comes in a range of types such as "fast page", "parity", "non-parity" and "EDO". EDO is the fastest. Be warned, not all computers takes all types of memory, and if you use more than one type of memory in your computer you are asking for trouble.

You should always use matched memory chips. On Pentium (686) motherboards you must use memory in matched pairs, so if you want 16 mb of memory, you must use two 8mb memory chips.

Mice come in two standard varieties, PS/2 (it has a small round connector) and standard which have a 9 pin "D" connector.

Keyboards also come with two different connectors, one is a small round PS/2 connector about 6mm in diameter, and the other is a larger round connector, about 12mm in diameter. You can get an adapter to change a large connector to a small connector, but you cannot get an adaptor to change the small connector to a large connector.

If you are offered an Apple Mac which is a very good make of computer, take care, as it uses non-standard parts which are only available from Apple Mac dealers, and they charge the earth!

A standards PC floppy drive cost around £25 fitted, whilst the Apple Mac version is over £140 for the same item.

TIME AND TIDE WIN

I've probably run out of both time and space so I'll wish you the best of PC until the next WACCI. Do write and ask questions (you may even get an answer.)

QUIZ ANSWERS

Please remember that the prices have dropped this much in just three years. And please note, these are the correct current prices when I wrote this, and they'll probably have dropped before you read this.

I buy my spares/parts/upgrades from Insight (Choice Peripherals) at Rhodesia, Near Worksop in North Nottinghamshire.

16 mb Simm: £20 save £427, Pentium Motherboard: £58 save £329, Soundcard: free, save £105. Sound cards now come as part of the Pentium Motherboards so you don't need to buy one. 14" Colour Monitor: £97 save £89, 2gb Hard Drive : £97 save £781

Please note you can send me PC letters on a PC disk in Word 6 or Lotus Amipro or WordPro format. Have a good holiday.

Best wishes,
Uncle Clive

Broken Rom?
by Jonty Jones

Eventually everything breaks, and it seems man-made items feature the worst off, but what would YOU do if you just happened to break the leg off a ROM? Or dafter buy a whole NEW ROM and software for ten quid plus? Here's a simple remedy I used with a small amount of solder, and there wasn't even a soldering iron in sight (I don't have one!).

YOU'LL REQUIRE

(1) Eye plucking tweezers. (2) A small piece of solder (3"-4" worth). (3) A small plate or the like. (4) A old metal fork or spoon. (5) A heat source (a gas or electric cooker). (6) The broken ROM

THE HEAT IS ON

The first thing to do is to remove the broken leg from the ROM slot in which it has become so fondly attached, this obviously is where the tweezers would come in handy.

Don't get too close when removing the broken ROM leg just in case the piece flies off at an awkward angle. There ARE a lot more WACCI's to come yet and and we wouldn't want you to miss them would we.
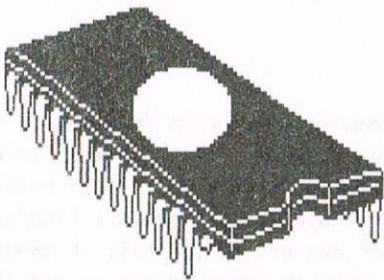
Now ignite your heat source and place your substitute soldering iron on (in?) the heat source to pre-heat. You DON'T want it glowing white, but it MUST be very hot or it will cool off BEFORE you accomplish the task at hand.

SOLDERING ON

Once the fork/spoon has become hot enough to work with, for at least thirty seconds, place the fork/spoon on the up turned plate. Hold the BROKEN ROM leg against the fork/spoon. As soon as possible take up the solder and draw it away from the TOP of the BROKEN ROM leg you MAY have to do this once or twice.

All you require is a small 1mm-2mm spike of solder. Don't worry if all you can achieve is a dull lump, as long as it is attached to the BROKEN ROM leg and NOT the other legs.

DRAW SOLDER DOWN
TO FORM A SPIKE
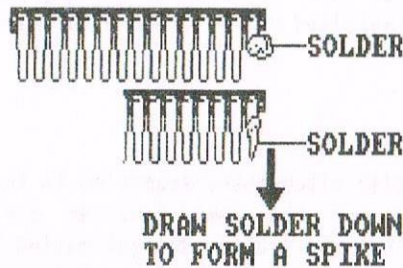
THANKS A-SLOT

Turn off the heat source, place the fork/spoon into some water *(especially if you have children)*, then take the ROM to the ROM slot in which you want it to go. Insert the ROM as per the normal procedure, if you have ENOUGH solder attached the the broken ROM leg when you fit the ROM into the slot the repaired leg should come up against the inner side of the slot, and in doing so allowing the ROM to log on once again? To make sure give the ROM a little extra push a visual check should confirm the ROM is in enough.

PLUGS, PLUGS, PLUGS

Their everywhere and the trouble is so are the cables from the plugs too. I have five plugs hanging off the back of my desk:- Monitor, External Tape Recorder, Printer, Lamp and last but not least a 3.5" power pack. Sometimes I have a Hi-Fi plugged in too. I spend more time un-tangling the things than I do using them. *Yeah right* just by a multi plug extension. I've tried that the trouble is I can only find a four-way version, and this still doesn't prevent the cables getting crossed over.

If you have a similar problem try this. Take a piece of stiff cardboard or thin corrugated card that will allow you to lay each cable side by side with at least half an inch gap and still have enough card to be able to fold over the cables. Tape the cables to the card leaving at least twelve inches of cable at the plug
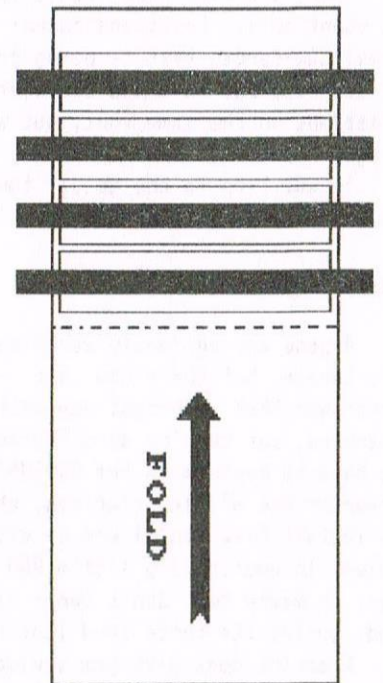
end free, then fold and tape the other half of the card over the cables. Either use coloured electricians tape or just plain tape with coloured card. Voila' no more tangles cables. A simple sticky label on the plugs will let you know who is who...

Jonty

PROPRINT RETRO-SPECTIVE
PETER ROGERSON

It was some time after the introduction of the saintly Protype that someone thought it necessary to introduce a second print enhancer onto an unsuspecting CPC public. It was called Proprint and it produced this wondrous document, which should give you an insight into one or two of the magical things it does.

To Start With...

Firstly let's look at what the disc includes. There's the program, of course, and something called CUSTOMIZ which you use to ... customise the details of what you want your document to look like. You use this to tell the program whether you've got an 8-bit printer port and what it's name is ... or something like that. I haven't got one, but if I had I'm pretty sure I'd call it Mervyn. You also use this program to load your fonts into memory, up to five of them if that's the way you feel. But a warning here: each font requires a bit of memory and with five installed you're left with less space for text than the disc version of Protext allows altogether.

Spoilt For Choice...

There are over forty fonts on the disc altogether, according to the handbook. I suppose this is true if you count bold, italic and normal versions of the same font as *different*. This page is an example. The normal day-to-day text is being printed using a little delight called SANS2-L. The italics are SANS2-LI and the sub-headings in SANS2-LB (where L stands for Large, I for italic and B for bold). To me these are three variations on the same font, but who am I when it comes to such sedentary occupations as the Naming of Things.

In addition to the actual font I'm using the Condensed version of it because I require to have around 100 characters per line and this is the best way of obtaining same.

Comparing Things...

Anyone who seriously wants to review this program will be forced into comparing it with Protype. It's unfortunate, but there you are - unfortunate because there isn't really a great deal of favourable comparison that an honest Joe with a pen in his hands can make. You see, both programs may be print enhancers, but they're as different as chalk is from ... cheese. To start with Proprint's a tad more fiddly. You have to begin with the CUSTOMIZ program in order to configure it to your own set up ... tell it about Mervyn or one of his relatives, choose fonts (up tp 5, but they are greedy when it comes to memory and if you install five you'll end up with less than 23k for text, and that's when you've got the ROM version of ProText in your cuddly little ROM box. A word of advice wouldn't go amiss here. If you're printing a single page, or maybe two, don't spoil it with too many fonts. Anyway, back to the argument. Proprint, on the other hand, buries its fonts (and itself) in the extra 64k and leaves text memory virtually untouched.

Proprint does give you a wider choice of fonts, though, and that can't be bad. In fact I'm going to register this fact as a definite plus.

Talking of Fonts...

Talking of fonts, there's another plus to Proprint's armoury. It has the most beautiful font designer I've ever seen. There's an endless load of features for the creation of the weird and wonderful on a 16 x 16 grid. Individual characters can be emboldened or thinned down in order to create that special effect at the touch of a single key - no pixel-by-pixel chores here. And even more glorious, if that's possible, the entire character set can be treated in such a way by the simple addition of there merest touch on the

CONTROL key. This is really good. I've forgotten how many times I've gone through the alphabet spending a boring afternoon making similar changes to the whole character set. And here CONTROL-B and the whole alphabet and the host of other characters that accompany it are thickened up - made more bold. With this one font designer we have the entire value of the program in our sticky fingers. A great deal of thought has gone into making it just about as user-friendly as any font designer can be.

Added to this cornucopia of praise is a foot-note. It also accepts Stop Press and MicroDesign fonts. Hallelujah!

Getting it in Proportion

I suppose that where Protype beats Proprint hands down is the proportional spaces that accompany proportional print so that a page can be right-justified even when the letters are printed in a proportional font. I'm not treating Proprint as a failure in this respect, though, because it doesn't try to do this. Instead it attempts to create legible and improved pages from a program that is already acknowledged as the best word processor for the CPC, and that can't be bad. At the end of the day it doesn't really matter what further refinements a print enhancing makes as long as it actually enhances the print quality. I know that might sound obvious, but that doesn't make it any the less true. In the same way Protype caters for just about every European language ever spawned and Proprint doesn't. If this is a weakness it's probably because you *need* characters from Hungarian, but I don't. At least, not at the moment.

External Commands

Proprint introduces around 20 new commands, from font instructions (CONTROL-X-i switches on Italics just as in Protext proper, but doesn't switch it off again. Instead, when you're fed up with Italics you have to point the program with a second but different CONTROL-X instruction in the direction of your desired font. Then there are external commands (>ex) for the arrangement of spacing, gaps and so on - all explained in concise detail in the instructions that accompany the program. The overall effect is to give you greater control over the appearance of your work, which is a good thing.

The Printer

Proprint caters for both 9-pin and 24-pin printers and is another program designed to compensate for the poorer qualities inherent in dot-matrix printing. And as with Protype, the printing takes a little longer than it would if you were using just the word processor and no enhancer. Protype takes an age to print a page, Proprint considerably less time. But those of us who still adhere to a printer from a forgotten generation must expect to have to pay some price for better output. And that price, members of the brother and sisterhood, is *time*. And wear and tear on our printers. And too much caffeine while we're waiting.

But printed output *can* be important. If we are in a competitive world (and observation tends to suggest that we just might be) then there must be occasions when the smart piece of paper carries more weight and has a greater bearing on our comfort and potential wealth than might the odd scrap of dotty nonsense. And therefore programs like Proprint might just pay for themselves many, many times over in almost invisible little ways. Take that CV you scribbled out last year, half an hour before the interview for a great job that you never got....

In Conclusion...

So there you have it. If you want a good quality print-out from your old Citizen or Star or whatever, then you'll need a program to enhance the default quality. You can choose between Proprint and Protype and both programs have their advantages. I doubt that Proprint will ever achieve quite the spectacular output of Protype, but what it does it does well, particularly in the field of font designing or modifying. Just look at these pages if you want evidence. Along with all other articles in this series I am using a CPC 6128, Protext on ROM, the relevant software, this time Proprint, Microdesign for the main headline and a Citizen 120D printer, all bundled together with a little bit of time. I print the pages out at home and send them (carefully packaged) to John for inclusion in a WACCI near you. So there you go - judge for yourselves. And remember that for reasons to do with conforming as closely as I can to the WACCI house style, this article has been prepared in condensed print.

And finally, you can still obtain this program commercially, from COMSOFT. Give Peter Campbell a ring on 0141 554 4735. I'm sure he'll be helpful.
Peter Rogerson

Programmers' Patch
by Hilary Phillips

In your more idle and lonely moments you may have wanted to hold a conversation with your computer. Much of the time, however, it will simply respond to your remarks with "Syntax error". For a more stimulating exchange you could try one of the psychiatric interview programs available in the PD library, but if you're anything like me you get sick of the invitation to "Tell me about your computer."

The program below is a simple way of teaching your computer to communicate. It has no applications that I am aware of, but it is quite entertaining to try out and you might learn more about using arrays if you can follow the explanation.

The basic listing is in Fig 1, and it will gradually be improved and enhanced over the next article or two. An explanatory list of variables is in Fig 2.

HISTORY LESSON

"AI" (or "Artificial Intelligence") was originally written by a friend of Matthew's for the BBC micro, adapted for use over a network and then rewritten for the Amstrad by Matthew. Once when I wanted to use it and we were unable to find the program it was rewritten once again by Matthew dictating it to me.

THE BASIC IDEA

To use the program, you type in a sentence. The computer looks at the sentence and stores the information about which words can follow which, and responds with a sentence based on this knowledge. (Until you have taught it a few sentences it is unlikely to come out with anything original.)

For example: Hilary: I am bored with this article Computer: I am bored with this article. (It only knows that "I" can begin a sentence and that "I" can be followed by "am", "am" by "bored", "bored" by "with", "with" by "this", "this" by "article", and "article" by the end of the sentence) Hilary: You can't be

bored Computer: I am bored. (It knows that "I" or "You" can start a sentence and chooses "I", which has to be followed by "am" and then "bored", but as "bored" can either finish a sentence or be followed by "with", it randomly opts for the former option.) And so the conversation continues...

PROGRAM ANALYSIS

How does it work, then? The arrays w$ and lk are used to store the information about words and word order. A word is stored in w$(a) and the "link" information about which word can follow it in lk(a,b).

The first number in the lk array, lk(a,0) gives the number of different words which can follow word a; the following numbers point to the words in w$ which can follow word a. Word 0 is blank and represents the start or end of a sentence. The table shows the values in the arrays after learning the two sentences in our example:

a	w$(a)	lk(a,0)	lk(a,1)	lk(a,2)
0		2	1	7
1	I	1	2	
2	am	1	3	
3	bored	2	4	0
4	with	1	5	
5	this	1	6	
6	article	1	0	
7	You	1	8	
8	can't	1	9	
9	be	1	3	

All the words apart from 0 and 3 can only be followed by one other. Word

0, the "start of sentence" word, can be followed by two words, 1 and 7 ("I" and "You"). Word 3, "bored" can be followed by 4 or 0 ("with" or the "end of sentence" word.)

IN MORE DETAIL

Now that you have an idea of how the data is stored in the arrays, let's look at the program in detail. The first three lines set things up.

10 DEFINT a-z speeds up the computer's "thought" processes. The rest of the line reserves a disc buffer, as later we may want to develop the program further and be able to load and save vocabulary.

20 sets limits on the number of words the computer can store and the number of links in the link array, and sets the number of words known to zero.

30 The arrays are dimensioned accordingly and another array is set up to hold the sentence typed in by the user for analysis.

40 We wait for the user's input of a$.

50 We GOSUB 300 which strips spaces from the front of a$. If anything then remains in a$, we GOSUB 100, which analyses the sentence and stores the information in the arrays.

60 Whatever the user typed, we then GOSUB 200, which prints up the computer's sentence, and GOTO 40 to await the user's input again.

SENTENCE ANALYSIS

The subroutine in lines 100-190 takes the user's sentence, held in a$, and pulls it apart.

100 WHILE there are any spaces in a$, i.e. more than one word, a is set to be the position of the first space.

110 The first word is put into the array tempw$ and taken out of a$.

120 The word count is increased by 1 and spaces are stripped from the front of a$ again. WEND takes us on to the next word, until we have no spaces left in a$.

130 If there is anything left in a$ then it is also put into the array tempw$.

Now we go through the array tempw$ looking to see if any of the words are already known (i.e. in w$). The variables cword and pword are used to point to the positions in the array w$ of the current word and previous word in the sentence respectively.

140 We go through the words in tempw$ one by one starting with tempw$(0), the first word in the sentence, and finishing with tempw$(wcount-1), the last word in the sentence. word$ is set to contain the current word under consideration.

150 We go through the array w$ backwards to make the loop simpler. Starting with the last word in the array, we try setting cword to that position in the array and see if the word word$ matches the word w$(cword) stored in the array. If not, we look at the previous word in w$. If we find a match, cword will be pointing to our word in w$.

160 If we have got back to the first word in w$ without a match, we add word$ onto the end of the array, increasing nwords accordingly, and set cword to point to our new word in w$. **170** The next thing we do is to GOSUB

400 to put in the link from the previous word in the sentence to the current word. If we are on the first word of the sentence, pword has been set to 0 to point to the "start of sentence" word in w$. After putting in the link we move onto the next word in tempw$.

180 Once we have reached the end of the words in tempw$ we have to put in a link from the last word to the "end of sentence" word in w$, so cword is set to 0 (pword is already pointing to the last word from tempw$) and we GOSUB 400 again to put in the link.

LINKING BITS

The link from word pword to word cword is stored in the array lk. The number of links from word pword is stored in lk(pword,0).

In lines 400-440 we check through the links from pword to see if cword is already listed there, and if it is not we add it on the end.

400 We set our counter j to the number of links from pword, lk(pword,0). The variable x will be used for the second coordinate in the lk array.

410 j counts down to zero to go through all the links. Starting with x=1 at the first link, we check whether cword is that link, and if it is not we move on to the next link by increasing x.

420 WEND is reached either when j=0 or when a match is found, in which case j>0 and we go to 440 which simply RETURNs us to the 100-190 subroutine as our analysis is finished for that word.

430 No match has been found, so we have to add a link for cword onto the end of the list of links.

We increase the number of links, lk(pword,0), by 1 and set lk(pword,x), which is the unallocated link which we reached when going through the WHILE-WEND loop, to be equal to cword.

TALKING BACK

We have dealt with storing the information. How do we use it? The subroutines in lines 200-230 and 500-510 produce the computer's sentence. 200 To start the sentence, we set pword to 0 which points to the "start of sentence" word in w$. GOSUBbing 500 fetches a randomly selected word which can start a sentence, and sets pword to point to the new word.

210 WHILE pword>0 we have not got to the "end of sentence" word. We print up w$(pword) followed by a space, and look for the next word by GOSUBbing 500. WEND is reached when pword=0, i.e. we have got to the end of the sentence.

220 We want to put a full stop on the end of the sentence, so we move the cursor back one and print a full stop over the last space. How do we decide on a word?

500 We look at the number of links from pword, lk(pword,0) and set i to be a random number from 1 to lk(pword,0) using the expression $i=INT(RND*lk(pword,0)+1)$.

510 We set pword to lk(pword,i), which points to a random word linked to by the previous word, and RETURN to print up the word w$(pword).

THAT'S ALL FOLKS

The only line which I haven't looked at for you in detail is 300, which does the space-stripping. I think you should be able to see how it works.

ANY QUESTIONS? Well, here are two simple ones answered:

WHAT ABOUT PUNCTUATION?

If you put a punctuation mark on the end of a word the computer won't realise that the punctuation is not a part of the word as there is no space between the word and the punctuation mark. For example, it will treat "however" and "however," as two different words. So if you end your sentences with a full stop, the

computer will end all of its sentences with two full stops. If you want to use punctuation then it may improve the computer's grammar, but you will need to increase the word limit and take out the line in the program which prints up the full stop.

AND CAPITAL LETTERS?

The program treats a capitalised word differently from a non-capitalised one, e.g. it will think that "You" and "you" are different words. This was the only way we could see to make sure it capitalised things correctly.

STILL TO COME

Next month we will look at some more ways of improving the program. The version here is very basic but should keep you busy for a month as long as you don't have too many bored or lonely moments. Remember that this program and all the others from the series are available on the Programmers' Patch disc, PD 98, from Dave Stitson.

AN ENHANCEMENT PREPARED EARLIER

I mentioned earlier that we might want to be able to load and save vocabulary. One way to do this while the program is running is to look out for commands typed by the user which wouldn't be used in normal speech. If you add in the extra lines 41, 42 and 600-740 below this will make the computer respond differently to sentences beginning with *LOAD and *SAVE.

Lines 600-640 deal with loading in vocabulary and 700-740 with saving it. Let's look at the saving routine. The loading routine works in exactly the same way, except that it uses INPUT#9 to input from disc instead of WRITE#9 to write to disc.

700 We look to see if the user has typed a filename after *SAVE, e.g. if they have typed *SAVE OurWords - in which case file$="OurWords". If there is nothing following the *SAVE command we use the default filename VOCAB.

710 We OPENOUT the filenamed file$ for writing, and the first thing

Figure 1

```
10 DEFINT a-z:OPENOUT"d":MEMORY HIMEM-1:CLOSEOUT
20 mwords=300:mlinks=20:nwords=0
30 DIM w$(mwords),lk(mwords,mlinks),tempw$(128)
40 LINE INPUT"",a$
50 wcount=0:GOSUB 300:IF a$<>"" THEN GOSUB 100
60 GOSUB 200:GOTO 40
100 WHILE INSTR(a$," ")>0:a=INSTR(a$," ")
110 tempw$(wcount)=LEFT$(a$,a-1):a$=MID$(a$,a+1)
120 wcount=wcount+1:GOSUB 300:WEND
130 IF a$<>"" THEN tempw$(wcount)=a$:wcount=wcount+1
140 pword=0:FOR i=0 TO wcount-1:word$=tempw$(i)
150 cword=nwords:WHILE cword>0 AND word$<>w$(cword):
cword=cword-1:WEND
160 IF cword=0 THEN nwords=nwords+1:w$(nwords)=word$:
cword=nwords
170 GOSUB 400:pword=cword:NEXT
180 cword=0:GOSUB 400
190 RETURN
200 pword=0:GOSUB 500
210 WHILE pword>0:PRINT w$(pword)" ";:GOSUB 500:WEND
220 PRINT CHR$(8);".":PRINT
230 RETURN
300 WHILE LEFT$(a$,1)=" ":a$=MID$(a$,2):WEND:RETURN
400 j=lk(pword,0):x=1
410 WHILE j>0 AND cword<>lk(pword,x):j=j-1:x=x+1
420 WEND:IF j>0 THEN 440
430 lk(pword,0)=lk(pword,0)+1:lk(pword,x)=cword
440 RETURN
500 i=INT(RND*lk(pword,0)+1)
510 pword=lk(pword,i):RETURN
```

Figure 2 TABLE OF VARIABLES

mwords maximum number of words held in w$
mlinks maximum number of links in the lk array
nwords current total number of words held in w$
wcount counter for words in tempw$
word$ temporary variable to hold the current word under analysis
pword   points to the position in w$ of previous word in the sentence
cword   points to the position in w$ of current word in the sentence
i, j   counter variables.
x   variable used in going through the links array

```
41 IF UPPER$(LEFT$(a$,5))="*LOAD" THEN GOSUB 600:GOTO 40
42 IF UPPER$(LEFT$(a$,5))="*SAVE" THEN GOSUB 700:GOTO 40
600 file$=MID$(a$,7):IF file$="" THEN file$="VOCAB"
610 OPENIN file$:INPUT#9,nwords
620 FOR i=0 TO nwords:INPUT#9,w$(i),lk(i,0)
630 FOR j=1 TO lk(i,0):INPUT#9,lk(i,j):NEXT:NEXT
640 CLOSEIN:RETURN
700 file$=MID$(a$,7):IF file$="" THEN file$="VOCAB"
710 OPENOUT file$:WRITE#9,nwords
720 FOR i=0 TO nwords:WRITE#9,w$(i),lk(i,0)
730 FOR j=1 TO lk(i,0):WRITE#9,lk(i,j):NEXT:NEXT
740 CLOSEOUT:RETURN
```

we WRITE is nwords, the number of words in the vocabulary. **720** Now for each word i, we WRITE w$(i) (the word itself) and lk(i,0) (the number of links from word i).**730** Following this we WRITE each lk(i,j) (all the links for the word i) in order, and then go back to 720 to do the NEXT word.

740 Now we have done it all, so we CLOSEOUT to finish with the file. Note that loading in vocabulary will overwrite any existing vocabulary in the arrays.

Widgets And Sealing Wax

A long time ago during Philips time as editor people were complaining about needing hardware supplies. As soon as I could find a supply of parts I had a batch of widgets made up and I still have them....

DIY Widgets

Here's how to do it for your self. The widget is an essential part of owning a PLUS model of our AMSTRADS it allows the add-ons to be used with these models as the expansion port is a much better type than the grey AMSTRADS edge type although the connection layout thankfully is the same so I know of no immediate problems.

First thing you require is a MAPLINS catalogue from the news agent. From the catalogue you must find the parts you require they are in there somewhere.

Bits you need

1) a length of 50 way flat cable.
XR79L 72p

2) a 50 way IDC Centronics plug.
JB56L £3.79p

3) a 50 way IDC card edge plug.
FT60Q £3.79p

4) a edge card reversing board.
Try veroboard

Assembly

Now to assemble these I use a small vice but there are special gadgets available. Take the centronics plug and slip a suitable length of say five inches through the plug along the short spikes take note of the red line this should go on the right as you look at the plug from the back.

Making The Connections

The connector should face away from you. Make sure the cable is at 90 degrees to the plug and when the lot is pressed together the spikes pop through the cable making the electrical contact. If we lay the cable on the desk with the face up the edge connector also can be slipped onto the cable face up. This gives a loop when connected to the computer thus avoiding any strain.

The assembly is now pressed in the vice to lock it in place. So why do we need the reversing board I here you say, well the odd thing is that what we have is reversed as if in a mirror. The grey Amstrad has the edge of the circuit board to attach extras on at this point we have a plug so to mimic the circuit board edge we need the reversing board.

After assembling I suggest trimming off the ribbon cable with a Stanley knife to give a neat edge to the job. At this point it is best to let you in on a top tip, take a magnifying glass an inspect the cut edge for any lose hairy wires that could cause a problem.

So that is it plug on to your PLUS at the back and slip on the expansion of you choice. The loop of cable helps to relieve strain on the connectors as you type on the keyboard this often lead to problems on the original AMSTRADS.

I have my 464+ with dd1, A and B drives, DK memory pack and rombo robot working well and some time ago the Bowley clan were having trouble and since I modified the connections I have no reports from them of problems.

Ready Made

Of course you can save yourself the bother by contacting me and buy a ready made WIDGET for £4.50 thus saving yourself all the above bother and even the cost is less than the purchase price of the parts.

Open a world of add-ons to your AMSTRAD PLUS. The only snag I have heard of was some Romboards don't like PLUS type as the chips have HS instead of LS this is low speed and high speed chips nothing to do with the widgets though and doesn't do any harm.

Stephen Potts

85 Thorold Ave.
Cranwell village,
Lincolnshire.
NG34 8DS

CPCing The Future

No, this is not another tired old article on the future of 8-bit computing, although that forms part of the subject in its wider sense.

What I am addressing here is what will happen to the equipment itself in years to come.

Computer disposal

Why this is becoming an issue in itself now is that new European *(wouldn't you know it!)* legislation is being formulated now with a view to enforcing responsible disposal of old computer equipment on its users, whether they be large corporations or individuals. Think of it as an extension of the philosophy of the *"bottle bank"* concept. By the way, have you ever tried to make a withdrawal from one on a Sunday?

Going to the scrapheap

Generally, I'm a very stroppy individual when it comes to being told what to do, especially by government bodies, but I can see that this particular legislation does have a point. We can probably all think of examples, either from personal experience or hearsay, where large quantities of high-power computing equipment has quite literally gone on the scrapheap because some nitwit in the company has justified his salary by forcing through an upgrade policy.

Having worked for many years in a senior position for a large company *(Athena, the poster people)* who TWICE IN SIX MONTHS were sold an inadequate system and then went through the trauma of completely scrapping the desktop units and all the associated hardware except the printers, I may be expected to have a particularly jaundiced view. Now,

with the benefit of hindsight, I can see that the problem each time was one of putting too much stress onto the system in terms of multi-tasking. This resulted in a crippling bottleneck as the system was backing up its data towards the end of each day. The terminals on every desk got slower and slower, and the staff got more and more frustrated.

Multi-Tasking

The smart move would have been to split the system into several smaller networks, but no, everyone apparently had to be able to exchange information instantly with everyone else, so *"stock maintenance"* and *"order processing"* suffered while *"invoicing"* and *"accounts"* watched their numbers scrolling up their screens. I found it ironic that it was the accountant who sanctioned the upgrade expenditure each time and somehow he managed to keep his job until well afterwards.

TWICE, the skips out the back were filled up with keyboards, monitors, and main units as the NEW consultancy company declared the PREVIOUS consultancy company's chosen kit to be woefully inadequate for the required performance and results. A wicked waste of money and resources, and *"the race for the new"* ensures that it still goes on in other companies.

But not in Athena it doesn't; it was taken over and later went spectacularly bust. The shops you still see on some high streets now with
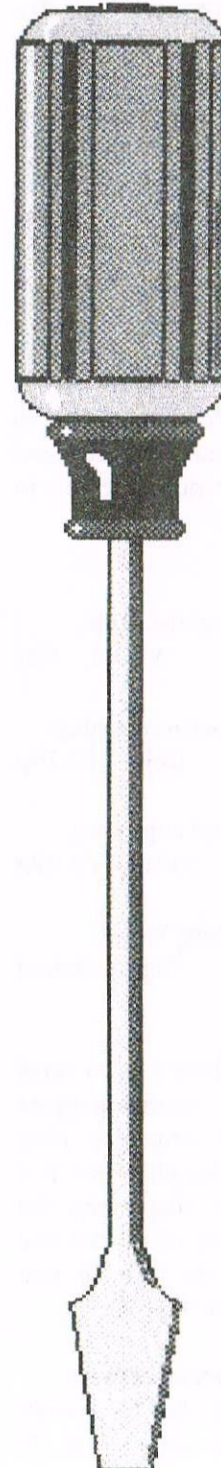
Athena over the door are franchises who paid to use the name after the crash. So, to impose some l e g a l l y e n f o r c e a b l e requirements on users to get rid of old computer equipment in a proper way just MAY give the nitwits pause.

As well as, we might hope, modifying the *"bin it and don't give a damn"* mentality, It will certainly cost them money, and that's going to be a major implication if the old system is a big one. It will be *"part of the deal"*, something to be costed on the balance sheet along with the purchase price of the new set up.

As the new l e g i s l a t i o n comes in, I anticipate a growth in

attractively worded part exchange arrangements to absolve the new computer purchaser of the need to get their hands dirty.

If you doubt it, look at the company car market; most of that is now administered by ongoing contracts, with a new car supplied as the old one reaches 60,000 miles and goes back to the supplier by disposal through the second-hand market.

If a company, or you and I, are offered an attractive enough deal by the shop to dispose of our embarrassingly large pieces of redundant technology, I reckon we'll take it. But what sort of trade in are we likely to be offered for a CPC, I wonder?

Disposal costs

This brings us back into the home. Without a desperate wish to buy a wholly new computer, a home user is likely to think twice about upgrading if there is a £10-20 fee attached for the local council to dispose of the OLD computer. Whether it be a CPC or whatever, they are rather large and awkward items to smuggle into a dustbin or tip.

And the actual disposal or recycling once the computer is in the hands of the authorities IS likely to be expensive without the option of simple landfilling.

On top of the labour charges there will be reclamation charges for the circuit boards and other modular parts which, at component level, include gold but also gallium arsenide and many other highly toxic and hard to remove chemicals.

So does this new culture do us CPC users any favours? Well, in a strange way, I think it does.

There is little point in getting rid of a computer if the DISPOSAL price is more than its REPLACEMENT COST price. It might as well be put to good use instead, if that is possible.

Doing the job properly

I have a particular contempt for multi-purpose power tools which purport to do a lot of jobs properly. In my experience they do them all barely adequately. If I need a circular saw, I should USE a circular saw, not a loosely attached deathtrap on the end of my drill, forcing the bearings to absorb loading from directions way outside their original specifications. Single purpose use is not something we generally associate with CPCs, but a slight change of attitude can be very worthwhile.

To quote one example, in our spare room I keep my old CPC with a rombox hanging off the back. All it contains is Protext, but it might as well be Brunword.

To the contrary

Despite all advice to the contrary, there is a 3" disc permanently in the disc drive so that if the phone rings and the TV is on too loudly in the living room, I can duck into the spare room, boot Protext with a flick of the "on" switch and |p, and be ready to save to disc a note arising out of the conversation in a matter of seconds. I find this is more efficient than using a notepad *(which can stray)* and a pen *(ditto)* and has the added benefit that I can later enlarge the note into a letter to act upon it.

So rather than be a potentially expensive liability, my old CPC is going to be a family asset for many years yet. The other members of the family use it too and for them it IS the notepad. By the way, if it's a VERY long call and you think you start to hear canon fire in the background, it may be *"Pirates!"*

Database

I have another computer *(a PcW16)* which is set up in the office and is principally used as my database. As you might expect, this means that I keep significant names, addresses, phone numbers, birthdays and other important dates on it, but it is ALSO the repository of my various price lists and product information packs too, ready for me to refer to if a potential customer needs detailed product features explained.

I keep other stuff there too, more LOOSELY described as *"information"*, which it is handy to have in a readily accessible form.

The '16 is a bit, shall we say, *"peculiar"* in some respects, but it's nearly as quick to boot from *"standby"* as the CPC is from *"off"*, and it has a Cabinet *(silicon/hard disc is the nearest CPC equivalent)* in which the principal documents I need to refer to can be stored for quick access and retrieval during a phone call. A CPC could be used in a very similar way.

For other purposes, using a CPC as an accounts *"station"* involving spreadsheets perhaps, programs can often be persuaded to self-boot not just into the program, but directly into a particular file. The few applications I suggest here are intended to be triggers to YOUR imagination for your requirements.

So this is my point. With the obligation to pay for the disposal of your old computer kit coming in at some stage soon, it makes sense to think in terms of USING it instead if you can, even if only for just one particular purpose in a location which may be away from your usual computing area.

If your main computer is of the same type as the secondary one you are ensuring immediate access to a spare, should you need one. And if it's a different type you can widen your usage options, although it makes sense to keep a utility program handy for transferring data between them.

Brian Watson

FIND by Stuart Paterson

Here is a Noddy program with at least four uses:- as a typing exercise, to explore BASIC, to convert to machine code, and to practice BASIC modifications on.

Lets take them in turn

1. Typing exercise. You will see that it is small, but when you have typed it correctly, it can be very useful.

```
65000 CLS:PRINT "Line No.";
65010 linad=368:targad=linad+4
65020 targlen=256*PEEK(linad+1)+PEEK(linad)-6
65030 targchr=PEEK(targad)
65050 linlen=256*PEEK(linad+1)+PEEK(linad):if linlen=0 then END
65060 FOR k=linad+4 TO linad+linlen
65070  p=PEEK(k):IF p=targchr THEN GOSUB 65100
65080  NEXT k
65090 linad=linad+linlen:GOTO 65050
65100 FOR m=1 TO targlen
65110    y=PEEK(targad+m)
65120    IF y=0 OR y=PEEK(k+m)  THEN NEXT m:PRINT 256*PEEK(linad+3)+PEEK(linad+2);
65130 RETURN
```

Having typed and SAVEd the Noddy, you can MERGE it with any program you are developing.

If you want to know where in the program you have used the variable <a$>, for instance, or a textual phrase, <"Hello">, for example, then put it in a dummy line 1, right at the start of your program. Then RUN 65000. The Noddy will give you a list of all lines in which your target appears. If it is in a line two or three times then the line number will be reported two or three times.

2. How is BASIC stored? In understanding how the Noddy works, you will be led to understand how BASIC is stored in memory.

The first thing to note is that this Noddy is written with line numbers starting at 65000. This is about as far as you can go, so it will always be MERGEd after any program you have. Taking each line in turn, 65000 just clears the screen and prints the name of what is to follow. 65010 sets the memory address of the first line as 368, because that is what it is, in a CPC. Locomotive BASIC stores commands and variables as tokens, so the only easy way to set up a potential Target is to incorporate it into the program and get ARNOR to convert it into the appropriate tokens for you. Because the first line contains the Target, the memory address of the first character of the Target is at 368+4. What, you may say, is at the addresses in between?

Well, part of the answer is on line 65020, i.e. the length of the line, or, more correctly, the number of addresses, or bytes, used up by the line. We have seen that some addresses are used up to store items other than the BASIC code, so the actual number of bytes containing the BASIC in the line is 6 less than the figure quoted. In the case of the first line the length we are interested in is the length of the Target.

Line 65030 PEEKs at the first character in the first line, that is the first character of the Target we are looking for. Line 65050 is the first line of the main loop of the Noddy. It does exactly the same as line 65020, but calls it the length of the line rather than the length of the Target. So, of course, the six bytes are not ignored this time. Line 65060 sets up a FOR/NEXT loop the length of the line (less the first four bytes). Then line 65070 PEEKs at each byte in turn. If it matches the first byte of the Target, then we transfer control to the subroutine at line 65100. If no match is found, then 65080 moves to the next byte. If there is no match in the line, then 65090 adds the length of the old line to the start address of the old line to find the start address of the next line, and transfers control back to 65050, the start of the main loop.

Line 65100, the start of the subroutine, sets up another FOR/NEXT loop, this time the length of the Target. The next line PEEKs at each byte of the Target in turn, and line 65120 looks for a match. If there is a match, then it loops back to check the next byte, until the end of the Target, which means that there is a perfect match, in which case the line number has to be reported. This is where the second and third bytes of the BASIC line come in. They hold the line number.

This particular FOR/NEXT loop is unconventional, not to say dangerous in its present layout, because control drops out if a match is not found, so that control can RETURN to look for the first byte of the Target further along the line of program that is being examined. It is unconventional in that it leaves the

FOR/NEXT loop incomplete, and doing that too often can cause the stack to overflow . So here is a nice little exercise in logic:-how do you rewrite this line to complete the loop every time?

3. A machine code exercise. All of you good readers who have been following Richard Fairhurst's course on machine code will be dying for a short routine to practise on. Well, here it is! I admit it runs a bit slow in BASIC, but code it up and you will have a super, fast way of exploring your programs.

4. Modifications. This Noddy is very "basic"! Just the bare minimum to get you started. The best way to learn is to do, and what better "do" to do than improving a utility that will help you in the future? No doubt you can think of a few improvements, but here are my suggestions.

(a)   How about having the option of reporting the line numbers to your printer?

(b)   Because it runs slow, you might like to indicate roughly which line it is searching.

(c)   You can explore Locomotive BASIC by looking at the actual bytes that go to make up the Target.

(d)   If you are going to use it repeatedly, you might like to arrange for Key f9 to run the Noddy for you using the KEYDEF facility.

(e)   Try speeding up the search by ending the search on each line sooner. (Hint: if there are not enough bytes left in the line being examined to contain the Target, then its not worth looking any further, is it?)

There you are, then. A tiny Noddy that can be accommodated on three lines, if you push very hard, but for such a small thing, it does a lot, and can lead you down lots of exciting avenues, each of which improves not only your basic BASIC skills, but improves the Noddy itself! Good typing - which is where we came in!

Stuart

Services

Homegrown, PD Extra, PD Discs 1-onwards:

Dave Stitson, 4 Connaught Avenue, Mutley, Plymouth, PL4 7BX Tele: 01752 254404

Homegrown

H1	Aj's Concept Disc
H2	Aj's Twilight Zone
H3-5	Utilities 1-3
H6-9	Games 1-4
H10-14	Utilities 4-8
H15	Pagesetter 64
H16	Family Tree
H17	Family History

PD Extra

X1	Maxidos, Procopy
X2	Super Meddler, Hack Pack
X3	Bonzo Doo Dah
X4	MS800, Bonzo Blitz
X5	Big Batch
X6	Flash Pack

Public Domain

1	Forth Programming
2	CP/M utilities
3	MBasic
4	PowerPage, RAMbase, PD Planner
5	CP/M Games
6	Games
7	CP/M+ utilities
8	Comms
9	CP/M utilities
10	VDE, I-Spell
11	Adventures
12	Cobol
13	Demos 1
14	Amiga Graphics 1
15	Flags, Animator Demo
16	PPC Pascal, ZBasic
17-8	JRT Pascal
19-21	MicroDesign Art 1-3
22-4	Pictures 1-3
25-6	MicroDesign Art 4-5
27	Mini Print Shop
28-9	MicroDesign Art 6-7
30	GPaint, RSXlib
31	Amiga Graphics 2
32-4	MicroDesign Art 8-10
35	Picture 4
36	Teach Yourself Pascal
37	Adventures 2
38	Games 2
39	Logos, Women & pics
40	MicroDesign Art 11
41	Eve of Shadows & advs
42	22disk/22nice for PC
43	Pictures 5
44	Demos 2
45	Paradise Demo (Paradox)
46	The Demo (Logon)
47	Utilities 1
48	Adventures 4
49-53	Pictures 6-10
54	Parrotry demo & clip art
55	Utilities 2
56	Star Trek pics, music
57	ZDE & CP/M utilities
58	Overscan Pictures
59	Demos 5, Face Hugger
60	Digital Dreams
61-62	Word Games 1-2
63	Adventures 5
64	Demos 6
65	Games 3
66	Overscan demo (18+)
67	Demos 7
68	Utilities 3
69	CCC 4 fanzine
70	Disc Full 6 fanzine
71	Disc Full 7 fanzine
72-3	Demos 8-9
74	Epic, utilities 4
75	Utilities 5
76	Megaparty Demo
77	Computext, pictures
78	Games 4
79	Games 5
80	Small C & Interpreter
81	Printmaster
82	CP/M utilities
83-84	Printmaster Graphics
85	MicroDesign fonts
86	DOScopy, DLan
87	Scrivener
88	Rundschlag PD disc
89	Pictures 11
90	Voyage 1993 demo
91	Protracker, Discomagic
92	Games 6
93	Utilities 8
94	Demos 12
95	Print Enhancer
96	CP/M PrintIt
97	Des demo
98	Programmers Patch Disc 1
99	Crossword Compiler
100	WACCI index
101	Promerge tutor, RAMbase 4
102	Utilities 9
103	Clip art 1
104-8	AMS music discs
109	AMS-MIDI, MusicPack
110	Word games
111	Pictures 12
112	Demo-makers, Basic tutor
113	CPC Emulator for PC
114	B-ASIC for CPC plus
115	Animal Quiz, Francis
116	Smart+, Fluff for CPC Plus
117	Bankbase
118	Games 7
119	Cribbage, Blackjack
120	Adventures 6
121	Soundhakker v2
122	Soundhakker modules
123	Otto 1-2 fanzine
124	Mr Men slideshow
125	CPC Guide
126	Street Warriors
127	PowerPage support
128	BTL 1 fanzine
129	BTL 2 fanzine
130	Ghoul's Writer

Other WACCI services:

Tapes: Arthur Cook, Brympton Cottage, Brunswick Road, Worthing 01903 206739
Books: John Bowley, 97 Tirrington, Bretton, Peterbrough, PE3 9XT 01733 702158
Past WACCI: Frank Neatherway, 3 Glebe Close, Rayleigh 01268 784742
Market Stall: Angie Hardwick, 23 Station Place, Bloxwich, WS3 2PH 01922 449730
Discs: Frank Frost, 4 Rectory Close, Wootton, Ryde, IoW, PO33 4QB 01983 882197
 £1.50 for 3", 35p for 3.5": minimum order two. Alternative Firmware Guide, £3.99p

---

Bob Moffatt *01454 329875*
Weekends only please. Masterfile III, Mastercalc, Qualitas and Protext.

Frank Frost *01983 882197*
4pm-10pm, VDE, CP/M Protext and WACCI Services.

Angie Hardwick *01922 449730*
Market Stall, Game Page and Convention tickets.

Jonty Jones *01670 852835*
5pm-8pm, SATURDAYS MicroDesign, OCP Art Studio and PowerPage.

Malcolm Brabbin *01502 716344*
8pm-11pm, but not Saturdays. Basic Programming.

Mark Stewart *01773 822599*
8pm-10pm, PAW Adventure Creator, Basic and General help.

Stuart Gascoigne *0117 9831361*
Basic, Machine Code. New PD is to be sent to Stuart please.

John Hudson *01484 423252*
MOII, CP/M, Wordstar, Newword, dBase, CP/M PD, CPC-PCW-PC.

John Bowley *01733 702158*
Subscriptions, Distribution and WACCI in general.

Dave Stitson *01752 254404*
Central Librarian. WACCI PD library enquires.

Richard Avery ....
Tasword 464 GPaint, Another World, 35, Marlborough View, Cove, Farnborough, GU14 9YA.

Nic Rayner .... *Mobile: 0850 992747*
CPC Emulators for the PC.

Arthur/Angela Cook *01903 206739*
8pm till 10pm, Tape Library, Repairs, General CPC, Merline-Serve/KAD and Angela's new magazine for the PC called Wibble.

Frank Neatherway *01268 784742*
Past WACCI's and Fair Comment letters.

Christine Raisin *0115 9133181*
Past WACCI's on disc, phone for what's available.

Do you want to be a Help-Liner? If YES then give me a call and this space could be yours.

PLEASE NOTE:
This help-line is provided on a voluntary basis by club members to assist other members. Please call between 7pm and 9pm, unless stated otherwise.

AMSTRAD

THE UK'S ONLY MONTHLY MAGAZINE
AND SUPPORTERS CLUB FOR THE AMSTRAD CPC

Help and advice on the following computers: 464, 464+, 664, 6128 and the 6128+

The club offers its members not only the best CPC magazine but also Public and Homegrown disk Library's, a Book Library and telephone help-line to help you get the most from your Amstrad CPC. WACCIs vast PD Disc Library is also open to non-members. For a free sample copy of the club magazine, send a 39p stamp to:

WACCI, 4 Lindsay Drive, Chorley, Lancashire. PR7 2QL

Chairman: **Doug Webb** Tel: **01257 411421**   Editor: **John Bowley** Tel: **01733 702158**

Dear Members,

When placing adverts into your local newspapers would you please also state somewhere in the advert that we are a *"Amstrad CPC club"*. I've had calls from some people who have seen these adverts and then enquiring about other Amstrad computers. *Thanks*

WACCI SERVICES

Frank Frost	Ex-software house discs ready
4 Rectory Close	formatted and labelled. £1.50 ea.
Wootton, Ryde	for 3" and 35p for 3.5". The
Isle of Wight	minimum order is TWO discs.
PO33 4QB	*The Alternative Firmware Guide*
Tel: 01983 882197	*and disc now only £3.99*

PAST WACCI

Frank Neatherway	01-03,20	50p
3 Glebe Close	40-42,44,46-49	50p
Rayleigh	52-62,65-69	50p
Essex	70,72-73,75-79	£1.00
SS6 9HL	80,83,88-98	£1.50
Tel: 01268 784742	99-124	£2.00

SUBSCRIPTION FORM (125)

UK	*issues @ £2.00 each (12 for £24)*
EUROPE	*issues @ £2.50 each (12 for £27.50)*
OVERSEAS	*issues @ £2.50 each (12 for £27.50)*

Name

Address

Postcode

From issue no   to

Interests:

What CPC do you have?

Telephone no.

E-mail:

Please send your cheque or Postal Order to: WACCI: 97 Tirrington, Bretton, Peterborough PE3 9XT

Subscription includes membership.

Seeya next month. **John**