SEGA® Computer

Christmas Issue 1987

The official magazine of the SEGA User Club of New Zealand

- *Reviews of the new 'my card' games*
- *Pattern Paint routine*
- *Machine code programming course*
- *Fast Disk Copy routine*
- *File Copy routine*

Published by MJH Software in association with Poseidon Software

Published bi-monthly by **MJH Software**
Auckland New Zealand
Telephone (09) 534-3379
All correspondence to **Poseidon Software**
P.O. Box 277
Tokoroa New Zealand

- All contributions are welcome, but please include your name, address and telephone number.
- A question and answer page in the form of **Letters To The Editor** is provided and we will do our best to answer any questions about software or programming.
- It is preferable that programs be submitted on tape or disk in a listable form. (No copyright protection please). A listing is useful but don't worry if you aren't lucky enough to own a printer. Where required please include instructions on how to type in the program.
- Please check your programs thoroughly for errors and spelling mistakes before sending it to us. Please send updates if any errors are discovered, so we can publish corrections.
- All software programs received by the magazine becomes the property of **MJH Software** unless by prior arrangement. They are accepted on the basis that they are the original work of the author.
- All contributions are subject to approval by the editor and may be edited to suit the magazine style. Submitted programs will be returned on request
- Each issue two prizes of NZ$40 and NZ$20 for the two feature programs are awarded in the following categories:
  Category 1 - Games. Judged on playability and use of graphics and sound.
  Category 2 - (i) Utilities. Judged on usefulness.
  (ii) Reviews. Judged on overall presentation.

SEGA USER CLUB

MEMBERSHIP YEAR

Oct 1987 - Sept 1988

You automatically become a member of the club when you subscribe to the magazine and this qualifies you for special club benefits.

New Zealand Subscription: NZ$25 incl GST
Australia Subscription: A$26 Airmail

See inside cover at the back

Poseidon Software

Geoff Crawford
Ph (0814) 67-105
P.O. Box 277
Tokoroa
New Zealand

SEGA®

HAVE YOU JOINED THE HIRE CLUB ???

Due to the great response to the hire club we have been unable to add new titles as we were very busy keeping up with the original six titles issued. BUT now we can !

AS at January the 25th the following titles will be added to the original six.

DRAGON WANG --- OK all you Karate buffs kick your way through this one. On your way theres plenty of action with guys that throw knives and Masters with nunchuckas and staffs and also warpman with a neat trick up his sleeve.

SEGA GALAGA --- Ever played Galaxians at the video parlour theres no need to now an extremely good copy of this game is now available. A real good shoot em up .

CHOPLIFTER --- Another good arcade game now available for the Sega. Fly your helicopter to rescue the soldiers Trapped in the battlefield. Be carefull of the enemy tanks and planes. If your good at it you can then rescue more at sea.

Now nine titles are available they are;

DROL LODERUNNER ELEVATOR ACTION CHAMPIONSHIP L/RUNNER

BANK PANIC DRAGON WANG PENGUINLAND SEGA GALAGA AND

CHOPLIFTER.

JOIN NOW $20 to join PLUS $7.50 a cartridge
(includes return P&P)

NAME.....

ADDRESS.....

.....

1st choice_____
2nd ohoice_____
3rd Choice_____

ENCLOSED IS CHEQUE/MONEY ORDER

PLEASE CHARGE BANKCARD / VISA

I Aggree to pay for damage or loss (except when in the care of NZ Post) and also agree to pay $1.50 per day for each day cartridge is not returned by due date.

SIGNED .....
(over 18 years)

NO ..... EXPIRY .....

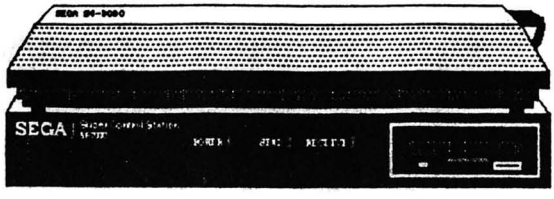
SIGNATURE.....

Welcome to the new look SEGA® Computer

The official magazine of the SEGA User Club of New Zealand

Contents

Cover illustration : Penguin Land - A 'my card' cartridge available from Poseidon Software Hire Club



Letters to the Editor

• **This question and answer page is provided to help you. So send me some questions. Remember that you can ask about software or programming.**

Dear Editor,
(a) I have had problems with my LSV program. Listing 1 seems to be OK, but when I run listing 2 I get a BAD FILE MODE ERROR IN 50 message. I have checked and rewritten it but can't find where it is wrong.
(b) Also in ASTRO ATTACK when my ship is hit I get a RETURN WITHOUT GOSUB IN LINE 1780. Where do I look for this problem .
J.I. Findlay, Whakatane

Editors reply,
(a) The error occurs because you are trying to load a Basic program as a machine language program. I can think of two ways inwhich this occured. (1) You have used a LOAD command instead of a LOADM command at line 50 or (2) You did not save each part of the program with the correct name. The machine language should have been saved as "LSV Dsk.Cde" so that line 50 can load it correctly.
(b) The error occurs because somewhere in the program a GOTO 1660 was used instead of a GOSUB 1660. There are three lines which contain this command. Check lines 1300, 1420 and 1540 for the problem
If you have any more problems send it in, or give me a call.

Dear Editor,
Could you please send me some information on how to continue on Transylvania Castle of Horror, down the stairway after digging the hole.
Patrick van de Pol

Editors reply,
Sorry I can't help you with this myself, as I have never played the game. But I am working on a solution for you.

Continued on next page

Dear Editor,

Congratulations on your New Magazine - a great effort. I am an Amateur Radio Operator and I am very interested in trying out my SEGA and Radio equipment on RTTY - using the International Murray Code.

(a) Do you know of a programme which will fill the bill?
(b) Can you possibly write a programme?
(c) Can anyone else?
(d) It is likely that the Japanese will produce a programme (in cassette)?

H. R. (Dusty) Miller, Marton

Editors reply,

(a) I beleive the Australians have produced software for modem handling. This may be of use, so I will find out more about this software.
(b) Yes! But I require some more information on what is required as I have little knowledge of this myself. For instance what type of interface does the modem you are constructing require (eg RS-232). Therefore do you have a Disk Drive (SF-7000 Super Control Station) to provide this interface. What would the software be required to do?

There are a number of other people who are interested in using there SEGA with RTTY, such as **W.J Downey** of Oamaru, **R.E. Templer** and **J.Linsay**. If you or anyone else interested would like to send me details, I will see what I can do.

(c) Is there anyone who has already written a program or can give help?
(d) Not likely!

Dear Editor,

Re David Pitmans cassette loading problems. The record / playback head, capstan (the little silver spindle beneath the black pinch roller) and the pinch roller itself must be kept clean. This is best done by means of cotton bud soaked in metholated spirits and should be done after every 8-10 hours of use.

The use of head cleaning tapes should be avoided, as most of them are slightly abrasive and will wear a groove in the head, which effects the drag and pressure. This therefore effects record and playback levels and cause *wow* and *flutter*. They also do not remove oxide build up from the pinch roller, the uneven build up of which causes the tape to run up off the roller, tangle and loop up. The head should be demagnetised every 12 months or so, depending on use.

Don't use music tapes, especially the poorer quality ones, as a slight oxide gap or miss will make little difference to a musical note, but to a computer, especially one as sensitive as the SEGA will certainly pick it up and give it error. Use only good quality low drop out computer tapes no more than C20, as the longer the tape the more sucseptible to tape stretch it is.

Take care of tapes. Keep them stored in a tape case away from dust. The little lugs in the case stop the tape from slackening. If you do notice any slackness use a thick pencil or ball point pen in the hub to retighten. Store them flat (opposite to video tapes).

Continued on page 7

EDITORIAL

After a few minor problems with our laser printer, which made the contents page and inverse characters program look like a *disaster area* the first issue turned out pretty well. Thanks to the number of people who wrote in to say so! Some of your letters arrived too late for publication.

Although the lack of programs being sent in has not yet hindered the magazine (as it did with previous publications), I cannot continue to write the whole magazine single handedly. I would appreciate any programs, no matter how small or silly they may seem to you. Thanks to Allan Clarke for a well written and informative article on arrays. There will be another article from Allan in the next magazine.

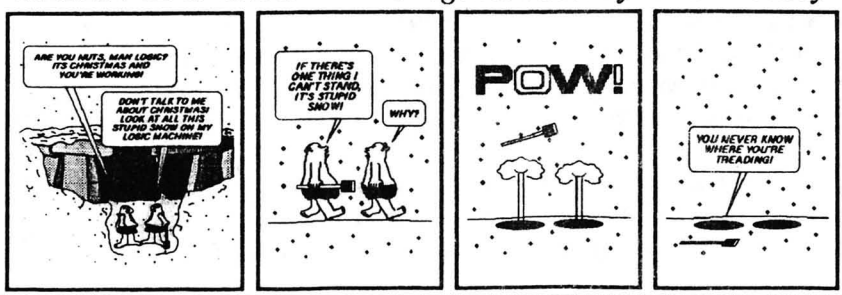
There has been a moderate response to the advertisement for new subscriptions, but not enough for the magazine to make the break-even point yet. However production of the magazine will continue.

For those of you who are interested, the magazine is designed completely on my Apple Macintosh computer system.This means I can't blame the typesetter for spelling mistakes, as I do all the typing!

There is no Adventure Section or Basic Programming in this issue because of the large Review Section and extra programs. Anyway I hope you like this issue and the cartoon below.

EDITOR: Michael Hadrup MJHadrup

MAN LOGIC *Original Idea by Neil Bradley*



News and Reviews

Mag Programmes on Cassette

For $20, you can forget about all the typing! With each new magazine you receive in this subscription year, a tape will also be provided which contains all the programs within that magazine. Simply follow the instructions as per the magazine articles and you will be able to use the programs immediately. I know that some of you do not have the time to type in the longer programs and this is a pity as some of the longer programs are the best. Hopefully the availability of these cassettes will let some of you make more use of your SEGA, rather than just for typing practice!

Penguin Land

This was one of my favourite cartridge games. As with most of these cartridge games the graphics and sound are brillant (music plays as you play), but what makes this game different is although the aim is simple (Take your egg from the top to the bottom of each scene where Mrs Penguin waits in your house), it requires a lot thinking to achieve this.

As you scroll downwards in each scene you must dig away ice cubes and let your egg drop to progress, but you can't let your egg drop too far otherwise it will break. There are menacing *Polar Bears* and *Sleeping Seals* who will try to smash your egg along the way. To help you there are rocks which can be dropped or pushed into the *Polar Bears* which kill them. You only get 3 eggs and if an egg breaks you must start all over again from the start of the scene.

Name : Penguin Land
Options : 1 / 2 Players, 25 different scenes
System : Cartridge - Poseidon Software Hire Club

Elevator Action

If you liked the arcade game then you will like this almost exact conversion of the Taito original. You must collect all the suitcases (which contain *Top Secret* information) hidden in the rooms with the red doors.

Work your way from the top floor to the ground level using the *Elevators* and *Escalators*, but watch out for the guards who won't hesitate to shoot first and ask questions later. Remember to collect all the suitcases of information, otherwise you will be sent back to get them.

Continued over page

If you make to the ground level, awaiting you is a *Porsche* so you can make a fast get away and collect a large bonus.

Name : Elevator Action
Options :1 / 2 Players
System : Cartridge - Poseidon Software Hire Club

Drol

Here is another exact arcade conversion of the Broderbund original with some of the most addictive music I have heard. In the first two rounds you (a robot) have to find the lost children - a girl with a balloon and a boy with a helicopter as well as their pets - an alligator and a lizard (strange children!). In the next round you have to rescue their mother who has been captured by the evil *Witch Doctor* and reunite her with her children.

Each round there are a number of creatures, such as *Deadly Scorpions* and *Snakes* which like the taste of robots. You can defend yourself against these menacing creatures by shooting them with your built-in laser. A really addictive game.

Name : Drol
Options :1 Player only
System : Cartridge - Poseidon Software Hire Club

Bank Panic

In have never seen this program in the arcades, but I am told that it is an arcade conversion and a very good one. As sheriff in a western town, it's your job to protect the innocent and shoot everything else! You must guard each of the twelve doors at the local bank, of which you can only see three doors at a time.

As customers (or bank robbers) arrive at each door, you must decide whether to shoot or not. As you play higher rounds the decision time gets shorter and shorter. It becomes a real test of reactions and this makes the game really addictive.

To make it even harder sometimes more than one shot is required to kill a robber and they may even take hostages, which get a little annoyed if you shoot them! You can get bonus points if you wait for the bank robber to draw (a fair shot), but be careful you have only half a second to respond.

Name : Bank Panic
Options : 1 / 2 Players, three starting levels (1,3 and 6)
System : Cartridge - Poseidon Software Hire Club

Continued on next page

Delta Fighter

Yor are the pilot of *Delta Fighter* and you must use this advanced fighter/bomber plane to penetrate the enemy outpost. To enable your main fleet to pass unnoticed, you must destroy all the scanning eyes while avoiding enemy aircraft and ground fire.

Your *Delta Fighter* is equipped with a photon cannon to destroy enemy aircraft and energy bombs to destroy ground installations and the scanning eyes. In mission one you must destroy eight scanning eyes and in the 12 x 12 screens this is quite a challange.

Delta Fighter the latest game from ATFUROS Software is a great cassette game and you'll love it. We must apologise for the mis-spelling of ATFUROS Software in the Adventure Section of the previous magazine.

Name : Delta Fighter
System : 16K tape for $32.95 Available from Poseidon Software Δ

Letters to the Editor Continued

Do not change modes ie. fast foward to play without first stopping the machine. If you are rewinding or fast fowarding to the end, stop the tape as soon as it reaches the end. As soon as a program has loaded, rewind the tape and put it away. Before recording on a new tape, play through both sides first to let it settle and gain initial stretch.

Keep tapes away from magnetic fields, television sets, loud speakers, all metals, shocks or jarring and excessive moisture of humidity.

Congratulations on getting SEGA up and running again. For a first attempt, a jolly fine effort and well done. So come on all you SEGA users, let's get together and in behind Michael and make this a big sucess as he deserves it.

I suggest we all put up a notice in our super market and community notice boards that SEGA is still alive and kicking, and give Poseidon's, your own or the Magazine phone number as contacts.

Merve Baucke, 3/124 Titirangi Road, New Lynn, Auckland 7. Ph 872-394

Editors reply

Although slighty edited the first part of your letter is printed as received. Thank you. Regarding your comments about LSV. LSV will help as it is designed to reduce the sensitivity of the SEGA to problems such as oxide gaps. One of my LSV tapes snapped in the middle of a program and I was able to rejoin the tape and still load the program sucessfully. As LSV is public domain software, it has no copyright protrction and it can be duplicated easily. Therefore there should always be at least one copy that works.

I must admit that the REM line did have 39 columns, but you were not asked to type that in and a check was provided anyway. I have not heard of anyone who had problems with this.

By the way, all my programs are written on the SEGA and I reserve the right to use ideas from other computers whch have been developed far further than SEGA ever will. Δ

Machine Code Programming

By Michael Hadrup

• **The second part of this courses covers registers, peeking, poking, adding and subtracting. There is a small example program to type in!**

Registers

A register is like a variable, in that it has a name and it can a store numbers like BASIC variables can. The big difference is that registers can only store numbers between 0 and #FF or 255. (The largest value that 8 bits or a 'byte' can hold). There are seven registers that we can use easily. Their names are **A, B, C, D, E, H and L.** (Don't ask me what happened to F, G, I, J and K).

Some of the registers may be used in pairs. B and C can be used as a pair and will hold four hex digits, (where B is called the 'high' byte and C the 'low' byte). There are three register pairs we can use, **BC, DE and HL.** If B contains #3B and C contains #14, then BC holds #3B14 when combined. Therefore a register pair can hold any number between 0 and #FFFF or 65535. (The largest value that 16 bits or a 'address' can hold.)

When working with decimal it's not just a question of joining the digits as above. If B contained 59d and C contained 20d then BC would contain 15124d. To work this out we need a calculator. We multiply the high byte by 256 and add this to the low byte. If you remember *Counting in Binary* from the previous issue, bit $7 = 2^7$ or 128. In the case of the high byte we consider this as starting at bit $8 = 2^8$ or 256. Hence the reason for multiplying by 256. Anyway, back to the calculation, 59*256+20 = 15124d.

There are another seven registers that we can use, which are in fact duplicates of A, B etc.. These are labeled **A', B', C', D', E', H' and L'**. You can only use one group at a time, and to change between groups there are two instructions

```
#08 EX AF, AF'
#D9 EXX
```

which swaps AF with AF' and AF' with AF

which is equivalent to EX BC, BC' EX DE, DE' and EX HL, HL'

LD - Machine Codes LET

In BASIC we would use A=5 or LET A=5 to assign values to a variable. In machine code we use the LD instruction. (This is an abbrievation for LOAD). For example A=5 becomes LD A,5 . There are 11 major forms of the LD instruction. Note that a comma is used instead of an equals sign.

LD Instruction

General from	*Example*	*In BASIC*
LD R,N	LD B,5	B=5
LD R,R'	LD B,A	B=A
LD RR,NN	LD BC,#F000	BC=&HF000
LD RR,(NN)	LD HL,(#F000)	HL=PEEK(#F000)+256*PEEK(#F001)
LD (NN),RR	LD (#F000),HL	POKE #F001,INT(HL/256)
		POKE #F000,HLMOD256
LD A,(NN)	LD A,(#F000)	A=PEEK(#F000)
LD (NN),A	LD (#F000),A	POKE #F000,A
LD R,(HL)	LD B,(HL)	B=PEEK(HL)
LD (HL),R	LD (HL),B	POKE HL,B
LD (HL),N	LD (HL),5	POKE HL,5
LD (RR),A	LD (DE),A	POKE DE,A
LD A,(RR)	LD A,(BC)	A=PEEK(BC)
R and R' are registers		*RR is a register pair*
N is an 8 bit number 0 - 255 or #FF		*NN is a 16 bit number 0 - 65535 or #FFFF*
(NN) is a 16 bit address		*(RR) is a location addressed by the register*

Figure 2.1

Each different LD instruction has a different code. For example the code for LD A,N is 3E followed by the data N - this instruction is two bytes in length whereas RET occupies only one byte #C9. Here are some of the hex codes:

LD R,N / LD RR,NN

```
06 00   LD B,0          01 00 FF   LD BC,#FF00
0E 00   LD C,0          11 00 FF   LD DE,#FF00
16 00   LD D,0          21 00 FF   LD HL,#F000
1E 00   LD E,0
26 00   LD H,0
2E 00   LD L,0
36 00   LD (HL),0
3E 00   LD A,0
```

NB With LD BC,#FF00 that the data is stored in reverse order 00 FF. Although this may seem strange, it is in fact usual for machine code. All two byte or 16 bit numbers are stored in reverse order - low byte followed by high byte.

Figure 2.2

In the table on the next page, you read the left-hand column registers first and the top row second. (All numbers are in hex). For example the code for LD L,E is #6B. Note that all of these instructions are one byte in length.

LD R,R'

LD	B	C	D	E	H	L	(HL)	A
B	40	41	42	43	44	45	46	47
C	48	49	4A	4B	4C	4D	4E	4F
D	50	51	52	53	54	55	56	57
E	58	59	5A	5B	5C	5D	5E	5F
H	60	61	62	63	64	65	66	67
L	68	69	6A	6B	6C	6D	6E	6F
(HL)	70	71	72	73	74	75	XX	77
A	78	79	7A	7B	7C	7D	7E	7F

(HL) is the value at the location addressed by HL ie PEEK (HL)

Figure 2.3

Brackets in machine code - Peeks and Pokes

If you look at figure 2.1 on the previous page you will see LD RR,NN and LD RR,(NN) as two forms of the LD instruction. The brackets are not just for variety or to make it look pretty, they do actually mean something! Brackets around a number or register pair refer to the contents of the address in the brackets and therefore peeks and pokes. eg.

```
LD HL,#F000        means   HL=&HF000
LD HL,(#F000)      means   HL=PEEK(&HF000)+256*PEEK(&HF001)
                   or      L=PEEK(&HF000):H=PEEK(&HF001)
LD (#F000),HL      means   POKE &HF001,INT(HL/256)
                           POKE &HF000,HL-256*PEEK(&HF001)
                   or      POKE &HF000,L:POKE #F001,H
```

```
0A          LD A,(BC)        02          LD (BC),A
1A          LD A,(DE)        12          LD (DE),A
2A 00 FF    LD HL<(#FF00)    22 00 FF    LD (#FF00),HL
3A 00 FF    LD A,(#FF00)     32 00 FF    LD (#FF00),A
ED 4B 00 FF LD BC,(#FF00)    ED 43 00 FF LD (#FF00),BC
ED 5B 00 FF LD DE,(#FF00)    ED 53 00 FF LD (#FF00),DE
```

Figure 2.4

Negative numbers

Sometimes it is useful to have negative numbers in machine code. One such time, as we will see later, is when moving sprites using an offset table. (To move a sprite to the left we would add -1 to its X coordinate). There is a special representation called *Two's Complement* for negative numbers in machine code. Using this method with decimal numbers we can represent the numbers -128 to 127 in 8 bits and -32768 to 32767 in 16 bits.

The way we do this is to write negative numbers as 256 (or 65536 for 16 bit numbers) minus the positve value. Therefore in 8 bits -1 = 255, -2 = 254, -3 = 253 and so on. In 16 bits -1 = 65535, -2 =65534 and -3 = 65533 and so on. If you think about this it is logical that negative numbers are represented this way. Consider the following using *Two's Complement*

2 - 1 = 1 (That's not too hard is it!)
In 8 bits 2 + 255 = 257 but 257 - 256 = 1 as the greatest value 8 bits can hold is 255, after which it starts counting again from zero.

24576 - 6144 = 18432 (That's a bit harder!)
In 16 bits 24576 + 59392 = 83968 but 83968 - 65536 = 18432 as the greatest value 16 bits can hold is 65535, after which it starts counting again from zero.

It is important to understand these concept, as it is essential when we start using Addition and Jump Relative instructions later on.

Simple arithmetic - Adding and Subtracting

Using the principles shown above 1 + 255 = 0 in 8 bits. The real answer in 16 bits is 256 or #0100 but only the low byte is kept with 8 bits. When an answer gets too big or too small, a *carry* occurs. At this stage it is time to introduce a new kind of machine code variable, called a **FLAG**. A flag is used to store two values, either a zero or a one.

One such flag is the *CARRY FLAG*. When a *carry* occurs with addition or subtraction, the *CARRY FLAG* is set to one. If answer is not too big or too small then there is no *carry* and the *CARRY FLAG* is reset to zero.

There are only two registers that you can add things to, A and HL. It is also important to note that only single registers may be added to A, and only register pairs to HL.

In other words you can't have ADD B,C or ADD DE,BC. There is a way you can ADD DE, BC and it involves adding the individual registers.

There is another instruction we can use instead of ADD. The instruction ADC stands for 'ADD with Carry'. Suppose the instruction ADC A,B is executed then A = A + B + *CARRY FLAG* (Previous value). As with ADD there are only two registers that you can ADC things to, A and HL.

Study these two programs. Sorry that they are on seperate pages.

Hex	*Assembler*	*In Basic*	
119644	LD DE,#4496	DE=&H4496	
21D88C	LD HL,#8CD8	HL=&H8CD8	
19	ADD HL,DE	HL=HL+DE	
		CARRY=0	
		IFHL>65535THENCARRY=1:HL=HL-65536	
CD9E7B	CALL #7B9E	PRINT HL	CD3A2B CALL #2B3A
C9	RET	STOP	*(Cartridge Basic Users)*

and

```
119644 LD DE,#4496          DE=&H4496
21D88C LD HL,#8CD8          HL=&H8CD8
7D     LD A,L               A=L
83     ADD A,E              A=A+E
                            CARRY=0
                            IFA>255THENCARRY=1:A=A-256
6F     LD L,A               L=A
7C     LD A,H               A=H
8A     ADC A,D              A=A+D+CARRY
                            CARRY=0
                            IFA>255THENCARRY=1:A=A-256
67     LD H,A               H=A
CD9E7B CALL #7B9E           PRINT HL        CD3A2B CALL #2B3A
C9     RET                  STOP            (Cartridge Basic Users)
```

The effect of both of these progams is the same. You can learn two things from this. Firstly the instruction LD does not affect or alter the value of the *CARRY FLAG* . Secondly the insruction ADD HL,DE is much shorter (and neater) than adding the registers seperately.

Let's run both of these programs to verify that they are the same. To do this type in the *MC Editor* from the previous issue (remember to fix the mistake as shown in this issue's Boo Boo page), and add one of the following lines.

```
1000 DATA 11964421D88C19CD9E7BC9
1010 DATA 11964421D88C7D836F7C8A67CD9E7BC9
```

or for Cartridge Basic Users

```
1000 DATA 11964421D88C19CD3A2BC9
1010 DATA 11964421D88C7D836F7C8A67CD3A2BC9
```

Now type RUN then type

```
CALL &HC000:PRINT:CALL &HC00B
```

The answer should be 53614 for both routines. If the answer is not 53614 then recheck the data statements. and try again.

Rather than continue to reproduce tables in the magazine which take up a lot of space, I have a set of **SUMMARY SHEETS** of all the machine code instructions and their opcodes in hex. One is ordered numerically for *Disassembly* and the other alphabetically for *Assembly* of programs and this makes them the best summary sheets I have seen. If you would like a copy of these send $1.00 to Poseidon Software using the magazine order form.

Next Issue : Next time I will cover the Stack, Jumping / Calling (Basic's GOTO AND GOSUB) and comparing (Basic's IF statement).

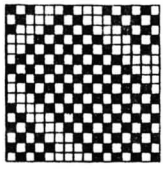
Δ

PATTERN PAINT

By Michael Hadrup

• For 32K and Disk Basic

The SEGA has a well designed Paint routine built into its Basic, which although fast it is sometimes not fast enough and it is restrictive because it paints only solid pixels. This program provides a Paint routine which is much faster and can *wash* the solid pixels with a pattern. Each pattern is designed from a 16x16 character (similar to a mag 1 sprite) and this is used to form a repeating pattern like tiles on a floor. See the example below



As you can see from this and the title above a number of interesting patterns can be designed. Remember that this program is compaitable with Print 64 and LSV and can therefore be used in conjunction with these two programs. In the next issue we will combine all three of these programs with a compressed pictures program eventually leading to an *GRAPHICS ADVENTURE* in later issues. The following information should be useful to some

PAINT	#EDE0-#EFFF
PRINT 64	#F000-#F8FF
LSV	#F900-#FFFF

Typing in the Program - Disk Basic Users

Type in listing 1 followed by listing 2 and listing 4, carefully. Each line in listing 4 contains 32 bytes of machine code followed by a checksum for that line. The program can be saved at any time by typing `SAVE "Paint.Dta"`

When you have finished type RUN and wait. The machine code is being poked to &HF000. If all goes well then the BASIC program will print the message "No errors!!". If you don't get this message then check the offending data line for typing errors, and type RUN again.

When you get (finally?), the message "No errors!!", insert a disk onto which the code can be saved and press space. The program will also automatically save itself.

Typing in the Program - Cartridge Basic Users Using LSV

If you have typed in LSV from the October '86 issue of SEGA Computer, then the process of typing Pattern Paint is simplified. Type in listing 1 followed by listing 4 and carefully make the alterations for Cartridge Basic by replacing those lines as shown in listing 5. The program can be saved at any time by typing `*SAVE "Paint.Data"`

When you have finished, save it, type RUN and wait. The machine code is being poked to &HEDE0. If all goes well then the BASIC program will print the message "No errors!!". If you don't get this message then check the offending data line for typing errors, and type RUN again.

When you get (finally?), the message "No errors!!", save the machine code using

```
*SAVEC "Paint.Code",&HEDE0,&HEFFF
```

Typing in the Program - Cartridge Basic Users without LSV

Start by typing a line "1 REM" followed by seven lines of zeros. The REM statements are where the machine code will be stored and this will allow us to load and save machine code in the form of a BASIC program.

When you list the program, Basic will have truncated the line to the required amount of characters and the line should look something like this...

```
1 REM 00000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000
```

Type `PRINT PEEK(&H9800)` . If the answer is not 251, then list the program and add some more characters to the REM statement and recheck.

Now duplicate line 1 as line 2 and line 2 as line 3. To make the listing easier to read, type as a direct command.

```
FORN=0TO2:POKE&H9806+N*257,13:NEXT
```

Now add to this Listing 3 then Listing 4 and carefully make the alterations for Cartridge Basic by replacing those lines as shown in listing 5. The program can be saved at any time by typing `SAVE "Paint.Data"`

When you have finished, save it, type RUN and wait. The machine code is being poked into the REM statements. If all goes well then the BASIC program will print the message "No errors!!". If you don't get this message then check the offending data line for typing errors, and type RUN again.

When you get (finally?), the message "No errors!!", delete lines 100 onwards and save the program using

```
DELETE 100-
SAVE "Paint"
```

Using the Program

Before you can use Pattern Paint, the machine code must be stored at &HEDE0. There are three ways to do this, depending on whether you are using LSV, Disk Basic or Cartridge Basic.

Disk Basic Users : Add this line to the start of your program to load the saved code.

```
LOADM "Paint.Cde",&HEDE0
```

Cartridge Basic with LSV : Add this line to the start of your program to load the saved code.

```
*LOADC "Paint.Code",&HEDE0
```

Cartridge Basic without LSV : Add your program after the REM statements and this line.

```
CALL &H9A69
```

The routine at &H9A69 copies the machine code from the REM statements to &HEDE0 .

Painting with Pattern Paint

Before you actually do any painting, you must give Patter Paint certain information. Take a look at Listing 6 - the demo program. Lines 10-100 define twelve addresses to which you must poke information for Pattern Paint - see below.

XC	= &HEE4A	X-Coordinate (where to start painting)
YC	= &HEE4B	Y-Coordinate
SCRN	= &HEDEB	16 bit address of storage screen. Use &HD500 for now
PTTRN	= &HEDF6	16 bit address points to start of pattern
MASK	= &HEF07	Determines whether colour already on screen is used
COLOUR	= &HEF13	Colour to paint with
X	= &HEE6D	Left margin of screen window
Y	= &HEE9E	Top margin of screen window
X1	= &HEF23	Right margin of screen window
Y1	= &HEE88	Bottom margin of screen window
PNT	= &HEDE0	Where you call to paint the screen
WASH	= &HEDE3	Where you call to wash the last pattern painted

Painting with Colour

MASK and *COLOUR* are used to determine the colour which Pattern Paint uses. If *MASK* is poked with

255 The colour already on the screen is used.
240 The foreground colour (ink) already on the screen is used.
15 The background colour (paper) already on the screen is used.
0 No colour from the screen is kept

The value of *COLOUR* is given by background + 16*foreground colour. If you are using the *MASK* (not zero) then the respective *COLOUR* values must be zero.

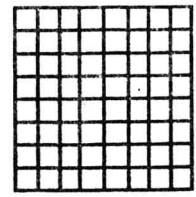
Lines 190-200 of the demo program set the paint colour as black on white with no colour kept from the screen.

Using the Patterns

As I described earlier the patterns are designed from 16x16 shapes. Each pattern requires 32 bytes stored like this (similar to a Mag1 sprite)

A	B
C	D

A	B	C	D
0	8	16	24
1	9	17	25
2	10	18	26
3	11	19	27
4	12	20	28
5	13	21	29
6	14	22	30
7	15	23	31



You must poke the variable PTTRN (&HEDF6) with the start address of the pattern. Line 350 of the demo program selects one of the patterns randomly and lines 360-370 poke the variable PTTRN.

I think that an explanation of how the demo program stores the patterns is essential. When the SEGA is using SCREEN 1 sprite patterns are stored in the *VRAM Save Area* . This is &HA316 or &H8B36 for Cartridge. When the SEGA changes to SCREEN 2, the sprite patterns are swapped with the character patterns (as defined with PATTERN C#) and are now stored at &H1800 in VRAM.

There is a special routine at &H64CA or &H2BD4 for cartridge which handles this swapping of sprite and character patterns. The demo program uses the PATTERN S# command to store the Paint patterns (as it is very quick) and uses the swap routine (lines 380 and 440) to switch these patterns into normal RAM - the *VRAM Save Area* so that Pattern Paint can use them.

Calling PNT (&HEDE0) starts painting at the point defined by the point XC,YC (&HEE4A) within the window (X,Y)-(X1,Y1). After completing the paint it will wash the pattern defined by PTTRN (&HEDF6) over all the solid pixels it filled.

Calling WASH (&HEDE3) will wash the last fill of solid pixels with the pattern defined by PTTRN. Note that washing a pattern does not affect the colour of the pixels.

Now you can try the demo program. The program starts by drawing 10 randomly sized ellipses on the graphics screen. It then paints 20 times at random positions with a random pattern and washs this pattern with another random pattern. The demo program uses the window (16,16)-(239,175). If you press Space it will clear the screen and start again.

If you press Break and the characters are all mudled then clear the screen and attempt to type CALL &H64CA or &H2B4D for cartridge. This will reset the character set.

Hopefully I have given you all the information required to use Pattern Paint. In the next issue Pattern Paint will be combined with a compressed pictures program. This is based on building up a picture as a set of lines, circles and boxes and using the patterns to fill these.

Listing 1 - Disk Basic and LSV Users

```
10 CLS
20 X=&HEDE0:RESTORE1000
30 FORN=1000TO1320STEP20:C=0
40 CURSOR0,0:PRINTHEX$(X)
50 FORM=0TO31:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
60 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N:STOP
70 NEXT
80 BEEP:BEEP:PRINT"No errors!!"
```

Listing 2 - Disk Basic Users only

```
90 PRINT:PRINT"Insert DISK and press space"
100 A$=INKEY$:IFA$<>" "THEN100
110 SAVEM"PAINT  .Cde",&HEDE0,&HEFFF
120 PRINT:PRINT"Saving """;
130 PRINT"PAINT  .Dta""":CALL&H21D4
140 END
```

Listing 3 - Cartridge Basic Users wihout LSV only

```
100 CLS
110 X=&H9807:RESTORE1000
120 FORN=1000TO1270STEP140
130 FORF=0TO6:C=0
140 CURSOR0,0:PRINTHEX$(X)
150 FORM=0TO31:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
160 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N+F*20:STOP
170 NEXTF:X=X+33
180 NEXTN
190 FORF=0TO3:C=0
200 CURSOR0,0:PRINTHEX$(X)
210 FORM=0TO31:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
220 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";1280+F*20:STOP
230 NEXTF
240 BEEP:BEEP:PRINT"No errors!!"
250 END
```

Listing 4 - Data Listing

```
1000 DATA CD,37,EE,F3,DB,BF,21,0,0,D9,11,0,D5,1,0,18,78,1F,DA,F8,ED,21,0,
D0,C5,41,D9,7D,D3,BF,7C,D3,FFC
1010 REM
1020 DATA BF,D9,1A,2F,4F,DB,BE,A1,4F,1A,A6,13,23,B1,D9,4F,7D,D3,BF,7C,C6,
40,D3,BF,23,79,D9,D3,BE,78,3D,E6,114C
1030 REM
1040 DATA F,C2,30,EE,5,CA,32,EE,4,7D,D6,10,6F,9F,84,67,10,C8,C1,10,BB,FB,
C9,F3,DB,BF,2A,EB,ED,ED,5B,EB,1228
1050 REM
1060 DATA ED,13,1,FF,17,36,0,ED,B0,11,80,5F,F3,DB,BF,D9,21,FF,FF,22,9C,EF
,23,D9,3A,EC,ED,32,1B,EF,2A,D5,1156
1070 REM
1080 DATA 9E,22,49,EF,CD,9E,EF,CD,BA,EF,C0,7B,FE,0,CA,7D,EE,CD,CB,EF,CD,B
A,EF,CA,6B,EE,CD,D4,EF,D9,11,1,15D1
```

```
1090 REM
1100 DATA 1,D9,D9,42,4B,D9,7A,FE,BF,3E,1,CA,99,EE,CD,E9,EF,CD,BA,EF,8,CD,
DD,EF,8,D9,5F,D9,7A,FE,0,3E,136B
1110 REM
1120 DATA 1,CA,AF,EE,CD,DD,EF,CD,BA,EF,8,CD,E9,EF,8,D9,57,EE,1,A0,CA,C0,E
E,23,D9,15,D5,CD,56,EF,14,D9,1448
1130 REM
1140 DATA 7B,EE,1,A1,CA,D0,EE,23,D9,14,D5,CD,56,EF,15,D9,D9,7D,D3,BF,7C,D
3,BF,E5,0,0,DB,BE,B1,47,0,7D,1261
1150 REM
1160 DATA D3,BF,7C,C6,40,D3,BF,78,D3,BE,3A,9C,EF,BD,C2,F8,EE,3A,9D,EF,BC,
CA,19,EF,7D,D3,BF,7C,C6,20,D3,BF,1530
1170 REM
1180 DATA 0,0,0,0,DB,BE,E6,FF,47,7D,D3,BF,7C,C6,60,D3,BF,78,F6,0,D3,BE,22
,9C,EF,7C,C6,0,67,7E,B1,77,1103
1190 REM
1200 DATA E1,7B,FE,FF,CA,33,EF,CD,D4,EF,CD,BA,EF,CA,82,EE,CD,CB,EF,D9,7C,
B5,CA,48,EF,2B,D9,D1,CD,9E,EF,CD,1813
1210 REM
1220 DATA BA,EF,CA,6B,EE,C3,33,EF,21,0,0,22,D5,9E,CD,CA,75,CD,71,75,FB,C9
,E5,21,6E,65,39,E1,D8,31,10,A3,1199
1230 REM
1240 DATA CD,48,EF,CD,6B,EF,3E,1B,C3,D3,75,F3,DB,BF,21,0,0,1,0,18,ED,5B,E
B,ED,7D,D3,BF,7C,D3,BF,1A,2F,10DC
1250 REM
1260 DATA D9,4F,DB,BE,A1,8,D9,7D,D3,BF,7C,F6,40,D3,BF,8,23,13,B,D3,BE,78,
B1,C2,78,EF,FB,C9,FF,FF,7A,E6,13E4
1270 REM
1280 DATA 7,67,7B,E6,F8,84,6F,7A,F,F,F,E6,1F,67,7B,E6,7,3C,47,3E,1,F,10,F
D,4F,C9,7D,D3,BF,7C,D3,BF,E48
1290 REM
1300 DATA 0,0,0,0,DB,BE,A1,C8,3E,1,C9,1D,CB,1,D0,7D,D6,8,6F,C9,1C,CB,9,D0
,7D,C6,8,6F,C9,15,7D,2D,D58
1310 REM
1320 DATA E6,7,C0,7D,C6,8,6F,25,C9,14,2C,7D,E6,7,C0,7D,D6,8,6F,24,C9,6F,2
C,7D,C9,0,0,0,0,0,0,0,B57
```

Listing 5 - Alterations to Listing 4. Cartridge Basic Users only

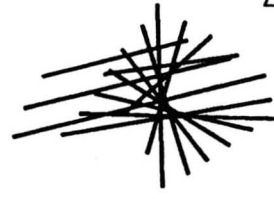
```
1060 DATA ED,13,1,FF,17,36,0,ED,B0,11,80,5F,F3,DB,BF,D9,21,FF,FF,22,9C,EF
,23,D9,3A,EC,ED,32,1B,EF,2A,E8,1169
1080 DATA 86,22,49,EF,CD,9E,EF,CD,BA,EF,C0,7B,FE,0,CA,7D,EE,CD,CB,EF,CD,B
A,EF,CA,6B,EE,CD,D4,EF,D9,11,1,15B9
1220 DATA BA,EF,CA,6B,EE,C3,33,EF,21,0,0,22,E8,86,CD,68,25,CD,F,25,FB,C9,
E5,21,5B,7D,39,E1,D8,31,30,8B,103D
1240 DATA CD,48,EF,CD,6B,EF,3E,1B,C3,71,25,F3,DB,BF,21,0,0,1,0,18,ED,5B,E
B,ED,7D,D3,BF,7C,D3,BF,1A,2F,102A
1320 DATA E6,7,C0,7D,C6,8,6F,25,C9,14,2C,7D,E6,7,C0,7D,D6,8,6F,24,C9,6F,2
C,7D,C9,0,0,0,0,0,0,0,B57
1340 DATA 21,7,98,11,E0,ED,6,2,C5,1,E0,0,ED,B0,1,21,0,9,C1,10,F3,1,60,0,E
D,B0,C9,0,0,0,0,0,A9F
```

Listing 5 - Demo Program

```
10 XC=&HEE4A
20 YC=&HEE4B
30 SCRN=&HEDEB
40 PTTRN=&HEDF6
50 MASK=&HEF07
60 COLOUR=&HEF13
70 X=&HEE6D
80 Y=&HEE9E
90 X1=&HEF23
100 Y1=&HEE88
110 PNT=&HEDE0
120 WASH=&HEDE3
130 REM
140 REM
150 POKESCRN,0
160 POKESCRN+1,&HD5
170 POKEX,16:POKEY,16
180 POKEX1,239:POKEY1,175
190 POKECOLOUR,31
200 POKEMASK,0
210 RESTORE510
220 FORN=0TO95
230 READA$
240 PATTERNS$N,A$
250 NEXT
260 REM
270 REM
280 SCREEN2,2:COLOR1,15:CLS
290 FORN=1TO10
300 CIRCLE(RND(1)*254+1,RND(1)*190+1)
,RND(1)*50+20,1,RND(1)+.4
310 NEXT
320 FORN=1TO20
330 POKEXC,RND(1)*224+16
340 POKEYC,RND(1)*160+16
350 A=INT(RND(1)*24)*32+&HA316+65536
360 POKEPTTRN,AMOD256
370 POKEPTTRN+1,INT(A/256)
380 CALL&H64CA
390 CALLPNT
400 A=INT(RND(1)*24)*32+&HA316+65536
410 POKEPTTRN,AMOD256
420 POKEPTTRN+1,INT(A/256)
430 CALLWASH
440 CALL&H64CA
450 IFINKEY$<>""THEN280
460 NEXTN
470 BEEP
480 FORN=1TO800
490 IFINKEY$=""THENNEXT
500 GOTO280
510 DATA FFFFFFFFFFFFFFFF
520 DATA FFFFFFFFFFFFFFFF
530 DATA FFFFFFFFFFFFFFFF
540 DATA FFFFFFFFFFFFFFFF
550 DATA 808080FF080808FF
560 DATA 808080FF080808FF
570 DATA 808080FF080808FF
580 DATA 808080FF080808FF
590 DATA 8040201008040201
600 DATA 8040201008040201
610 DATA 8040201008040201
620 DATA 8040201008040201
630 DATA 8142241818244281
640 DATA 8142241818244281
650 DATA 8142241818244281
660 DATA 8142241818244281
670 DATA FF000000FF000000
680 DATA FF000000FF000000
690 DATA FF000000FF000000
700 DATA FF000000FF000000
710 DATA 55AA55AA55AA55AA
720 DATA 55AA55AA55AA55AA
730 DATA 55AA55AA55AA55AA
740 DATA 55AA55AA55AA55AA
750 DATA AAAAAAAAAAAAAAAA
760 DATA AAAAAAAAAAAAAAAA
770 DATA AAAAAAAAAAAAAAAA
780 DATA AAAAAAAAAAAAAAAA
790 DATA F0F0F0F00F0F0F0F
800 DATA F0F0F0F00F0F0F0F
810 DATA F0F0F0F00F0F0F0F
820 DATA F0F0F0F00F0F0F0F
830 DATA 01010101010101FF
840 DATA 80808080808080FF
850 DATA FF01010101010101
860 DATA FF80808080808080
870 DATA 8800000088000000
880 DATA 8800000088000000
890 DATA 8800000088000000
900 DATA 8800000088000000
910 DATA 10698000224C0000
920 DATA 0020580304205086
930 DATA 0020D10A0008D600
940 DATA 0040320002046800
950 DATA 0018244242241800
960 DATA 0018244242241800
970 DATA 0018244242241800
980 DATA 0018244242241800
990 DATA 8080403C02010101
1000 DATA C020101010080807
1010 DATA C020101010080807
```

```
1020 DATA 8080403C02010101
1030 DATA 8080808040201807
1040 DATA 0000000001020CF0
1050 DATA 0000000001020CF0
1060 DATA 8080808040201807
1070 DATA 8041221408102040
1080 DATA 8041221408102040
1090 DATA 8041221408102040
1100 DATA 8041221408102040
1110 DATA 3E5C88C5E3D1881D
1120 DATA 3E5C88C5E3D1881D
1130 DATA 3E5C88C5E3D1881D
1140 DATA 3E5C88C5E3D1881D
1150 DATA 0F10274853545555
1160 DATA F008E412C9259555
1170 DATA 54534827904F201F
1180 DATA 559525C912E408F0
1190 DATA 55AA55BF5FBA58BA
1200 DATA 55AA55FEFDAEAD8E
1210 DATA 5ABA5FBF55AA55AA
1220 DATA ADAEFDFE55AA55AA
1230 DATA 0000103854103854
1240 DATA 5410385410385410
1250 DATA 1038541010100000
1260 DATA 1010100000001038
1270 DATA FEFEC6C6C6FEFE00
1280 DATA FEFEC6C6C6FEFE00
1290 DATA FEFEC6C6C6FEFE00
1300 DATA FEFEC6C6C6FEFE00
1310 DATA 77EEDDAA77EEDDAA
1320 DATA 77EEDDAA77EEDDAA
1330 DATA 77EEDDAA77EEDDAA
1340 DATA 77EEDDAA77EEDDAA
1350 DATA 00040E0E1F1F3F3F
1360 DATA 000000207070F8F8
1370 DATA 7F7FFFFF00000000
1380 DATA FCFCFEFE00000000
1390 DATA 000101110905037F
1400 DATA 00000010204080FC
1410 DATA 0305091101010000
1420 DATA 8040201000000000
1430 DATA 00903317444A7218
1440 DATA 027044CEA0186C0C
1450 DATA 3036180573220E40
1460 DATA 184E5222E8CC0900
```

Δ



```
10 REM         Graphics demo
20 REM
30 REM       By Michael Hadrup
40 REM
100 IF(PEEK(&HF701)XORPEEK(&HF702))=119THEN170
110 CLS:X=&HF700:RESTORE1000
120 FORN=1000TO1100STEP10:C=0
130 CURSOR0,0:PRINTHEX$(X)
140 FORM=0TO31:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
150 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N:STOP
160 NEXT
170 SCREEN2,2:COLOR15,1,,1:CLS
171 SCREEN1,1:CLS
180 PRINT"Use the enter key to return to BASIC
190 PRINT:PRINT"Use the space bar to trap the quix
200 PRINT"( This forces the quix to the bottom    right corner )
210 PRINT:PRINT:PRINT
220 INPUT"Number of lines ";N
230 IFN=0THEN220
240 POKE&HF70D,NMOD256:POKE&HF70E,INT(N/256)
250 CALL&HF700
260 GOTO170
1000 DATA CD,14,63,3E,F1,32,1A,AB,CD,CE,4F,FB,21,18,0,29,29,44,4D,21,48,F
8,11,49,F8,36,0,ED,B0,21,48,F8,D52
1010 DATA 6,2,CD,A0,F7,77,23,CD,A0,F7,FE,C0,D2,27,F7,77,23,10,EF,CD,35,F8
,21,0,0,E5,29,29,11,48,F8,19,F6D
```

```
1020 DATA E5,5E,23,56,23,7E,23,66,6F,AF,47,CD,F0,60,D1,E1,E5,D5,7C,B5,C2,
5A,F7,2A,D,F7,2B,29,29,11,48,F8,1014
1030 DATA 19,3E,1,D3,DE,DB,DC,E6,10,CC,35,F8,FD,21,44,F8,D1,E5,CD,A6,F7,E
1,5E,23,56,23,7E,23,66,6F,AF,47,1170
1040 DATA 3C,CD,F0,60,E1,3E,5,D3,DE,DB,DC,E6,40,C8,23,ED,5B,D,F7,7A,BC,C2
,39,F7,7B,BD,D2,39,F7,C3,36,F7,1394
1050 DATA ED,5F,90,1F,80,C9,CD,A9,F7,7E,FD,CB,0,7E,C2,BD,F7,FD,86,0,D2,C3
,F7,CD,E8,F7,C3,A9,F7,FD,86,0,158C
1060 DATA D2,B7,F7,CD,E2,F7,7E,FD,CB,0,7E,C2,DC,F7,FD,86,0,FE,C0,DA,E2,F7
,CD,E8,F7,C3,C6,F7,FD,86,0,D2,17F4
1070 DATA D6,F7,12,23,13,FD,23,C9,FD,7E,0,ED,44,FD,77,0,D9,3E,80,D3,7F,ED
,5F,E6,1F,D3,7F,16,F,5A,1C,3E,F83
1080 DATA 90,B2,D3,7F,1,64,0,B,78,B1,C2,7,F8,1D,20,FD,15,7A,FE,FF,20,E7,1
6,0,3E,10,92,5F,3E,90,B2,D3,E63
1090 DATA 7F,1,C8,0,B,78,B1,C2,24,F8,1D,20,FD,14,7A,FE,10,20,E5,D9,C9,11,
44,F8,6,4,CD,A0,F7,E6,8F,12,F19
1100 DATA 13,10,F7,C9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,1E3
```

Alterations for Cartridge Basic

```
100 IF(PEEK(&HF701)XORPEEK(&HF702))=173THEN170
1000 DATA CD,90,3D,3E,F1,32,3A,93,CD,6C,53,FB,21,18,0,29,29,44,4D,21,48,F
8,11,49,F8,36,0,ED,B0,21,48,F8,D52
1020 DATA E5,5E,23,56,23,7E,23,66,6F,AF,47,CD,DA,3B,D1,E1,E5,D5,7C,B5,C2,
5A,F7,2A,D,F7,2B,29,29,11,48,F8,FD9
1040 DATA 3C,CD,DA,3B,E1,3E,5,D3,DE,DB,DC,E6,40,C8,23,ED,5B,D,F7,7A,BC,C2
,39,F7,7B,BD,D2,39,F7,C3,36,F7,1359
```

Δ

Fast Disk Copy

By Michael Hadrup

Type in and save the small program below using `SAVE "DiskCopy"` Next type in the large listing, which can be saved at any time with `SAVE "DiskCopy.Dta"`

When you have finished save it, type RUN and wait. If you don't get the message "No errors!!" then check the offending line and type RUN again. When you do get the message "No errors!!" insert a disk onto which the code can be saved. To use the program type `RUN "DiskCopy"` and follow the instructions.

```
10 LOADM "DiskCopy.Cde",&HFA00:CALL &HFF00
```

```
10 CLS
20 X=&HFA00:RESTORE1000
30 FORN=1000TO1230STEP10:C=0
40 CURSOR0,0:PRINTHEX$(X)
50 FORM=0TO63:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
60 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N:STOP
70 NEXT
80 BEEP:BEEP:PRINT"No errors!!"
90 PRINT:PRINT"Insert DISK and press space"
100 A$=INKEY$:IFA$<>" "THEN100
110 SAVEM"DiskCopy.Cde",&HFA00,-1
120 PRINT:PRINT"Saving ""
```

```
130 PRINT"DiskCopy.Dta""":CALL&H21D4
140 END
1000 DATA 3E,C,D3,E7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C3,2C,0,11,1,0,305
1010 DATA 6,A,3E,F,CD,C2,0,CD,C2,0,3E,A,CD,C2,0,CD,C9,1,1,7,2A,2A,20,43,6
F,70,79,20,63,6F,6D,70,6C,65,74,65,64,20,2A,2A,80,CD,C9,1,3,48,69,74,20,7
3,70,61,63,65,20,62,61,72,20,74,6F,20,63,6F,1606
1020 DATA 70,79,20,61,6E,6F,74,68,65,72,20,64,69,73,6B,82,CD,5A,2,C3,69,0
,C5,D5,F5,CD,C9,1,1,49,6E,73,65,72,74,20,73,6F,75,72,63,65,20,64,69,73,6B
,20,69,6E,20,64,69,73,6B,20,64,72,69,76,65,D,61,6E,198C
1030 DATA 64,20,68,69,74,20,74,68,65,20,73,70,61,63,65,20,62,61,72,82,CD,
A6,2,F1,F5,21,0,10,8,1,0,10,CD,8C,4,C2,77,1,14,8,3D,C2,8,1,CD,C9,1,1,49,6
E,73,65,72,74,20,64,65,73,74,69,6E,61,74,69,167F
1040 DATA 6F,6E,20,64,69,73,6B,20,69,6E,20,64,69,73,6B,20,64,72,69,76,65,
D,61,6E,64,20,68,69,74,20,74,68,65,20,73,70,61,63,65,20,62,61,72,82,CD,A6
,2,CD,CD,2,F1,D1,F5,21,0,10,8,1,0,10,CD,3A,4,C2,178C
1050 DATA 77,1,14,8,3D,C2,64,1,F1,C1,C9,1,0,F0,9,CD,C9,1,1,7,3F,3F,20,43,
61,6E,6E,6F,74,20,63,6F,70,79,20,74,68,69,73,20,64,69,73,6B,20,3F,3F,D,2A
,2A,20,28,20,55,6E,66,6F,72,6D,61,74,65,64,20,1527
1060 DATA 6F,72,20,70,72,6F,74,65,63,74,65,64,20,64,69,73,6B,20,29,20,2A,
2A,80,F1,F1,F1,C3,95,0,E3,7E,E6,7F,CD,D8,1,CB,7E,23,CA,CA,1,E3,C9,C5,D5,E
5,1,0,0,FE,20,DA,1A,2,F5,21,0,7C,11,28,0,78,A7,1C95
1070 DATA CA,F4,1,19,5,C3,EA,1,9,7D,D3,BF,7C,D3,BF,F1,D3,BE,2A,DC,1,2C,7D
,FE,28,C2,13,2,2E,2,24,7C,FE,18,C2,13,2,26,0,22,DC,1,E1,D1,C1,C9,21,3D,2,
E5,FE,1,CA,41,2,FE,2,CA,60,2,FE,3,CA,57,1C9D
1080 DATA 2,FE,7,CA,6C,2,FE,C,CA,83,2,FE,D,CA,9E,2,C9,E1,D1,C1,C9,21,2,B,
CD,6C,2,22,DC,1,6,4E,3E,20,CD,D8,1,10,F9,22,DC,1,C9,CD,6C,2,21,2,F,C3,47,
2,3E,1,D3,DE,DB,DC,E6,10,C8,C3,64,2,1BAA
1090 DATA 1,10,27,3E,90,D3,7F,B,78,B1,20,FB,E5,21,D,5,1,7F,6,ED,B3,E1,C9,
21,2,0,22,DC,1,AF,D3,BF,3E,7C,D3,BF,1,C0,3,3E,20,D3,BE,B,78,B1,C2,93,2,C9
,E,2,4,ED,43,DC,1,C9,CD,FE,2,D0,CD,C9,1C62
1100 DATA 1,3,7,3F,3F,20,44,69,73,6B,20,6E,6F,74,20,72,65,61,64,79,20,3F,
3F,80,CD,FE,2,DA,C4,2,C3,5A,2,CD,80,3,D0,CD,C9,1,3,7,3F,3F,20,44,69,73,6B
,20,77,72,69,74,65,20,70,72,6F,74,65,63,74,65,173F
1110 DATA 64,20,3F,3F,80,6,0,CD,80,3,DA,F1,2,10,F8,C3,5A,2,C5,D5,E5,DB,E6
,E6,F2,F6,20,D3,E6,CB,4F,CA,20,3,3E,2,D3,E7,1,65,3,AF,3D,C2,16,3,B,78,B1,
C2,15,3,11,5,64,2E,0,CD,70,3,CA,63,3,2E,1B38
1120 DATA 4,CD,70,3,CA,63,3,1,51,3C,2E,0,CD,73,3,CA,4E,3,21,F7,A,A7,ED,42
,D2,55,3,15,C2,23,3,C3,63,3,1D,C2,23,3,C3,63,3,CD,C0,3,21,3,5,3E,3,CD,9F,
3,CA,6B,3,CD,CA,3,37,E1,D1,C1,C9,A7,1859
1130 DATA E1,D1,C1,C9,1,0,0,DB,E4,E6,4,BD,C0,B,78,B1,C2,73,3,C9,C5,D5,E5,
CD,C0,3,21,6,5,3E,2,CD,9F,3,C2,63,3,CD,10,4,3A,1B,5,E6,40,C2,63,3,C3,55,3
,C5,47,E,E1,DB,E0,CB,67,C2,A3,3,DB,E0,1E59
1140 DATA 7,D2,AA,3,7,DA,BC,3,ED,A3,C2,AA,3,C1,AF,C9,C1,C3,10,4,DB,E4,F,D
0,CD,10,4,C3,C0,3,3E,7,D3,E7,3D,D3,E7,DB,E6,E6,F0,F6,20,D3,E6,E5,CD,C0,3,
21,3,5,3E,3,CD,9F,3,C2,DA,3,CD,F0,3,C2,21C6
1150 DATA DA,3,E1,C9,CD,C0,3,E5,21,B,5,3E,2,C3,6,4,7A,E5,32,A,5,21,8,5,3E
,3,CD,9F,3,E1,DB,E4,F,D2,A,4,E5,21,1B,5,DB,E0,7,D2,14,4,7,DA,2C,4,3E,8,D3
,E1,DB,E0,7,D2,22,4,7,D2,33,4,17EE
1160 DATA DB,E1,77,23,C3,22,4,3A,1B,5,E6,C0,E1,C9,E5,21,5,5,E3,C5,D5,E5,C
D,5D,4,E1,D1,C1,E3,CA,5A,4,2D,C2,3E,4,CD,CA,3,2E,5,25,C2,3E,4,25,E1,9,C9,
CD,FC,3,C0,3E,45,CD,DE,4,C0,1E,40,50,14,41,1D87
```

```
1170 DATA E,E1,78,A7,CA,7F,4,DB,E0,7,30,FB,A3,28,8F,ED,A3,20,F4,15,20,F1,
3E,5,D3,E7,3D,D3,E7,C3,A,4,E5,21,5,5,E3,C5,D5,E5,CD,AF,4,E1,D1,C1,E3,CA,A
C,4,2D,C2,90,4,CD,CA,3,2E,5,25,C2,90,4,25,1FE3
1180 DATA E1,9,C9,CD,FC,3,C0,3E,46,CD,DE,4,C0,1E,40,50,14,41,E,E1,78,A7,C
A,D1,4,DB,E0,7,30,FB,A3,28,E,ED,A2,20,F4,15,20,F1,3E,5,D3,E7,3D,D3,E7,C3,
A,4,E5,21,1B,5,77,2C,36,0,2C,72,2C,36,0,2C,1B61
1190 DATA 73,2C,36,1,2C,36,10,2C,36,E,2C,36,FF,21,1B,5,3E,9,CD,9F,3,E1,C9
,3,B0,B,4,0,F,0,0,7,0,9F,BF,DF,FF,88,7,0,F0,F,FF,3,76,3,F1,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,EC8
1200 DATA F3,31,2C,0,21,0,FA,11,2C,0,1,EF,4,ED,B0,CD,78,2,3E,92,D3,DF,3E,
90,D3,E7,3E,D,D3,E7,CD,CA,3,DB,BF,21,13,5,6,80,7E,23,D3,BF,78,D3,BF,4,FE,
87,C2,28,FF,3E,C,D3,E7,AF,D3,BF,3E,58,D3,BF,1F9E
1210 DATA 1,BE,0,78,ED,79,10,FC,21,7C,9,1E,7,ED,A3,20,FC,1D,C2,4D,FF,3E,D
,D3,E7,AF,D3,BF,47,3E,7B,D3,BF,3E,C0,D3,BE,0,0,10,FA,CD,C9,1,C,20,20,20,2
0,20,20,20,20,20,2A,2A,20,46,41,53,54,20,44,49,18CE
1220 DATA 53,4B,20,43,4F,50,59,20,2A,2A,D,D,D,20,20,20,20,20,20,20,20,20,
20,42,79,20,20,4D,69,63,68,61,65,6C,20,48,61,64,72,75,70,D,D,20,20,20,20,
20,20,20,20,28,43,29,20,20,31,39,38,37,20,4D,4A,48,DCC
1230 DATA 20,53,6F,66,74,77,61,72,65,D,8D,C3,69,0,F,B1,32,1A,AB,E1,22,B9,
AE,D1,D9,7C,B7,20,8,78,D9,CD,D1,41,D9,18,3D,D9,7A,BC,38,2,54,67,7B,BD,38,
2,5D,6F,D9,7D,B7,20,1D,78,D9,44,4D,62,CD,D1,41,6B,1CCF
```

Δ

File Copy

By Michael Hadrup

This program provides an extended command which will copy any type of Basic disk file. It will copy ASCII files for those of you who have SEGA WORD 3 and machine language files. The syntax of the extended command is `*C "Filename","New name"`

The new name is optional, but can be used to rename a file when it is copied, or simply to duplicate a file on the same disk.

Type in the small listing and save it with `SAVE "DiskCopy"` Next type in the large listing, which can be saved at any time with `SAVE "DiskCopy.Dta"`

When you have finished save it, type RUN and wait. If you don't get the message "No errors!!" then check the offending line and type RUN again. When you do get the message "No errors!!" insert a disk onto which the code can be saved.

To use the program type `RUN "FileCopy"` to initialise the extended command.

```
1 REM File Copy
2 REM
3 REM By Michael Hadrup
4 REM
10 LOADM"FileCopy.Cde",&HFE00
20 REM
30 REM Start buffer at &HBA00
40 REM
50 POKE&HFEA4,&HBA
60 POKE&HFF08,&HBA
70 REM
80 REM Length of buffer = 17K
90 REM
100 POKE&HFEA7,17
110 POKE&HFF0C,17
120 REM
130 REM Initialise extended command
140 REM
150 CALL&HFE00
160 REM
170 REM NEW
180 REM
190 CALL&H623C
```

```
10 CLS
20 X=&HFE00:RESTORE1000
30 FORN=1000TO1070STEP10:C=0
40 CURSOR0,0:PRINTHEX$(X)
50 FORM=0TO63:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
60 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N:STOP
70 NEXT
80 BEEP:BEEP:PRINT"No errors!!"
90 PRINT:PRINT"Insert DISK and press space"
100 A$=INKEY$:IFA$<>" "THEN100
110 SAVEM"FileCopy.Cde",&HFA00,-1
120 PRINT:PRINT"Saving """;
130 PRINT"FileCopy.Dta""":CALL&H21D4
1000 DATA 21,7,FE,22,D5,9E,C9,FE,1F,C2,48,7,1A,13,FE,C1,20,5,1A,FE,43,28,
5,3E,1F,C3,48,7,CD,3D,84,AF,CD,38,68,DA,5A,FF,3E,1,CD,EB,74,3A,8F,9A,A7,C
A,5A,FF,CD,97,34,3E,D,32,9C,9A,AF,32,9D,9A,21,90,1D43
1010 DATA 9A,CD,17,4B,CD,7E,FF,CD,3A,35,21,90,9A,CD,7,34,DA,60,FF,3E,FF,3
2,2D,B2,1,C,0,9,7E,32,93,FF,23,7E,32,26,B0,CD,3E,84,FE,2C,20,19,CD,3D,84,
AF,CD,38,68,DA,5A,FF,3E,1,CD,EB,74,3A,8F,9A,A7,CA,1E38
1020 DATA 5A,FF,CD,97,34,CD,3E,84,FE,D,28,5,FE,3A,C2,5A,FF,D5,21,2C,B1,11
,2F,B2,1,A0,0,ED,B0,18,3,CD,7E,FF,11,0,B6,D9,6,12,3A,93,FF,D9,CD,6E,FF,CD
,1F,36,DA,66,FF,14,78,4,E6,3,20,F3,3A,93,FF,C6,1F89
1030 DATA 2F,6F,26,B2,7E,32,93,FF,D9,5,FE,C0,30,3,4,10,DA,C5,21,AC,FF,CD,
81,FF,3A,2D,B2,A7,28,28,CD,3A,35,21,90,9A,CD,7,34,D2,69,FF,21,90,9A,CD,54
,34,DA,63,FF,1,C,0,9,22,94,FF,23,3A,26,B0,77,AF,1D2C
1040 DATA 32,96,FF,32,2D,B2,11,0,B6,D9,C1,3E,12,90,47,3A,96,FF,CD,DF,35,3
8,4C,ED,5B,94,FF,22,94,FF,12,32,96,FF,36,FE,D9,CD,6E,FF,CD,85,36,38,39,14
,78,4,E6,3,20,F4,D9,10,D8,3A,93,FF,FE,C0,DA,9F,FE,77,2203
1050 DATA CD,8C,35,E,14,3A,2B,B1,47,11,2B,B0,CD,85,36,38,C,CD,3E,38,D1,1A
,13,C3,16,7,3E,1F,1,3E,47,1,3E,48,1,3E,49,1,3E,4B,1,3E,4C,CD,D3,75,47,E6,
FC,CB,3F,CB,3F,4F,78,E6,3,7,7,3C,47,C9,21,97,1601
1060 DATA FF,CD,3E,38,CD,17,4B,CD,6A,56,A7,28,FA,CD,19,37,D0,18,CA,C3,B5,
B1,89,7,49,6E,73,65,72,74,20,73,6F,75,72,63,65,20,64,69,73,6B,D,0,7,49,6E
,73,65,72,74,20,64,65,73,74,69,6E,61,74,69,6F,6E,20,1AAB
1070 DATA 64,69,73,6B,D,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1B8
```

Print 64 Applications

```
1 REM Print 64 BASIC Program List
2 REM
3 REM By Denver Scott
4 REM
5 REM
10 POKE&HF016,64
20 POKE&HF017,246
30 POKE&HF640,4
40 POKE&HF641,0
50 POKE&HF642,63
60 POKE&HF643,23
70 DATA 21,FF,FF,11,01,00,C3,1F,4B
```

```
70 DATA 21,FF,FF,11,01,00,C3,2F,01
80 RESTORE70
90 FORI=0TO8
100 READA$
110 POKE&HF800+I,VAL("&H"+A$)
120 NEXT
130 SCREEN2,2:CLS
140 CALL&HF000
150 PRINTCHR$(12);
160 CALL&HF800
170 GOTO170
```

Alterations for Cartridge Basic

Print 64 as an Editor

The program below shows how to use Print 64 as an Editor. The program is a shell of a text adventure and it allows input of lines to an input window and printing of messages in an output window. #F016 points to the start of the window data, which is stored as X, Y, X1, Y1, XC, YC. Where (X,Y)-(X1,Y1) is the window and (XC,YC) are the cursor coordinates.

```
10 REM Print 64 Editor
20 REM
30 REM By Michael Hadrup
40 REM
50 REM (C) MJH Software 1987
60 REM
100 REM Store Machine code
110 REM
120 IF(PEEK(&HED00)XORPEEK(&HED01)
)=247THEN180
130 RESTORE2030:FORN=&HED00TO&HED44
140 READA$:POKEN,VAL("&H"+A$):NEXT
150 REM
160 REM Set screen using Normal PRINT
170 REM
180 CALL&HF007
190 SCREEN2,2:COLOR1,3,,1:CLS
200 LINE(0,0)-(15,191),,BF
210 CURSOR40,4:PRINT"Input window"
220 CURSOR160,4:PRINT"Output window"
230 REM
240 REM Set window sizes
250 REM
260 RESTORE1030
270 FORN=0TO2:READX,Y,X1,Y1
280 POKE&HF640+N*6,X
290 POKE&HF641+N*6,Y
300 POKE&HF642+N*6,X1
310 POKE&HF643+N*6,Y1
320 NEXT
330 POKE&HF017,&HF6
340 REM
350 REM Cls each window
360 REM
370 CALL&HF000
380 POKE&HF016,&H40:POKE&HF017,&HF6
390 COLOR15,4:PRINTCHR$(12);CHR$(1);
400 POKE&HF016,&H46:POKE&HF017,&HF6
410 COLOR15,1:PRINTCHR$(12);CHR$(1);
420 POKE&HF016,&H4C:POKE&HF017,&HF6
430 COLOR15,6:PRINTCHR$(12);CHR$(1);
440 PRINT"This is an example of
Print 64 being used as an Editor,
possibly for a   text adventure."
:PRINT"Type QUIT to exit.":PRINT
450 REM
460 REM Main loop
470 REM
480 REM
490 REM Call EDITOR in input window
500 REM
510 POKE&HF016,64:PRINT
520 CALL&HED00
530 REM
540 REM Write line to command window
550 REM
560 POKE&HF016,&H46
570 CALL&HED16
580 REM
590 REM Get first command and print
600 REM in output window
610 REM
620 A$="":N=&HF800+PEEK(&HF640)
630 IFPEEK(N)=32THENN=N+1:GOTO630
640 A=PEEK(N):IFA=0ORA=32THEN660
650 A$=A$+CHR$(A):N=N+1:GOTO640
660 IFA$=""THEN510
670 POKE&HF016,&H4C:PRINT"First com
mand= ":PRINTA$:PRINT
680 IFA$="QUIT"THENSTOP
690 GOTO510
1000 REM
1010 REM Window data
1020 REM
1030 DATA 6,2,31,20
1040 DATA 6,22,31,22
1050 DATA 36,2,61,22
```

```
2000 DATA DD,2A,16,F0,F3,CD,82,F2,FB,7E,5F,CD,24,ED,CD,E,F0,FE,D,C8,18,EE
2010 DATA DD,2A,16,F0,D5,E5,F3,CD,43,F2,FB,E1,D1,C9
2020 DATA 36,7F,CD,1A,ED,6,A,CD,6A,56,A7,20,2,10,F8,F5,73,CD,1A,ED,F1,C0,6
,A,CD,6A,56,A7,C0,10,F9,18,DF
```

```
2020 DATA 36,7F,CD,1A,ED,6,A,CD,6D,43,A7,20,2,10,F8,F5,73,CD,1A,ED,F1,C0,6
,A,CD,6A,56,A7,C0,10,F9,18,DF
```

Alterations for Cartridge Basic Δ

Sorting Program

• As promised here is the sorting program and disection. This program will sort alphabetically each letter of the word. The method used is a modified Shell Sort.

```
10 RESTORE200:READN
20 DIMA$(N)
30 FORI=1TON:READA$(I):NEXT
40 GOSUB70
50 FORI=1TON:PRINTI,A$(I):NEXT
60 END
70 D=N
80 D=INT(D/2):IFD<1THENRETURN
90 FORK=1TON-D
100 J=K
110 IFA$(J)<A$(J+D)THEN180
130 N$=A$(J)
140 A$(J)=A$(J+D)
150 A$(J+D)=N$
160 J=J-D
170 IFJ>=1THEN110
180 NEXTK
190 GOTO80
200 DATA 18
210 DATA SUN,MERCURY,VENUS,EARTH,MOON,MARS,JUPITER,SATURN,URANUS,NEPTUNE,
PLUTO,ASTEROID,MILKY WAY,GALAXY,SPACESHUTTLE,ASTRONAUT,SPACESHIP,STARS
```

Line 10 Gets the number of items to sort

Line 20 Dimensions an array A$ for the number of items

Line 30 Reads each item from the list at Line 210 and stores them in the array A$

Line 40 Calls the SORT routine

Line 50 Prints each item of the sorted array

Line 70 Set the inital distance to the number of items

Line 80 Divides the distance by 2. If this is less than one then the sort has finished

Line 90 K counts each item of the array up to N-D (the N-D is to ensure that when the two items are compared they are both within the array).

Line 100 J is used as a temporary value of K which can be change when looping back. (This is the modified part of the Shell Sort).

Line 110 Compares the (J+Distance) th item with the Jth item. If greater then it goes to 180

Line 130-150 Swap the (J+Distance) th ietm with the Jth item. N$ is used as a temporary store

Line 160 *THE MODIFICATION* J is shifted back by the distance ie it loops back

Line 170 This line checks if the J-Distance points to an item in the array and if it does it loops back to compare the Jth item with (J-Distance) th item.

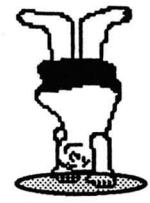
Line 180 Otherwise it continues to bubble up the array using K

Line 190 The distance is halved and the sorting process is repeated at line 80

Line 200 The number of items

Line 210 The list of items.

Graphics Screen FLIP



By David Gladstone

```
1 REM Use CALL &HED00 to FLIP the Graphics Screen
10 CLS
20 X=&HED00:RESTORE1000
30 FORN=1000TO1040STEP10:C=0
40 CURSOR0,0:PRINTHEX$(X)
50 FORM=0TO31:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
60 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N:STOP
70 NEXT
80 BEEP:BEEP:PRINT"No errors!!"
90 PRINT:PRINT"Insert DISK and press space"
100 A$=INKEY$:IFA$<>" "THEN100
110 SAVEM"Flip  .Cde",&HED00,&HED9F
120 PRINT:PRINT"Saving """;
130 PRINT"Flip  .Dta""":CALL&H21D4
140 END
1000 DATA DB,BF,21,FF,17,D9,21,0,0,CD,46,ED,21,FF,1F,D9,21,0,18,11,0,4,CD
,49,ED,21,0,3B,6,20,F3,CD,C76
1010 DATA 65,ED,3E,B6,91,CD,84,ED,23,CD,65,ED,3E,F7,91,CD,84,ED,23,CD,65,
ED,2F,CD,84,ED,23,23,10,E1,FB,21,125D
1020 DATA FF,37,D9,21,0,20,11,0,C,F3,CD,65,ED,D9,CD,65,ED,79,D9,CD,84,ED,
23,79,D9,CD,84,ED,FB,2B,D9,1B,11D4
1030 DATA 7A,B3,20,E5,C9,7D,D3,BF,7C,D3,BF,0,0,0,0,0,DB,BE,8,7C,FE,20,30,
9,8,E,1,1F,CB,11,30,FB,CC9
1040 DATA C9,8,4F,C9,8,7D,D3,BF,7C,F6,40,D3,BF,0,0,0,0,8,D3,BE,0,0,0,0,C9
,0,0,0,0,0,0,0,9A6
```

Δ

Lode Runner Screen

Here is a Lode Runner screen (pattern) designed by J.N. How of Ranui, Auckland. The numbers represent the keys you press while in the Lode Runner screen designer. If you have designed any screens write them out similar to below and send them in.

```
6 0 0 0 0 0 0 0 2 0 0 0 4 4 4 4 4 0 4 0 4 0 0 0 0 0 0 0 0
6 0 0 0 7 8 6 1 6 2 3 2 2 0 0 6 0 0 3 0 3 0 3 0 0 4 4 4 4 3
6 9 0 2 2 1 1 1 6 0 3 0 0 2 2 6 2 2 2 6 1 7 2 1 1 0 8 0 0 3
3 1 1 2 1 2 2 2 2 2 3 2 1 0 0 6 0 0 2 6 1 0 2 1 7 1 0 8 0 3
3 1 7 2 0 0 0 0 0 0 3 2 7 0 0 6 7 7 0 6 1 0 2 0 0 7 1 0 8 3
3 1 7 2 3 1 1 1 1 1 1 1 2 1 6 1 6 2 2 2 6 1 0 2 0 0 0 7 1 0 3
3 1 7 2 3 1 1 1 1 1 1 7 2 0 6 1 6 2 0 0 6 1 0 2 0 0 0 0 7 1 3
3 1 7 2 3 2 2 2 2 2 2 2 0 6 1 6 2 7 7 6 1 0 2 0 0 0 0 0 0 3
3 1 8 2 3 1 0 0 0 0 0 0 0 6 1 6 2 1 2 6 1 0 2 7 6 2 2 2 2 3
3 5 2 2 3 0 6 1 3 1 1 1 1 1 1 6 0 0 0 6 1 0 2 2 2 2 2 2 2 3
2 0 0 0 3 2 1 1 3 0 0 7 0 0 0 1 1 1 1 6 0 0 0 0 0 0 0 0 0 3
```

Δ

BOO BOO'S

Due to some problems with our laser printer the following lines from the **Inverse Characters** program were not printed correctly ...

```
30 PRINT " INVERSE CHARACTER SET ",,,
40 PRINT " REPLACES ENG DIER'S CHARACTERS ",,,
50 PRINT " BY MICHAEL HADRUP ",,,,,,
170 PRINT ,,,"** FINSHED **"
```

Where underlined text is typed with the ENG DIER's key

If you wish to add the inverse characters to your own program then delete lines 10-60 and lines 170-180.

Cartridge Basic Users using LSV may have had some problems when typing in **Print 64.** An additional statement should have been added to the instructions for typing in the program. .... Type in listing 1 followed by listing 3 and carefully make the alterations for Catridge Basic by replacing those lines as shown in listing 4 *except lines 100 to 250. These lines poke the data into REM statements and are not required.* The program can be saved at any time by typing ....

Thanks to someone in Christchurch (whose name I have forgotten) for helping to locate this problem after a long toll call.

Line 180 of the MC Editor (Figure 1.2) should have read `150 PRINT:GOTO50`

and of course a few spelling mistakes.

In the next issue

Tank Battle - A machine code game

for 16K, 32K and Disk Basic Users

Compressed picture drawing

for 32K and Disk Basic Users

More on machine code and BASIC programming

Due out in March

Poseidon Software

NZ SEGA DISTRIBUTORS

FREEPOST 243 P.O. BOX 277 TOKOROA NEW ZEALAND

Specials

Aerobat	*32K only*	$21.00
Burgular Bill	*16K and 32K*	$19.00
Carverns of Karanor	*16K and 32K*	$21.00
Vortex Blaster	*32K only*	$15.00
Vermin Invaders	*16K and 32K*	$15.00
Castle of Fear	*32K only*	$12.00
Castaway	*32K only*	$12.00
Orb of Fear	*32K only*	$12.00
Burgular Bill	*Disk version*	$33.00
Caverns of Karanor	*Disk version*	$33.00
Michael Howard's - More than 50 Programs		$5.00
Book and Tape - Teach Yourself BASIC Programming		$6.00
LSV, Print 64 and Pattern Paint	*Disk / Tape versions*	$15.00
Magazine programs on cassette		$20.00
Machine Code Summary Sheets		$1.00

	Australia	New Zealand	
Magazine subscription (6 issues)	A$26	NZ$25	GST incl
One issue	A$7	NZ$6	

Name _____ **Phone** _____

Address _____

_____

Item	No. of items	$ Total

Add Postage **$2.50**

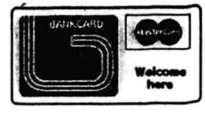
Payment by (circle one) **Total Enclosed $**_____

Cheque Cash Bankcard Visa

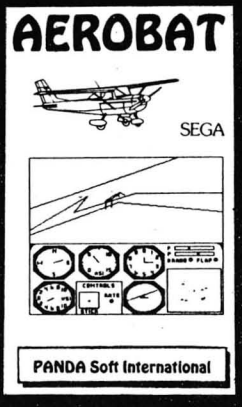
Orders over $20.00 only

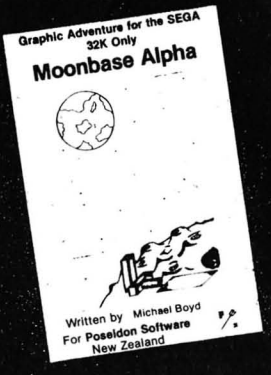
Credit Card No. _____

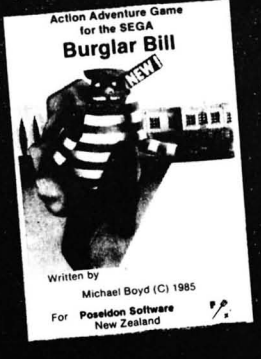
Signed _____ Expires / /



GET THE MOST OUT OF YOUR SEGA







Poseidon Software

NZ SEGA DISTRIBUTORS

P.O. BOX 277 TOKOROA NEW ZEALAND