

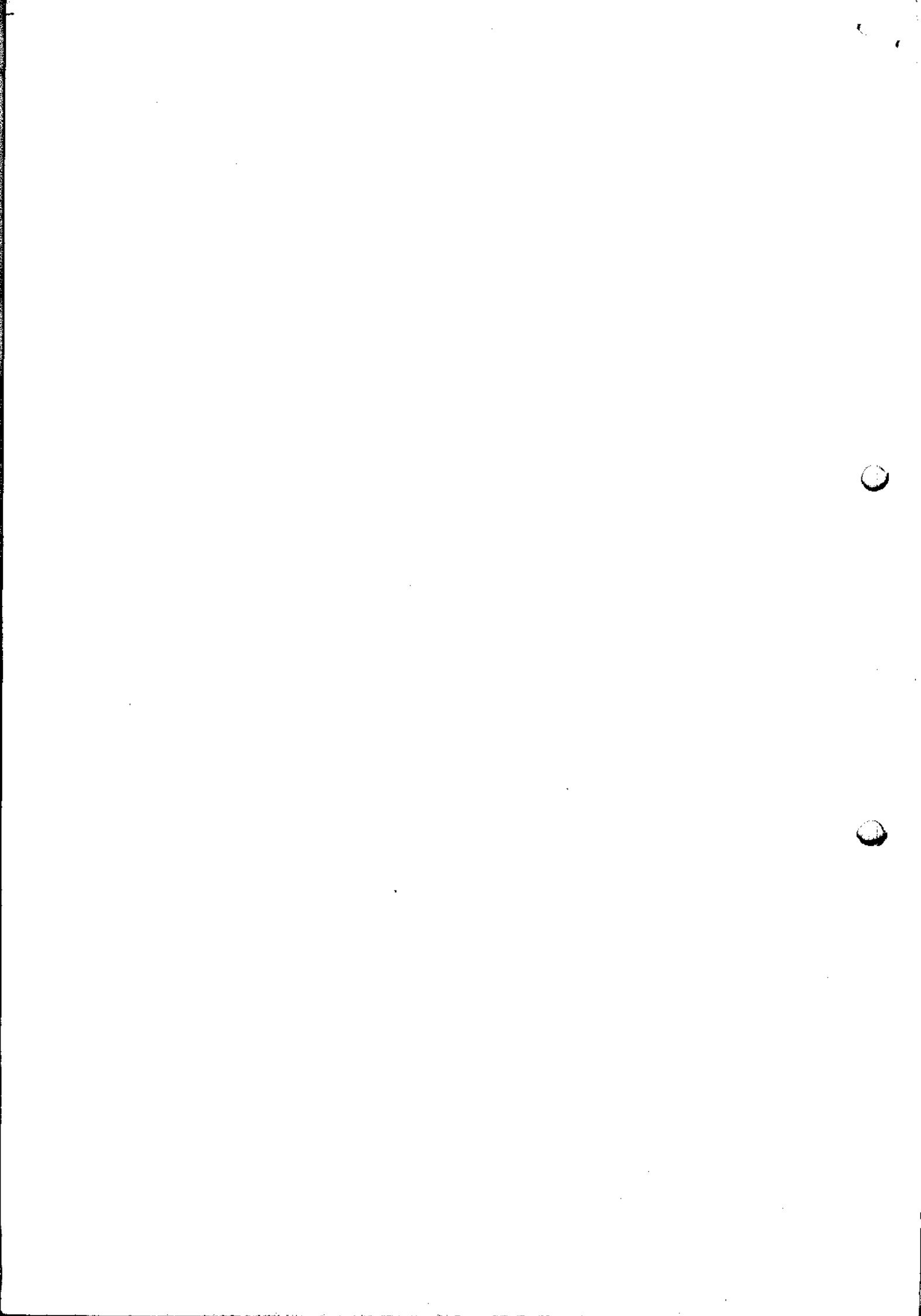
nm

nascom microcomputers limited

NAS-DIS

**disassembler
for NASCOM
microcomputers**

PART NO.	DATE	ISSUE
230-300	21-04-80	2



INTERACTIVE OPTION SUMMARY SHEET

Source form only
Tape output (or printer)
ZEAP file to memory
Xreference listing
Labels
Paginated listing
Delay at end-of-line
Range limitation
Unmodified RST handling

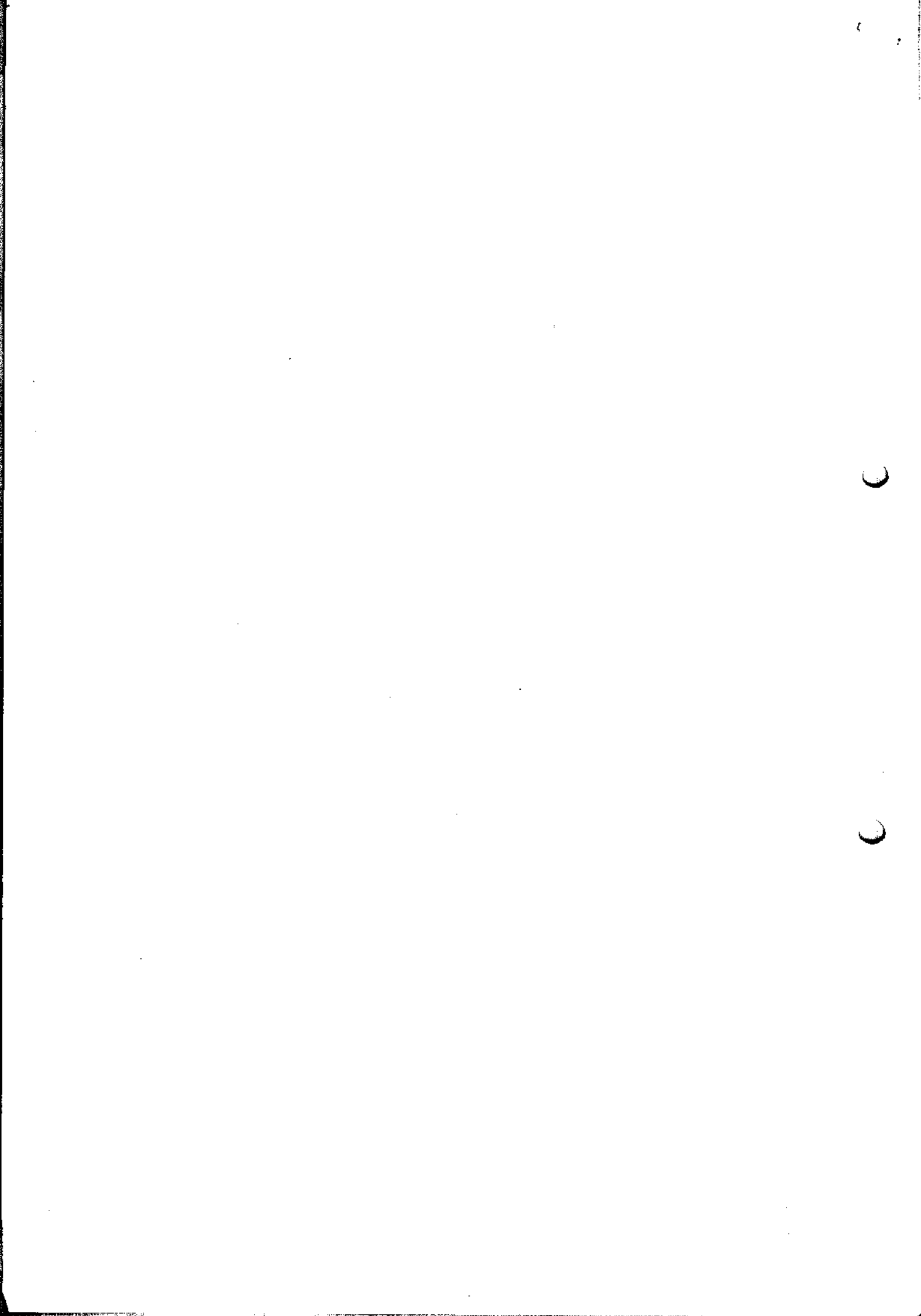
Options for source file

ZEAP file:- Z (RU)
Tape file:- STLD (RU)

Full listing to printer

TLP(X) (RU)

Note: All address pairs entered are inclusive. ie Start End.



CONTENTS

	page
1. INTRODUCTION	2
2. MEMORY REQUIREMENTS	3
3. RUNNING THE DIS-ASSEMBLER	4
3.1 Simple mode	4
3.2 Interactive mode	5
4. NOTES ON USING THE DIS-ASSEMBLER	9
5. PRODUCING SOURCE FILES	10
5.1 ZEAP files direct to memory	10
6. USING A PRINTER	11
APPENDICES	
A IDIOSYNCRASIES AND BUGS	12
B MODIFICATIONS	13
C WRITING YOUR OWN PROGRAM FOR REVAS	14
D SAMPLE OUTPUT - SIMPLE MODE	15
E SAMPLE OUTPUT - INTERACTIVE MODE	16
F PROGRAM LISTINGS	18

1. INTRODUCTION

An assembler takes in instructions such as LD (HL),A INC HL CPDR and converts them to the corresponding machine code (eg 77 23 ED B1). A dis-assembler, or reverse assembler, performs the complementary function. It translates machine code back to mnemonics.

The dis-assembler is a useful tool. It lets you examine programs in memory in an easily readable format. You can produce "listings" of programs for which you do not have the source, and, working from them, you can make changes to programs to tailor them more to your own personal requirements. If the changes you wish to make are extensive, you can generate a source file for the program, which can be edited and re-assembled, perhaps with new origins.

The dis-assembler can be run in two modes, a simple one for direct examination of memory, and a more complex one for producing labelled listings and source files. You will probably find yourself using the simple mode often as it provides a very useful extension to the monitor in that it produces easily readable tabulations of programs in memory. The full interactive mode is of use when you buy a new piece of software for which you wish to produce a source listing/file, and, as a result, is likely to be used less frequently.

One thing must be realised from the outset. An original assembly language program contains a lot of information that is only of use to the programmer. These are the comments and label names which are used to document the program and to make it readable. The dis-assembler has no way of re-inserting these as they have no representation in the machine code, but it can generate a form of label and add a little more information to its output which is of help in following a listing. (When you come to examine a large dis-assembled listing you will realise the value of plentiful comments and informative labels!).

The format of the output produced by the dis-assembler is:-

```
AAAA OP OP OP OP LLLLL MMMM OPERAND ;CCCC
```

Where:AAAA is the address of the instruction.
OP are the bytes (1 to 4) making up the instruction
LLLLL is a label (if present).
MMMM is a 2 to 4 character mnemonic.
OPERAND is the operand field.(eg A,(HL) or HL,£1025)
CCCC are the 7-bit ASCII characters corresponding to bytes 1 to 4 of the instruction. All control characters appear as a period.

The kernel of the package is the reverse assembler subroutine REVAS, and if you wish to write your own programs to drive it the necessary information is given in appendix C. (Don't just sit there thinkingif only it..... : Sit down and write your own control program to do what you want).

REVAS has been written to produce ZEAP compatible mnemonics, and in addition, recognises three special instructions. These are the three Restart codes D7,DF,EF which, if they occur in a program that runs with Nas-Sys, are followed by one or more parameters. (The routines are RCAL, SCAL, and PRS). REVAS lists the parameters directly after printing the restart instruction, and does not attempt to interpret them as program.

2. MEMORY REQUIREMENTS

NAS-DIS is supplied in three 2708s and is coded to run at address C400. When used with the NASCOM2 it is expected that these will be inserted into the spare sockets on the main processor board. Refer to the N2 hardware manual and check that you have the header plug wired correctly, the strapping fields (LKB2-4) correctly set, and LSW1 7&8 suitably set.

Assuming that you have the eight RAM/EPROM sockets decoded for address range C000-DFFF insert the three EPROMs as follows, making sure that you have them correctly orientated:-

NASDIS-1 in socket IC36
NASDIS-2 in socket IC37
NASDIS-3 in socket IC38

Remember that EPROMs are MOS devices and can be damaged by static electricity, so take reasonable care while you are handling them.

When running in the full interactive mode the program uses workspace from E00 to EC2. When only run in the simple mode the workspace used is E00 to E43. In addition some memory is required for the dis-assembler's stack. NAS-DIS does not set the stack pointer at all, it just accepts the value passed to it on entry. Conventionally the program is run by the Nas-sys "E" command, and in this case the stack area used is from FFF downwards (towards 0).

In the simple mode the stack can reach FE0, while in the interactive mode it can reach FC8. However in the interactive mode any address pairs entered in response to the prompt "Data Areas?" are saved on the stack, so the stack depth will increase by four bytes for every pair entered. This means that there is room for about 60 address pairs before the dis-assembler is likely to overwrite its own workspace, and about 36 address pairs before it overwrites ZEAP's workspace (see section 5.1).

C000
C400
DFFF

C4000

3. RUNNING THE DIS-ASSEMBLER

The dis-assembler has two modes of operation, a simple one for direct examination of areas of memory, and a more complex one for producing labelled listings and/or assembler files.

REVAS is run by using the monitor E command. Three possible formats are acceptable:

```
E C400
E C400 XXXX
E C400 XXXX YY
```

In the first instance, where no parameters are passed to the program, the dis-assembler runs in its interactive mode and will prompt you for further information (see on). Where one or two extra parameters are included in the command line the simple mode is executed.

3.1 Simple mode

Where both parameters are included in the command line the dis-assembler will dis-assemble code starting at address XXXX and will pause after listing YY lines in mnemonic-type format. If the second extra parameter is omitted then a default value of 1 line is assumed.

Depressing any key on the keyboard will produce the next YY lines, return to the monitor only occurring when "Escape" (shift/enter) is typed.

This mode provides an easy way of displaying areas of memory in a mnemonic format and, amongst other uses, provides an admirable way of checking that any Hex codes you have entered manually to memory are actually the instructions you intended.

In this mode the dis-assembler assumes that the three RST instructions D7,DF and EF are RCAL,SCAL and PRS.

3.2 INTERACTIVE MODE

You will be prompted first to enter the options to be used for the dis-assembly as shown below:

Options (STZXLPDRU) -

You select options by entering the corresponding letters followed by a carriage return. If you require more than one option, enter them all before the carriage return. Any unrecognised character will cause the question to be repeated. In this case all the options previously entered will be cleared.

The option letters have the following significance:

OPTION X (Cross-reference)

A listing of all referenced addresses and the corresponding addresses of the instructions that refer to them will be produced at the end of the dis-assembly. See the sample output for details.

OPTION L (Labelled)

The dis-assembler outputs a label symbol to the left of any instruction which is referred to elsewhere in the dis-assembled program. The form of the label is an L followed by the address of the label. See the sample output.

OPTION T (Tape/typed output)

Selecting this option causes the screen output to be echoed to the serial output port where it could be printed, or it could be recorded on tape for re-input later (say to an assembler). See also options D,L and S which may be relevant if option T is selected.

OPTION S (Source form only)

This option is useful if you want to re-input the data as source for an assembler, when only the text and label references are required. It causes the listing of the location of each instruction and the code bytes of each instruction to be omitted. Also all non-significant blanks are removed from the output and, except in the case of DEFB lines, the comment field is deleted. See also options D and T which may be relevant.

OPTION D (Delay)

This option introduces a short delay after each line is printed. It is used for printers that require a small delay at the end a line, and also when producing a source file to tape (see on).

OPTION Z (ZEAP source file)

This option causes a ZEAP compatible source file to be produced directly in memory complete with all necessary line numbers and pointers. This is covered in more detail in section 5.

OPTION P (Paginated output)

This option formats the output into pages with a title at the top of each page and a page number.

OPTION R (Range limitation)

This option allows you to restrict the range of the dis-assembly listing whilst actually dis-assembling a larger area of memory. (If you wish to modify a small part of a large program this allows you to list only the area of interest whilst ensuring that any possible entry points in it are labelled).

OPTION U (Unmodified RST handling)

This option cancels the special handling of the three RST instructions - RCAL, SCAL and PRS. It would be used if you are dis-assembling a program that was written to run with a monitor that used the RST instructions in a different manner to Nas-sys.

If you selected options L or X you will now see the prompt:

Symbol table area?

The dis-assembler now wants the start and finish address of some memory that it can use as temporary workspace to hold the symbol table that it has to generate. Efficient use is made of the memory allocated, but if insufficient memory is specified an error message will appear during Pass 1 (see on).

Enter the two addresses followed by a carriage return.

If you selected option Z you will see the prompt:

ZEAP file area?

This prompt appears in place of the one above. The program uses the top of this area to hold its symbol table, and stores the ZEAP source file at the bottom. The addresses entered here are also passed on to ZEAP. In this instance more memory is required to hold the file in addition to the symbol table, and so overflow may also occur during pass 2, while the source file is actually being stored in the file area (see section 5.1).

If you selected option P you will now see the prompt:

Title?

Enter the title you wish to appear on the listing. At this time you also have the opportunity to redefine the lines/page and lines between pages figures used by the program. If you wish to do this you should start the title line:-

=AABB T.....T

Where AA is the hexadecimal number of lines between pages+1

and BB is the hexadecimal number of lines/page

and T.....T is the title.

The total number of lines per page are AA+BB+1.

You will now see the prompt:

What on?

The program is now asking you for the start and finish address of the area of memory you wish to dis-assemble. You also have the option of entering a third address as well. If you only enter two addresses REVAS will do a straightforward dis-assembly of the area specified. However if the third address is entered, the program will still dis-assemble the same area of memory, but the addresses on the listing will start with the third address. ie The area of memory will be dis-assembled as though it were actually located at the third address entered, rather than the first.

This allows you to correctly dis-assemble any program, even if it is not at its execution address. (That area of memory may already be in use in your system, or may even be absent!).

Enter the two (or three) addresses separated by spaces and finish with a carriage return.

Any addresses you enter from now on refer to the listing address. This address will differ from the memory address if you responded with three parameters to the last request.

If you selected option R you will now see the prompt:

Listing range?

Enter the start and finish address of the desired range.

NOTE If the addresses do not overlap the full dis-assembly range then nothing will be listed except the cross-reference table (assuming the X option has also been selected).

Data areas?

DATA AREAS in this context refers to areas of data within the bounds of the START and FINISH addresses which you do not want to be dis-assembled into instruction format. Typically they would be data tables, messages etc. If you know of such areas in the program enter a pair of start and finish addresses for each area, one pair per line. The address pairs should be entered in ascending order. You will get an error message if the pair of addresses are unacceptable and the cursor will be positioned to the start of the line in error. Enter a carriage return on its own to terminate the data area list. If you notice that you have entered an incorrect address in a previous entry typing "-" (followed by enter) will delete the last entry. This can be used repeatedly until you reach the line in error.

NOTE. The parameters following the restart calls RCAL/SCAL/PRS are dealt with automatically by REVAS and they should not be identified here, although remember that this may have been cancelled by option U.

Finally the program prompts:

Go?

The program is waiting for you to press a key on the keyboard before it runs. If you press the spacebar the program will start to run in a single step mode, any other key resets the single step flag.

If you have selected options L, Z or X the blinking cursor will disappear and there will be a delay of a few seconds or more while the dis-assembler makes a first pass of the code building up its symbol table. In the event of it running out of memory for the symbol table the message "Overflow" will appear on the screen along with the line currently being dis-assembled displayed below it. This shows you how far the program has got and should let you make a reasonable estimate of how much more memory is required for the symbol table. In the event of overflow the program has to be re-run from the beginning.

At the start of Pass 2, (immediately if neither option L, Z or X were entered), the listing will start to appear on the screen. While the program is listing you can switch easily between a continuous and a single step mode. During a continuous listing pressing any key on the keyboard halts it at the end of the line currently being listed. (The keyboard is only checked at the end of the line, so at low listing speeds make sure you hold the key down until the end of the line is reached). Pressing another key allows the listing to resume. If the 'resume' key is the spacebar the program will automatically halt after listing the next line. With any other key it will continue to list until you interrupt it again by pressing a key on the keyboard. Typing "Escape" (shift/enter) at this point will abort the program and return control to the monitor.

4. NOTES ON USING THE DIS-ASSEMBLER

1. Any program, including the operating system, REVAS itself, ZEAP etc, can be dis-assembled if it is in memory with the dis-assembler. Be careful that the program to be dis-assembled does not overlap the workspace used by REVAS. If it does, move it to a different area of memory and respond with three addresses to the prompt - What on?.
2. Unless you specify data areas, REVAS has no way of knowing whether the bytes it is trying to decode are program or data. It assumes that they are program, unless instructed otherwise, so if a program contains unspecified embedded data, then this data will be decoded as program.

Due to the structure of the Z80 instruction set, it is unlikely that invalid instructions will be detected in such areas, but they are easily recognisable as the sequence of instructions will make no sense at all. In the case of an ASCII message the text will be plainly visible in the comment field on the right of the listing.

At the end of the data, it may decode a few instructions incorrectly. This is because it may be getting the boundary between individual instructions wrong. A similar effect occurs if you ask it to start half way through an instruction.

3. LABELS: There is no easy way for the dis-assembler to decide whether the operand in an instruction such as LD HL,£1000 is an address, byte count, or arithmetic constant, and so it assumes that it is an address and generates a label for it. As a result any source files produced by the dis-assembler will probably require a little editing before they can be re-assembled to run at another address.

NOTE: Any labels that lie outside the bounds of the dis-assembled code are equated to their absolute values (see appendix E). If a label lies within the bounds specified, but does not line up with an instruction boundary, then a program relocatable EQU statement is generated.

eg L147A EQU \$-2

4. For those who wish to examine a full dis-assembly listing, but do not possess a printer, the suggested procedure is to generate the listing to tape. This tape can then be played back through the serial port and sections of the listing examined on the screen. It is a simple matter to stop/start the tape recorder, and to fast wind forward and backward to locate any place in the listing.
5. The delay given by option D should be adequate for most cases, but if it is found to be too short it can be effectively doubled by halving the CPU clock rate before running the dis-assembler (and doubling it afterwards).

5. PRODUCING SOURCE FILES

The following is a suggested procedure for generating source files.

First of all, using the simple mode, quickly run through the program to be dis-assembled and note down all the obvious data areas within the program. (This will not be necessary if you have already produced a source listing of the program as you can work from that).

Next run the dis-assembler in its interactive mode and, if your assembler is ZEAP (version 2.0) and you have sufficient room in memory for the source file and the symbol table, use only option Z. Otherwise use options LSTD (Labels,Source,Tape,Delay), and record the file on tape. Answer the remaining prompts and run the program.

If you have created the file on tape you will have to read it back into your assembler/editor. If (like ZEAP) it has an "auto line number" mode, or accepts lines directly, then enter your editor, set Auto mode if necessary, and play the source tape back through the tape interface. The delay at the end of the line gives the editor time to move the line into the file and generate the next line number before the next line starts. If your editor will not accept the tape directly, you will have to write a small program of your own to read the tape in. (Either load it directly to memory in the correct format, or if the problem is a minor one write your own driver to pre-process the incoming line - see Nas-sys U command).

Having got the source file into your editor the first thing to do is to set an ORG statement. Then, no doubt, there will be various areas you will want to edit - Data areas may need changing to DEFMs and DEFWs (strings and address tables) and some extra labels added as a result. Before you make any change to the source code which will result in different object code (such as a new ORG or a small modification), it is advisable to assemble the file and, using a simple program, check that the object code produced is identical to the original. This confirms that you have made no errors in your editing so far (like leaving a significant trailing space out of a DEFM) and could save you a lot of time later.

5.1 ZEAP files direct to memory

By using option Z a ZEAP file can be created directly in memory without passing through the intermediate stage of a tape file. Before this option is used ZEAP should have been entered and exited to ensure that it has initialised its own workspace. The actual file area to be used is set by the addresses entered in response to the prompt from the dis-assembler.

During Pass 1 the dis-assembler constructs the symbol table at the top of this area. In the event of it having insufficient room the message "Overflow" appears. (NB At this stage it hasn't even started trying to fit the source in as well). During Pass 2 after each line is listed to the screen it is numbered and appended to the source file. At this point a check is done to ensure that the end of the file has not

started to overwrite the symbol table. If it has the "Overflow" message appears and dis-assembly stops. Although a partial source file exists at this time, one ZEAP parameter, the label count, will be incorrectly set.

The dis-assembler requires a little more memory to create the file than ZEAP does to hold it as the dis-assembler symbol table uses 3 bytes/entry while ZEAP uses only 2 bytes/entry.

If you wish to enter more than 36 address pairs in response to "DATA areas?" you will have to temporarily save ZEAP's workspace (&F04-&F2F) elsewhere and restore it afterwards. (See section 2).

6. USING A PRINTER

The "T" option causes the screen listing to be output to the serial port as well as to the screen, with a line feed output after every carriage return. As a result a serial printer, such as a Teletype, can be connected directly to the serial port. The characters are output through the SRLX routine within Nas-sys. (SCAL SRLX).

If you have a printer with a parallel interface connected to your system, then this can be driven through the Nas-sys User output routine. Either the control program can be altered so that when the "T" option is set the dis-assembler calls UOUT rather than SRLX, or the "U" command can be given in the monitor before running the dis-assembler. In the latter case the printed output will include all the program prompts and your replies.

APPENDIX A

Idiosyncrasies and Bugs

1. In common with a lot of software, if the dis-assembler is asked to dis-assemble code up to FFFF it will not stop at FFFF, but wraps round to 0 and will continue ad infinitum. The address to be entered to ensure that this does not happen is FFFF-length of instruction ending at FFFF. (eg FFFC if FFFD is the start of a JP instruction).
2. If you dis-assemble a very large program with option "P" set you will discover that the control program cannot count properly past page number 99. Page 100 appears as :0, page 110 as ;0, and so on. (See the ascii equivalents of 3A, 3B, 3C,).
3. If you dis-assemble an area of memory without setting option "U", (ie RCAL/SCAL/PRS assumed), then the following code sequences will not be detected as invalid code:-
..DD D7../..DD DF../..DD EF../..FD D7../..FD DF../..FD EF..
The DD/FD prefix is ignored and the following code is treated as though it were a valid RCAL/SCAL/PRS instruction.

APPENDIX B

Modifications

Here is a short list of points in the program that you may wish to change to suit your own particular circumstances.

The number of printed lines per page is set at CB5A and is at present set to 3C (dec. 60).

The number of lines between pages is set at CB5B. This is actually set to one more than the number of lines wanted, and is presently set to 5 to give a gap of 4 lines.

(The total number of lines per page is $60+5+\text{heading}=66$ (decimal)).

The end-of-line delay is set by the value at CFD7 which is at present set to 80 (128 decimal)).

The SRLX routine number (6F) is held at CFE3. This should be changed to 75 for UOUT (see section 6).

APPENDIX C

Writing your own control program for REVAS

The reverse assembler subroutine REVAS does the actual dis-assembly of the code in memory. The addition of labels, tape output, cross reference tables etc are functions performed by the external control routine. REVAS dis-assembles a specified block of memory, line by line, and after generating each line of source it calls an external output routine. In the case of the simple control program this just lists the line directly to the VDU. In the interactive control program two completely different "output" routines are used, depending on whether Pass 1 or Pass 2 is being executed.

REVAS requires three parameters to be passed to it on calling. These are set in the three register pairs BC DE HL and are:-

BC=The start address to be used on the listing.
DE=The start memory address of the code to be dis-assembled.
HL=The end memory address of the code to be dis-assembled.
(REVAS returns when this address is passed).

A jump to REVAS is held in the last three bytes of the program, and so you should CALL CFFD.

Before you call REVAS you have to set two items in its workspace. Firstly the variable RSTFLG (E03) has to be set. Bit 0 of it controls how REVAS handles RST instructions. If bit 0 is set then RCAL, SCAL and PRS are assumed, if bit 0 is clear then all RSTs are treated as single byte instructions with no following parameters. Secondly, when REVAS comes to output a line it sets register pair HL to point to the start of the output buffer and then calls address E00. Your program must set a JP instruction at this address to point to your "output" routine.

When REVAS calls the output routine there are several variables that could contain values of interest.

LINEA (E0E) holds the address of the instruction currently being listed.

If the line contains a sixteen-bit operand, (ie the instruction is JP, JR, CALL, LD HL,... etc), LABELA (E10) holds its value and the variable LABELP (E12) points to the leading £ of the operand in the output buffer. If there is no sixteen-bit operand then LABELP is set to zero.

LABELP/LABELA/LINEA would be used by a program that generated labels.

For example:

Pass1:If LABELP is non-zero then LABELA holds a 16-bit reference to be entered to the symbol table, and LINEA holds the address for the cross-reference entry.

Pass2:Before outputting the line - check LINEA value and see if the line needs labelling. Also if LABELP is non-zero then a label needs to be set in the buffer at the point where LABELP is pointing.



APPENDIX D

Sample dis-assembly - Simple mode.

NB All user input is underlined.

address	code bytes	mnemonic	operand	character equivalents
<u>EC400 CDD6 2D</u>				
CDD6	CD E9 CD	CALL	SCDE9	;MIM
CDD9	36 00	LD	(HL),200	;6.
CDDB	ED 5B 10 0E	LD	DE,(20E10)	;m1..
CDDF	2B	DEC	HL	;+
CDE0	73	LD	(HL),E	;a
CDE1	2B	DEC	HL	;+
CDE2	72	LD	(HL),D	;F
CDE3	2B	DEC	HL	;+
CDE4	CB 5F	BIT	3,A	;K
CDE6	C8	RET	Z	;H
CDE7	18 CC	JR	2CDB5	;.L
CDE9	ED 5B 52 0E	LD	DE,(20E52)	;m1R.
CDED	23	INC	HL	;2
CDEE	A7	AND	A	;.
CDEF	06 00	LD	B,200	;..
CDF1	ED 52	SBC	HL,DE	;mR
CDF3	EB	EX	DE,HL	;k
CDF4	ED 42	SBC	HL,BC	;mB
CDF6	42	LD	B,D	;B
CDF7	4B	LD	C,E	;K
CDF8	EB	EX	DE,HL	;k
CDF9	2A 4E 0E	LD	HL,(20E4E)	;*N.
CDFC	2B	DEC	HL	;+
CDFD	ED 52	SBC	HL,DE	;mR
CDFE	30 0B	JR	NC,2CE0C	;0.
CE01	2A 52 0E	LD	HL,(20E52)	;*R.
CE04	ED 53 52 0E	LD	(20E52),DE	;mSR.
CE08	ED B0	LDIR		;m0
CE0A	2B	DEC	HL	;+
CE0B	C9	RET		;I
CE0C	EF	RST	228	;o
CE0D	4F 76 65 72	DEFM	/Over/	;Over
CE11	66 6C 6F 77	DEFM	/flow/	;flow
CE15	20 61 74	DEFM	/ at/	; at
CE18	0D 00	DEFB	20D,200	;..
CE1A	21 14 0E	LD	HL,20E14	;1..
CE1D	CD F3 CE	CALL	2CEF3	;MaN
CE20	DF	RST	218	;.
CE21	5B	DEFB	25B	;1
CE22	01 13 00	LD	BC,20013	;...
CE25	FD 21 58 0E	LD	IX,20E58	;1x.
CE29	E5	PUSH	HL	;e
CE2A	DD E1	POP	IX	;2a
CE2C	DD 09	ADD	IX,BC	;2.
CE2E	3A 44 0E	LD	A,(20E44)	;:D.

APPENDIX E

Sample output - Interactive mode

NB All user input is underlined.

EC400

Options? (STZLPDRU)-LXP

Symbol table area?

1000 1FFF

Title?

Demonstration

What on?

CDD6 CE2E

DATA areas?

Go?

address	code	bytes	label	mne.	operand	character equivalents
↓	↓	↓	↓	↓	↓	↓
PAGE 01			Demonstration			
			LC013	EQU	LC013	
			LC0E10	EQU	LC0E10	
			LC0E14	EQU	LC0E14	
			LC0E44	EQU	LC0E44	
			LC0E4E	EQU	LC0E4E	
			LC0E52	EQU	LC0E52	
			LC0E58	EQU	LC0E58	
			LC0DB5	EQU	LC0DB5	
CDD6	CD	E9 CD		CALL	LCDE9	;M1M
CDD9	36	00		LD	(HL),LC00	;6.
CDDB	ED	5B 10 0E		LD	DE,(LC0E10)	;m1..
CDDF	2B			DEC	HL	;+
CDE0	73			LD	(HL),E	;s
CDE1	2B			DEC	HL	;+
CDE2	72			LD	(HL),D	;r
CDE3	2B			DEC	HL	;+
CDE4	CB	5F		BIT	3,A	;K
CDE6	C8			RET	Z	;H
CDE7	18	CC		JR	LCDB5	;.L
CDE9	ED	5B 52 0E	LCDE9	LD	DE,(LC0E52)	;m;R.
CDED	23			INC	HL	;E
CDEE	A7			AND	A	;.
CDEF	06	00		LD	B,LC00	;..
CDF1	ED	52		SBC	HL,DE	;mR
CDF3	EB			EX	DE,HL	;k
CDF4	ED	42		SBC	HL,BC	;mB
CDF6	42			LD	B,D	;B
CDF7	4B			LD	C,E	;K
CDF8	EB			EX	DE,HL	;k
CDF9	2A	4E 0E		LD	HL,(LC0E4E)	;*N.
CDFC	2B			DEC	HL	;+
CDFD	ED	52		SBC	HL,DE	;mR
CDFE	30	0B		JR	NC,LC0E0C	;0.
CE01	2A	52 0E		LD	HL,(LC0E52)	;*R.

Unresolved label refs
outside the range of
code dis-assembled.

```

CE04 ED 53 52 0E      LD (LOE52),DE ;mSR.
CE08 ED B0            LDIR ;m0
CE0A 2B              DEC HL ;+
CE0B C9              RET ;I
CE0C EF              LCEOC RST £28 ;o
CE0D 4F 76 65 72    DEFM /Over/ ;Over
CE11 66 6C 6F 77    DEFM /flow/ ;flow
CE15 20 61 74        DEFM / at/ ; at
CE18 0D 00           DEFB £0D,£00 ;..
CE1A 21 14 0E        LD HL,LOE14 ;!..
CE1D CD F3 CE        CALL LCEF3 ;MaN
CE20 DF              RST £18 ;_
CE21 5B              DEFB £5B ;!
CE22 01 13 00        LD BC,L0013 ;...
CE25 FD 21 58 0E     LD IX,LOE58 ;.IX.
CE29 E5              PUSH HL ;e
CE2A DD E1           POP IX ;!a
CE2C DD 09           ADD IX,BC ;!.
CE2E 3A 44 0E        LD A,(LOE44) ;:D.
                    LCEF3 EQU £CEF3

```

PAGE 02 Demonstration

```

0013 CE22
0E10 CDBB
0E14 CE1A
0E44 CE2E
0E4E CDF9
0E52 CDE9 CE01 CE04
0E58 CE25
CDB5 CDE7
CDE9 CDD6
CE0C CDFE
CEF3 CE1D
NAS-SYS 1

```

↑ ↑
address Location of references to the address

Simple Control Program for REVAS

```

C400          0002 ;*****
C400          0003 ; SIMPLE CONTROL PROGRAM FOR REVAS 21-02-80
C400          0004 ;   ENTERED BY "E" COMMAND
C400          0005 ;
C400          0006 ;   THREE POSSIBLE FORMATS -
C400          0007 ;   <E NNNN>   - TRANSFERS CONTROL TO REVASC
C400          0008 ;   <E NNNN AAAA> - STARTS SIMPLE DISASSEMBLY AT LINE AAAA
C400          0009 ;   <E NNNN AAAA BB>- AS ABOVE BUT DISPLAYS
C400          0010 ;   <BB> LINES AT A TIME
C400          0011 ;*****
C400 3E C3     0012      LD   A,%C3      ;SET RAM JUMP
C400 32 00 0E   0013      LD   (PATCH),A
C400 32 03 0E   0014      LD   (RSTFLG),A ;SET RSTFLG=NAS-SYS
C400 21 28 CA   0015      LD   HL,PRINT ;PATCH ROUTINE
C400 22 01 0E   0016      LD   (PATCH+1),HL
C400 3A 00 0C   0017      LD   A,(NUMARG) ;CHECK OPTIONS
C400 FE 02     0018      CP   2          ;TWO?
C400 DA 00 CA   0019      JP   C,REVASC ;ONE,GO TO REVASC
C400 DF        0020      RST  SCAL      ;GET ARGUMENTS TO REGS.
C400 60        0021      DEFB  ARG3
C400 20 02     0022      JR   NZ,++2    ;SKIP IF 3 ENTERED
C400 0E 01     0023      LD   C,1        ;SET DEFAULT
C400 41        0024      LD   B,C        ;SET COUNT
C400 ED 43 10 0C 0025      LD   (ARG3),3C ;SAVE
C400 42        0026      LD   B,D        ;GET START ADDRESS...
C400 48        0027      LD   C,E        ;..TO 0C AS WELL
C400 21 FF FF   0028      LD   HL,-1      ;SET END ADDRESS
C400 10 16     0029      JR   REVAS    ;GO TO REVAS
C428          0030 ;
C428          0031 ; PRINT (HL) UNTIL A CR, THEN CHECK LINE COUNT
C428          0032 ;
C428 7E        0033 PRINT: LD   A,(HL)    ;GET BYTE
C428 F7        0034      RST  %30      ;OUTPUT A CHARACTER
C42A 23        0035      INC  HL
C42B FE 00     0036      CP   CR        ;WAS IT A CR?
C42B 20 F9     0037      JR   NZ,PRINT ;NO,CONTINUE
C42F 21 10 0C  0038      LD   HL,ARG3   ;CHECK COUNT
C432 35        0039      DEC  (HL)     ;DONE?
C433 C0        0040      RET  NZ        ;NO
C434 23        0041      INC  HL        ;YES,RESET COUNT
C435 7E        0042      LD   A,(HL)
C436 20        0043      DEC  HL
C437 77        0044      LD   (HL),A
C438 CF        0045      RST  RIN      ;WAIT FOR KEY
C439 06 10     0046      SUB  ESC      ;FINISH?
C43B C0        0047      RET  NZ        ;NO,CONTINUE
C43C 0F        0048      RST  SCAL    ;YES,DONE
C43D 5B        0049      DEFB  RETURN
C43E          0050 ;*****
C43E          0051 ;   EQUATES TO NAS-SYS ROUTINES
C43E          0052 ;*****
0000          0053 RIN:   EQU  0
0010          0054 SCAL:  EQU  %10
005B          0055 RETURN: EQU  %5B
0060          0056 ARG3:  EQU  %60
%C0B          0057 NUMARG: EQU  %C0B
%C10          0058 ARG3:  EQU  %C10
001B          0059 ESC:   EQU  %1B
0000          0060 CR:   EQU  %D

```

REVAS subroutine version N1.1 88-93-88

```

C43E      0062 ;*****
C43E      0063 ; A REVERSE ASSEMBLER FOR 8080/Z80 CODES
C43E      0064 ;
C43E      0065 ; WRITTEN BY DAVID PARKINSON
C43E      0066 ; VERSION N1.1 88-93-88
C43E      0067 ;
C43E      0068 ; THE OUTPUT IS IN Z80 COMPATIBLE MNEMONICS
C43E      0069 ;
C43E      0070 ; THE PROGRAM IS IN THE FORM OF A SUBROUTINE
C43E      0071 ; REVAS IS ENTERED WITH THE FOLLOWING SEQUENCE....
C43E      0072 ;
C43E      0073 ; LD BC WITH THE "PROGRAM" ADDRESS
C43E      0074 ; (THE STARTING ADDRESS USED IN THE LISTING)
C43E      0075 ; LD DE WITH THE START MEMORY ADDRESS
C43E      0076 ; (ACTUAL LOCATION OF CODE TO BE REVERSE ASSEMBLED)
C43E      0077 ; LD HL WITH THE END ADDRESS
C43E      0078 ; (REVAS RETURNS WHEN THIS ADDRESS IS PASSED)
C43E      0079 ; CALL REVAS
C43E      0080 ;*****
C43E      0081 ; VARIABLE STORAGE
C43E      0082 ;*****
C43E      C43E 0083 SAVE: EQU *
C43E      0084 ORG $E00 ;WORKSPACE
0E99 0003 0085 PATCH: DEFS 3 ;RAM PATCH TO ROUTINE
0E93 0001 0086 RSTFLG: DEFS 1 ;RST HANDLING FLAG
0E94 0002 0087 ADDR: DEFS 2 ;PROGRAM COUNTER
0E96 0002 0088 MADDR: DEFS 2 ;CURRENT MEMORY ADDRESS
0E98 0002 0089 EADDR: DEFS 2 ;WHERE TO STOP
0E9A 0002 0090 OPCADR: DEFS 2 ;POINTER IN BUFFER
0E9C 0002 0091 HXYFLG: DEFS 2 ;HL,IX,IY FLAG
0E9E 0002 0092 LINEA: DEFS 2 ;ADDR. LINE REF. LABEL
0E10 0002 0093 LABELA: DEFS 2 ;LABEL VALUE
0E12 0002 0094 LABELP: DEFS 2 ;POINTER TO POS. IN BUFFER
0E14 0003 0095 BUFFER: DEFS 48 ;OUTPUT BUFFER
0E44      0096 ;*****
0E44      0097 ; REVERSE ASSEMBLER ENTRY POINT
0E44      0098 ;*****
0E44      0099 ORG SAVE
C43E ED 43 04 0E 0100 REVAS: LD (ADDR),BC ;SAVE DATA
C442 ED 33 06 0E 0101 LD (MADDR),DE
C446 22 08 0E 0102 LD (EADDR),HL
C449      0103 ; NEXT LINE OF OUTPUT
C449 2A 08 0E 0104 NEXTL: LD HL,(EADDR) ;CHECK FOR END
C44C ED 5B 06 0E 0105 LD DE,(MADDR)
C450 AF 0106 XOR A ;CLEAR CARRY
C451 ED 52 0107 SBC HL,DE
C453 08 0108 RET C ;RETURN IF PAST
C454 CD 4E C4 0109 CALL INITB ;INITIALISE OUTPUT BUFFER
C457 CD 08 C4 0110 CALL BYTE ;GET NEXT BYTE
C45A CD 15 C5 0111 CALL DECODE ;DECODE IT
C45D 21 0C 0E 0112 LD HL,HXYFLG ;CHECK IF SET THEN USED
C460 7E 0113 LD A,(HL)
C461 23 0114 INC HL
C462 3E 0115 CP (HL)
C463 C4 53 C9 0116 CALL NZ,NOTVAL ;INVALID IF UNUSED
C466 21 14 0E 0117 LD HL,BUFFER ;SET ADDRESS
C469 CB 08 0E 0118 CALL PATCH ;PRINT OR WHATEVER
C46C 18 08 0119 JR NEXTL ;LOOP
C46E      0120 ;*****
C46E      0121 ; INITIALISE THE OUTPUT BUFFER (AND CERTAIN VARIABLES)

```


REVAS subroutine version N1.1 88-03-88

```

C46E                                0122 ;*****
C46E 11 14 0E                       0123 INITB: LD DE,BUFFER ;GET START ADDRESS
C471 05                               0124         PUSH DE ;GET TO IX
C472 DD E1                           0125         POP IX
C474 21 00 00                         0126         LD HL,0 ;CLEAR LABEL POINTER
C477 22 12 0E                         0127         LD (LABELP),HL
C47A 22 0C 0E                         0128         LD (HXYFLG),HL ;CLEAR FLAG
C47D 2A 04 0E                         0129         LD HL,(ADDR) ;WRITE ADDRESS
C480 22 0E 0E                         0130         LD (LINEA),HL ;KEEP ADDRESS
C483 CD 9D C4                         0131         CALL HEX4
C484 EB                               0132         EX DE,HL ;NOW CLEAR REST OF BUFFER
C487 23                               0133         INC HL
C488 22 9A 0E                         0134         LD (OPCADR),HL ;SET POINTER FOR OPCODES
C48B 2B                               0135         DEC HL
C48C 06 2B                             0136         LD B,43
C48E 36 20                             0137         LD (HL),' '
C490 23                               0138         INC HL
C491 10 FB                             0139         DJNZ *-5
C493 00 36 2A 3D                      0140         LD (IX+42),';' ;COMMENT DELIM.
C497 36 0D                             0141         LD (HL),CR ;SET CR AT END
C499 11 2D 0E                         0142         LD DE,BUFFER+25 ;SET ADDRESS FOR MNEONIC
C49C C9                               0143         RET ;DONE
C49D                                0144 ;*****
C49D                                0145 ;UTILITY SUBROUTINES
C49D                                0146 ;*****
C49D                                0147 ;
C49D                                0148 ;HL TO BUFFER AS FOUR HEX CHARACTERS
C49D                                0149 ;
C49D 7C                               0150 HEX4: LD A,H ;GET HI BYTE
C49E CD A2 C4                         0151         CALL HEX2 ;PRINT IT
C4A1 7D                               0152         LD A,L ;GET LO BYTE
C4A2                                0153 ;TRANSFER A AS TWO HEX CHARACTERS TO
C4A2                                0154 ;OUTPUT BUFFER (DE POINTS TO CURRENT POSITION
C4A2                                0155 ;IN THE BUFFER).
C4A2                                0156 ;
C4A2 F5                               0157 HEX2: PUSH AF ;SAVE A
C4A3 0F                               0158         RRCA ;SHIFT HI' NIBBLE DOWN.
C4A4 0F                               0159         RRCA
C4A5 0F                               0160         RRCA
C4A6 0F                               0161         RRCA
C4A7 CD AB C4                         0162         CALL HEX1 ;PRINT IT
C4AA F1                               0163         POP AF ;RECOVER A
C4AB F5                               0164 HEX1: PUSH AF ;SAVE A
C4AC E6 0F                           0165         AND 00F ;ISOLATE LO' NIBBLE
C4AE C6 90                             0166         ADD 90H ;MAKE ASCII
C4B0 27                               0167         DAA
C4B1 CE 40                             0168         ADC 40H
C4B3 27                               0169         DAA
C4B4 12                               0170         LD (DE),A ;PUT IN BUFFER
C4B5 13                               0171         INC DE ;BUMP ADDRESS
C4B6 F1                               0172         POP AF ;RECOVER A
C4B7 C9                               0173         RET
C4B8                                0174 ;
C4B8                                0175 ;GET NEXT BYTE OF DATA. PUT IN OUTPUT
C4B8                                0176 ; STRING ALONG WITH ANY ASCII
C4B8                                0177 ; EQUIVALENT
C4B8                                0178 ;
C4B8 00 23                             0179 BYTE: INC IX ;BUMP POINTER
C4BA 2A 04 0E                         0180         LD HL,(ADDR) ;UPDATE PC
C4BD 23                               0181         INC HL

```

REVAS subroutine version N1.1 08-03-80

```

C4BE 22 04 0E      0182      LD      (ADDR),HL
C4C1 2A 06 0E      0183      LD      HL,(HADDR) ;LOAD BYTE ADDRESS
C4C4 7E            0184      LD      A,(HL) ;LOAD BYTE
C4C5 23            0185      INC     HL ;UPDATE ADDRESS
C4C6 22 06 0E      0186      LD      (HADDR),HL
C4C9 2A 0A 0E      0187      LD      HL,(OPCADR) ;NOW PUT IN OUTPUT...
C4CC 23            0188      INC     HL ;SPACE BETWEEN OPCODES
C4CD EB           0189      EX      DE,HL ;..STRING IN CORRECT..
C4CE CD A2 C4      0190      CALL   HEX2 ;..PLACE.
C4D1 EB           0191      EX      DE,HL ;RESTORE POINTER
C4D2 22 0A 0E      0192      LD      (OPCADR),HL ;SAVE IN CASE NEEDED AGAIN
C4D5 F5           0193      PUSH   AF ;SAVE
C4D6 3C           0194      INC     A ;ROLL "RUB OUT" ROUND
C4D7 E6 7F        0195      AND     $7F ;STRIP OFF PARITY
C4D9 FE 21        0196      CP      $21 ;PRINTABLE?
C4DB 3D           0197      DEC     A ;(CORRECT A)
C4DC 30 02        0198      JR      NC,++2 ;YES,SKIP
C4DE 3E 2E        0199      LD      A,'.' ;NO,SET '.'
C4E0 DD 77 2A     0200      LD      (IX+42),A ;APPEND TO BUFFER
C4E3 F1           0201      POP    AF ;RECOVER A
C4E4 C9           0202      RET
C4E5             0203 ;
C4E5             0204 ; WRITE "EX"
C4E5             0205 ;
C4E5 EB          0206 WREX:  EX  DE,HL
C4E6 36 45        0207      LD      (HL),'E'
C4E8 23          0208      INC     HL
C4E9 36 58        0209      LD      (HL),'X'
C4EB 18 06        0210      JR      WRLD0 ;REST IN WRLD
C4ED             0211 ;
C4ED             0212 ;WRITE "LD"
C4ED             0213 ;
C4ED EB          0214 WRLD:  EX  DE,HL
C4EE 36 4C        0215      LD      (HL),'L'
C4F0 23          0216      INC     HL
C4F1 36 44        0217      LD      (HL),'D'
C4F3 11 32 0E     0218 WRLD0:  LD  DE,BUFFER+30 ;SET OPERAND ADDRESS
C4F6 C9           0219      RET
C4F7             0220 ;
C4F7             0221 ; WRITE ','
C4F7             0222 ;
C4F7 EB          0223 COMMA:  EX  DE,HL
C4F8 36 2C        0224      LD      (HL),','
C4FA 23          0225      INC     HL
C4FB EB          0226      EX      DE,HL
C4FC C9           0227      RET
C4FD             0228 ;
C4FD             0229 ;WRITE "*"
C4FD             0230 ;
C4FD EB          0231 POUND:  EX  DE,HL
C4FE 36 23        0232      LD      (HL),'#'
C500 23          0233      INC     HL
C501 EB          0234      EX      DE,HL
C502 C9           0235      RET
C503             0236 ;
C503             0237 ;COPY OVER CHARACTERS
C503             0238 ;
C503 ED A9        0239 COPY6:  LDI
C505 ED A8        0240 COPY5:  LDI
C507 ED A8        0241 COPY4:  LDI

```

REVAS subroutine version N1.1 88-03-89

```

C509 ED A0      0242 COPY3:  LDI
C50B ED A0      0243 COPY2:  LDI
C50D ED A0      0244      LDI
C50F C9         0245      RET
C510           0246 ;
C510           0247 ;FORM ADDRESS FOR TABLE LOOK UP
C510           0248 ;
C510           0249 FTADR:  ADD  L
C511 6F         0250      LD   L,A
C512 D0         0251      RET  NC
C513 24         0252      INC  H
C514 C9         0253      RET
C515           0254 ;*****
C515           0255 ; PRELIMINARY DECODE OF BYTE
C515           0256 ;*****
C515 FS        0257 DECODE:  PUSH AF          ;SAVE
C516 E6 C0      0258      AND  $C0          ;ISOLATE MSBS
C518 FE 40      0259      CP   $40          ;01XXXXXX?
C51A CA 9D C5   0260      JP   Z,LOAD8      ;YES,8-BIT LOAD
C51B FE 80      0261      CP   $80          ;10XXXXXX?
C51F CA 94 C6   0262      JP   Z,ARITH8     ;YES,8-BIT ARITHMETIC
C522 F1         0263      POP  AF          ;GET BYTE BACK
C523 F3         0264      PUSH AF         ;PUT ON STACK FOR LATER
C524 E6 8F      0265      AND  $8F          ;ISOLATE 7XXX3210
C526 87         0266      RLCA          ;FORM AS ..32107.
C527 87         0267      RLCA
C528 21 7F C9   0268      LD   HL,TABLE     ;LOAD BASE ADDRESS
C529 CD 19 C5   0269      CALL FTADR       ;FORM ADDRESS
C52E 7E         0270      LD   A,(HL)       ;GET ADDRESS AT THAT ADDRESS
C52F 23         0271      INC  HL
C530 66         0272      LD   H,(HL)
C531 6F         0273      LD   L,A
C532 E9         0274      JP   (HL)          ;GO TO IT
C533           0275 ;*****
C533           0276 ; DECODE OPCODES FULLY
C533           0277 ;*****
C533           0278 ;
C533           0279 ;INC/DEC REGISTER/REGISTER PAIR
C533           0280 ;*****
C533 21 BF C9   0281 INC:   LD   HL,INCH     ;POINT TO "INC"
C534 18 03      0282      JR   ++3          ;SKIP
C538 21 C2 C9   0283 DEC:   LD   HL,DECH     ;POINT TO "DEC"
C53B CD 38 C7   0284      CALL COPY3S      ;COPY OVER
C53E F1         0285      POP  AF          ;RECOVER OPCODE
C53F C0 57      0286      BIT  2,A          ;TEST 0/16 BIT
C541 28 34      0287      JR   Z,REGPR     ;JUMP IF 16 BIT
C543 0F         0288      RRCA          ;SHIFT REG. ID....
C544 0F         0289      RRCA          ;...DOWN TO LOW A
C545 0F         0290      RRCA
C546 18 4C      0291      JR   SREG        ;RETURN AFTER PRINTING REGISTER
C548           0292 ;*****
C548           0293 ;16-BIT DIRECT LOAD
C548           0294 ;*****
C548 CB ED C4   0295 LD16:  CALL URLD        ;WRITE "LD"
C54B F1         0296      POP  AF          ;RECOVER OPCODE
C54C CD 77 C5   0297      CALL REGPR       ;PRINT REGISTER PAIR
C54F CB F7 C4   0298      CALL CDMA        ;" "
C552 CD B8 C4   0299 LD16A: CALL BYTE        ;GET LO BYTE
C555 4F         0300      LD   C,A          ;SAVE IN C
C556 CD B8 C4   0301      CALL BYTE        ;GET HI BYTE

```

REVAS subroutine version N1.1 00-03-00

```

C559 67          0302      LD      H,A          ;PUT IN H
C55A 69          0303      LD      L,C          ;RECOVER FIRST BYTE
C55B ED 53 12 9E 0304 LD16B: LD      (LABELP),DE ;SAVE POINTER
C55F 22 10 9E    0305      LD      (LABELA),HL ;SAVE ADDRESS (?)
C562 CD FD C4    0306      CALL   POUND          ;"#"
C565 C3 9D C4    0307      JP      HEX4          ;RETURN,PRINTING HL.
C568            0308 ;*****
C568            0309 ;      ADD HL
C568            0310 ;*****
C568 21 B0 C9    0311 ADDHL: LD      HL,ARTAB ;POINT TO "ADD"
C56B CD 3B C7    0312      CALL   COPY35        ;WRITE IT
C56E 3E 20       0313      LD      A,$20         ;MAKE LIKE "HL"
C570 CD 77 C5    0314      CALL   REGPR         ;WRITE HL,IX,OR IY
C573 CD F7 C4    0315      CALL   CONMA         ;", "
C576 F1         0316      POP     AF           ;RECOVER OPCODE
C577            0317 ;*****
C577            0318 ;REGISTER PAIR DECODE/PRINT
C577            0319 ;*****
C577 47          0320 REGPR: LD      B,A          ;SAVE IN B
C578 0F          0321      RRCA          ;SHIFT PAIR ID BITS DOWN.
C579 0F          0322      RRCA
C57A 0F          0323      RRCA
C57B C6 02       0324      ADD     2           ;ALTER REG. CODING
C57D 0E 04       0325      LD      C,4         ;PUT 4 IN C
C57F A1          0326      AND     C           ;ISOLATE REG. ID
C580 21 C8 C9    0327      LD      HL,RPRTAB   ;LOAD BASE ADDRESS
C583 20 00       0328      JR      NZ,NOTSP    ;SKIP IF NOT 'SP'
C585 20         0329      DEC     HL          ;POINT TO "SP"
C586 C0 70       0330      BIT     7,B        ;IS IT PUSH/POP?
C588 CA 0B C5    0331      JP      Z,COPY2     ;NO DONE
C58B 20         0332      DEC     HL          ;YES,POINT...
C58C 20         0333      DEC     HL          ;...TO "AF"
C58D 09         0334 NOTSP: CP      C           ;IS IT "HL"?
C58E 20 07       0335      JR      NZ,RP2     ;NO,SKIP
C590 3A 0C 0E    0336      LD      A,(HXYFLG) ;LOAD FLAG
C593 32 0D 0E    0337      LD      (HXYFLG+1),A ;(SET 'USED')
C596 01         0338      ADD     C           ;ADJUST IF IX/IY
C597 CD 10 C5    0339 RP2:  CALL   FTADR     ;FORM ADDRESS
C59A C3 0D C5    0340      JP      COPY2     ;COPY OVER
C59D            0341 ;*****
C59D            0342 ;8-BIT LOAD INSTRUCTIONS
C59D            0343 ;*****
C59D F1          0344 LOAD8: POP     AF           ;RECOVER OPCODE
C59E FE 76       0345      CP      $76        ;HALT?
C5A0 20 5C       0346      JR      Z,HALT     ;YES,JUMP
C5A2 F5         0347      PUSH   AF          ;SAVE OPCODE
C5A3 CD ED C4    0348      CALL   WRLD        ;WRITE "LD"
C5A6 0F          0349      RRCA          ;SHIFT REG ID DOWN
C5A7 0F          0350      RRCA
C5A8 0F          0351      RRCA
C5A9 CD B4 C5    0352      CALL   SREG        ;PRINT IT
C5AC CD F7 C4    0353      CALL   CONMA       ;", "
C5AF F1         0354 L8B:  POP     AF           ;GET OPCODE BACK
C5B0 CB 77       0355      BIT     6,A        ;TEST FOR IMMED. DATA
C5B2 20 41       0356      JR      Z,INN      ;JUMP IF SO
C5B4            0357 ;*****
C5B4            0358 ;SINGLE REGISTER DECODE
C5B4            0359 ;*****
C5B4 3C          0360 SREG:  INC     A           ;ADJUST CODING
C5B5 E6 07       0361      AND     7           ;ISOLATE IT

```

REVAS subroutine version M1.1 88-93-88

```

C5B7 FE 87          #362      CP      7          ;IS IT MEMORY?
C5B9 28 89          #363      JR      Z, MEM    ;YES, JUMP
C5BB 21 C9 C9       #364      LD      HL, RPRTAB+1 ;LOAD BASE ADDRESS
C5BE CD 19 C3       #365      CALL   FTADR     ;FORM ADDRESS
C5C1 ED A8          #366      LBI     ;COPY OVER REG
C5C3 C9            #367      RET
C5C4 3E 28          #368 MEM:  LD      A, '('     ;MEM INDIRECT
C5C4 12            #369      LD      (DE), A    ;WRITE "("
C5C7 13            #370      INC     DE         ;BUMP ADDRESS
C5C8 21 CE C9       #371      LD      HL, HXYTAB
C5CB 3A 9C 9E       #372      LD      A, (HXYFLG) ;OFFSET IF INDEX
C5CE 32 8D 9E       #373      LD      (HXYFLG+1), A ;SET 'USED FLAG'
C5D1 F5            #374      PUSH   AF         ;SAVE
C5D2 CD 97 C5       #375      CALL   RP2       ;COPY REG. PR OVER
C5D5 F1            #376      POP    AF         ;GET IT BACK
C5D6 A7            #377      AND    A          ;IS IT INDEX?
C5D7 28 17          #378      JR      Z, NOTIXY ;NO, JUMP
C5D9              #379 ;
C5D9 CD 88 C4       #380      CALL   BYTE      ;YES, GET OFFSET
C5DC A7            #381      AND    A          ;TEST IT
C5DD 9E 28          #382      LD      C, '+'    ;SET POSITIVE
C5DF 28 9F          #383      JR      Z, NOTIXY ;SKIP IF ZERO
C5E1 F2 EB C5       #384      JP     P, PLUS   ;JUMP IF IT WAS +VE
C5E4 ED 44          #385      NEG
C5E6 9E 2D          #386      LD      B, '-'    ;2'S COMP
C5E8 47            #387 PLUS:  LD      B, A      ;CHANGE SIGN
C5E9 79            #388      LD      A, C      ;SAVE MAGNITUDE
C5EA 12            #389      LD      (DE), A   ;GET SIGN
C5EB 13            #390      INC     DE         ;PRINT IT
C5EC 78            #391      LD      A, B      ;BUMP ADDRESS
C5ED CD FB C5       #392      CALL   PHEX2     ;GET MAGNITUDE
C5F0 3E 29          #393 NOTIXY: LD      A, '('   ;WRITE "BXX"
C5F2 12            #394      LD      (DE), A   ;CLOSE BRACKETS
C5F3 13            #395      INC     DE
C5F4 C9            #396      RET
C5F5              #397 ; *****
C5F5              #398 ; GET SINGLE BYTE IMMEDIATE
C5F5              #399 ; *****
C5F5 CD 88 C4       #400 IMM:  CALL   , BYTE   ;GET DATA
C5F8 CD FD C4       #401 PHEX2: CALL   POUND ;PRINT "B"
C5FB C3 A2 C4       #402      JP     HEX2      ;PRINT DATA
C5FE              #403 ; *****
C5FE              #404 ; HALT INSTRUCTION
C5FE              #405 ; *****
C5FE 21 D4 C9       #406 HALT:  LD      HL, HALTN
C601 C3 87 C5       #407      JP     COPY4     ;EXIT VIA COPY4
C604              #408 ; *****
C604              #409 ; 8-BIT ARITHMETIC
C604              #410 ; *****
C604 F1            #411 ARITH: POP    AF          ;RECOVER OPCODE
C605 EE 48          #412      XOR    48        ;TOGGLE BIT
C607 F5            #413      PUSH   AF         ;PUT IT BACK
C608 9F            #414      RRCA
C609 9F            #415      RRCA
C60A E6 9E          #416      AND    9E        ;ISOLATE IT
C60C 47            #417      LD      B, A      ;*3 IT
C60D 9F            #418      RRCA
C60E 88            #419      ADD    B
C60F 21 D8 C9       #420      LD      HL, ARTAB ;LOAD BASE ADDRESS
C612 C4 18 C5       #421      CALL   NZ, FTADR  ;FORM ADDRESS

```

REVAS subroutine version N1.1 08-03-88

```

C615 78          0422      LD  A,B          ;RELOAD OP ID
C616 CD 38 C7   0423      CALL COPY3S      ;COPY MNEMONIC
C619 21 98 CA   0424      LD  HL,ACONMA    ;POINT TO "A,"
C61C FE 96     0425      CP   4          ;SBC?
C61E CC 98 C5   0426      CALL Z,COPY2    ;YES,DO IT
C621 FE 93     0427      CP   3          ;ADD,ADC?
C623 DC 98 C5   0428      CALL C,COPY2    ;YES,DO IT.
C626 18 87     0429      JR   LBB        ;PRINT OPERAND
C628           0430 ;*****
C628           0431 ; POP AND PUSH
C628           0432 ;*****
C628 21 F4 C9   0433 POP:   LD  HL,POPH    ;LOAD POINTER
C629 18 93     0434      JR   ++3       ;SKIP
C62D 21 F8 C9   0435 PUSH:  LD  HL,PUSHM   ;LOAD POINTER
C630 CD 87 C5   0436      CALL COPY4      ;PRINT MNEMONIC
C633 13        0437      INC  DE        ;SPACE
C634 C3 76 C5   0438      JP   REGPR-1   ;GET OP & PR. REG PAIR
C637           0439 ;*****
C637           0440 ; CALL/IX/EXTENDED/IY
C637           0441 ;*****
C637 F1        0442 CALET: POP  AF          ;RECOVER OPCODE
C638 FE ED     0443      CP   9ED       ;EXTENDED?
C63A CA C9 C8   0444      JP   Z,EXTND   ;YES,JUMP
C63D FE CD     0445      CP   9CD       ;CALL?
C63F 28 17     0446      JR   Z,CJR+1   ;YES,JUMP
C641           0447 ; SET IX/IY FLAG
C641 D6 F9     0448      SUB  9F9       ;LEAVES 4 IF IY
C643 38 82     0449      JR   NC,++2    ;SKIP IF 4
C645 3E 82     0450      LD  A,2       ;ELSE LOAD 2
C647 32 9C 8E   0451      LD  (HXYFLG),A ;SET FLAG
C64A CD 98 C4   0452      CALL BYTE      ;GET NEXT BYTE
C64D 47        0453      LD  B,A       ;TEMP SAVE
C64E E6 9F     0454      AND  9F       ;CHECK LO' NIBBLE
C650 FE 8D     0455      CP   9D       ;IS IT -D?
C652 78        0456      LD  A,B       ;(RESET OPCODE)
C653 C8        0457      RET  Z        ;YES,INVALID CODE
C654 C3 15 C5   0458      JP   DECODE   ;PERHAPS NOT,DECODE IT
C657           0459 ;
C657           0460 ; CALL/JUMP/RETURN
C657           0461 ;
C657 F1        0462 CJR:   POP  AF          ;RECOVER OPCODE
C658 F5        0463      PUSH AF        ;SAVE IT
C659 E6 86     0464      AND  6         ;ISOLATE ID
C65B 87        0465      RLCA         ;
C65C 21 F8 C9   0466      LD  HL,CJRTAB ;LOAD BASE ADDRESS
C65F C4 18 C5   0467      CALL NZ,FTADR ;FORM ADDRESS
C662 CD 87 C5   0468      CALL COPY4     ;WRITE MNEMONIC
C665 13        0469      INC  DE        ;SPACE
C666 F1        0470      POP  AF        ;GET OPCODE BACK
C667 FE C9     0471      CP   9C9       ;UNCOND RETURN?
C669 C8        0472      RET  Z        ;YES,RETURN
C66A C9 47     0473      BIT  9,A       ;UNCOND. CALL/JUMP?
C66C 28 8B     0474      JR   NZ,UNCND ;YES,SKIP
C66E F5        0475      PUSH AF        ;SAVE AGAIN
C66F CD 7C C6   0476      CALL CCODES    ;ADD CONDITION
C672 F1        0477      POP  AF        ;RECOVER AGAIN
C673 E6 97     0478      AND  7         ;CHECK FOR RETURN?
C675 C8        0479      RET  Z        ;YES,RETURN
C676 CD F7 C4   0480      CALL CONMA     ;" "
C679 C3 52 C5   0481 UNCD:  JP   LD16A     ;EXIT WRITING "NNNN"

```

REVAS subroutine version N1.1 88-03-89

```

C67C          0482 ;
C67C          0483 ; DECODE CONDITION CODES
C67C          0484 ;
C67C 0F      0485 CCODES: RRCA          ;SHIFT DOWN
C67D 0F      0486          RRCA          ;ISOLATE
C67E E6 0E   0487          AND  #E          ;ISOLATE
C680 21 04 CA 0488          LD  HL,CCTAB      ;LOAD BASE ADDRESS
C683 C3 97 C5 0489          JP  RP2          ;COPY OVER
C686          0490 ;*****
C686          0491 ; SINGLE BYTE ROTATE & MISC.
C686          0492 ;*****
C686 F1      0493 ROTNIS: POP  AF          ;RECOVER OPCODE
C687 0F      0494          RRCA          ;ISOLATE ID
C688 E6 1C   0495          AND  #1C         ;ISOLATE ID
C68A 21 14 CA 0496          LD  HL,RMTAB      ;LOAD BASE ADDRESS
C68D C4 10 C5 0497          CALL NZ,FTADR     ;FORM ADDRESS
C690 C3 07 C5 0498          JP  COPY4          ;WRITE TO BUFFER
C693          0499 ;*****
C693          0500 ; RESTART
C693          0501 ;*****
C693 21 34 CA 0502 RST: LD  HL,RSTH      ;WRITE "RST"
C694 CD 38 C7 0503          CALL COPY3S
C699 F1      0504          POP  AF          ;RECOVER OPCODE
C69A E6 38   0505          AND  #38         ;ISOLATE ADDRESS
C69C CD F8 C5 0506          CALL PHEX2      ;WRITE "XXX"
C69F 21 03 0E 0507          LD  HL,RSTFLG   ;CHECK HANDLING FLAG
C6A2 C8 46   0508          BIT  #,(HL)    ;HAS-SYS?
C6A4 C8      0509          RET  Z          ;RETURN IF NOT
C6A5 FE 10   0510          CP  #10         ;<RCAL?
C6A7 D8      0511          RET  C          ;YES,DONE
C6A8 FE 20   0512          CP  #20         ;BRKPT?
C6AA C8      0513          RET  Z          ;YES,RETURN
C6AB 38 3C   0514          JR  C,FLUSH    ;SKIP IF RCAL/SCAL
C6AD FE 28   0515          CP  #28         ;PRG?
C6AF C8      0516          RET  NZ         ;NO,RETURN
C6B0          0517 ; DECODE RST AS NASCON "PRS"
C6B0          0518 ; DECODE FOLLOWING STRING AS DEFB OR DEFB
C6B0 CD E9 C6 0519 BORN: CALL FLUSH      ;PRINT "RST",GET NEXT BYTE
C6B3 06 03   0520          LD  B,3         ;SET FOR 3 'DEFB'S
C6B5 38 22   0521          JR  C,UNPRN    ;SKIP IF DEFB
C6B7 04      0522          INC  B          ;4 IF 'DEFB'
C6B8 21 30 0E 0523          LD  HL,BUFFER+2B ;SET HL
C6BB 36 4D   0524          LD  (HL),'M'    ;OVERWRITE 'B' WITH 'M'
C6BD 23      0525          INC  HL         ;SPACE ON
C6BE 23      0526          INC  HL
C6BF 36 2F   0527          LD  (HL),'/'    ;SET DELIMITER
C6C1 23      0528          INC  HL
C6C2 EB      0529          EX  DE,HL       ;POINTER TO DE
C6C3 18 08   0530          JR  NS1         ;SKIP ON
C6C5 CD FC C6 0531 MORENS: CALL PRTABL     ;NEXT BYTE PRINTABLE?
C6C8 38 07   0532          JR  C,DONEN    ;NO,PRINT LINE
C6CA CD B0 C4 0533          CALL BYTE      ;YES,GET IT
C6CD 12      0534 NS1: LD  (DE),A      ;SET CHARAC. IN O/P
C6CE 13      0535          INC  DE
C6CF 10 FA   0536          DJNZ MORENS    ;LOOP FOR MORE
C6D1 EB      0537 DONEN: EX  DE,HL       ;CLOSE STRING
C6D2 36 2F   0538          LD  (HL),'/'
C6D4 18 DA   0539          JR  BORN       ;LOOP
C6D6          0540 ; DEFB SECTION
C6D6 CD 08 C4 0541 IN0: CALL BYTE      ;GET NEXT BYTE

```

REVAS subroutine version N1.1 08-03-80

```

C6D9 B7          0542 UNPRN:  OR   A           ;CHECK FOR EOS
C6DA CA 53 C9    0543          JP   Z,NOTVAL   ;YES,EXIT THRU' NOTVAL
C6DB CD FC C4    0544          CALL PRTABL    ;NEXT BYTE PRINTABLE?
C6E0 39 02       0545          JR   NC,UN1    ;NO,PRINT WHAT'S THERE
C6E2 10 F2       0546          DJNZ UNS      ;LOOP IF ROOM FOR MORE
C6E4 CD 53 C9    0547 UN1:   CALL NOTVAL    ;PRINT LINE
C6E7 18 C7       0548          JR   BORN     ;LOOP
C6E9            0549 ;PRINT BUFFER,GET NEXT BYTE & CHECK IF PRINTABLE
C6E9 21 14 0E    0550 FLUSH: LD   HL,BUFFER
C6EC CD 00 0E    0551          CALL PATCH
C6EF CD 4E C4    0552          CALL INITB
C6F2 CD 00 C4    0553          CALL BYTE
C6F3 F3         0554          PUSH AF       ;SAVE BYTE
C6F6 CD 53 C9    0555          CALL NOTVAL   ;SET 'DEFB'
C6F9 F1         0556          POP  AF       ;RESET BYTE
C6FA 10 04       0557          JR   PRTB1    ;SKIP
C6FC 2A 06 0E    0558 PRTAB1: LD  HL,(MADDR) ;GET NEXT BYTE
C6FF 7E         0559          LD   A,(HL)
C700 FE 20       0560 PRTB1: CP   '/'      ;>=SPACE?
C702 D0         0561          RET  C        ;NO,RETURN
C703 FE 2F       0562          CP   '/'      ;BAR DEFB DELIMITER
C705 37         0563          SCF
C706 C8         0564          RET  Z
C707 FE 7F       0565          CP   $7F
C709 3F         0566          CCF
C70A C7         0567          RET
C70B            0568 ;*****
C70B            0569 ; NOP/DJNZ/JR NZ/JR NC
C70B            0570 ;*****
C70B F1         0571 NOPETC: POP AF      ;RECOVER OPCODE
C70C FE 10       0572          CP   $10     ;DJNZ?
C70E 20 2E       0573          JR   Z,DJNZ   ;YES,JUMP
C710 30 23       0574          JR   C,NOP    ;JUMP ALSO IF<10.
C712            0575 ;
C712            0576 ; JUMP RELATIVE
C712            0577 ;
C712 E0         0578 JR:   EX   DE,HL   ;WRITE "JR"
C713 36 4A       0579          LD   (HL),'J'
C715 23         0580          INC  HL
C716 36 52       0581          LD   (HL),'R'
C718 11 32 0E    0582          LD   DE,BUFFER+30 ;RESET DE
C71B FE 10       0583          CP   $10     ;UNCONDITIONAL?
C71D 20 00       0584          JR   Z,UCD   ;YES,JUMP
C71F E6 10       0585          AND  $10     ;ISOLATE COND. CODES
C721 CD 7C C6    0586          CALL CCODES   ;WRITE APPROPRIATE ONE
C724 CD F7 C4    0587          CALL COMMA    ;","
C727 CD 00 C4    0588 UCD:  CALL BYTE   ;GET OFFSET
C72A 4F         0589          LD   C,A     ;PUT IN C
C72B 07         0590          RLCA        ;EXTEND SIGN THRU' A
C72C 9F         0591          SBC  A
C72D 47         0592          LD   B,A     ;PUT IN B
C72E 2A 04 0E    0593          LD   HL,(ADDR) ;LOAD CURRENT ADDRESS
C731 09         0594          ADD  HL,BC   ;COMPUTE TARGET ADDRESS
C732 C3 5B C5    0595          JP   LD16B   ;WRITE IT ETC.
C735            0596 ;
C735            0597 ; WRITE "NOP"
C735            0598 ;
C735            0599 ;
C735 21 37 CA    0599 NOP:   LD   HL,NOPM
C738 CD 09 C5    0600 COPY3S: CALL COPY3 ;COPY OVER MNEMONIC
C73B 13         0601          INC  DE     ;SPACE ON

```


REVAS subroutine version N1.1 88-83-89

```

C73C 13          $682      INC  DE
C73D  C9          $683      RET
C73E             $684      ;
C73E             $685      ; DJNZ
C73E             $686      ;
C73E 21 3A CA    $687  DJNZ:  LD   HL,DJNZH  ;WRITE "DJNZ"
C741  CD 87 C5   $688      CALL  COPY4
C744  13         $689      INC  DE      ;SPACE
C745  18 E8      $618      JR   UCD     ;REST IN "JR" ROUTINE
C747             $611      ;*****
C747             $612      ; EX AF/AF'/JR/JR Z/JR C
C747             $613      ;*****
C747  F1         $614  EXAETC: POP  AF      ;RECOVER OPCODE
C748  FE 88      $615      CP   8        ;JRT
C74A  28 C6      $616      JR   NZ,JR     ;YES,GO TO THEM
C74C             $617      ;
C74C  CD E5 C4   $618      CALL  UREX    ;WRITE "EX"
C74F  21 3E CA   $619      LD   HL,EXAFH ;WRITE "AF,AF'"
C752  C3 93 C5   $628      JP   COPY6
C755             $621      ;*****
C755             $622      ; JP/OUT/EX (SP),HL/DI
C755             $623      ;*****
C755  F1         $624  JPETC:  POP  AF      ;RECOVER OPCODE
C756  FE D3      $625      CP   $03     ;JUMP?
C758  DA 58 C6   $626      JP   C,CJR+1 ;JUMP IF "JP"
C758  CA EC C8   $627      JP   Z,OUT    ;JUMP IF "OUT"
C75E  FE F3      $628      CP   $F3     ;"DI"?
C768  28 8E      $629      JR   Z,DI     ;YES,JUMP
C762             $630      ;
C762  CD E5 CA   $631      CALL  UREX    ;WRITE "EX"
C763  21 44 CA   $632      LD   HL,BSP8M ;WRITE "(SP),"
C768  CD 95 C5   $633      CALL  COPY5
C768  3E 28      $634      LD   A,$28   ;SET FOR HL/IX/IY
C768  C3 77 C5   $635      JP   REGPR   ;WRITE APPROPRIATE ONE.
C778             $636      ;*****
C778             $637      ; DI/EI
C778             $638      ;*****
C778  3E 44      $639  DI:   LD   A,'D'   ;WRITE "DI"...
C772  18 82      $640      JR   ++2
C774  3E 45      $641  EI:   LD   A,'E'   ;..OR "EI"
C776  EB         $642      EX   DE,HL
C777  77         $643      LD   (HL),A
C778  23         $644      INC  HL
C779  36 49      $645      LD   (HL),'I'
C77B  C9         $646      RET
C77C             $647      ;*****
C77C             $648      ; ROTATE & TEST/IN/EX DE,HL/EI
C77C             $649      ;*****
C77C  F1         $650  CBETC:  POP  AF      ;RECOVER OPCODE
C77D  FE DB      $651      CP   $DB     ;IN?
C77F  CA A8 C8   $652      JP   Z,IN    ;YES,JUMP
C782  DA 8D C8   $653      JP   C,CB    ;JUMP IF "CB"
C785  FE FB      $654      CP   $FB     ;"EI"?
C787  28 EB      $655      JR   Z,EI    ;YES,JUMP
C789             $656      ; EX DE,HL
C789  CD E5 C4   $657      CALL  UREX    ;WRITE "EX"
C78C  21 CC C9   $658      LD   HL,RPRTAB+4 ;WRITE "DE"
C78F  CD 8B C5   $659      CALL  COPY2
C792  CD F7 C4   $660      CALL  COMHA   ;",,"
C795  C3 8B C5   $661      JP   COPY2    ;WRITE "HL" & RETURN

```

REVAS subroutine version M1.1 88-93-88

```

C798          0662 ;*****
C798          0663 ; RET/EXX/JP (HL)/LD SP,HL
C798          0664 ;*****
C798 F1       0665 RETETC: POP AF          ;RECOVER OPCODE
C799 FE D9    0666          CP $D9          ;"RET" OR "EXX"?
C79B DA 58 C6 0667          JP C,CJR+1      ;"RET",SO JUMP
C79E 28 1D    0668          JR Z,EXX        ;"EXX" SO JUMP
C7A0 FE F9    0669          CP $F9          ;"LD SP"?
C7A2 28 8B    0670          JR Z,LDSP       ;YES,JUMP
C7A4 21 FC C9 0671          LD HL,CJRTAB+4 ;WRITE "JP"
C7A7 CD 8B C5 0672          CALL COPY2
C7AA 21 32 8E 0673          LD HL,BUFFER+30 ;RESET HL
C7AD 18 3D    0674          JR LD1+3        ;WRITE HL,IX OR IY
C7AF          0675 ;
C7AF          0676 ; LOAD SP
C7AF          0677 ;
C7AF CD ED C4 0678 LDSP: CALL WRLD          ;WRITE "LD"
C7B2 21 4F CA 0679          LD HL,SPH
C7B3 CD 09 C3 0680          CALL COPY3      ;WRITE "SP"
C7B8 3E 28    0681          LD A,$28        ;MAKE LIKE HL
C7BA C3 77 C5 0682          JP REGPR        ;WRITE HL,IX,OR IY
C7BD          0683 ;
C7BD          0684 ; WRITE "EXX"
C7BD          0685 ;
C7BD CD E5 C4 0686 EXX: CALL UREX          ;WRITE "EX"
C7C0 23       0687          INC HL          ;BUMP ADDRESS
C7C1 34 58    0688          LD (HL),'X'     ;ADD ANOTHER "X"
C7C3 C9       0689          RET
C7C4          0690 ;*****
C7C4          0691 ; LD (BC),A LD (DE),A LD(NN),HL LD,(NN),A
C7C4          0692 ;*****
C7C4 CD ED C4 0693 STIND: CALL WRLD          ;WRITE "LD"
C7C7 F1       0694          POP AF          ;RECOVER OPCODE
C7C8 F5       0695          PUSH AF         ;SAVE IT FOR ST16I
C7C9 FE 22    0696          CP $22          ;16 BIT?
C7CB 28 2D    0697          JR Z,ST16I     ;YES,JUMP
C7CD F1       0698          POP AF          ;NO,THROW AWAY SAVE
C7CE CD E9 C7 0699          CALL LD1        ;WRITE INDIRECT PART
C7D1 CD F7 C4 0700          CALL COMMA      ;","
C7D4 3E 41    0701 ST1: LD A,'A'          ;WRITE "A"
C7D6 12       0702          LD (DE),A
C7D7 13       0703          INC DE
C7D8 C9       0704          RET
C7D9          0705 ;*****
C7D9          0706 ; LD A,(BC) LD A,(DE) LD HL,(NN) LD A,(NN)
C7D9          0707 ;*****
C7D9 CD ED C4 0708 LDIND: CALL WRLD          ;WRITE "LD"
C7DC F1       0709          POP AF          ;RECOVER OPCODE
C7DD F5       0710          PUSH AF         ;SAVE
C7DE FE 2A    0711          CP $2A          ;16 BIT?
C7E0 28 21    0712          JR Z,LD16I     ;YES,JUMP
C7E2 CD D4 C7 0713          CALL ST1        ;"A"
C7E3 CD F7 C4 0714          CALL COMMA      ;","
C7E8 F1       0715          POP AF          ;RECOVER OPCODE
C7E9 FE 22    0716 LD1: CP $22          ;REGPR OR EXTENDED?
C7EB EB       0717          EX DE,HL       ;WRITE "("
C7EC 34 28    0718          LD (HL),'('
C7EE 23       0719          INC HL
C7EF EB       0720          EX DE,HL
C7F0 3F       0721          CCF            ;REVERSE RESULT OF COMPARE

```

REVAS subroutine version Ni.1 88-03-88

```

C7F1 04 77 C5      0722      CALL NC,REGPR      ;(RETURNS WITH NC)
C7F4 0C 32 C5      0723      CALL C,LD16A      ;EXTENDED ADDRESS
C7F7 C3 F0 C5      0724      JP      NOTIXY    ;CLOSE BRACKETS
C7FA              0725 ;
C7FA              0726 ; 16-BIT INDIRECT STORE LD (NNNN),PP
C7FA              0727 ;
C7FA CD E9 C7      0728 ST16I: CALL LD1      ;DO INDIRECT BIT
C7FB CD F7 C4      0729      CALL COMHA        ;", "
C800 C3 76 C5      0730      JP      REGPR-1   ;GET OP & PR. REG PAIR
C803              0731 ;
C803              0732 ; 16-BIT INDIRECT LOAD LD PP,(NNNN)
C803              0733 ;
C803 F1            0734 LD16I: POP AF          ;GET OPCODE
C804 CD 77 C5      0735      CALL REGPR        ;PRINT REG PR.
C807 CD F7 C4      0736      CALL COMHA        ;", "
C80A A7            0737      AND A            ;CLEAR CARRY
C80B 10 0E         0738      JR      LD1+2     ;DO INDIRECT BIT
C80D              0739 ;*****
C80D              0740 ; ROTATE/SHIFT/BIT/SET/RESET
C80D              0741 ; NB. IF INDEXED THEN OFFSET PRECEDES OPCODE
C80D              0742 ;*****
C80D 3A 9C 0E      0743 CB: LD A,(HXYFLG)
C810 A7            0744      AND A            ;INDEXED?
C811 F5            0745      PUSH AF         ;SAVE FLAG
C812 20 00         0746      JR      Z,NOTXY ;NO,SKIP
C814 11 34 0E      0747      LD DE,BUFFER+32 ;YES,WRITE..
C817 3E 06         0748      LD A,6          ;...REG. FIRST.
C819 CD 04 C5      0749      CALL SREG
C81C 11 20 0E      0750      LD DE,BUFFER+25 ;RESET POINTER FOR MNEMONIC.
C81F CD 00 C4      0751 NOTXY: CALL BYTE      ;GET OPCODE
C822 F5            0752      PUSH AF         ;SAVE IT
C823 FE 40         0753      CP      #40     ;(<40?
C825 30 2C         0754      JR      C,ROTATE ;YES,JUMP
C827 21 4F CA      0755      LD HL,BRSTAB-3 ;LOAD POINTER
C82A 07            0756      RLCA           ;SHIFT OPCODE DOWN
C82B 07            0757      RLCA
C82C E6 03         0758      AND 3          ;ISOLATE ID
C82E 47            0759      LD B,A         ;MAKE 3,4,OR ?
C82F 07            0760      RLCA
C830 00           0761      ADD B
C831 CD 10 C5      0762      CALL FTADR      ;FORM ADDRESS
C834 CD 30 C7      0763      CALL COPY3S     ;WRITE MNEMONIC
C837 F1            0764      POP AF         ;GET OPCODE
C838 F5            0765      PUSH AF        ;SAVE AGAIN
C839 0F            0766      RRCA          ;PRINT BIT NUMBER
C83A 0F            0767      RRCA
C83B 0F            0768      RRCA
C83C E6 07         0769      AND 7          ;ISOLATE BIT ID
C83E F6 30         0770      OR      #30     ;MAKE ASCII
C840 12            0771      LD (DE),A      ;WRITE IT
C841 13            0772      INC DE
C842 CD F7 C4      0773      CALL COMHA      ;", "
C843 C1            0774 TESTXY: POP BC          ;RECOVER OPCODE
C844 F1            0775      POP AF         ;RECOVER HXY FLAG
C847 7B            0776      LD A,B         ;LOAD OPCODE
C848 CA 04 C5      0777      JP      Z,SREG  ;EXIT IF NOT HXY
C84B E6 07         0778      AND 7          ;ISOLATE LD BITS OF OP
C84D FE 06         0779      CP      6       ;IS IT AN (HL) ONE?
C84F C8            0780      RET Z          ;YES,RETURN
C850 C3 53 C9      0781      JP      NOTVAL  ;NO,INVALID

```

REVAS subroutine version M1.1 88-93-88

```

C853 9F          9782 ROTATE: RRCA          ;SHIFT DOWN
C854 9F          9783          RRCA
C855 C4 92      9784          ADD 2          ;ROLL CODING ROUND
C857 E6 9E      9785          AND 9E         ;ISOLATE ID
C859 FE 9E      9786          CP 9E          ;IS IT 9E?
C85B CA 31 C9   9787          JP Z,NTVL     ;YES,INVALID CODE
C85E 47         9788          LD B,A        ;DO ID*3 AGAIN
C85F 9F         9789          RRCA
C860 99         9790          ADD B
C861 21 3B CA   9791          LD HL,ROTTAB ;LOAD BASE ADDRESS
C864 CD 18 C5   9792          CALL FTADR    ;FORM ADDRESS
C867 CB 38 C7   9793          CALL COPY3S   ;WRITE MNEMONIC
C86A 18 D9      9794          JR TESTXY    ;EXIT WRITING REGISTER
C86C           9795 ;*****
C86C           9796 ; AUTO CP LD IN OUT
C86C           9797 ;*****
C86C CB 57      9798 AUTO:  BIT 2,A          ;TEST FOR VALIDITY
C86E C2 53 C9   9799          JP NZ,NOTVAL ;JUMP IF NOT
C871 F5         9800          PUSH AF       ;SAVE OPCODE
C872 E6 93      9801          AND 3         ;ISOLATE OP ID
C874 97         9802          RLCA         ;*2
C875 21 79 CA   9803          LD HL,OPTAB  ;LOAD BASE ADDRESS
C878 CB 97 C5   9804          CALL RP2      ;FORM ADDR. & COPY OVER
C87B F1         9805          POP AF        ;RECOVER OPCODE
C87C F5         9806          PUSH AF       ;SAVE AGAIN
C87D 3C         9807          INC A        ;INC UP
C87E E4 13      9808          AND 13       ;ISOLATE
C880 28 94      9809          JR NZ,++4    ;SKIP IF NOT "OUT*"
C882 18         9810          DEC DE       ;BACKSPACE POINTER
C883 CD 8B C5   9811          CALL COPY2    ;MODIFY MNEMONIC
C886 F1         9812          POP AF        ;RESET OPCODE
C887 21 7A CA   9813          LD HL,OPTAB+10 ;LOAD BASE ADDRESS
C88A 9F         9814          RRCA         ;SHIFT ID DOWN
C88B 9F         9815          RRCA
C88C E6 96      9816          AND 6        ;ISOLATE IT
C88E C3 97 C5   9817          JP RP2       ;FINISH MNEMONIC
C891           9818 ;*****
C891           9819 ; ADC/SBC
C891           9820 ;*****
C891 21 E1 C9   9821 ADCSBC: LD HL,ARTAB+9 ;POINTER TO "SBC"
C894 79         9822          LD A,C       ;LOAD OPCODE
C895 F5         9823          PUSH AF      ;SAVE FOR "ADDHL"
C896 CB 5F      9824          BIT 3,A     ;SBC?
C898 28 93      9825          JR Z,++3    ;YES,SKIP
C89A 21 D9 C9   9826          LD HL,ARTAB+3 ;POINT TO "ADC"
C89D C3 6B C5   9827          JP ADDHL+3  ;REST IN "ADDHL"
C8A0           9828 ;*****
C8A0           9829 ; IN A,(HXX) IN r,(C)
C8A0           9830 ;*****
C8A0 3E FF      9831 IN:  LD A,9FF   ;"IN A," ENTRY POINT
C8A2 FE 9E      9832 INRC: CP 9E     ;TRY FOR (HL)?
C8A4 CA 53 C9   9833          JP Z,NOTVAL  ;YES,INVALID
C8A7 21 49 CA   9834          LD HL,INH    ;WRITE "IN"
C8AA CD 38 C7   9835          CALL COPY3S
C8AD F5         9836          PUSH AF      ;SAVE OP
C8AE CD 94 C5   9837          CALL SREG    ;WRITE REG.
C8B1 CD F7 C4   9838          CALL COMHA   ;ADD ", "
C8B4 F1         9839          POP AF       ;RECOVER A
C8B5 EB         9840 PORT: EX DE,HL ;OPEN BRACKETS
C8B6 36 28      9841          LD (HL),('

```

REVAS subroutine version M1.1 98-03-80

```

C8B8 23          9842      INC  HL
C8B9 3C          9843      INC  A
C8BA 28 97      9844      JR   Z,INA
C8BC 36 43      9845      LD   (HL),C
C8BE 23          9846      INC  HL
C8BF EB        9847      EX  DE,HL
C8C0 C3 F9 C5   9848      JP  NOTIXY
C8C1 EB        9849      INAI: EX DE,HL
C8C4 CD F5 C5   9858      CALL INH
C8C7 18 F7      9851      JR   INA-3
C8C9           9852      ;*****
C8C9           9853      ; EXTENDED DECODING
C8C9           9854      ;*****
C8C9 CD B8 C4   9855      EXTND: CALL BYTE
C8CC FE C8      9856      CP   $C8
C8CE D2 53 C9   9857      JP  NC,NOTVAL
C8D1 FE 48      9858      CP   $48
C8D3 DA 53 C9   9859      JP  C,NOTVAL
C8D4 FE A8      9868      CP   $A8
C8D8 38 92      9861      JR  NC,AUTO
C8DA FE 88      9862      CP   $88
C8DC D2 53 C9   9863      JP  NC,NOTVAL
C8DF 4F        9864      LD   C,A
C8E0 E6 97      9865      AND  7
C8E2 47        9866      LD   B,A
C8E3 79        9867      LD   A,C
C8E4 9F        9868      RRCA
C8E5 9F        9869      RRCA
C8E6 9F        9870      RRCA
C8E7 28 B9      9871      JR   Z,INRC
C8E9 18 19      9872      NOTIN: DJNZ NOTOUT
C8EB           9873      ;*****
C8EB           9874      ; OUT (RXX),A OUT (C),R
C8EB           9875      ;*****
C8EB 21        9876      DEFB $21
C8EC 3E FF      9877      OUT: LD  A,$FF
C8EE FE 2E      9878      CP   $2E
C8F0 CA 53 C9   9879      JP  Z,NOTVAL
C8F3 21 4C CA   9888      LD  HL,OUTH
C8F6 CD 38 C7   9881      CALL COPY3S
C8F9 F5        9882      PUSH AF
C8FA CD B5 C8   9883      CALL PORT
C8FD CD F7 C4   9884      CALL CONHA
C900 F1        9885      POP  AF
C901 C3 B4 C5   9886      JP  SREG
C904           9887      ;
C904           9888      ;
C904 05        9889      NOTOUT: DEC  B
C905 28 8A      9890      JR   Z,ADCSBC
C907 18 14      9891      DJNZ NOTLD
C909           9892      ;*****
C909           9893      ; LD PP,(8NNNN) LD (8NNNN),PP
C909           9894      ;*****
C909 E6 8E      9895      AND  $E
C90B FE 8C      9896      CP   $C
C90D CA 53 C9   9897      JP  Z,NOTVAL
C910 CD ED C4   9898      CALL URLD
C913 CB 59      9899      BIT  3,C
C915 79        9900      LD   A,C
C916 C2 84 C8   9901      JP  NZ,LD16I+1

```

```

;TEST OP.
;SKIP IF "IN A" (OR "OUT .,A")
;SET "C"
;RESET ADDRESSES
;CLOSE BRACKETS
;GET & WRITE...
;...PORT NUMBER
;CLOSE BRACKETS

```

```

;GET OPCODE
;>=C8?
;YES,NOT VALID
;<$48?
;YES,NOT VALID
;>=A8?
;YES,AUTO INC/DEC
;>=88?
;YES,NOT VALID
;SAVE OPCODE
;ISOLATE L8 BITS
;PUT IN B
;RELOAD OPCODE
;SHIFT DOWN

```

```

;JUMP IF "IN"
;TEST AGAIN
;*****
;SKIP THE NEXT 2 BYTES
;"OUT -,A" ENTRY POINT
;TRY FOR (HL)?
;YES,INVALID
;POINT TO "OUT"
;WRITE IT
;SAVE OP
;WRITE PORT (OR (C))
;ADD ,
;RECOVER A
;WRITE REG.

```

```

;TEST
;JUMP IF "ADC" OR "SBC"
;TEST AGAIN
;*****
;WRITE "LD"
;WHICH WAY?
;RELOAD OPCODE FOR LD/ST16
;JUMP IF LOAD

```

REVAS subroutine version M1.1 88-83-88

```

C919 F3          8982      PUSH AF          ;AND PUT ON STACK
C91A C3 FA C7   8983      JP ST1&I        ;JUMP TO ST1&I
C91B           8984 ;*****
C91B           8985 ; CLEAR UP THE REST OF EXETENBEB DECODING
C91B           8986 ;*****
C91B 79         8987 NOTLD: LD A,C          ;RELOAD OPCODE
C91E 81 8C 88   8988      LD BC,12        ;REMAINING OPS
C921 21 82 CA   8989      LD HL,REHEX    ;TABLE BASE
C924 ED 81      898A      CPIR           ;CHECK TABLE
C926 28 28     898B      JR NZ,NOTVAL  ;NOT THERE,ERROR
C928 79        898C      LD A,C         ;SET REMAINING COUNT
C929 87        898D      RLCA          ;*2
C92A FE 88     898E      CP 8          ;LD I OR R?
C92C 38 8E     898F      JR C,LDRI     ;YES,SKIP
C92E 87        8990      RLCA          ;*2 AGAIN
C92F 21 7E CA   8991      LD HL,EXTHE-16 ;BASE OF TABLE-16
C932 FE 1C     8992      CP 28        ;IM?
C934 38 12     8993      JR C,INTNOD  ;YES,SKIP
C936 C9 18 C5   8994      CALL FTADR    ;INDEX INTO TABLE
C939 C3 87 C5   8995      JP COPY4     ;COPY MNEMONIC OVER
C93C           8996 ;
C93C           8997 ; LD A,R LD A,I LD R,A LD I,A
C93C           8998 ;
C93C           8999 ;
C93C CD ED C4   899A LDRI: CALL WRLD      ;WRITE "LD"
C93F 21 AE CA   899B LD HL,IRTAB    ;LOAD BASE ADDRESS
C942 CD 18 C5   899C CALL FTADR    ;FORM ADDRESS
C943 C3 89 C5   899D JP COPY3     ;COPY OVER
C948           899E ;
C948           899F ; INTERRUPT MODE
C948           89A0 ;
C948 CD 97 C5   89A1 INTNOD: CALL RP2     ;COPY OVER "IN"
C948 11 32 8E   89A2 LD DE,BUFFER+38 ;SET OPERAND ADDRESS
C94E ED A8      89A3 LDI           ;COPY OVER NODE #
C950 C9        89A4 RET           ;DONE
C951           89A5 ;*****
C951           89A6 ;INVALID CODE. WRITE 'DEFB' AND COPY CODE OVER.
C951           89A7 ;*****
C951 E1        89A8 NTVL: POP HL          ;THROW AWAY TWO RETURNS
C952 E1        89A9 POP HL
C953 21 87 CA   89AA NOTVAL: LD HL,DEFB    ;POINT TO "DEFB"
C956 11 2D 8E   89AB LD DE,BUFFER+25 ;SET FOR MNEMONIC
C959 C8 87 C5   89AC CALL COPY4     ;COPY IT OVER
C95C 13        89AD INC DE         ;SPACE ON
C95D 21 1A 8E   89AE LD HL,BUFFER+6  ;SET FOR OPCODES
C960 8E 15     89AF LD C,CR+8      ;SET TO COUNT DEFBs
C962 3E 28     89B0 LD A,' '
C964 C8 F8 C4   89B1 NVLP: CALL POUND    ;WRITE 'N'
C967 C8 88 C5   89B2 CALL COPY2     ;COPY OVER OP
C96A 23        89B3 INC HL         ;PAST SPACE
C96B 8E        89B4 CP (HL)        ;IS ANOTHER THERE?
C96C 28 85     89B5 JR Z,DNV       ;NO,DONE
C96E C8 F7 C4   89B6 CALL COMMA    ;WRITE ','
C971 18 F1     89B7 JR NVLP       ;COPY IT OVER
C973 21 88 88   89B8 DNV: LD HL,0      ;Clear any pointer
C976 22 12 8E   89B9 LD (LABELP),HL ;SEE HOW MANY ON LINE
C979 79        89BA LD A,C         ;DONE 4?
C97A FE 8D     89BB CP CR         ;NO,RETURN
C97C C9        89BC RET NZ        ;NO,RETURN
C97D 12        89BD LD (DE),A     ;YES,SET CR SO NO RUBBISH
C97E C9        89BE RET

```

REVAS subroutine version M1.1 88-83-88

```

C97F          8962 ;*****
C97F          8963 ; TABLE AREA
C97F          8964 ;*****
C97F          8965 ;
C97F          8966 ;JUMP TABLE FOR DECODE
C97F          8967 ;32 ENTRIES
C97F          8968 ;
C97F 88 C7    8969 TABLE:  DEFU NOPETC
C981 57 C6    8970         DEFU CJR
C983 48 C5    8971         DEFU LD16
C985 28 C6    8972         DEFU POP
C987 C4 C7    8973         DEFU STIND
C989 57 C6    8974         DEFU CJR
C98B 33 C5    8975         DEFU INC
C98D 55 C7    8976         DEFU JPETC
C98F 33 C5    8977         DEFU INC
C991 57 C6    8978         DEFU CJR
C993 38 C5    8979         DEFU DEC
C995 2D C6    8980         DEFU PUSH
C997 9D C5    8981         DEFU LOAD8
C999 84 C6    8982         DEFU ARITH8
C99B 86 C6    8983         DEFU ROTHIS
C99D 93 C6    8984         DEFU RST
C99F 47 C7    8985         DEFU EXAETC
C9A1 57 C6    8986         DEFU CJR
C9A3 68 C5    8987         DEFU ADDHL
C9A5 98 C7    8988         DEFU RETETC
C9A7 D9 C7    8989         DEFU LDIND
C9A9 57 C6    8990         DEFU CJR
C9AB 38 C5    8991         DEFU DEC
C9AD 7C C7    8992         DEFU CBETC
C9AF 33 C5    8993         DEFU INC
C9B1 57 C6    8994         DEFU CJR
C9B3 38 C5    8995         DEFU DEC
C9B5 37 C6    8996         DEFU CALETC
C9B7 9D C5    8997         DEFU LOAD8
C9B9 84 C6    8998         DEFU ARITH8
C9BB 86 C6    8999         DEFU ROTHIS
C9BD 93 C6    9000         DEFU RST
C9BF          9001 ;*****
C9BF 49 4E 43 9002 INCH:  DEFB 'INC'
C9C2 44 45 43 9003 DECH:  DEFB 'DEC'
C9C5 41 46 53 9004         DEFB 'AFS'
C9C8 58 41 42 43 9005 RPRTAB: DEFB 'PABCDE'
      44 45
C9CE 48 4C 49 58 9006 HXYTAB: DEFB 'HLIXY'
      49 59
C9D4 48 41 4C 54 9007 HALTM:  DEFB 'HALT'
C9D8 41 44 44 41 9008 ARTAB:  DEFB 'ADDADCSUBSBC'
      44 43 53 55
      42 53 42 43
C9E4 41 4E 44 58 9009         DEFB 'ANDXOROR CP '
      4F 52 4F 52
      28 43 58 28
C9F0 58 55 53 48 9010 PUSHM:  DEFB 'PUSH'
C9F4 58 4F 58 28 9011 POPM:   DEFB 'POP '
C9F8 32 45 54 28 9012 CJRTAB: DEFB 'RET JP CALL'
      4A 58 28 28
      43 41 4C 4C
CA04 4E 5A 28 5A 9013 CCTAB:  DEFB 'NZ ZNC CPOPE P M'

```

REVAS subroutine version M1.1 88-03-86

	4E 43 28 43		
	58 4F 58 45		
	28 58 28 4D		
CA14	52 4C 43 41	1014 RNTAB:	DEFB 'RLCARRCARLA RRA '
	52 52 43 41		
	52 4C 41 28		
	52 52 41 28		
CA24	44 41 41 28	1015	DEFB 'DAA CPL SCF CCF '
	43 58 4C 28		
	53 43 46 28		
	43 43 46 28		
CA34	52 53 54	1016 RSTM:	DEFB 'RST'
CA37	4E 4F 58	1017 NOPH:	DEFB 'NOP'
CA3A	44 4A 4E 5A	1018 DJNZH:	DEFB 'DJNZ'
CA3E	41 46 2C 41	1019 EXAFH:	DEFB 'AF,AF''''
	46 27		
CA44	28 53 58 29	1020 BSPBH:	DEFB '(SP),'
	2C		
CA49	49 4E 28	1021 INH:	DEFB 'IN '
CA4C	4F 53 54	1022 OUTH:	DEFB 'OUT'
CA4F	53 58 2C	1023 SPH:	DEFB 'SP, '
CA52	42 49 54 52	1024 BRSTAB:	DEFB 'BITRESSET'
	45 53 53 45		
	54		
CA58	53 52 4C 52	1025 ROTTAB:	DEFB 'SRLRLCRRRRL RR SLASRA'
	4C 43 52 52		
	43 52 4C 28		
	52 52 28 53		
	4C 41 53 52		
	41		
CA78	4C 44 43 58	1026 OPTAB:	DEFB 'LBCPINOTUTI D IRDR'
	49 4E 4F 54		
	53 54 49 28		
	44 28 49 52		
	44 52		
CA82	44 45 4D 67	1027 REMEXT:	DEFB \$44,\$45,\$4D,\$47,\$4F,\$46
	6F 46		
CA88	54 5E 57 4F	1028	DEFB \$56,\$5E,\$57,\$4F,\$5F,\$47
	5F 47		
CA8E	49 4D 32 28	1029 EXTNHE:	DEFB 'IM2 IM1 IN9 RLD '
	49 4D 31 28		
	49 4D 38 28		
	52 4C 44 28		
CA9E	52 52 44 28	1030	DEFB 'RRD RETIRETNNEG '
	52 45 54 49		
	52 45 54 4E		
	4E 45 47 28		
CAAE	49 2C 41 2C	1031 IRTAB:	DEFB 'I,A,R,A,I'
	52 2C 41 2C		
	49		
CAB8		1032 ACOHMA:	EQU IRTAB+2
CAB7	44 45 46 42	1033 DEFB:	DEFB 'DEFB'
CABB		1034	*****
	CABB	1035 REVASC:	EQU *
		1036	END

CROSS REFERENCE

ACONMA	CAB0	0424
ADCSBC	C871	0890
ADDHL	C568	0827 0987
ADDR	0E04	0100 0129 0180 0182 0593
ARG3	0C10	0825 0838
ARG5	0060	0021
ARITH0	C604	0262 0982 0998
ARTAB	C9D0	0311 0420 0821 0826
AUTO	C86C	0861
BORN	C6B0	0539 0548
BRSTAB	CA52	0735
BSPBH	CA44	0632
BUFFER	0E14	0117 0123 0142 0218 0523 0550 0582 0673 0747 0750 0933 0942 0945
BYTE	C4B0	0110 0299 0301 0380 0400 0452 0533 0541 0553 0588 0751 0855
CALETC	C437	0996
CA	C80D	0633
CBETC	C77C	0992
CCODES	C67C	0476 0586
CCTAB	CA04	0480
CJR	C457	0446 0626 0667 0970 0974 0978 0986 0990 0994
CJR TAB	C9F0	0466 0671
CONMA	C4F7	0290 0315 0353 0480 0507 0660 0700 0714 0729 0736 0773 0838 0884 0953
COPY2	C50B	0331 0340 0426 0428 0659 0661 0672 0811 0949
COPY3	C509	0600 0680 0920
COPY3S	C738	0284 0312 0423 0503 0763 0793 0835 0881
COPY4	C507	0407 0436 0468 0490 0600 0921 0943
COPY5	C505	0633
COPY6	C503	0620
CR	000D	0036 0141 0946 0950
DEC	C538	0979 0991 0995
DECH	C9C2	0283
DECODE	C515	0111 0450
DEFB	CAB7	0941
DI	C770	0629
DJNZ	C73E	0573
DJNZH	CA3A	0607
DNV	C973	0952
DONEN	C6D1	0532
EADDR	0E90	0102 0104
EI	C774	0655
ESC	0010	0046
EXAETC	C747	0985
EXAFH	CA3E	0619
EXTNHE	CABE	0917
EXTND	C8C9	0444
EXX	C70D	0660
FLUSH	C6E9	0514 0519
FTADR	C510	0269 0339 0365 0421 0467 0497 0762 0792 0920 0927
HALT	C5FE	0346
HALTH	C9D4	0406
HEX1	C4A0	0162
HEX2	C4A2	0151 0190 0402
HEX4	C49D	0131 0307
HXYFLG	0E0C	0112 0120 0336 0337 0372 0373 0451 0743
HXYTAB	C9CE	0371
INH	C5F5	0356 0850
IN	C8A0	0652

CROSS REFERENCE

INA	C8C3	0844 0851
INC	C533	0975 0977 0993
INCH	C98F	0281
INITB	C46E	0109 0552
INM	CA49	0834
INRC	C8A2	0871
INTMOD	C948	0919
IRTAB	CAAE	0926 1032
JPETC	C753	0976
JR	C712	0616
L8B	C5AF	0429
LABELA	0E10	0305
LABELP	0E12	0127 0304 0956
LD1	C7E9	0674 0699 0728 0738
LD16	C548	0971
LD16A	C552	0401 0723
LD16B	C55B	0595
LD16I	C803	0712 0901
LDIND	C7D9	0909
LBRI	C93C	0915
L8SP	C7AF	0670
LINEA	0E0E	0130
LOADS	C59D	0260 0981 0997
NADDR	0E06	0101 0105 0183 0186 0550
MEH	C5C4	0363
MORENS	C6C5	0336
MS1	C6CD	0530
NEXTL	C449	0119
NOF	C735	0574
NOFETC	C70B	0969
NOFH	CA37	0599
NOTIN	C8E9	
NOTIXY	C5F0	0378 0383 0724 0840
NOTLD	C91D	0891
NOTOUT	C944	0872
NOTSP	C58D	0328
NOTVAL	C953	0116 0343 0347 0555 0701 0799 0833 0857 0859 0863 0879 0897 0911
NOTXY	C81F	0746
NTVL	C951	0707
NUMARG	0C0B	0017
NVLP	C964	0934
OPCADR	0E0A	0134 0187 0192
OPTAB	CA70	0803 0813
OUT	C8EC	0627
OUTH	CA4C	0880
PATCH	0E00	0013 0016 0110 0551
PNEX2	C5FB	0392 0506
PLUS	C5E8	0304
POP	C620	0972
POPH	C9F4	0433
PORT	C805	0803
POUND	C4FD	0306 0401 0940
PRINT	C420	0015 0037
PRTABL	C6FC	0531 0544
PRTBI	C700	0557
PUSH	C62D	0900
PUSHH	C9F0	0435
REGPR	C577	0207 0297 0314 0430 0635 0602 0722 0730 0735
RENEXT	CA02	0909

CROSS REFERENCE

RETETC	C798	8988					
RETURN	885B	8849					
REVAS	C43E	8829					
REVASC	CABB	8819					
RIN	8888	8845					
RNTAB	CA14	8496					
ROTATE	C853	8754					
ROTNIS	C686	8983	8999				
ROTTAB	CA5B	8791					
RP2	C597	8335	8375	8489	8884	8817	8932
RPRTAB	C9C8	8327	8364	8658			
RST	C693	8984	1888				
RSTFL8	8E93	8814	8587				
RSTH	CA34	8582					
SAVE	C43E	8899					
SCAL	8818	8828	8848				
SPH	CA4F	8679					
SREG	C884	8291	8352	8749	8777	8837	8886
ST1	C7D4	8713					
ST16I	C7FA	8697	8983				
STIND	C7C4	8973					
TABLE	C97F	8268					
TESTXY	C845	8794					
UCD	C727	8584	8618				
UN8	C4D6	8546					
UN1	C6E4	8545					
UNCND	C679	8474					
UNPRN	C6D9	8521					
UREX	C4E3	8618	8631	8657	8686		
WRLD	C4ED	8295	8348	8678	8693	8788	8898
WRLD8	C4F3	8218					

Interactive control program for REVAS

```

C400          0002 ;
C400          0003 ; EQUATES TO MAS-SYS ROUTINES
C400          0004 ;
          0005 RIN:      EQU    0
          0010          0006 SCAL:    EQU    $18
          0020          0007 PRS:     EQU    $20
          0030          0008 ROUT:    EQU    $30
          0030          0009 RDEL:    EQU    $38
          0050          0010 RETURN:  EQU    $50
          0060          0011 ARSS:    EQU    $60
          0062          0012 POLLK:   EQU    $62
          0063          0013 INLINE:  EQU    $63
          0064          0014 NUH:     EQU    $64
          006A          0015 CRLF:    EQU    $6A
          006B          0016 ERRH:    EQU    $6B
          006F          0017 SRLX:    EQU    $6F
          0079          0018 RLIN:    EQU    $79
          007B          0019 BLINK:   EQU    $7B
          0C0B          0020 ARGN:    EQU    $C0B
          0C21          0021 NUMV:    EQU    $C21
C400          0022 ;
C400          0023 ; EQUATES TO REVAS ROUTINES
C400          0024 ;
          C43E          0025 REVAS:   EQU    ++$3E
          C46E          0026 INITB:   EQU    ++$4E
          C49D          0027 HEX4:    EQU    ++$9D
          C4B8          0028 BYTE:    EQU    ++$B8
          C953          0029 NOTVAL:  EQU    ++$53
C400          0030 ;
          000A          0031 LF:      EQU    $A
          000D          0032 CR:      EQU    $D
          001B          0033 ESC:     EQU    $1B
C400          0034 ;*****
C400          0035 ; VARIABLES AND BUFFERS
C400          0036 ;*****
          CABB          0037 SAVE:    EQU    ++$6B
          0038          0038 ORG     $E00
          0E00 0003      0039 PATCH:  DEFS  3          ;RAM PATCH TO O/P ROUTINE
          0E03 0001      0040 RSTFLG: DEFS  1          ;RST HANDLING FLAG
          0E04 0002      0041 ADDR:   DEFS  2
          0E06 0002      0042 MADDR:  DEFS  2
          0E08 0002      0043 EADDR:  DEFS  2
          0E0A 0002      0044 OPCADR:  DEFS  2
          0E0C 0002      0045 HXYFLG: DEFS  2
          0E0E 0002      0046 LINEA:  DEFS  2          ;ADDR. LINE REF. LABEL
          0E10 0002      0047 LABELA: DEFS  2          ;LABEL VALUE
          0E12 0002      0048 LABELP: DEFS  2          ;POINTER TO POS IN BUFFER
          0E14 0030      0049 BUFFER:  DEFS  40         ;OUTPUT BUFFER
          0E44 0001      0050 FLAG:   DEFS  1          ;FLAG BYTE
          0E45 0001      0051 STEP:   DEFS  1          ;SINGLE STEP FLAG
          0E46 0002      0052 START:  DEFS  2          ;START OF DATA ON STACK
          0E48 0002      0053 FIRST:  DEFS  2          ;RANGE
          0E4A 0002      0054 LAST:   DEFS  2          ;....
          0E4C 0002      0055 POINT:  DEFS  2
          0E4E 0002      0056 SYMST:  DEFS  2          ;SYMBOL TABLE START
          0E50 0002      0057 SYMEND: DEFS  2          ;SYMBOL TABLE END
          0E52 0002      0058 PSYN:   DEFS  2          ;PRESENT POSITION IN TABLE
          0E54 0002      0059 OFFSET: DEFS  2
          0E56 0002      0060 ZEAPP:  DEFS  2          ;POINTER INTO ZEAP FILE
          0E58 0024      0061 SEQU:   DEFS  36

```

Interactive control program for REVAS

0E7C	0001	0062 LC:	DEFS 1	;LINE COUNT
0E7D	0001	0063 GAP:	DEFS 1	
0E7E	0002	0064 LINES:	DEFS 2	
0E80	0002	0065 LINENO:	DEFS 2	;ZEAP LINE NUMBER
0E82	0002	0066 ZLABEL:	DEFS 2	;ZEAP LABEL COUNT*2
0E84	0008	0067 HEAD:	DEFS 8	
0E8C	0006	0068 PAGE:	DEFS 6	
0E92	0031	0069 TITLE:	DEFS 49	;ROOM FOR TITLE

Interactive control program for REVAS

```

0EC3          0071 ;
0EC3          0072 ;   A CONTROL PROGRAM FOR REVAS
0EC3          0073 ;
0EC3          0074 ;   WRITTEN BY DAVID PARKINSON
0EC3          0075 ;
0EC3          0076 ;   VERSION N1.1 00-03-00
0EC3          0077 ;
0EC3          0078 ;   ORG   SAVE
CABB          0079 ;*****
CABB          0080 ;   INITIALISATION FOR DATA AREA
CABB          0081 ;*****
CABB 11 00 0E 0082 REVASC: LD   DE,LINENO
CABE 21 E5 CF 0083         LD   HL,RAMLN
CAC1 01 12 00 0084         LD   BC,MOVLNG
CAC4 ED 00 0085         LDIR
CAC6          0086 ;*****
CAC6          0087 ;   FIRST SET UP ALL PROGRAM CONTROL VARIABLES
CAC6          0088 ;*****
CAC6 EF 0089 OPTION: RST  PRS           ;ASK FOR OPTIONS
CAC7 00 4F 70 74 0090         DEFB CR,'Options? ('
      6F 6E 6E 73
      3F 20 20
CAD2 53 54 5A 50 0091 OTAB:  DEFB 'STZXPDR' ;OPTION TABLE
      4C 30 44 32
CADA 53 29 2D 00 0092         DEFB 'U)-',0
CADE 11 00 00 0093         LD   DE,0           ;CLEAR D & E
CAE1 3E 01 0094         LD   A,1           ;SET LSB OF A
CAE3 01 0095         DEFB 1           ;SKIP THE NEXT TWO BYTES
CAE4 06 55 0096 FOUND: SUB  'U'         ;IS IT U?
CAE6 20 03 0097         JR   NZ,++3       ;NO,LEAVE ALONE
CAE8 32 03 0E 0098         LD   (RSTFLG),A      ;SET FLAG
CAEB 70 0099         LD   A,E           ;SET APP. BIT IN FLAG
CAEC 02 0100         OR    0
CAED 57 0101         LD   D,A
CAEE CF 0102         RST  RIN           ;READ A CHARACTER
CAEF F7 0103         RST  ROUT         ;ECHO THE CHARACTER
CAF0 37 0104         SCF                    ;SET FLAG
CAF1 1E 00 0105         LD   E,0           ;CLEAR E
CAF3 21 D2 CA 0106         LD   HL,OTAB        ;LOOK UP OPTION
CAF4 CB 13 0107 LUOP:  RL   E           ;ROTATE E
CAF8 ED A1 0108         CPI                    ;CHECK AGAINST TABLE
CAFA 20 E0 0109         JR   Z,FOUND        ;JUMP IF THERE
CAFC 30 F0 0110         JR   NC,LUOP        ;TRY AGAIN IF FLAG NOT THRU'
CAFE FE 0D 0111         CP    CR           ;END?
CB00 20 C4 0112         JR   NZ,OPTION      ;UNRECOGNISED,TRY AGAIN
CB02 7A 0113         LD   A,D           ;GET OPTIONS
CB03 CB 57 0114         BIT  2,A           ;ZEAP FILE?
CB05 20 02 0115         JR   Z,++2         ;NO,SKIP
CB07 F6 11 0116         OR    $11         ;YES,SET LABEL & SOURCE
CB09 32 44 0E 0117         LD   (FLAG),A
CB0C F3 0118         PUSH AF           ;SAVE FLAG
CB0D          0119 ;
CB0D          0120 ; OPTIONS SET IN FLAG
CB0D          0121 ;
CB0D          0122 ;   BIT           IF SET
CB0D          0123 ;"S" 0   - Source file.(No addr/ops or format)
CB0D          0124 ;"T" 1   - Tape output
CB0D          0125 ;"Z" 2   - ZEAP source file to memory
CB0D          0126 ;"X" 3   - Xreference table
CB0D          0127 ;"L" 4   - Labelled listing

```

Interactive control program for REVAS

```

CB08          0128 ;"P" 5 - Paginated listing
CB08          0129 ;"B" 6 - Insert end-of-line delay
CB08          0130 ;"R" 7 - Range limitation on listing
CB08          0131 ;
CB08 E6 1C    0132      AND  01C      ;XREF,LABELS OR ZEAP?
CB0F 28 38    0133      JR    Z,NLASS  ;NO,SKIP
CB11          0134 ;
CB11          0135 ; GET SYMBOL TABLE MEMORY BOUNDS
CB11          0136 ;
CB11 F1       0137 GETSTA: POP  AF      ;RESET FLAG IN A
CB12 F5       0138      PUSH AF
CB13 C8 57    0139      BIT   2,A      ;ZEAP FILE?
CB15 28 08    0140      JR    Z,ST     ;NO,DIFFERENT MESSAGE
CB17 EF       0141      RST   PRS
CB18 5A 45 41 58 0142      DEFB  'ZEAP fl',0
      28 66 69 00
CB20 18 0C    0143      JR    ST0
CB22 EF       0144 ST:   RST   PRS
CB23 53 79 6D 62 0145      DEFB  'Symbol tab',0
      6F 6C 28 74
      61 62 00
CB2E EF       0146 ST0:  RST   PRS
CB2F 6C 65 28 61 0147      DEFB  'Is area?',CR,0
      72 65 61 3F
      00 00
CB39 C8 FE CC  0148      CALL  GETTWO  ;GET BOUNDS
CB3C 38 03    0149      JR    C,GETSTA ;REPEAT ON ERROR
CB3E EB 43 4E 0E 0150      LD   (SYNST),BC
CB42 ED 53 58 0E 0151      LD   (SYHEND),DE
CB46 3E FF    0152      LD   A,0FF   ;SET EOT MARKER
CB48 12       0153      LD   (DE),A
CB49          0154 ;
CB49          0155 ; LOAD TITLE IF PAGINATED & SET PAGE NO.
CB49          0156 ;
CB49 F1       0157 NLABS: POP  AF      ;RECOVER FLAG
CB4A C8 6F    0158      BIT   5,A
CB4C 28 38    0159      JR    Z,NOITL  ;SKIP IF NONE
CB4E EF       0160 ASKT: RST   PRS     ;ELSE ASK FOR IT
CB4F 54 69 74 6C 0161      DEFB  'Title?',CR,0
      65 3F 00 00
CB57 DF       0162      RST   SCAL   ;READ IT
CB58 63       0163      DEFB  INLINE
CB59 21 3C 05 0164      LD   HL,053C  ;DEFAULT VALUES FOR LISTING
CB5C 1A       0165      LD   A,(DE)  ;GET FIRST CHARACTER
CB5D FE 3D    0166      CP    'a'    ;REDEFINE?
CB5F 28 08    0167      JR    NZ,SETLPP ;NO,SKIP
CB61 13       0168      INC   DE     ;YES,SPACE PAST "a"
CB62 DF       0169      RST   SCAL
CB63 64       0170      DEFB  NUM    ;READ VALUE
CB64 38 EB    0171      JR    C,ASKT  ;REPEAT ON ERROR
CB66 2A 21 0C 0172      LD   HL,(NUMV) ;GET NEW VALUE
CB69 22 7E 0E 0173 SETLPP: LD   (LINES),HL ;SET VALUES
CB6C 21 01 01 0174      LD   HL,0101  ;SET FOR FIRST ONE
CB6F 22 7C 0E 0175      LD   (LC),HL
CB72 EB       0176      EX   DE,HL   ;SOURCE TO HL
CB73 11 92 0E 0177      LD   DE,TITLE ;BUFFER
CB76 AF       0178      XOR   A      ;CLEAR A
CB77 EB A0    0179      LDI  ;COPY OVER UNTIL NULL
CB79 BE       0180      CP   (HL)
CB7A 28 FB    0181      JR   NZ,*-5

```

Interactive control program for REVAS

```

CB7C EB          #182      EX  DE,HL      ;END TO HL
CB7D 3E 29      #183      LD  A,' '    ;NOW GO BACK DOWN BUFFER...
CB7F 2B          #184      DEC  HL
CB80 0E          #185      CP   (HL)    ;..LOOKING FOR FIRST...
CB81 2B FC      #186      JR   Z,+4    ;...NON SPACE.
CB83 23          #187      INC  HL      ;CORRECT HL
CB84 36 9D      #188      LD   (HL),CR ;SET CR AT END
CB86            #189 ;
CB86            #190 ; NOW FIND OUT WHAT TO RUN ON
CB86            #191 ;
CB86 EF        #192 NOTITL: RST  PRS      ;GET ADDRESSES FOR REVAS
CB87 57 68 61 74 #193      DEFB  'What on?',CR,0
      29 6F 6E 3F
      9D 99

CB91 DF        #194      RST  SCAL    ;READ LINE
CB92 63        #195      DEFB  INLINE
CB93 DF        #196      RST  SCAL    ;GET ADDRESSES TO ARGS
CB94 79        #197      DEFB  RLIN
CB95 38 EF     #198      JR   C,NOTITL ;TRY AGAIN IF NON-HEX
CB97 3A 9B 9C  #199      LD   A,(ARGN) ;NO. OF ARGS
CB9A FE 82     #200      CP   2       ;CHECK >=2 ENTERED
CB9C 38 E9     #201      JR   C,NOTITL ;NO,REPEAT QUESTION
CB9E DF        #202      RST  SCAL    ;GET ARGUMENTS
CB9F 60        #203      DEFB  ARGS
CBA0 13        #204      INC  DE      ;SAVE END ADDRESS
CBA1 ED 53 14 0E #205      LD   (BUFFER),DE
CBA5 34        #206      LD   D,H     ;HL->DE
CBA6 5D        #207      LD   E,L
CBA7 2B        #208      DEC  HL     ;ADJUST
CBA8 ED 73 46 0E #209      LD   (START),SP ;SET POINTER
CBAC E5        #210      PUSH HL    ;SAVE START
CBAD 23        #211      INC  HL     ;CORRECT
CBAE 28 82     #212      JR   Z,+2    ;DON'T LOAD THIRD IF ONLY 2
CBB0 69        #213      LD   H,B     ;OVERWRITE WITH 3RD
CBB1 69        #214      LD   L,C
CBB2 ED 52     #215      SBC  HL,DE   ;COMPUTE OFFSET
CBB4 22 54 0E  #216      LD   (OFFSET),HL ;SAVE
CBB7            #217 ;
CBB7            #218 ; CHECK FOR RESTRICTED RANGE TO PRINT
CBB7            #219 ;
CBB7            #220 ;
CBB7 01 00 00  #220      LD   BC,0    ;LOAD DEFAULT VALUES
CBBA 11 FF FF  #221      LD   DE,-1
CBBB 21 44 0E  #222      LD   HL,FLAG ;POINT TO FLAG BYTE
CBC0 CB 7E     #223      BIT  7,(HL) ;RANGE WANTED?
CBC2 28 18     #224      JR   Z,DA    ;NONE,SKIP
CBC4 CB 9E     #225      RES  7,(HL) ;CLEAR FOR PASS 1 FLAG
CBC6 EF       #226 RNRGR: RST  PRS      ;ASK FOR RANGE
CBC7 4C 69 73 74 #227      DEFB  'Listing range?',CR,0
      69 6E 67 29
      72 61 4E 67
      65 3F 9D 99

CBD7 CD FE CC  #228      CALL GETTWO  ;GET IT
CBDA 38 EA     #229      JR   C,RNRGR ;REPEAT ON ERROR
CBDC ED 43 48 0E #230 DA:  LD   (FIRST),BC ;SET VALUES
CBE0 ED 53 4A 0E #231      LD   (LAST),DE
CBE4            #232 ;
CBE4            #233 ; ASK FOR ANY DATA AREAS
CBE4            #234 ;
CBE4 EF       #235 BAREA: RST  PRS      ;ASK FOR DATA AREAS
CBE5 44 41 54 41 #236      DEFB  'DATA areas?',CR,0

```


Interactive control program for REVAS

```

20 61 72 65
61 73 3F 8D
00
CBF2 AF          #237      XOR   A          ;CLEAR COUNT
CBF3 F5          #238      PUSH  AF         ;SAVE COUNT
CBF4 CD FE CC    #239 DAREA0: CALL  GETTWO     ;GET NEXT PAIR
CBF7 38 27      #240      JR    C,DAREA1  ;SKIP ON ERROR
CBF9 60          #241      LD    H,B        ;ADJUST VALUES BY OFFSET
CBFA 69          #242      LD    L,C
CBFB ED 4B 54 0E #243      LD    BC,(OFFSET)
CBFF C5          #244      PUSH  BC
CC00 ED 42      #245      SBC  HL,BC
CC02 E3          #246      EX   (SP),HL
CC03 C1          #247      POP  BC
CC04 EB          #248      EX   DE,HL
CC05 B7          #249      DR   A          ;CLEAR CARRY
CC06 ED 52      #250      SBC  HL,DE
CC08 EB          #251      EX   DE,HL
CC09 F1          #252      POP  AF         ;RECOVER COUNT
CC0A E1          #253      POP  HL         ;GET PREVIOUS
CC0B E5          #254      PUSH HL
CC0C F5          #255      PUSH AF        ;RESET
CC0D 37          #256      SCF          ;COMPARE THEN
CC0E 23          #257      INC  HL
CC0F ED 42      #258      SBC  HL,BC
CC11 30 24      #259      JR   NC,DAREA2  ;JUMP ON ERROR
CC13 2A 14 0E   #260      LD   HL,(BUFFER) ;CHECK END>=2ND
CC14 ED 52      #261      SBC  HL,DE
CC18 38 10      #262      JR   C,DAREA2
CC1A F1          #263      POP  AF         ;REMOVE COUNT
CC1B C5          #264      PUSH BC        ;OK,SO SAVE
CC1C D5          #265      PUSH DE
CC1D 3C          #266      INC  A         ;BUMP COUNT
CC1E 10 D3      #267      JR   DAREA0-1   ;GO GET NEXT
CC20 A7          #268 DAREA1: AND  A         ;TEST A
CC21 28 10      #269      JR   Z,DAREA3   ;CONTINUE IF END
CC23 FE 2D      #270      CP   '-'        ;DELETE LAST?
CC25 20 10      #271      JR   NZ,DAREA2  ;NO,ERROR
CC27 F1          #272      POP  AF         ;RECOVER COUNT
CC28 A7          #273      AND  A         ;ZERO?
CC29 28 B9      #274      JR   Z,DAREA    ;YES,START AGAIN
CC2B E1          #275      POP  HL         ;NO,DELETE LAST
CC2C E1          #276      POP  HL
CC2D 3D          #277      DEC  A
CC2E F5          #278      PUSH AF        ;RESET COUNT
CC2F EF          #279      RST  PRS       ;MOVE CURSOR BACK
CC30 13 1B 13 1B #280      DEFB $13,$ESC,$13,$ESC,$
00
CC35 10 8D      #281      JR   DAREA0     ;LOOP
CC37 DF          #282 DAREA2: RST  SCAL    ;PRINT 'Error'
CC38 60          #283      DEFB ERRH
CC39 EF          #284      RST  PRS       ;MOVE CURSOR BACK
CC3A 13 13 17 00 #285      DEFB $13,$13,$17,$
CC3E 10 B4      #286      JR   DAREA0     ;TRY AGAIN
CC40          #287 ;
CC40 F1          #288 DAREA3: POP  AF         ;REMOVE COUNT
CC41 2A 14 0E   #289      LD   HL,(BUFFER) ;GET END ADDRESS
CC44 E5          #290      PUSH HL        ;ADD TO LIST
CC45 21 FF FF   #291      LD   HL,-1     ;SET EOT MARKER
CC48 E5          #292      PUSH HL

```

Interactive control program for REVAS

```

CC49          0293 ;
CC49          0294 ; GET SINGLE STEP/RUN CHARACTER
CC49          0295 ;
CC49 EF      0296      RST PRS
CC4A 47 6F 3F 00 0297      DEFB 'Go?',0
CC4E DF      0298      RST SCAL
CC4F 78      0299      DEFB BLINK
CC50 32 45 0E 0300      LD (STEP),A      ;SAVE IN STEP FLAG
CC53 DF      0301      RST SCAL      ;NEW LINE
CC54 6A      0302      DEFB CRLF
CC55          0303 ;*****
CC55          0304 ; EVERYTHING NOW SET UP, LET'S GO
CC55          0305 ;*****
CC55 3A 44 0E 0306      LD A,(FLAG)      ;SEE IF PASS1 NEEDED
CC58 E6 18 0307      AND $18      ;XREF OR LABELST
CC5A 21 88 CD 0308      LD HL,PASS1      ;(SET O/P DRIVER ADDR)
CC5D C4 37 CD 0309      CALL NZ,PASS      ;YES,DO FIRST PASS
CC60 21 44 0E 0310      LD HL,FLAG      ;POINT TO FLAG
CC63 C8 FE 0311      SET 7,(HL)      ;SET PASS2 FLAG
CC65 C8 56 0312      BIT 2,(HL)      ;ZEAP FILE?
CC67 28 23 0313      JR Z,ALLIN0      ;NO,SKIP
CC69          0314 ;
CC69          0315 ; FOR ZEAP FILE - SET ADDRESSES IN ZEAP
CC69          0316 ; WORKSPACE AND INITIALISE FILE
CC69          0317 ;
CC69 2A 58 0E 0318      LD HL,(SYEND)      ;SET END FILE AREA
CC6C 22 02 0F 0319      LD ($F02),HL
CC6F 2A 4E 0E 0320      LD HL,(SYMST)      ;SET START OF AREA
CC72 22 08 0F 0321      LD ($F08),HL
CC75 ED 58 52 0E 0322      LD DE,(PSYN)      ;GET CURRENT END OF TABLE
CC79 ED 53 4E 0E 0323      LD (SYNST),DE      ;SET IN SYNST
CC7D ED 58 82 0E 0324      LD DE,(ZLABEL)      ;SET LABEL COUNT
CC81 23 0325      INC HL      ;...AT START OF FILE.
CC82 23 0326      INC HL
CC83 73 0327      LD (HL),E
CC84 23 0328      INC HL
CC85 72 0329      LD (HL),D
CC86 23 0330      INC HL
CC87 36 00 0331      LD (HL),0      ;SET INITIAL NULL
CC89 22 56 0E 0332      LD (ZEAPP),HL      ;SET FILE POINTER
CC8C          0333 ;
CC8C 21 25 CE 0334 ALLIN0: LD HL,PASS2      ;(SET O/P DRIVER ADDR.)
CC8F CD 37 CD 0335      CALL PASS      ;GO DO IT
CC92 21 44 0E 0336      LD HL,FLAG      ;POINT TO FLAG
CC95 C8 0E 0337      RES 7,(HL)      ;CLEAR PASS 2 FLAG
CC97 C8 66 0338      BIT 4,(HL)      ;LABELS?
CC99 E5 0339      PUSH HL      ;(SAVE HL)
CC9A C4 02 CF 0340      CALL NZ,LABEL      ;YES,PRINT REMAINING.
CC9D E1 0341      POP HL      ;(RESET HL)
CC9E C8 5E 0342      BIT 3,(HL)
CCA0 28 02 0343      JR NZ,++2      ;YES,DO IT
CCA2 DF 0344 EXIT: RST SCAL      ;NO,RETURN
CCA3 58 0345      DEFB RETURN
CCA4          0346 ;
CCA4          0347 ; PRINT THE CROSS REFERENCE TABLE
CCA4          0348 ;
CCA4 7E 0349      LD A,(HL)      ;GET FLAG
CCA5 21 7D 0E 0350      LD HL,GAP      ;POINT TO GAP
CCA8 35 0351      DEC (HL)      ;REDUCE BY ONE
CCA9 28 0352      DEC HL      ;POINT TO LC

```

Interactive control program for REVAS

```

CCAA ES          #353      PUSH HL          ;(SAVE HL)
CCAD CB 6F      #354      BIT 5,A          ;CHECK IF PAGED
CCAD C4 80 CF   #355      CALL NZ,EJECT   ;NEW PAGE IF PAGED
CCB0 E1         #356      POP HL          ;RECOVER POINTER TO LC
CCB1 34         #357      INC (HL)        ;INCREASE BY ONE
CCB2 21 FF FF   #358      LD HL,$FFFF    ;RESET RANGE HI LIMIT
CCB5 22 4A 0E   #359      LD (LAST),HL
CCB8 FB 2A 50 0E #360      LD IY,(SYNEND) ;SET POINTER TO TABLE
CCBC FB 7E 00   #361 XREF1: LD A,(IY)        ;CHECK FOR EOT
CCBF 3C         #362      INC A          ;THERE?
CCC0 28 E0      #363      JR Z,EXIT     ;YES,FINISHED
CCC2 FD 2B      #364      DEC IY        ;SPACE ON
CCC4 4F         #365      LD C,A        ;PICK UP COUNT
CCC5 04 09      #366      LD B,9        ;SET 9 ENTRIES MAX ON LINE
CCC7 11 14 0E   #367      LD B,BUFFER   ;POINT TO BUFFER
CCCA FD 6E 00   #368 XREF2: LD L,(IY)        ;LOAD REFERENCE
CCCD FD 2B      #369      DEC IY
CCCF FD 66 00   #370      LD H,(IY)
CCD2 FD 2B      #371      DEC IY
CCD4 CD 9B C4   #372      CALL HEX4     ;WRITE TO BUFFER
CCD7 3E 20      #373      LD A,$20     ;SPACE
CCD9 12         #374      LD (DE),A
CCDA 13         #375      INC DE
CCDB 00         #376      DEC C          ;MORE?
CCDC 28 02      #377      JR Z,+2      ;NO, DONE ENTRY
CCDE 10 EA      #378      BJNZ XREF2   ;LOOP IF MORE ON LINE
CCE0 C5         #379      PUSH BC      ;SAVE COUNT
CCE1 E9         #380      EX DE,HL    ;PUT CR AT END
CCE2 36 00      #381      LD (HL),CR
CCE4 21 14 0E   #382      LD HL,BUFFER ;POINT TO BUFFER
CCE7 CB 40 CE   #383      CALL OUTPUT  ;PRINT LINE
CCEA C1         #384      POP BC      ;GET COUNT BACK
CCEB 79         #385      LD A,C        ;WAS IT ALL?
CCEC A7         #386      AND A        ;CHECK
CCE0 28 CD      #387      JR Z,XREF1   ;YES, TRY NEXT
CCEF 21 14 0E   #388      LD HL,BUFFER ;CLEAR START OF BUFFER
CCF2 06 05      #389      LD B,5
CCF4 36 20      #390      LD (HL),
CCF6 23         #391      INC HL
CCF7 10 FB      #392      BJNZ #-5
CCF9 EB         #393      EX DE,HL    ;ADDRESS TO DE
CCFA 06 00      #394      LD B,0       ;RESET B
CCFC 10 CC      #395      JR XREF2     ;DO NEXT LINE
CCFE           #396 ;*****
CCFE           #397 ; UTILITY SUBROUTINE - GETS TWO ADDRESSES
CCFE           #398 ; TO BC & DE. CHECKS BC>=DE
CCFE           #399 ;*****
CCFE DF         #400 GETTWO: RST SCAL      ;READ LINE
CCFF 43         #401      DEFB INLN
CDB0 3E 1B      #402      LD A,ESC     ;ENSURE LINE CLEAR
CDB2 F7         #403      RST ROUT
CDB3 05         #404      PUSH DE      ;SAVE POINTER
CDB4 0F         #405      RST SCAL    ;GET ADDRESSES FROM IT
CDB5 79         #406      DEFB RLIN
CDB6 01         #407      POP DE      ;RESET DE
CDB7 1A         #408      LD A,(DE)   ;LOAD FIRST CHARAC. IN LINE
CDB8 08         #409      RET C        ;RETURN ON ERROR
CDB9 3A 0B 0C   #410      LD A,(ARGN) ;GET # ENTERED
CDBC FE 02      #411      CP 2         ;TWO INT?
CDBE 37         #412      SCF         ;SET ERROR FLAG

```

Interactive control program for REVAS

```

CB0F C0          #413      RET    NZ          ;RETURN IF NOT TWO
CB10 0F          #414      RST    SCAL        ;OK,GET THEM
CB11 60          #415      DEFB  ARG5
CB12 44          #416      LD     B,H          ;HL->BC
CB13 4B          #417      LD     C,L
CB14 B7          #418      OR     A            ;CLEAR CARRY
CB15 2B          #419      DEC   HL
CB16 EB 52       #420      SBC   HL,DE        ;CHECK 2ND>=1ST
CB18 3F          #421      CCF
CB19 C9          #422      RET
CB1A            #423 ;*****
CB1A            #424 ;   ROUTINES TO PERFORM A COMPLETE PASS
CB1A            #425 ;*****
CB1A            #426 ;
CB1A            #427 ;   SUBROUTINE TO GET NEXT SET OF ADDRESSES
CB1A            #428 ;   FOR PASS
CB1A            #429 ;
CB1A 2A 4C 0E    #430 NEXTAD: LD   HL,(POINT) ;GET ADDRESS POINTER
CB1B 2B          #431      DEC   HL
CB1E 56          #432      LB   D,(HL)        ;PICK UP START ADDRESS
CB1F 2B          #433      DEC   HL
CB20 5E          #434      LB   E,(HL)
CB21 22 4C 0E    #435      LD   (POINT),HL   ;PUT POINTER BACK
CB24 E5          #436      PUSH HL           ;HL->IX
CB25 0D E1       #437      POP  IX
CB27 62          #438      LB   H,D           ;HL=DE
CB28 4B          #439      LB   L,E
CB29 ED 4B 34 0E #440      LD   BC,(OFFSET) ;ADDR. OFFSET
CB2D 09          #441      ADD  HL,BC        ;COMPUTE PC
CB2E 44          #442      LB   B,H
CB2F 4D          #443      LB   C,L
CB30 DD 46 FF    #444      LD   H,(IX-1)     ;GET END ADDRESS
CB33 DD 4E FE    #445      LD   L,(IX-2)
CB36 C9          #446      RET
CB37            #447 ;
CB37            #448 ;   DO COMPLETE PASS
CB37            #449 ;
CB37 22 01 0E    #450 PASS:  LD   (PATCH+1),HL ;SET O/P DRIVER ADDR.
CB3A 2A 50 0E    #451      LD   HL,(SYMBOL) ;INIT. SYMBOL TABLE POINTER
CB3B 22 52 0E    #452      LD   (PSYN),HL
CB3D 2A 46 0E    #453      LD   HL,(START)  ;INITIALISE POINTER ADDRESS..
CB43 22 4C 0E    #454      LD   (POINT),HL ;...TO MEMORY SECTIONS
CB46 C0 1A CB    #455 PASSL: CALL NEXTAD ;GET PAIR OF ADDRESSES
CB49 03          #456      INC  BC           ;CORRECT THEM
CB4A 13          #457      INC  DE
CB4B 2B          #458      DEC  HL
CB4C CD 3E C4    #459      CALL REVAS       ;DO THEM
CB4F CD 1A CB    #460      CALL NEXTAD      ;GET DATA PAIR
CB52 7B          #461      LB   A,L          ;CHECK FOR END OF TABLE
CB53 A4          #462      AND  H
CB54 3C          #463      INC  A
CB55 C8          #464      RET  Z            ;RETURN IF HL=$FFFF
CB56 3A 44 0E    #465      LD   A,(FLAG)    ;CHECK PASS#
CB59 E6 09       #466      AND  $00         ;IF PASS 1, DON'T BOTHER...
CB5B 2B E9       #467      JR   Z,PASSL     ;...WITH DATA AREAS.
CB5D            #468 ;
CB5D            #469 ;   SECTION TO DEAL WITH DATA AREAS
CB5D            #470 ;   PRINT AS:-
CB5D            #471 ;   AAAA XX XX XX (LABEL) DB $XX,$XX,$XX ;C
CB5D            #472 ;

```

Interactive control program for REVAS

```

CD5B ED 3B 06 0E      0473      LD   DE,(MADDR)  ;GET NEXT ADDRESS
CD61 ED 52            0474      SBC  HL,DE       ;COMPUTE BYTE COUNT
CD63 3B E1           0475      JR   C,PASSL     ;LOOP IF -VE
CD65 23             0476      INC  HL          ;CORRECT IT
CD66 E5             0477      PUSH HL         ;SAVE COUNT
CD67 CD 6E C4       0478 DATA1: CALL INITB      ;INITIALISE BUFFER
CD6A 06 03          0479      LD   B,3        ;3 BYTES/LINE
CD6C CD 08 C4       0480 DATA2: CALL BYTE     ;GET NEXT BYTE
CD6F E1            0481      POP  HL         ;RECOVER COUNT
CD70 2B            0482      DEC  HL         ;MORE?
CD71 7D            0483      LD   A,L
CD72 04            0484      OR   H
CD73 E5            0485      PUSH HL         ;SAVE COUNT
CD74 2B 02         0486      JR   Z,**2      ;SKIP CHECK ON B IF DONE
CD76 1B F4         0487      DJNZ DATA2     ;MORE ON LINE?
CD78 F5            0488      PUSH AF        ;SAVE FLAG
CD79 CD 53 C9       0489      CALL NOTVAL     ;PUT IN 'DEFB' BIT
CD7C 21 14 0E      0490      LD   HL,BUFFER  ;PRINT
CD7F CD 25 CE       0491      CALL PASS2
CD82 F1            0492      POP  AF        ;RECOVER FLAG
CD83 2B E2         0493      JR   NZ,DATA1  ;LOOP IF MORE
CD85 E1            0494      POP  HL        ;(REMOVE COUNT)
CD86 1B BE         0495      JR   PASSL     ;ELSE NEXT SECTION
CD88              0496 ; *****
CD89              0497 ; PASS1 OUTPUT SECTION - BUILDS SYMBOL TABLE
CD8B              0498 ; *****
CD8B 2A 12 0E      0499 PASS1: LD   HL,(LABELP) ;CHECK IF ANY REF.
CD8D 7D            0500      LD   A,L
CD8E 04            0501      OR   H
CD8F C8            0502      RET  Z         ;RETURN IF NONE
CD90 2A 50 0E      0503      LD   HL,(SYNEND) ;LOAD TABLE ADDRESS
CD91              0504 ;
CD91              0505 ; NOW FIND SYMBOL IN TABLE
CD91              0506 ; TABLE IS ORGANISED AS:-
CD91              0507 ; <NENTRIES> <LABEL><ADDR.> <REF><ADDR1> <REF><ADDR2> ...
CD91              0508 ; "LABELS" ARE IN ENTERED IN ASCENDING NUMERICAL ORDER
CD91              0509 ; BECAUSE OF ZEAP FILE THE TABLE STARTS AT THE
CD91              0510 ; TOP OF MEMORY AND WORKS IT WAY DOWN.
CD91              0511 ; EDT MARKED BY %FF
CD91              0512 ;
CD91 7E            0513 SRCHL: LD   A,(HL) . ;LOAD # ENTRIES
CD92 E5            0514      PUSH HL         ;GET HL TO IY
CD93 FB E1         0515      POP  IY
CD95 3C            0516      INC  A         ;TEST
CD96 2B 2E         0517      JR   Z,PAST     ;EDT
CD98 47            0518      LD   B,A        ;COUNT IN B
CD99 2B            0519      DEC  HL
CD9A 5E            0520      LD   E,(HL)     ;LOAD LABEL IN TABLE
CD9B 2B            0521      DEC  HL
CD9C 36            0522      LD   D,(HL)
CD9D 2B            0523      DEC  HL
CD9E E5            0524      PUSH HL         ;SAVE POINTER
CD9F 2A 10 0E      0525      LD   HL,(LABELA) ;GET CURRENT LABEL
CBA2 A7            0526      AND  A         ;CLEAR CARRY
CBA3 ED 52         0527      SBC  HL,DE     ;COMPARE
CBA5 E1            0528      POP  HL        ;RECOVER POINTER
CBA6 1B 02         0529      JR   **2      ;SKIP IN CASE NONE
CBA8 2B            0530      DEC  HL        ;SPACE TO END OF ENTRY
CBA9 2B            0531      DEC  HL
CDA0 1B FC         0532      DJNZ **4

```

Interactive control program for REVAS

```

CDAC 3A 44 0E      0533      LD  A,(FLAG)      ;GET FLAG TO A
CDAF 38 15         0534      JR  C,PAST        ;NEW < TABLE ENTRY
CDB1 28 DE         0535      JR  NZ,SRCHL     ;TRY NEXT ENTRY
CDB3              0536 ;
CDB3              0537 ; FOUND ENTRY, SO APPEND XREF TO IT
CDB3              0538 ; ONLY IF REQUIRED.
CDB3              0539 ;
CDB3 CB 5F        0540      BIT  3,A          ;CHECK IF "X" OPTION
CDB5 C8           0541      RET  Z           ;RETURN IF NOT
CDB6 0E 02        0542      LD  C,2         ;MAKE ROOM FOR IT
CDB8 CD EF CD     0543      CALL HVUP
CDB9 ED 58 0E 0E 0544 FND:  LD  DE,(LINEA) ;INSERT XREF
CDBF 73           0545      LD  (HL),E
CDC# 2B           0546      DEC  HL
CDC1 72           0547      LD  (HL),D
CDC2 FD 34 00     0548      INC  (IY)       ;BUMP COUNT
CDC3 C9           0549      RET             ;DONE
CDC6              0550 ;
CDC6              0551 ; NEW ENTRY WANTED,MAKE
CDC6              0552 ; ROOM AND INSERT IT
CDC6              0553 ; BUT FIRST UPDATE LABEL COUNT IN CASE ZEAP FILE
CDC6              0554 ;
CDC6 2A 82 0E     0555 PAST:  LD  HL,(ZLABEL)
CDC9 23           0556      INC  HL
CDBA 23           0557      INC  HL
CDBB 22 82 0E     0558      LD  (ZLABEL),HL
CDBE FD E5        0559      PUSH IY        ;BACK TO START OF ENTRY
CDD0 E1           0560      POP  HL
CDD1 3A 44 0E     0561      LD  A,(FLAG)   ;GET FLAG BYTE
CDD4 0E 05        0562      LD  C,5        ;MAKE ROOM FOR NEW ENTRY
CDD6 CB 5F        0563      BIT  3,A       ;XREF?
CDD8 28 82        0564      JR  NZ,++2     ;YES SKIP
CDDA 0E 03        0565      LD  C,3        ;NO,SO NO XREF ENTRY
CDDC CD EF CD     0566      CALL HVUP
CDDF 36 00        0567      LD  (HL),0     ;COUNT=0
CDE1 ED 58 18 0E 0568      LD  DE,(LABELA) ;PUT LABEL IN
CDE5 2B           0569      DEC  HL
CDE6 73           0570      LD  (HL),E
CDE7 2B           0571      DEC  HL
CDE8 72           0572      LD  (HL),D
CDE9 2B           0573      DEC  HL
CDEA CB 5F        0574      BIT  3,A       ;"XREF" GOING?
CDEC C8           0575      RET  Z         ;NO,DONE
CDED 18 CC        0576      JR  FND        ;REST IN 'FOUND'
CDEF              0577 ;
CDEF              0578 ; MAKE SPACE WITHIN THE TABLE
CDEF              0579 ; FOR A NEW ENTRY & CHECK FOR
CDEF              0580 ; OVERFLOW OF THE TABLE
CDEF              0581 ;
CDEF ED 58 52 0E 0582 HVUP:  LD  DE,(PSYM)   ;SEE HOW MUCH TO MOVE
CDF3 23           0583      INC  HL
CDF4 A7           0584      AND  A         ;CLEAR CARRY
CDF5 06 00        0585      LD  B,0        ;CLEAR B
CDF7 ED 52        0586      SBC  HL,DE     ;HL=BYTE COUNT
CDF9 EB           0587      EX  DE,HL     ;COUNT TO DE
CDFA ED 42        0588      SBC  HL,BC     ;HL=NEW END
CDFC 42           0589      LD  B,D        ;COUNT TO BC
CDFD 4B           0590      LD  C,E
CDFE EB           0591      EX  DE,HL     ;NEW END TO DE
CDFE 2A 4E 0E     0592      LD  HL,(SYNST) ;CHECK FOR TABLE OVERFLOW

```

Interactive control program for REVAS

```

CE02 2B          #593      DEC  HL
CE03 ED 52      #594      SBC  HL,DE
CE05 30 08      #595      JR   NC,OVRFLU
CE07 2A 52 0E   #596      LD   HL,(PSYN) ;RESET SOURCE ADDRESS
CE0A ED 53 52 0E #597      LD   (PSYN),DE ;UPDATE PSYN
CE0E ED 80      #598      LDIR ;MOVE TABLE DOWN
CE10 2B         #599      DEC  HL ;ADJUST ADDRESS
CE11 C9         #600      RET
CE12           #601 ;
CE12 EF         #602 OVRFLU: RST  PRS ;ERROR MESSAGE
CE13 4F 76 65 72 #603      DEFB 'Overflow',CR,#
      66 6C 6F 77
      8D 88

CE1B 21 14 0E   #604      LD   HL,BUFFER ;PRINT CURRENT LINE
CE20 CD FB CE   #605      CALL OUTLIN
CE23 0F         #606      RST  SCAL
CE24 5B         #607      DEFB RETURN
CE25           #608 ;*****
CE25           #609 ; PASS 2 - PRINTS OUTPUT & ADDS LABELS
CE25           #610 ;*****
CE25 01 13 08   #611 PASS2: LD   BC,19 ;SPACE PAST OPCODES ETC
CE26 FD 21 58 0E #612      LD   IX,SEQU ;IN CASE EQU WANTED
CE2C E5         #613      PUSH HL ;GET HL TO IX
CE2D DD E1      #614      POP  IX
CE2F DD 09      #615      ADD  IX,BC ;IX POINTS TO LABEL POSITION
CE31 3A 44 0E   #616      LD   A,(FLAG)
CE34 CB 47      #617      BIT  0,A ;OPCODES ETC WANTED?
CE36 28 03      #618      JR   Z,+3 ;YES
CE38 09         #619      AND  HL,BC ;NO,ALTER BUFFER START ADDR.
CE39 FD 09      #620      ADD  IX,BC
CE3B CB 67      #621      BIT  4,A ;LABELS?
CE3D 28 0E      #622      JR   Z,OUTPUT ;NONE,SO SKIP
CE3F E5         #623      PUSH HL ;SAVE ADDRESS
CE40 CD 02 CF   #624      CALL LABEL ;DO LABEL CHECK
CE43 2A 12 0E   #625      LD   HL,(LABELP) ;CHECK OPERAND FOR...
CE46 7B         #626      LD   A,L ;..LABEL THERE
CE47 B4         #627      OR   H
CE48 28 02      #628      JR   Z,+2 ;SKIP IF NONE
CE4A 36 4C      #629      LD   (HL),'L' ;CHANGE '0' TO 'L'
CE4C E1         #630      POP  HL ;RECOVER ADDRESS
CE4D           #631 ;
CE4D           #632 ; NOW OUTPUT THE LINE
CE4D           #633 ;
CE4D 44         #634 OUTPUT: LD   B,H ;TEMP SAVE IN BC
CE4E 40         #635      LD   C,L
CE4F 2A 4A 0E   #636      LD   HL,(LAST) ;CHECK AGAINST RANGE
CE52 ED 5B 0E 0E #637      LD   DE,(LINEA)
CE56 A7         #638      AND  A ;CLEAR CARRY
CE57 ED 52      #639      SBC  HL,DE
CE59 DB         #640      RET  C ;RETURN IF LINEA>LAST
CE5A 2A 48 0E   #641      LD   HL,(FIRST)
CE5D EB         #642      EX  DE,HL
CE5E ED 52      #643      SBC  HL,DE
CE60 DB         #644      RET  C ;RETURN IF FIRST<LINEA
CE61 C5         #645      PUSH BC ;SAVE POINTER ON STACK
CE62           #646 ;
CE62 3A 44 0E   #647      LD   A,(FLAG) ;CHECK OPTIONS
CE65 F5         #648      PUSH AF ;SAVE ON STACK
CE66 CB 4F      #649      BIT  5,A ;PAGED?
CE68 C4 7B CF   #650      CALL NZ,BPAGE ;YES,DEAL WITH IT

```

Interactive control program for REVAS

```

CE6B F1          0651      POP  AF          ;(RESET FLAG)
CE6C E1          0652      POP  HL          ;RECOVER POINTER
CE6D F5          0653      PUSH AF          ;SAVE FLAG AGAIN
CE6E 0F          0654      RRCA
CE6F 30 3D      0655      JR    NC,ALLIN  ;SKIP IF ALL WANTED
CE71            0656 ;*****
CE71            0657 ; CRUNCH SOURCE INTO BUFFER (USE EQU ONE)
CE71            0658 ;*****
CE71 EB          0659      EX    DE,HL      ;TEMP SAVE HL
CE72 21 09 00   0660      LD    HL,9        ;OFFSET TO 4TH BYTE OF MNE.
CE73 19          0661      ADD  HL,DE        ;POINT TO IT
CE74 7E          0662      LD    A,(HL)     ;GET IT
CE77 EB          0663      EX    DE,HL      ;RESET HL
CE78 11 38 0E   0664      LD    DE,SEQU    ;START OF BUFFER
CE7B 01 38 20   0665      LD    BC,' '     ;SET FOR NOT 'DEFB'
CE7E 06 42      0666      SUB  'B'         ;DEFB?
CE80 20 01      0667      JR    NZ,**1     ;NO,SKIP
CE82 4F          0668      LD    C,A        ;YES,CLEAR DELIMITER
CE83 7E          0669      LD    A,(HL)     ;LOAD BYTE
CE84 FE 00      0670      CP    CR         ;REACHED END?
CE86 28 19      0671      JR    Z,SQSH3   ;YES
CE88 09          0672      CP    C          ;REACHED DELIMITER?
CE89 28 16      0673      JR    Z,SQSH3   ;YES,JUMP ON
CE8B 12          0674      LD    (DE),A     ;NO,COPY IT OVER
CE8C 13          0675      INC  DE         ;UPDATE ADDRESSES
CE8D 23          0676      INC  HL         ;
CE8E FE 2F      0677      CP    '/'        ;WAS IT DEFB DELIMITER?
CE90 20 06      0678      JR    NZ,SQSH2  ;NO,SKIP
CE92 70          0679      LD    A,B        ;YES,MODIFY OR RESET
CE93 2F          0680      CPL
CE94 47          0681      LD    B,A
CE95 79          0682      LD    A,C
CE96 2F          0683      CPL
CE97 4F          0684      LD    C,A
CE98 08          0685      CP    8          ;SPACE?
CE99 20 EB      0686      JR    NZ,SQSH1  ;NO,CONTINUE
CE9B 0E          0687      CP    (HL)       ;NEXT A SPACE?
CE9C 20 E5     0688      JR    NZ,SQSH1  ;NO,COPY IT OVER(?)
CE9E 23          0689      INC  HL         ;YES,IGNORE IT
CE9F 18 F7      0690      JR    SQSH2     ;GO CHECK NEXT
CEA1            0691 ; DELETE ANY ODD SPACE AT THE END OF THE LINE
CEA1 EB          0692      EX    DE,HL      ;DE->HL
CEA2 2B          0693      DEC  HL
CEA3 7E          0694      LD    A,(HL)     ;GET LAST BYTE
CEA4 FE 20      0695      CP    '/'        ;SPACE?
CEA6 28 01      0696      JR    Z,**1     ;YES,SKIP
CEA8 23          0697      INC  HL         ;NO SET CR AT END
CEA9 36 00      0698      LD    (HL),CR   ;
CEAB 21 38 0E   0699      LD    HL,SEQU    ;RESET BUFFER POINTER
CEAE            0700 ;*****
CEAE            0701 ; PRINT THE LINE AND ALSO MOVE TO ZEAP FILE (?)
CEAE            0702 ;*****
CEAE CD FB CE   0703      ALLIN: CALL  OUTLIN
CEB1 F1          0704      POP  AF          ;RECOVER FLAG
CEB2 CB 57      0705      BIT  2,A        ;CHECK FOR ZEAP FILE
CEB4 C8          0706      RET  Z          ;NONE,SO RETURN
CEB5 ED 5B 00 0E 0707      LD    DE,(LINENO) ;GET CURRENT LINE NUMBER
CEB9 2A 56 0E   0708      LD    HL,(ZEAPP) ;AND POINTER INTO ZEAP FILE
CEBC 23          0709      INC  HL         ;SPACE PAST NULL
CEBD 73          0710      LD    (HL),E     ;PUT LINE NO INTO FILE

```


Interactive control program for REVAS

```

CEBE 23          0711      INC  HL
CEBF 72          0712      LD   (HL),D
CEC0 23          0713      INC  HL
CEC1 EB          0714      EX   DE,HL      ;FILE POINTER TO DE
CEC2 3E 01      0715      LD   A,1      ;UPDATE LINE NO.
CEC4 85          0716      ADD  L
CEC5 27          0717      DAA
CEC6 6F          0718      LD   L,A
CEC7 3E 00      0719      LD   A,0
CEC9 BC          0720      ADC  H
CECA 27          0721      DAA
CECB 67          0722      LD   H,A
CECC 22 00 0E   0723      LD   (LINENO),HL
CECF 21 58 0E   0724      LD   HL,SEQU      ;SET SOURCE POINTER
CED2 3E 0D      0725      LD   A,CR      ;MOVE LINE TO FILE
CED4 ED A0      0726 Z1:  LDI
CED6 0E          0727      CP   (HL)      ;END YET?
CED7 20 FB      0728      JR   NZ,Z1      ;NO, LOOP
CED9           0729 ; NOW SAVE POINTER AND UPDATE FILE START & END
CED9 EB          0730      EX   DE,HL      ;POINTER BACK TO HL
CEDA 22 56 0E   0731      LD   (ZEAPP),HL ;UPDATE IN MEM.
CEDD 36 90      0732      LD   (HL),0
CEDF 23          0733      INC  HL
CEE0 36 FF      0734      LD   (HL),+FF
CEE2 23          0735      INC  HL
CEE3 E5          0736      PUSH HL      ;SAVE FOR OVERFLOW CHECK
CEE4 ED 58 00 0F 0737      LD   DE,(+FF00) ;COMPUTE REL. OFFSET
CEE8 ED 52      0738      SBC  HL,DE
CEEA EB          0739      EX   DE,HL
CEE9 73          0740      LD   (HL),E      ;SET AT FILE START
CEEC 23          0741      INC  HL
CEED 72          0742      LD   (HL),D
EEEE D1          0743      POP  DE      ;RECOVER END POINTER
CEEF 2A 4E 0E   0744      LD   HL,(SYHST) ;CHECK FOR OVERFLOW
CEF2 ED 52      0745      SBC  HL,DE
CEF4 D0          0746      RET  NC      ;OK IF BELOW SYMBOL TABLE
CEF5 C3 12 CE   0747      JP   OVRFLW
CEFB           0748 ;*****
CEFB           0749 ;  OUTPUT A LINE TO THE SCREEN (AND TAPE?)
CEFB           0750 ;*****
CEFB 46          0751 OUTLIN: LD   B,(HL)      ;GET BYTE
CEFB CD AA CF   0752      CALL CHROUT      ;OUTPUT IT
CEFC 23          0753      INC  HL
CEFD FE 0D      0754      CP   CR      ;FINISHED YET?
CEFF 20 F7      0755      JR   NZ,OUTLIN ;LOOP IF MORE
CF01 C9          0756      RET
CF02           0757 ;*****
CF02           0758 ;  PASS 2 - LABEL HANDLING
CF02           0759 ;*****
CF02 2A 52 0E   0760 LABEL: LD   HL,(PSYH) ;POINT TO CURRENT SYMBOL
CF05 7E          0761      LD   A,(HL)      ;LOAD # ENTRIES
CF06 3C          0762      INC  A      ;CHECK FOR EOT
CF07 C0          0763      RET  Z      ;RETURN IF THERE
CF08 47          0764      LD   B,A      ;SAVE COUNT
CF09 2B          0765      DEC  HL      ;GET SYMBOL
CF0A 5E          0766      LD   E,(HL)      ;...TO DE
CF0B 2B          0767      DEC  HL
CF0C 54          0768      LD   D,(HL)
CF0D 2B          0769      DEC  HL
CF0E 10 02      0770      JR   ++2      ;SKIP IN CASE NONE

```

Interactive control program for REVAS

```

CF10 2B          0771      DEC HL          ;SPACE TO END OF ENTRY
CF11 2B          0772      DEC HL
CF12 10 FC       0773      DJNZ *-4
CF14 44          0774      LD B,H          ;SAVE POINTER IN BC
CF15 4D          0775      LD C,L
CF16 2A 9E 0E    0776      LD HL,(LINEA)  ;GET CURRENT LINE
CF19 3A 44 0E    0777      LD A,(FLAG)    ;CHECK IF FORCING REMAINING..
CF1C 07          0778      RLCA           ;...EQU'S. PASS IN CARRY.
CF1D 3F          0779      CCF           ;0=NORMAL PASS2,1=FORCE EQU'S
CF1E 38 03       0780      JR C,+3        ;1=SKIP CHECK
CF20 ED 52       0781      SBC HL,DE      ;COMPARE WITH ENTRY
CF22 08          0782      RET C         ;RETURN,NOT THERE YET
CF23 ED 43 32 0E 0783      LD (PSYN),BC   ;SET NEXT ENTRY ADDRESS
CF27 28 48       0784      JR Z,ALAB     ;EQUAL,SO APPEND LABEL
CF29             0785 ;
CF29             0786 ; PAST LABEL,SO INSERT 'EQU'
CF29             0787 ;
CF29 44          0788      LD B,H          ;GET OFFSET TO BC
CF2A 4D          0789      LD C,L
CF2B 03          0790      PUSH BC        ;SAVE BC
CF2C 06 13       0791      LD B,19
CF2E 21 58 0E    0792      LD HL,SEQU     ;CLEAR BUFFER FOR IT
CF31 36 38       0793      LD (HL),' '
CF33 23          0794      INC HL
CF34 18 FB       0795      DJNZ *-5
CF36 F5          0796      PUSH AF        ;PRESERVE A
CF37 CD 74 CF    0797      CALL ALAB1     ;WRITE LABEL
CF3A F1          0798      POP AF         ;RESET IT
CF3B E5          0799      PUSH HL        ;SAVE HL
CF3C 21 F7 CF    0800      LD HL,EQU      ;PUT IN 'EQU'
CF3F 01 06 08    0801      LD BC,6
CF42 ED 00       0802      LDIR
CF44 EB          0803      EX DE,HL
CF45 D1          0804      POP DE         ;'LABEL' TO DE
CF46 C1          0805      POP BC         ;RESET BC
CF47 0F          0806      RRCA          ;"FORCE" FLG STILL IN A
CF48 30 15       0807      JR NC,ABSOL   ;"FORCE"=ABSOLUTE LABEL
CF4A 78          0808      LD A,B         ;CHECK FOR RANGE
CF4B A7          0809      AND A         ;ABSOLUTE IF >255
CF4C 28 11       0810      JR NZ,ABSOL
CF4E 79          0811      LD A,C         ;ABSOLUTE IF >4
CF4F FE 05       0812      CP 5
CF51 30 0C       0813      JR NC,ABSOL
CF53 36 24       0814      LD (HL),'4'   ;INSERT PC. RELATIVE
CF55 23          0815      INC HL
CF56 36 2D       0816      LD (HL),'-'
CF58 23          0817      INC HL
CF59 F6 30       0818      OR $30        ;MAKE OFFSET ASCII
CF5B 77          0819      LD (HL),A     ;PUT IN BUFFER
CF5C 23          0820      INC HL        ;ADD A CR
CF5D 18 08       0821      JR PEQU      ;PRINT LINE
CF5F             0822 ;
CF5F             0823 ; ABSOLUTE LABEL
CF5F             0824 ;
CF5F 36 23       0825 ABSOL: LD (HL),'#' ;HEX
CF61 23          0826      INC HL        ;INSERT ADDRESS
CF62 EB          0827      EX DE,HL
CF63 CD 9D C4    0828      CALL HEX4
CF66 EB          0829      EX DE,HL
CF67 36 0D       0830 PEQU: LD (HL),CR ;SET CR AT END

```

Interactive control program for REVAS

```

CF49 FD E5          0831      PUSH IY          ;GET IY TO HL
CF6B E1            0832      POP HL
CF6C CD 4D CE      0833      CALL OUTPUT      ;PRINT LINE
CF6F 18 91         0834      JR LABEL        ;CHECK NEXT LABEL
CF71              0835 ;
CF71              0836 ; WRITE LABEL
CF71              0837 ;
CF71 BB E5         0838 ALAB:  PUSH IX          ;GET IX TO HL
CF73 E1            0839      POP HL
CF74 36 4C         0840 ALAB1: LD (HL),'L'      ;LEADING L
CF76 23           0841      INC HL
CF77 EB           0842      EX DE,HL       ;WRITE TO BUFFER
CF78 C3 9D CA      0843      JP HEX4        ;RETURN THRU' HEX4
CF7B              0844 ;*****
CF7B              0845 ; PAGED,CHECK LINE COUNT ETC
CF7B              0846 ;*****
CF7B 21 7C 0E      0847 DPAGE: LD HL,LC          ;POINT TO LC
CF7E 35           0848      DEC (HL)
CF7F CB           0849      RET NZ        ;RETURN IF NOT EOP
CF80 3A 7D 0E      0850 EJECT: LD A,(GAP)      ;COMPUTE # LINE FEEDS
CF83 86           0851      ADD (HL)
CF84 57           0852      LD B,A        ;SAVE IN B
CF85 86 0D         0853      LD B,CR
CF87 18 03         0854      JR ++3       ;SKIP
CF89 CD AA CF      0855      CALL CHROUT   ;DO REQUISITE #
CF8C 15           0856      DEC B
CF8D 28 FA         0857      JR NZ,+-6
CF8F 2A 7E 0E      0858      LD HL,(LINES) ;RESET LINE COUNT
CF92 22 7C 0E      0859      LD (LC),HL    ;...(AND GAP)
CF95 21 8D 0E      0860      LD HL,PAGE+1  ;INC PAGE NUMBER
CF98 34           0861      INC (HL)
CF99 7E           0862      LD A,(HL)    ;CHECK FOR CARRY
CF9A FE 3A         0863      CP '9'+1
CF9C 28 04         0864      JR NZ,NOINC
CF9E 36 30         0865      LD (HL),'0'   ;RESET
CFA0 2B           0866      DEC HL
CFA1 34           0867      INC (HL)     ;BUMP PAGE NUMBER
CFA2 21 84 0E      0868 NOINC: LD HL,HEAD    ;PRINT HEADING
CFA5 CB F8 CE      0869      CALL OUTLIN
CFA8 86 0D         0870      LD B,CR      ;SPACE ONE LINE
CFAA              0871 ;*****
CFAA              0872 ; CHARACTER OUTPUT ROUTINE
CFAA              0873 ;*****
CFAA 78           0874 CHROUT: LD A,B          ;O/P TO SCREEN
CFAB F7           0875      RST ROUT
CFAC E5           0876      PUSH HL      ;SAVE HL
CFAD 21 44 0E      0877      LD HL,FLAG   ;POINT TO FLAG
CFB0 CD 0F CF      0878      CALL PUNCH   ;PUNCH IT MAYBE
CFB3 FE 0D         0879      CP CR       ;EOLT
CFB5 28 26         0880      JR NZ,EXCHR ;NO,EXIT
CFB7 3E 0A         0881      LD A,LF     ;ADD LF MAYBE
CFB9 CB 0F CF      0882      CALL PUNCH
CFBC 3A 45 0E      0883      LD A,(STEP) ;CHECK SINGLE STEP FLAG
CFBF FE 20         0884      CP ' '      ;SPACE?
CFC1 28 04         0885      JR Z,WAIT   ;IF S/STEP SKIP NEXT
CFC3 0F           0886      RST SCAL    ;POLL THE KEYBOARD
CFC4 62           0887      BEFB POLLX
CFC5 38 04         0888      JR NC,++4   ;IF NOTHING-CNTUE
CFC7 CF           0889 WAIT: RST RIN    ;READ A CHARACTER
CFC8 32 45 0E      0890      LD (STEP),A ;SAVE KEY

```

Interactive control program for REVAS

```

CFCB FE 1B          0891      CP      ESC      ;ESCAPE PRESSED?
CFCD CA A2 CC      0892      JP      Z,EXIT  ;YES,RETURN
CFDE 78            0893      LD      A,B    ;(RESET A)
CFD1 CB 76         0894      BIT    4,(HL)  ;DELAY GOING?
CFD3 28 08         0895      JR      Z,EXCHR;NO,EXIT
CFD5              0896 ; SHORT DELAY ROUTINE
CFD6 C5            0897      PUSH   BC     ;PRESERVE BC -(FIRST LOOP SHORT)
CFD8 06 08         0898      LD      B,128 ;LOOP COUNT
CFDB FF           0899 DLOOP: RST   RDEL ;DELAY
CFD9 10 FD         0900      DJNZ  DLOOP  ;DELAY
CFDB C1            0901      POP    BC     ;RESTORE BC
CFDC 78            0902      LD      A,B    ;AND A
CFDD E1            0903 EXCHR: POP    HL     ;RESET HL
CFDE C9            0904      RET
CFDF              0905 ;
CFDF CB 4E         0906 PUNCH: BIT   1,(HL)  ;TAPE OPTION ON?
CFE1 C8            0907      RET    Z      ;NO,RETURN
CFE2 DF           0908      RST   SCAL   ;PUNCH IT
CFE3 6F           0909      DEFB  SRLX   ;O/P TO SERIAL PORT
CFE4 C9            0910      RET
CFE5              0911 ;*****
CFE5              0912 ; RAM INITIALISATION IMAGE
CFE5              0913 ;*****
CFE5 10 00 00 00   0914 RAMLD:  DEFW  $10,0
CFE9 20 20 20 50   0915      DEFB  ' PAGE 00
          41 47 45 20
          30 30 20 20
          20 20
          0012
CFE7 20 45 51 55   0916 MOVLD: EQU  *-RAMLD
          20 20      0917 EQU:  DEFB  ' EQU
CFE8 C3 3E C4      0918      JP      REVAS ;fixed transfer to REVAS
          0920      END

```

CROSS REFERENCE

ABSOL	CF5F	0007 0010 0013
ADDR	0E04	
ALAB	CF71	0784
ALAB1	CF7A	0797
ALLIN0	CC8C	0313
ALLIN	CEAE	0635
ARGN	0C0B	0199 0410
ARGS	0060	0203 0415
ASKT	CB4E	0171
BLINK	007B	0299
BUFFER	0E14	0205 0260 0289 0367 0382 0388 0490 0604
BYTE	C400	0400
CHROUT	CFAA	0752 0855
CR	000D	0090 0111 0147 0161 0180 0193 0227 0236 0301 0603 0670 0690 0725 0754 0830 0853 0870 0879
CRLF	006A	0302
DA	CBDC	0224
DAREA	CBE4	0274
DAREA0	CBF4	0267 0281 0206
DAREA1	CC20	0240
DAREA2	CC37	0259 0262 0271
DAREA3	CC40	0269
DATA1	CD67	0493
DATA2	CD6C	0487
DLOOP	CFDB	0900
DPAGE	CF7B	0650
EADDR	0E08	
EJECT	CF80	0355
EDU	CFF7	0000
ERRM	006B	0203
ESC	001B	0200 0200 0402 0891
EXCHR	CFDD	0000 0095
EXIT	CCA2	0363 0092
FIRST	0E40	0230 0641
FLAG	0E44	0117 0222 0306 0310 0336 0465 0533 0561 0616 0647 0777 0077
FND	CDBB	0576
FOUND	CAE4	0100
GAP	0E7D	0350 0050
GETSTA	CB11	0149
GETTWO	CCFE	0140 0220 0239
HEAD	0E04	0060
HEX4	C49D	0372 0020 0043
HXYFL0	0E0C	
INITB	C46E	0470
INLINE	0063	0163 0195 0401
LABEL	CF02	0340 0624 0034
LABELA	0E10	0325 0560
LABELP	0E12	0499 0625
LAST	0E4A	0231 0359 0636
LC	0E7C	0173 0047 0059
LF	000A	0001
LINEA	0E0E	0544 0637 0776
LINENO	0E09	0002 0707 0723
LINES	0E7E	0173 0050
LUOP	CAF6	0110
MADDR	0E06	0473
MOVING	0012	0004
MVUP	CDEF	0543 0566
NEXTAD	CD1A	0455 0460

CROSS REFERENCE

NLABS	CB49	#133
NOINC	CFA2	#864
NOTITL	C886	#159 #198 #291
NOTVAL	C953	#489
NUM	#864	#178
NUMV	#C21	#172
OFFSET	#E54	#216 #243 #448
OPCADR	#E9A	
OPTION	CAC6	#112
OTAB	CAD2	#186
OUTLIN	CEF8	#685 #793 #755 #869
OUTPUT	CE4D	#383 #622 #833
OVRFLW	CE12	#595 #747
PAGE	#E8C	#868
PASS	CD37	#389 #335
PASS1	CD88	#388
PASS2	CE25	#334 #491
PASSL	CB46	#467 #475 #495
PAST	CDC6	#517 #534
PATCH	#E98	#458
PEQU	CF67	#821
POINT	#E4C	#438 #435 #454
POLLK	#862	#887
PRS	#828	#889 #141 #144 #146 #168 #192 #226 #235 #279 #284 #296 #682
PSYH	#E52	#322 #452 #582 #596 #597 #768 #783
PUNCH	CFDF	#878 #882
RANLD	CFE5	#883 #916
RDEL	#838	#899
RETURN	#85B	#345 #687
REVAS	CA3E	#459 #918
REVASC	CAB8	
RIN	#888	#182 #889
RLIN	#879	#197 #486
RNGRQ	CBC6	#229
ROUT	#938	#183 #483 #875
RSTFLG	#E#3	#898
SAVE	CABB	#878
SCAL	#818	#162 #169 #194 #196 #282 #282 #298 #381 #344 #488 #495 #414 #686 #886 #988
SEQU	#E58	#612 #664 #699 #724 #792
SETLPP	CB69	#167
SOSH1	CE83	#686 #688
SOSH2	CE98	#678 #699
SOSH3	CEA1	#671 #673
SRCML	CD91	#535
SRLX	#86F	#989
ST	CB22	#148
ST#	CB2E	#143
START	#E46	#289 #453
STEP	#E45	#388 #883 #898
SYMEND	#E58	#151 #318 #368 #451 #583
SYNST	#E4E	#158 #328 #323 #592 #744
TITLE	#E92	#177
WAIT	CFC7	#885
XREF1	CCBC	#387
XREF2	CCCA	#378 #395
ZI	CEB4	#728
ZEAPP	#E56	#332 #788 #731
ZLABEL	#E82	#324 #555 #558