

```

1070 lhld lsbas #man/auto core status base
1080 call wrdbt #make bc be a word/bit offset for a
1090 call setwb #see if this one is manual or automatic
1100 ora a
1110 Jz mn303 #automatic
1111 *MANUAL SLIDEPOT PROCESSING
1112 mov a,e
1113 call rdsld
1117 lhld rrbas #hl->scratchpad
1118 dad d #point to scratchpad word
1119 mov b,a #b=slidpot
1120 mov a,m #scratchpad value
1121 sub b #find difference
1122 inr a #a=a-1
1123 cpi 5 #if < 5, don't update
1124 Jc mn306
1125 lxi h,rbas #sample/hold base
1126 dad d
1127 mov m,b #update
1128 lhld rrbas #scratchpad base
1129 dad d
1130 mov m,b #update scratchpad, too
1131 mn306 Jmp mn304
1132 *AUTOMATIC SLIDEPOT PROCESSING
1139 mn303 mov a,e #restore a from e
1140 call rdsld #read that slidpot
1150 lhld rrbas #hl->scratchpad

```

? exec 4006

```

MEM START: 2000
MEM END: 2332
T/S: 3800
LAST TRACK SECTOR: 38 06
?

```

exec 4000 ~

? exec 4003

```

T/S START: 00
T/S END: 00
MEM START: 00? hexec 4003
WHAT?
? exec 4003

```

```

T/S START: 0~1800
T/S END: 181f
MEM START: 0000? exec 4003

```

```

T/S START: 2000
T/S END: 2009
MEM START: 2000? exec 4000

```

```

? ecex 4003
WHAT?
? exec 4003

```

```

T/S START: 2000
T/S END: 2009

```

MEM START: 2000
LAST MEM ADDR: 24 ff

? list

```
0000  org 03dfh
0010  jmp binit
0100  org 02afh
0110  ****
0120  *
0130  * Module BINIT
0140  *
0150  * Written 7/30/76 by Bruce Cichowles
0160  * Copyright Circle C - ARF Instruments, 7/30/76
0170  *
0180  * BINIT - this call initializes all the voice bins used by
0190  * the polyphonic driver programs.
0200  * There are no arguments.
0210  *
0220  ****
0230  binit equ $
0240  push b
0250  push d
0260  push h
0270  lxi d,binst+vn
0280  lxi h,notel
0290  mvi b,vn
0300  loopi mvi m,0ffh
0310  inc h
0320  mov m,e
0330  inc h
0340  mov m,d
0350  inc h
0360  *add vn to de
0370  push b
0380  lxi b,vn
0390  xchg
0400  dad b
0410  xchg
0420  pop b
0430  dec b
0440  jnz loopi
0450  lxi h,bins
0460  mvi b,verod
0470  mvi a,0ffh
0480  loopj mov m,a
0490  inc h
0500  dec b
0510  jnz loopj
0520  mvi a,vn
0530  call bnsset
0540  lxi h,bins
0550  mvi b,vn
0560  xra a
0570  loopk mov m,a
0580  inc h
0590  inc a
0600  cmp b
0610  jnz loopk
0620  pop h
0630  pop d
0640  pop b
0650  ret
5000  vn equ 8
5010  verod equ 72 $vn*(vn+1)
6000  *RAM
6010  notel equ 0897h
6020  bins equ 08afh
```

9000 bnset equ 03f1h
9999 *END

? ? exec 4003

T/S START: 2200
T/S END: 220f
MEM START: 2000
LAST MEM ADDR: 27 ff

? list

```
0000  org 03d9h
0010  jmp lstrn
0100  org 00e2h
0101  ****
0102  *
0103  * Module LSTRN
0104  *
0105  * Written 7/19/76 by Bruce Cichowlas
0106  * Last modified 7/30/76 by Bruce Cichowlas
0107  * Copyright Circle C - ARP Instruments, Inc. 7/19/76
0108  *
0109  * LSTRN -
0110  *This is a routine to run down a note list as prepared by
0120  *the scan program and call another given
0130  *program for each note that is found to be on.
0140  *The address of the routine to call is given in HL.
0150  *The address of the list to process is given
0160  *in DE.
0172  *
0180  ****
0199  lstrn equ $
0200  shld dync1+1 ;put the program's address in the modifiable transfer vect
0201  mvi a,0c3h ;code for 'JMP'
0202  sta dync1 ; dynamic call
0210  push h
0220  push d
0230  push b
0240  xchg ;set the list addr into HL
0250  lxi b,7 ;BC=7 for the last item in the list.
0260  mvi e,7fh ;highest note
0270  lf push h
0280  dad b ;develop address in list
0290  mvi d,8 ;8 bits in word.
0300  mov a,m ;will rotate word in accumulator
0310  lf2 rlc
0320  push m
0324  mov a,e ;call other routine with pitch-octave code in A
0335  cc dync1 ;call chosen routine if note was on
0340  dcr e ;lower pitch-octave counter
0350  mov a,e ;check for illegal pitch-octave code
0351  cpi 06fh
0352  jnz l3
0353  mvi e,03fh
0354  l3 equ $
12 0360  ani 0fh
11 0370  cpi 0fh
10 0380  jnz cn1 ;pitch-octave code is legit
9 0390  dcr e
8 0400  dcr e
7 0410  dcr e
6 0420  dcr e ;now code will be for b-natural of next lower octave.
5 0430  cn1 POP m
4 0440  dcr d
0450  jnz lf2
0460  dcr c
0470  POP h
```

0480 JP lP
0490 POP b
0500 POP d
0510 POP h
0520 ret
6000 *RAM
6010 denc1 eau 0878h ;dynamically generated call vector
9999 *END

? -
exec 2400
? exec 4003
DDR: ? exec 4000
? exec 4003
DDR: 2000? exec 4003
DDR: ? exec 40000
WHAT?
STACK IS'LOW.
? exec 4000
? exec 4003
DDR: 1? 2
WHAT?
STACK IS'LOW.
? exec 4003
DDR: exec e^4000? ? exec 4000
? exec4^ 4003
DDR: ? exec 4009
? exec 4003
DDR: ? ? rdt=ADDR? 4000 NAME? disc.obj?

rdt=ADDR? 4000 NAME? dil^sk

rd t=? rd tP
WHAT?
STACK IS'LOW.
? rdt=ADDR? 4000 NAME? disk!? file
f 2000 2e16

? file disk~~~~/disk 4000
WHAT?
STACK IS'LOW.
? ? files

? file /dil^sk/ 4000
disk 4000 4000

? rdt=ADDR? 4000 NAME? disk!
? fix=ADDR? 4000
? files
disk 4000 d059

? dump d000 d0ff
d000 64 69 73 6b 20 00 40 59 d0 32 35 37 30 00 00 00
d010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
d020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
d030 00 00 00 00 00 00 00 00 00 00 00?

tprd
WHAT?
? rdt#
ff
WHAT?
? rdt#ADDR? 4000 NAME? #disk!
?

dump 4000
4000 c3
? dump 4000 43ce

4000	c3	c1	42	c3	4f	42	c3	b1	41	c3	33	40	c3	47	41	c3	-
4010	01	41	c3	9d	41	e5	d5	c5	7a	cd	54	40	cd	cd	40	c1	-
4020	d1	e1	c9	e5	d5	c5	7a	cd	54	40	3e	87	cd	91	40	c1	-
4030	d1	e1	c9	d3	08	db	08	e6	40	ca	4e	40	db	08	e6	02	-
4040	c2	3c	40	3e	02	d3	09	3e	00	d3	09	c3	35	40	3e	00	-
4050	32	3a	43	c9	c5	47	3a	3a	43	4f	b8	ca	8f	40	d2	78	-
4060	40	db	08	e6	02	c2	61	40	3e	01	d3	09	3e	00	d3	09	-
4070	0c	79	32	3a	43	c3	56	40	db	08	e6	02	c2	78	40	3e	-
4080	02	d3	09	3e	00	d3	09	0d	79	32	3a	43	c3	56	40	c1	-
4090	c9	4f	3e	88	91	47	cd	ef	40	3e	80	d3	09	16	01	3e	-
40a0	80	b6	5f	23	db	08	a2	c2	a4	40	83	d3	0a	7e	23	5e	-
40b0	23	0d	ca	bb	40	0d	d3	0a	c2	a4	40	db	08	a2	c2	bb	-
40c0	40	d3	0a	05	c2	bb	40	fb	3e	08	d3	09	c9	cd	ef	40	-
40d0	0e	89	db	08	b7	fa	d2	40	db	0a	77	23	0d	ca	e9	40	-
40e0	0d	00	db	0a	77	23	c2	d2	40	fb	3e	08	d3	09	c9	3e	-
40f0	04	d3	09	f3	db	09	1f	da	f4	40	e6	1f	bb	c2	f4	40	-
4100	c9	c5	d5	e5	3e	ff	32	46	43	7a	32	47	43	7b	32	48	-
4110	43	3e	00	32	4a	43	32	4b	43	e1	e5	11	4c	43	06	80	-
4120	7e	12	23	13	05	c2	20	41	06	86	21	46	43	3e	00	4e	-
4130	81	23	05	c2	2f	41	32	cc	43	e1	d1	d5	e5	21	46	43	-
4140	cd	23	40	e1	d1	c1	c9	c5	d5	e5	3e	01	32	3b	43	21	-
4150	46	43	cd	15	40	11	4c	43	e1	e5	3a	3b	43	b7	ca	6b	-
4160	41	06	80	1a	77	23	13	05	c2	63	41	e1	d1	d5	e5	3a	-
4170	47	43	ba	c2	98	41	3a	48	43	bb	c2	98	41	06	86	21	-
4180	46	43	3e	00	4e	81	23	05	c2	84	41	4f	3a	cc	43	b9	-
4190	c2	98	41	e1	d1	c1	b7	c9	e1	d1	c1	37	c9	c5	d5	e5	-
41a0	3e	00	32	3b	43	c3	4f	41	1c	7b	fe	20	c0	1e	00	14	-
41b0	c9	cd	c7	42	21	d4	42	cd	e5	ff	eb	22	3c	43	cd	c7	-
41c0	42	21	e0	42	cd	e5	ff	eb	22	3e	43	cd	c7	42	21	ea	-
41d0	42	cd	e5	ff	eb	22	40	43	2a	3e	43	eb	2a	3c	43	2b	-
41e0	cd	20	42	da	31	42	2a	40	43	eb	2a	3c	43	cd	01	41	-
41f0	cd	9d	41	d2	fc	41	cd	0e	42	c3	d8	41	cd	a8	41	eb	-
4200	22	40	43	eb	11	80	00	19	22	3c	43	c3	d8	41	cd	c7	-
4210	42	21	f0	42	cd	df	ff	7a	cd	be	ff	7b	cd	be	ff	c9	-
4220	7c	92	da	2f	42	c2	2d	42	7d	93	da	2f	42	37	c9	b7	-
4230	c9	cd	c7	42	21	00	43	cd	df	ff	2a	40	43	2b	7d	fe	-
4240	ff	c2	46	42	2e	1f	7c	cd	be	ff	7d	cd	be	ff	c9	cd	-
4250	c7	42	21	14	43	cd	e5	ff	eb	22	42	43	cd	c7	42	21	-
4260	20	43	cd	e5	ff	eb	22	44	43	cd	c7	42	21	d4	42	cd	-
4270	e5	ff	eb	22	3c	43	2a	44	43	eb	cd	a8	41	2a	42	43	-
4280	cd	20	42	da	ab	42	2a	42	43	eb	2a	3c	43	cd	47	41	-
4290	d2	99	42	cd	0e	42	c3	76	42	cd	a8	41	eb	22	42	43	-
42a0	eb	11	80	00	19	22	3c	43	c3	76	42	cd	c7	42	21	2a	-
42b0	43	cd	df	ff	2a	3c	43	2b	7c	cd	be	ff	7d	cd	be	ff	-
42c0	c9	3e	00	cd	33	40	c9	e5	21	d0	42	cd	df	ff	e1	c9	-
42d0	0d	0a	03	03	4d	45	4d	20	53	54	41	52	54	3a	20	03	-
42e0	4d	45	4d	20	45	4e	44	3a	20	03	54	2f	53	3a	20	03	-
42f0	45	52	52	4f	52	20	41	54	20	54	2f	53	3a	20	03	03	-
4300	4c	41	53	54	20	54	52	41	43	4b	20	53	45	43	54	4f	-
4310	52	3a	20	03	54	2f	53	20	53	54	41	52	54	3a	20	03	-

```

4320 54 2f 53 20 45 4e 44 3a 20 03 4c 41 53 54 20 4d -
4330 45 4d 20 41 44 44 52 3a 20 03 00 01 00 30 ff ff -
4340 00 01 00 01 1f 00 ff 00 1f 00 00 00 23 c3 77 ff -
4350 cd d9 ff fe 03 ca 7f ff 3e 21 cd ee ff c9 00 00 -
4360 41 44 44 52 3f 20 03 03 20 4e 41 4d 45 3f 20 03 -
4370 f5 d5 e5 c5 cd e3 f1 c1 e1 d1 f1 c9 fb c3 c9 e3 -
4380 fc c3 93 f0 c3 c2 fd c3 a2 fe c3 a4 ff c3 d4 fe -
4390 c3 bc c9 c9 00 00 c3 0f ff c3 ee fd c3 ee fc c3 -
43a0 9f fd c3 94 fd c3 c6 fc c3 7e fd c3 b5 fd c3 6e -
43b0 fd c3 14 fe c3 64 fd c3 4f fd c3 d1 fc c3 a0 f0 -
43c0 c3 36 fd c3 23 fd c3 3f ff c3 20 fe 37 00 00 00 -
?
```

Bu ffer

```

tprwt
WHAT?
? wrtte
WHAT?
? wrtSTART? 4000 END? 43ce
NAME? disk.obj!
?
```

```

rdtADDR? 4000 NAME? disk.obj?
rdtADDR? 4000 NAME? #disk? rdtADDR? 4000 NAME? #disk!
? wrtSTART? 4000 END? 43ce
NAME? disk.obj!
? rdtADDR? 4000 NAME? disk.obj!
```

```

? dump 4000 43ce
4000 c3 c1 42 c3 4f 42 c3 b1 41 c3 33 40 c3 47 41 c3
4010 0f 41 c3 9d 41 e5 d5 c5 7a cd 54 40 cd cd 40 c1
4020 d1 e1 c9 e5 d5 c5 7a cd 54 40 3e 87 cd 91 40 c1
4030 d1 e1 c9 d3 08 db 08 e6 40 ca 4e 40 db 08 e6 02
4040 c2 3c 40 3e 02 d3 09 3e 00 d3 09 c3 35 40 3e 00
4050 32 3a 43 c9 c5 47 3a 3a 43 4f b8 ca 8f 40 d2 78
4060 40 db 08 e6 02 c2 61 40 3e 01 d3 09 3e 00 d3 09
4070 0c 79 32 3a 43 c3 56 40 db 08 e6 02 c2 78 40 3e
4080 02 d3 09 3e 00 d3 09 0d 79 32 3a 43 c3 56 40 c1
4090 c9 4f 3e 88 91 47 cd ef 40 3e 80 d3 09 16 01 3e
40a0 80 b6 5f 23 db 08 a2 c2 a4 40 83 d3 0a 7e 23 5e
40b0 23 0d ca bb 40 0d d3 0a c2 a4 40 db 08 a2 c2 bb
40c0 40 d3 0a 05 c2 bb 40 fb 3e 08 d3 09 c9 cd ef 40
40d0 0e 89 db 08 b7 fa d2 40 db 0a 77 23 0d ca e9 40
40e0 0d 00 db 0a 77 23 c2 d2 40 fb 3e 08 d3 09 c9 3e
40f0 04 d3 09 f3 db 09 1f da f4 40 e6 1f bb c2 f4 40
4100 c9 c5 d5 e5 3e ff 32 46 43 7a 32 47 43 7b 32 48
4110 43 3e 00 32 4a 43 32 4b 43 e1 e5 11 4c 43 06 80
4120 7e 12 23 13 05 c2 20 41 06 86 21 46 43 3e 00 4e
4130 81 23 05 c2 2f 41 32 cc 43 e1 d1 d5 e5 21 46 43
4140 cd 23 40 e1 d1 c1 c9 c5 d5 e5 3e 01 32 3b 43 21
4150 46 43 cd 15 40 11 4c 43 e1 e5 3a 3b 43 b7 ca 6b
4160 41 06 80 1a 77 23 13 05 c2 63 41 e1 d1 d5 e5 3a
4170 47 43 ba c2 98 41 3a 48 43 bb c2 98 41 06 86 21
4180 46 43 3e 00 4e 81 23 05 c2 84 41 4f 3a cc 43 b9
4190 c2 98 41 e1 d1 c1 b7 c9 e1 d1 c1 37 c9 c5 d5 e5
41a0 3e 00 32 3b 43 c3 4f 41 1c 7b fe 20 c0 1e 00 14
41b0 c9 cd c7 42 21 d4 42 cd e5 ff eb 22 3c 43 cd c7
41c0 42 21 e0 42 cd e5 ff eb 22 3e 43 cd c7 42 21 ea
41d0 42 cd e5 ff eb 22 40 43 2a 3e 43 eb 2a 3c 43 2b
41e0 cd 20 42 da 31 42 2a 40 43 eb 2a 3c 43 cd 01 41
41f0 cd 9d 41 d2 fc 41 cd 0e 42 c3 d8 41 cd a8 41 eb
4200 22 40 43 eb 11 80 00 19 22 3c 43 c3 d8 41 cd c7
4210 42 21 f0 42 cd df ff 7a cd be ff 7b cd be ff c9
4220 7c 92 da 2f 42 c2 2d 42 7d 93 da 2f 42 37 c9 b7
```

12
11
10
9
8
7
6
5
4

4230 c9 cd c7 42 21 00 43 cd df ff 2a 40 43 2b 7d fe
4240 ff c2 46 42 2e 1f 7c cd be ff 7d cd be ff c9 cd
4250 c7 42 21 14 43 cd e5 ff eb 22 42 43 cd c7 42 21
4260 20 43 cd e5 ff eb 22 44 43 cd c7 42 21 d4 42 cd
4270 e5 ff eb 22 3c 43 2a 44 43 eb cd a8 41 2a 42 43
4280 cd 20 42 da ab 42 2a 42 43 eb 2a 3c 43 cd 47 41
4290 d2 99 42 cd 0e 42 c3 76 42 cd a8 41 eb 22 42 43
42a0 eb 11 80 00 19 22 3c 43 c3 76 42 cd c7 42 21 2a
42b0 43 cd df ff 2a 3c 43 2b 7c cd be ff 7d cd be ff
42c0 c9 3e 00 cd 33 40 c9 e5 21 d0 42 cd df ff e1 c9
42d0 0d 0a 03 03 4d 45 4d 20 53 54 41 52 54 3a 20 03
42e0 4d 45 4d 20 45 4e 44 3a 20 03 54 2f 53 3a 20 03
42f0 45 52 52 4f 52 20 41 54 20 54 2f 53 3a 20 03 03
4300 4c 41 53 54 20 54 52 41 43 4b 20 53 45 43 54 4f
4310 52 3a 20 03 54 2f 53 20 53 54 41 52 54 3a 20 03
4320 54 2f 53 20 45 4e 44 3a 20 03 4c 41 53 54 20 4d
4330 45 4d 20 41 44 44 52 3a 20 03 00 01 00 30 ff ff
4340 00 01 00 01 1f 00 ff 00 1f 00 00 00 23 c3 77 ff
4350 cd d9 ff fe 03 ca 7f ff 3e 21 cd ee ff c9 00 00
4360 41 44 44 52 3f 20 03 03 20 4e 41 4d 45 3f 20 03
4370 f5 d5 e5 c5 cd e3 f1 c1 e1 d1 f1 c9 fb c3 c9 e3
4380 fc c3 93 f0 c3 c2 fd c3 a2 fe c3 a4 ff c3 d4 fe
4390 c3 bc c9 c9 00 00 c3 0f ff c3 ee fd c3 ee fc c3
43a0 9f fd c3 94 fd c3 c6 fc c3 7e fd c3 b5 fd c3 6e
43b0 fd c3 14 fe c3 64 fd c3 4f fd c3 d1 fc c3 a0 f0
43c0 c3 36 fd c3 23 fd c3 3f ff c3 20 fe 37 00 00
? exec 4000

?

exec 4003

T/S START: 14 00
T/S END: 14 19
MEM START: 0000
ERROR AT T/S: 13 00
ERROR AT T/S: 13 00
ERROR AT T/S: 13 00
ERROR AT T/S: 13 00
ERROR AT T/S: 13 00
ERROR AT T/S: 13 00
ERROR AT T/S: 13 00
ERROR AT T/S: ? ec^ex

d

exec 4000

? exec 4003

T/S START: 1400
T/S END: 1419
MEM START: 2000
LAST MEM ADDR: 2c ff
? list

0062 jmp init
12 0065 jmp din
11 0068 jmp dout
10 0071 jmp dver
9 0100 dkin push h
8 0110 push d
7 0120 push b
6 0130 mov a,d #track number
5 0140 call nwtrk
4 0150 call dski
0160 pop b
0170 pop d
0180 pop h

```
0190 ret
0200 dkout push h
0210 push d
0220 push b
0230 mov a,d
0240 call nwtrk
0250 mvi a,1?
```

files

? list

```
0062 jmp init
0065 jmp din
0068 jmp dout
0071 jmp dver
0100 dkin push h
0110 push d
0120 push b
0130 mov a,d ;track number
0140 call nwtrk? file
```

? exec 4003

```
T/S START: 1200
T/S END: 1209
MEM START: 2000
LAST MEM ADDR: 24 ff
```

? list

```
0062 jmp init
0065 jmp din
0068 jmp dout
0071 jmp dver
0100 dkin push h
0110 push d
0120 push b
0130 mov a,d ;track ? diu
```

```
dump d000 md00f
WHAT?
```

```
? dump d000 d00f
d000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
? file presc 2000
WHAT?
```

```
? file P~/presj~k/2000
presc 2000 2000
```

```
? file /presj/ 0
```

```
? file /presk ~c/ 2000
presc 2000 2000
```

```
? fixfADDR? 2000
```

```
12 ? file
11 presc 2000 2000
```

```
10
9 ? fixfADDR? 20~4ff
```

```
8 ? file
7 presc 24ff 2cae
```

```
6
5 ? fixfADDR? 2000
```

```
4 ? file
presc 2000 2000
```

? exec 4000

? exec 403^03

T/S START: 1200

T/S END: 1209

MEM START: 2000

LAST MEM ADDR: 24 ff

? list

0000 org 03fdh

0010 jmp presc

0100 org 0040h

0110 ****

0120 *

0130 * Module PRESC

0140 *

0150 * Written by Bruce Cichowles 8/25/76

0160 * Copyright Circle C - ARP Instruments 9/15/76

0170 *

0180 * PRESC - checks the preset panel and sets the previous

0190 * preset readings (prvpr), as well as the preset request

0200 * word (prera). This routine should be invoked frequently,

0210 * as it actually does the preset panel scanning.

0220 *

0230 * No arguments

0240 *

0250 ****

1000 presc equ \$

1010 push b

1020 lda paddr ;preset panel addr

1030 ora a ;set flag

1040 jz pres1 ;none set

1050 mov b,a

1060 lda prvpr ;load previous reading

1070 ora b

1080 sta prvpr

1090 pres2 pop b

1100 ret

1110 pres1 lda prvpr ;find out if any were set previously

1120 ora a

1130 jz pres2 ;none before

1140 mov b,a

1150 lda prera

1160 ora b ;include bits set in b

1170 sta prera

1180 xra a

1190 sta prvpr ;store 0 for the previous reading

1200 pop b

1210 ret

5000 paddr equ 1800h

6000 *RAM

6001 prvpr equ 08ffh

6002 prera equ 08feh

? file

presc 2000 2000

? fix

WHAT?

? fixfADDR?

? file /presc/ 0

? file pres^*/presc/ 1000

presc 1000 1000

? fixfADDR? 2000

? file

T/S END: 2007

? exec 4003

T/S START: 3200

T/S END: 3210

MEM START: 1000

ERROR AT T/S: 32 00

ERROR AT T/S: 32 00

ERROR AT T/S: 32 00

ERROR AT T/S: 32 00

ERROR AT T/S: 32 00

ERROR AT T/S: 32 00

ERROR AT T/S: 32 00

ERROR AT T/S: 32 00

ERRO?

```

mem1  file mem~~~/mem1/3000
      3000 3000

```

? rdfil

WHAT?

? rdfil

WHAT?

? rdf1

WHAT?

? ece

WHAT?

?

exec ff3f

ADDR? 4000 NAME? mem1? rdfil

WHAT? --

? exec ff3f

ADDR? 3000~~~~4000 NAME? mem1!

? fixfADDR? 4000

? file

mem1 4000 433b

? list

0000 mem1 nop

0005 mvi e, 0ffh

0010 assign mvi h,0

0020 mvi b, 03fh

0030 xra a

0035 writl mvi l,00

0040 mvi c, 0ffh

0045 ll mov m,a

0050 inr l

0060 dcr c

0070 jnz ll

0080 writh inr h

0090 call writl

0100 dcr b

0110 jnz writh

0115 lxi h,00h

0120 tst nop

0125 call fack

0128 nop

0130 mov a,m

0140 cpi 00h

0150 jnz err

0160 inx h

0170 jmp tst

0200 err mov a,h

0210 call outnc