

**VERYSOFT**

**presents**

**COMPARATOR**

**Version 1.0**

**Programmed by Martyn Davis**

## **COMPARATOR V1.0**

**Copyright Verysoft 1993**

**All rights reserved. It is illegal to reproduce or transmit either this manual or the accompanying computer program in any form without the written permission of the copyright holder. Software piracy is theft.**

**If you have written any software that you believe could be marketed commercially, please send a copy to Verysoft at the address below. If we think it is of good enough quality, we can handle the marketing for you.**

**Verysoft.  
450 Roughwood Road,  
Wingfield Estate,  
Rotherham,  
South Yorkshire  
S61 4LA**

## CONTENTS

<b>PAGE</b>	
2	<b>Introduction</b>
3	<b>General Control</b>
4	<b>COMPARE</b>
11	<b>FILE MENU</b>
13	<b>UTILITIES MENU</b>
15	<b>EDITOR MENU</b>
19	<b>PRINT MENU</b>
20	<b>SEARCH MENU</b>
22	<b>PRINTER MENU</b>
23	<b>Appendix A: Key Summary</b>

## **INTRODUCTION**

COMPARATOR is a program to help you find pokes, specifically for use in finding Multiface pokes. Included in this program are utilities that will help you edit/examine programs. To use COMPARATOR you must have a MULTIFACE, although you can load in and edit programs in the standard AMSDOS format you will not be able to use the COMPARE option without a MULTIFACE.

When using COMPARATOR you will require a blank formatted disk to save the Multiface files on. The Compare option requires you to save out the game code three times so the blank disk must be totally blank. From now on I will refer to this blank disk as the WORK DISK.

To use COMPARATOR fully you do need to have a reasonable knowledge of machine code.

### **IMPORTANT**

+++++

Do not use this program to infringe copyright laws. Remember back-up copies of programs are for personal and archival purposes only.

COMPARATOR works only on disk based machines with 128K of memory. It is compatible with one or two disk drives.

**ALL EXTERNAL ROMS  
MUST BE DISABLED BEFORE RUNNING THIS PROGRAM**

## **GENERAL CONTROL AND OPERATION**

All the options in this program are accessed through a series of menus. If an option takes you to a further menu it will be identified in the menu by '>>'.

To select an option move the cursor bar up or down with the cursor up/down keys. To execute your selection press either the space bar or the return/enter keys.

Once in a feature to escape press the escape key or 'E'. In the Multi-View option press 'Q' to quit.

The first menu you see after booting up is the Main menu. From here you access all the other areas of the program.

### **MAIN**

<b>Compare</b>	
<b>File</b>	>>
<b>Editor</b>	>>
<b>Printer</b>	>>

## COMPARE

This is where you find all the locations that contain the life counters etc.

COMPARATOR works by comparing the entire program three times. You can search for any series of three bytes or a series that increases or decreases by 1.

For example a life counter will usually decrease by 1 every time you loose a life. To find this location you would search for a series of bytes that decreases each time by 1. COMPARATOR will automatically search for any references to this location, it's then up to you to then decide which part of the program to alter.

### FOLLOW THIS EXAMPLE:

Suppose you wanted to gain infinite lives in a shoot-em-up.

#### Step 1

Prepare a blank work disk and have it ready.

#### Step 2

Run the shoot-em-up program.

Stop the program using the Multiface as soon as you start playing the game, after the title screens etc.

Insert your work disk into the drive.

Select the (C)lear option to erase the second banks.

(COMPARATOR only works on games that don't use the extra memory).

Select the save option in the Multiface.

Save the program calling it FILEA (You must ALWAYS call it this)

**Make a note of the number of lives left at this stage.**

**Return to the game. Play it until you loose a life then stop it again.**

**Save the program calling it FILEB. Again make a note of the number of lives remaining.**

**Again return to the program and play it until you loose another life.**

**Stop the program again using the Multiface and save it calling it FILEC. Again making a note of the number of lives left.**

**You should now have a work disk containing the files:**

**FILEA.BIN  
FILEA1.BIN  
FILEA2.BIN  
FILEA3.BIN  
FILEA4.BIN  
FILEB.BIN  
FILEB1.BIN  
FILEB2.BIN  
FILEB3.BIN  
FILEB4.BIN  
FILEC.BIN  
FILEC1.BIN  
FILEC2.BIN  
FILEC3.BIN  
FILEC4.BIN**

### Step 3

Reset the computer. Either switch it on & off (removing any disks first!) or use the reset button on the Multiface.

Load and run COMPARATOR.

Select the COMPARE option from the main menu.

A series of requests will then appear:

#### No of Parts 3 or 4

This is included to save time. Most games use the area of memory #C000 to #FFFF for the screen memory (the default) as we are only wanting to look in the program for locations that change with regards to lives etc. there is no need to load part4 of the saved files. So initially enter 3 at this prompt.

If however you suspect that the default screen memory is used to store program data/code instead of the screen image then select 4 in response to this request.

#### Select type of search

##### 1= Increment

This will search the saved files for a series of bytes that increase by 1 in the same location.

i.e. Location #9000 in FILEA contains 4

Location #9000 in FILEB contains 5

Location #9000 in FILEC contains 6



## 2=Decrement

This will search the saved files for a series of bytes that decrease by 1 in the same location.

i.e. Location #7000 in FILEA contains 6

Location #7000 in FILEB contains 5

Location #7000 in FILEC contains 4

## 3=Specific

This will bring up another requester that asks you to enter any 3 bytes to the search for. You will be asked first for the byte to look for in FILEA then a byte to look for in FILEB then FILEC.

As we are looking for a life counter that we know decreases by 1 and we know the number of lives at each stage (you did make a note didn't you?) we can use search type 3.

So select number 3 then enter the values as follows:

For FILEA enter the number of lives left when you saved FILEA.

For FILEB enter the number of lives left when you saved FILEB.

For FILEC enter the number of lives left when you saved FILEC.

Easy isn't it?

Insert your work disk when prompted and watch COMPARATOR work.

A list of information will scroll past telling you exactly what's happening. Firstly COMPARATOR loads in the first part of each file i.e. FILEA1.BIN, FILEB1.BIN AND FILEC1.BIN. Then it searches through each part for the values you specified, if it finds them it will print something like [#14F5,05,04,03]. The first number is the location (#1F45) that in each of the files contains the following bytes 05 in FILEA1.BIN, 04 in FILEB1.BIN and 03 in FILEC1.BIN, the numbers are in hexadecimal by the way.

This stage is repeated for the second and third parts (and the 4th part if selected).

Stage 2 will then load all four parts of FILEA.

Stage 3 then searches for any reference to the locations found. It may for example come up with:

Reference to #14F5 at #6200

This means that a legal instruction including the bytes #F5,#14 (LOW BYTE,HIGH BYTE REMEMBER) was found with the bytes located at #6200.

#### Step 4

This is where your skill in understanding machine code will come in handy. Don't worry if you're not very experienced in Z80 code there are only a few instructions you'll need to recognise to come up with a final location to poke.

Go to the EDITORS menu and select MULTI-VIEW, five windows will appear on screen four vertical and one (the command line) one line high at the bottom.

Using our example of Reference to #14f5 found at #6200.

If you press L a list of all the locations that have been found will then appear including #14f5.

Press R, then you will be prompted to enter a location address, enter one of the listed locations. Either 'No References' or a list of references will then appear. If 'No References' appears then select another location. Using our example #6200 should appear.

Press W then 2 to select window 2.

Now we need to enter a start address to look at the program from. Press S and then enter a start address about 20 bytes less than the reference address you are looking for. So in our example enter #61F0.

Now press D to start a Disassembly listing, press a key to stop it when you can see the area you are interested in, i.e. #6200. Remember that #6200 is the address of the first byte of the location not the address of the instruction containing the location. So the address to look for will be at least 1 byte less than #6200 i.e. #61FF you should easily spot the location anyhow.

What we now need to look for is the instruction that decrements our life counter by 1 each time we loose a life. Unfortunately there is no one standard way of doing this but usually it will be of the form:

```
LD A,(location) ;Where location is the address containing the life
DEC A           ;counter
LD (location),A
```

The DEC A is the instruction to remove here. Make a note of the address of the instruction DEC A.

If you can't find a DEC(rement) instruction then look for a SUB(tract) instruction (1 byte) or a SUB 1 instruction (2 bytes) or SBC instruction (check how many bytes by counting em!). When it comes to poking the game you will have to remove the bytes containing the DEC or SBC instructions etc. by poking the address of the instruction with 0.

Other things to check for are losing all your lives therefore checking for instructions that check for zero in some way such as:

```
LD A,(location) LD A,(location) LD HL,(location)
DEC A           DEC HL           OR L
CP 0            AND A            LD A,H
JP Z,address   JP Z,address
LD (location),A LD (location),A
JP Z,address   LD (location),HL
```

Other variants could be:

```
LD HL,counter LD HL,counter
LD A,(HL)     DEC (HL)
DEC A
etc., etc.
```

All load a register (A) or register pair (HL) with the number of lives (stored in 'location') then decrement the value and then check for the result equalling 0 if it does then they jump of to another address.

If you feel totally baffled by all this then all I can suggest is getting hold of a good book on beginners machine code. Don't be afraid to use this program to examine other peoples code, its how most people learn machine code -including me! Remember what you're looking for are instructions that in some way take 1 off your life counter.

When you feel that you've found the instructions that change the life counter, make a note of the addresses that contain these instructions exit COMPARATOR, run the program, stop it using the Multiface and using the Multiface poke the addresses you've noted with 0 then restart the program, hopefully when you now get killed you wont loose a life. If nothing different happens then try again with a different set of addresses, there will usually be more than one reference to a life counter, for example the initial start value has to be set up somewhere, 1 could be added to it for a bonus life etc. these are all things that you should look for.

If you've think that you have found the location that contains the life counter but are not sure then make a note of the location of the life counter, exit COMPARATOR, restart the game stop it using the Multiface, using the tool commands poke the location with any number, say 20, play the game and loose a life if the life counter drops to 19 then you've definitely found the right location.

Using this location you could use COMPARATOR's search functions to find where the number of lives is originally set up then you could alter this value so that you always start the game with say 20 lives!

## FILE MENU

### Options

**Load Multiface**  
**Save Multiface**  
**Load Program**  
**Save Program**  
**Load Data**  
**Save Data**  
**Utilities >>**  
**Clear Memory**  
**EXIT**

All the programs file handling features are accessed via this menu and the subsidiary Utilities menu.

### Load Multiface

Use this program to load in a program previously saved out with the use of a Multiface.

This program only works, fully, on 64k games. i.e. It only loads in the first 4 blocks of a Multiface saved game. On programs that use the second banks of memory (128k programs) then you will be able to load in the first 4 blocks.

A Multiface saves out either 5 or 9 files depending on the size of program, one file is the loader the other 4 or 8 are 16k blocks of the computers memory. If a program uses only the normal 64K of memory then when you save it from the Multiface press 'C' to clear the extra banks. (See Multiface instructions).

### Save Multiface

Selecting this option will compress the data and save it out in Multiface format so that you can run it again at a later date. This option only works on programs loaded in with the Load Multiface option.

No protection is removed by COMPARATOR. You are given the opportunity to rename the file. At the Change Name prompt type 'Y' to change the name or 'N' not to, the press enter.

### Load Program

This option is used to load a standard Amsdos file. The file will be loaded automatically to the address it was original saved from. The files' parameters are preserved for use with the Save Program option outlined below.

### Save Program

Saves a program previously loaded with the above option. The file will be automatically saved with the correct file length, start address and execution address.

### Load Data

If you wish to specify where in memory you wish to load a file then you should use this option. Any address is available from #0000 to #FFFF, the full 64k is available to the user.

### Save Data

If you wish to save a section of memory then this option is to be selected. you will be requested for a start address, end address and execution address.

### Utilities >>

Selects the utilities menu. (See next page).

### Clear Memory

Warning:

Be very careful when using this option. It irreversibly clears the user memory.

## UTILITIES

### Options

**Catalogue**  
**Erase**  
**Erase Backups**  
**Rename**  
**Information**  
**Drive A } Only available on**  
**Drive B } dual drive machines.**  
**EXIT**

### Catalogue

Displays a directory of the disk in the currently selected drive.

### Erase

Displays a directory of the disk in the currently selected drive then prompts for a file name to erase. Standard Amsdos wild cards are allowed (e.g. ? and \*).

### Erase Backups

Deletes all files with the extension .BAK. The equivalent of |ERA,\*\*.BAK".

### Rename

Renames a file. Prompts for a new filename then the old filename (the one to rename) and then renames the file.

## Information

Displays a directory of the current disk and then prompts for a file name. The following information regarding the file is then displayed.

User number  
File Name  
File type  
Start Address  
File Length  
Execution Address

## Drive A

(Only available on dual drive machines).

Makes drive A the current drive.

## Drive B

(Only available on dual drive machines).

Makes drive B the current drive.

## Exit

Returns to the File menu.



## EDITOR

### Options

**Ascii Dump**  
**Hex & Ascii Dump**  
**Edit**  
**Disassemble**  
**Multi View**  
**View Memory**  
**Print >>**  
**Search >>**  
**View Mface Info**

### Ascii Dump

#### **Keys:**

**Any key pauses - Same key again exits - Different key continues**  
**Escape or E Exits**

**This option allows you to examine the computers memory as a series of characters, thereby revealing any messages contained in a program.**

**You are requested for a start address, an end address, the text type and whether the top bit should be set or reset.**

**Text Type, chooses wether to display characters between 0 and 127 (Text Type 0) or characters between 0 and 255 (Text Type 1).**

**Top bit, when reset will set the top bit of a character byte to 0 therefore forcing the character into the range 0-127. When set the character remains in the range 0-255.**

**It is useful to be able to reset the top bit when looking for messages, for the sake of clarity. Some, if not most, programmers will mark the last character in a string by setting the top bit of the last byte. Resetting the top bit makes messages easier to read.**

## Hex & Ascii Dump

### Keys:

Any key pauses - Same key again exits - Different key continues  
Escape or E Exits

Dumps to the screen a Hexadecimal and Ascii Dump of the computer memory between two specified addresses.

## Edit

### Keys:

\ (Back slash) To start entering text (border changes to red).  
Return To finish entering text.  
. (Full stop) To enter a byte in hex (border changes to orange).

Cursor keys Move cursors  
Ctrl+cursor up/down Move up/down one page.

Any key pauses - Same key again exits - Different key continues  
Escape or E Exits

This is where you can alter the memory contents. Either by typing in a hexadecimal number or a typing in a string.

### For example:

Assume you have been examining a game and you have found the high score table data, and you wish to place your name in the top spot. Move the cursor to the start of the name you wish to overwrite. Then press \ (next to the right shift key), the border will turn red indicating that anything you now type will be entered into the code. Type your name, finish by pressing return. Anything altered will be shown in inverse. That's all there is to it.

## Disassemble

### Keys:

Any key pauses - Same key again exits - Different key continues  
Escape or E Exits

This is the main disassembly option, it gives you a choice of 3 different styles of output:

Type 1: Address and instruction only.

i.e.: #0000 LD HL,#4049

Type 2: Address, Ascii and instruction.

i.e.: #0000 !I@ LD HL,#4049

Type 3: Address, hex, Ascii and instruction.

i.e.: #0000 21 49 40 !I@ LD HL,#4049

The disassembler will disassemble all the Z80 instructions including all the undocumented ones.

## Multi-View

### Keys:

W	Select window (1 to 4)
S	Enter start address
E	Enter end address
D	Display a disassembler listing
A	Display an Ascii dump
H	Display a hexadecimal dump
L	Display a list of all locations found using COMPARE
R	Display a list of all references to above location
?	Display commands
Q	Quit back to Editor menu

In Multi-view you have four windows available, each of which can display either a disassembly, an Ascii dump or a hex dump. It is very useful when tracing machine code to be able to have more than one listing on screen at any time. To view an area of memory first set the (S)tart address then the (E)nd address then the command you require.

### View Memory

Display a graphical representation of the contents of memory between locations #0000 to #C000.

As you use this option more and more you will learn to recognise which areas are program code and which are graphic data etc. Generally program code will appear more solid than graphic data which will appear more patchy.

Print >>

Enter Print Menu. (See next page).

Search >>

Enter Search menu. (See later page).

### View Mface Info

Displays information as to the status of the computer when the program was initially stopped, with the Multiface, and then saved.

The information displayed is:

Z80 registers

6845CRTC registers

Palette

Multiface additional info. (see Multiface instructions)

Exit

Return back to Main Menu

## PRINT

### Options

Ascii Dump  
Hex & Ascii  
Disassemble  
Printer >>  
EXIT

### Ascii Dump + Hex & Ascii + Disassemble

Do relevant dumps to the printer as well as to the screen.

Note:

Only characters with an Ascii Value between 32 and 127 can be printed from an Amstrad computer.

Printer >>

Enter Printer Menu. (See later page).

EXIT

Return to Editor menu.

## SEARCH

### Options

**Wild Byte**  
**Byte**  
**String**  
**EXIT**

### Wild Byte

Allows the user to search through memory for a series of bytes allowing for 'wild bytes' to be searched.

A wild byte is specified by entering a 0 in the search sequence.

For example:

Say you are searching a program for the instructions that decrement a life, which could possibly be something like this:

```
LD A,(addr)
DEC A
LD (addr),A
```

As you will not know the address that stores the life value, you need to search for all occurrences of the instructions LD A,(addr):DEC A:LD (addr),A, specifying the address as wild bytes:

e.g. Search for: #3A,000,000,#3D,#32,000,000

This will find all the occurrences of bytes that fit the above routine.

To enter the bytes, simply type them in, in either hexadecimal or decimal, separating each byte by pressing return. Press return again after the last byte to end the sequence.

## Byte

Allows the user to search for a specified sequence of bytes. No Wild cards are allowed so that searching for :

#3A,000,000,#FE,000

Will only find that particular sequence of bytes.

## String

To search for a series of Ascii characters or a string then use this option. Wild bytes are allowed and are identified in the search string by a '?'.  
Wild bytes are allowed and are identified in the search string by a '?'.

For example:

Searching for: LI?E

will find; LIFE LIKE and LIVE

Note all search's are case sensitive, i.e. they treat a and A as different characters.

## EXIT

Returns to the Editor Menu.

## PRINTER

### Options

**Reset Printer**  
**Select Condensed**  
**Output Ctrl Codes**  
**Title**  
**EXIT**

### Reset Printer

Resets the printer to its initial default state.

### Select Condensed

Sends the control codes 27,15 to the printer to select the condensed typeface.

### Output Ctrl Codes

Allows the user to output any control codes to the printer. To enter the codes type them in separating each one by pressing return and end by pressing return again.

### Title

Allows the user to input a string which will be immediately output to the printer.

### Exit

Return to the previous menu.



## COMPARATOR KEY SUMMARY

### ALL MENUS:

Move cursor: Cursor up/down  
Select option: Space or Return or Enter

### EDITOR

Line up/down: Cursor up/down  
Page up/down: Control+cursor up/down  
Enter text: \ (Enter to end)  
Enter byte: .  
Exit: Escape/E

### DISASSEMBLER

Type 1: #0000 LD (#4544),A  
Type 2: #0000 2DE LD (#4544),A  
Type 3: #0000 32 44 45 2DE LD (#4544),A

Any key to pause any different key to resume.  
Same key twice exits.

### MULTI-VIEW

Select Window (1-4): W  
Enter start address: S  
Enter end address: E  
Disassemble: D  
Hexadecimal dump: H  
Ascii character dump: A  
List References: R  
List Locations: L  
Help: ?  
Quit: Q

## **PRINTER**

**Entering control codes:**

**Enter codes separated by pressing return, return/enter to end.**

## **SEARCH**

**Entering Search bytes:**

**Enter codes separated by pressing return, return/enter to end.**