```
1 REM    Z-80 Disassembler
2 REM (C) 1984 Stephen Devine
3 REM
10 MODE 1:BORDER 12:INK 0,12:INK 1,0
20 DIM s$(14,7),a$(3),mcode%(8),byte%(5)
30 GOSUB 5000
40 CLS
50 ON ERROR GOTO 4500
60 LINE INPUT"START ADDRESS:";start$
70 start=INT(VAL(start$)):IF start)65535 THEN 4500
80 IF start(0 THEN start=start+65536:IF start(0 THEN 4500
90 dec%=(ASC(start$)()ASC("&"))
100 LINE INPUT"END ADDRESS:";last$
110 IF last$="" THEN last=65535:GOTO 140
120 last=INT(VAL(last$)):IF last(0 THEN last=last+65536
130 IF last)65535 OR last(start THEN 4500
140 LINE INPUT"PRINTER (y/n)?";pr$
150 IF UPPER$(pr$)="Y" THEN pr%=8 ELSE pr%=0
160 PRINT
170 ON ERROR GOTO 4510
180 baddr=start
199 REM Start of main loop to determine mnemonic
200 type%=0:rep%=0:n%=-1
210 mcode%(1)=@byte%(0)
220 CALL @mcode%(0),baddr
230 n%=n%+1
240 IF df% THEN mn$="DATA":df%=df%-1:GOTO 3000
250 IF byte%(n%)=&76 THEN mn$="HALT":GOTO 1000
259 REM Determine if instruction is of specific type
260 IF byte%(n%)=&CB THEN type%=1:IF rep% THEN n%=n%+1:GOTO 230 ELSE 2
30
270 IF byte%(n%)=&ED THEN type%=2:GOTO 230
280 IF byte%(n%)=&DD THEN rep%=1:GOTO 230
290 IF byte%(n%)=&FD THEN rep%=2:GOTO 230
299 REM Convert to binary and isolate important bit patterns
300 b$=BIN$(byte%(n%))
310 WHILE LEN(b$)(8:b$="0"+b$:WEND
320 bh%=VAL("&x"+LEFT$(b$,2))
330 bm%=VAL("&x"+MID$(b$,3,3))
340 bl%=VAL("&x"+RIGHT$(b$,3))
350 bmh%=bm%\2
360 bml%=bm%-2*bmh%
369 REM Main routine to determine mnemonic from bit patterns
370 ON bh% GOTO 510,520,530
380 ON bl% GOTO 420,440,450,470,480,490,500
390 rel%=bm%)1
400 IF bm%)3 THEN mn$="JR "+s$(3,bm%-4)+",V" ELSE mn$=s$(5,bm%)
410 GOTO 1000
420 IF bml%=0 THEN mn$="LD "+s$(1,bmh%)+",W" ELSE mn$="ADD *,"+s$(1,bm
h%)
430 GOTO 1000
440 mn$="LD "+s$(6,bm%):GOTO 1000
450 IF bml%=0 THEN mn$="INC "+s$(1,bmh%) ELSE mn$="DEC "+s$(1,bmh%)
460 GOTO 1000
470 mn$="INC "+s$(0,bm%):GOTO 1000
480 mn$="DEC "+s$(0,bm%):GOTO 1000
490 mn$="LD "+s$(0,bm%)+",V":GOTO 1000
500 IF bl%=7 THEN mn$=s$(7,bm%):GOTO 1000
510 mn$="LD "+s$(0,bm%)+","+s$(0,bl%):GOTO 1000
520 mn$=s$(4,bm%)+s$(0,bl%):GOTO 1000
530 ON bl% GOTO 550,570,580,590,600,620,630
540 mn$="RET "+s$(3,bm%):GOTO 1000
550 IF bml%=0 THEN mn$="POP "+s$(2,bmh%) ELSE mn$=s$(8,bmh%)
560 GOTO 1000
570 mn$="JP "+s$(3,bm%)+",W":GOTO 1000
580 mn$=s$(9,bm%):GOTO 1000
590 mn$="CALL "+s$(3,bm%)+",W":GOTO 1000
600 IF bml%=0 THEN mn$="PUSH "+s$(2,bmh%) ELSE mn$="CALL W"
610 GOTO 1000
620 mn$=s$(4,bm%)+"V":GOTO 1000
630 mn$="RST"
640 IF NOT dec% THEN mn$=mn$+" ":IF bm%(2 THEN mn$=mn$+"0"
650 IF NOT dec% THEN mn$=mn$+HEX$(bm%*8) ELSE mn$=mn$+STR$(bm%*8)
660 IF bm%=1 OR bm%=2 OR bm%=5 THEN df%=2
670 IF bm%=3 THEN df%=3
999 REM This part caters for opcodes beginning with &CB or &ED
1000 IF type%=0 THEN 2000
1010 mn$="DATA"
1020 IF type%)1 THEN 1090
1030 IF bh%=0 THEN IF bm%()6 THEN mn$=s$(10,bm%)+" "+s$(0,bl%):GOTO 20
00 ELSE 2000
1050 IF bh%=2 THEN mn$="RES"
1060 IF bh%=3 THEN mn$="SET"
1070 mn$=mn$+STR$(bm%)+","+s$(0,bl%)
1080 GOTO 2000
1090 IF bh%=0 OR bh%=3 THEN mn$="DATA":GOTO 2000
1100 IF bh%=2 THEN 1250
1110 ON bl% GOTO 1130,1150,1170,1190,1200,1230,1240
1120 IF bl%=0 AND bm%()6 THEN mn$="IN "+s$(0,bm%)+",(C)"
1130 IF bl%=1 AND bm%()6 THEN mn$="OUT (C),"+s$(0,bm%)
1140 GOTO 2000
1150 IF bml%=0 THEN mn$="SBC HL,"+s$(1,bmh%) ELSE mn$="ADC HL,"+s$(1,b
mh%)
1160 GOTO 2000
1170 IF bml%=0 THEN mn$="LD (W),"+s$(1,bmh%) ELSE mn$="LD "+s$(1,bmh%)
+",(W)"
1180 GOTO 2000
1190 IF bm%=0 THEN mn$="NEG":GOTO 2000 ELSE 2000
1200 IF bm%=0 THEN mn$="RETN":GOTO 2000
1210 IF bm%=1 THEN mn$="RETI"
1220 GOTO 2000
1230 mn$=s$(13,bm%):GOTO 2000
1240 mn$=s$(14,bm%):GOTO 2000
1250 mn$=s$(11,bl%):IF mn$="DATA"THEN 2000
1260 IF bl%=3 AND bm%)5 THEN mn$="OT"
1270 IF bm%(4 THEN mn$=""
1280 mn$=mn$+s$(12,bm%)
1999 REM This part caters for HL and the index registers
2000 IF mn$="DATA" THEN df%=1:n%=0:GOTO 240
2010 IF rep%)0 THEN 2050
2020 r1$="*":r2$="HL":GOSUB 4000
2030 r1$="@":r2$="(HL)":GOSUB 4000
```

```
2040 GOTO 2200
2050 b%=byte%(2)
2060 IF b%)127 THEN b%=b%-256
2070 dd$=STR$(b%):IF b%)=0 THEN dd$="+"+RIGHT$(dd$,LEN(dd$)-1)
2080 rf%=0
2090 IF rep%)1 THEN 2130
2100 r1$="*":r2$="IX":GOSUB 4000:rf%=f%
2110 r1$="@":r2$="(IX"+dd$+")":GOSUB 4000:IF f% THEN rf%=1
2120 GOTO 2150
2130 r1$="*":r2$="IY":GOSUB 4000:rf%=f%
2140 r1$="@":r2$="(IY"+dd$+")":GOSUB 4000:IF f% THEN rf%=1
2150 type%=0 AND f%=-1 THEN n%=n%+1
2160 IF rf%=0 THEN mn$="DATA":n%=0:GOTO 3000
2199 REM Replace all V's and W's with their equivalent bytes
2200 r1$="V"
2210 IF NOT rel% THEN 2280
2220 offset%=byte%(1)
2230 IF offset%)127 THEN offset%=offset%-256
2240 taddr=baddr+2+offset%:IF taddr)65535 THEN taddr=taddr-65536
2250 IF NOT dec% THEN r2$=HEX$(taddr) ELSE r2$=STR$(taddr) :r2$=RIGHT$
(r2$,LEN(r2$)-1):GOTO 2270
2260 WHILE LEN(r2$)(4:r2$="0"+r2$:WEND
2270 rel%=0:GOTO 2300
2280 IF NOT dec% THEN r2$=HEX$(byte%(n%+1)) ELSE r2$=STR$(byte%(n%+1))
:r2$=RIGHT$(r2$,LEN(r2$)-1):GOTO 2300
2290 IF LEN(r2$)=1 THEN r2$="0"+r2$
2300 GOSUB 4000
2310 IF f% THEN n%=n%+1:GOTO 3000
2320 r1$="W"
2330 IF dec% THEN r2$=STR$(byte%(n%+1)+256*byte%(n%+2)):r2$=RIGHT$(r2$
,LEN(r2$)-1):GOTO 2380
2340 r2$=HEX$(byte%(n%+1))
2350 IF LEN(r2$)=1 THEN r2$="0"+r2$
2360 r2$=HEX$(byte%(n%+2))+r2$
2370 IF LEN(r2$)=3 THEN r2$="0"+r2$
2380 GOSUB 4000
2390 IF f% THEN n%=n%+2
2999 REM Print out disassembled mnemonic
3000 addr$=HEX$(baddr)
3010 IF dec% THEN addr$=STR$(baddr):addr$=RIGHT$(addr$,LEN(addr$)-1)
3020 WHILE LEN(addr$)(4-dec%:addr$="0"+addr$:WEND
3030 PRINT#pr%,addr$;TAB(7);
3040 FOR i%=0 TO n%
3050  b%=byte%(i%):a$(i%)=","
3060  IF b%)=32 AND b%(128 THEN a$(i%)=CHR$(b%)
3070  c$=HEX$(b%)
3080  IF LEN(c$)=1 THEN c$="0"+c$
3090  PRINT#pr%,c$;" ";
3100 NEXT i%
3110 p%=INSTR(mn$," "):PRINT#pr%,TAB(20);
3120 IF p% THEN PRINT#pr%,LEFT$(mn$,p%);TAB(25);RIGHT$(mn$,LEN(mn$)-p%
);ELSE PRINT#pr%,mn$;
3130 PRINT#pr%,TAB(35);";";:FOR i%=0 TO n%:PRINT#pr%,a$(i%);:NEXT i%:P
RINT#pr%
3140 IF INKEY$=" "THEN WHILE INKEY$="":WEND
3149 REM Update location pointer and repeat
3150 baddr=baddr+n%+1:IF baddr(=last THEN 200
3160 IF pr% THEN PRINT#8,CHR$(12);
3170 GOTO 50
3999 REM Replace every occurance of r1$ in mn$ with r2$
4000 f%=0
4010 p%=INSTR(mn$,r1$):IF p%=0 THEN RETURN
4020 f%=-1:mn$=LEFT$(mn$,p%-1)+r2$+RIGHT$(mn$,LEN(mn$)-p%)
4030 GOTO 4010
4499 REM Error traps
4500 RESUME 40
4510 RESUME 60
4999 REM Initialize string array and set up machine code
5000 FOR i%=0 TO 14
5010  FOR j%=0 TO 7:READ s$(i%,j%):NEXT j%
5020 NEXT i%
5030 mcode%(0)=&2100
5040 mcode%(1)=@byte%(0)
5050 FOR i%=2 TO 8:READ v:mcode%(i%)=v:NEXT
5060 RETURN
7999 FOR i=@mcode%(0) TO @mcode%(0)+20:PRINT HEX$(PEEK(i));" ";:NEXT
9999 REM Data for mnemonics
10000 DATA B,C,D,E,H,L,@,A
10010 DATA BC,DE,*,SP,,,,
10020 DATA BC,DE,*,AF,,,,
10030 DATA NZ,Z,NC,C,PO,PE,P,M
10040 DATA "ADD A,","ADC A,","SUB ","SBC A,","AND ","XOR ","OR ","CP "
10050 DATA NOP,"EX AF,AF'",DJNZ V,JR V,,,,
10060 DATA "(BC),A","A,(BC)","(DE),A","A,(DE)","(W),*","*,(W)","(W),A"
,"A,(W)"
10070 DATA RLCA,RRCA,RLA,RRA,DAA,CPL,SCF,CCF
10080 DATA RET,EXX,JP (*),"LD SP,*",,,,
10090 DATA JP W,DATA,"OUT (V),A","IN A,(V)","EX (SP),*","EX DE,HL",DI,
EI
10100 DATA RLC,RRC,RL,RR,SLA,SRA,DATA,SRL
10110 DATA LD,CP,IN,OUT,DATA,DATA,DATA,DATA
10120 DATA DATA,DATA,DATA,DATA,I,D,IR,DR
10130 DATA IM 0,DATA,IM 1,IM 2,DATA,DATA,DATA,DATA
10140 DATA "LD I,A","LD R,A","LD A,I","LD A,R",RRD,RLD,DATA,DATA
10199 REM Data for machine code
10200 DATA &cd,&06b9,&1a04,&1377,&2323,&f910,&c9
```

SAMPLE OUTPUT

```
C000   80              ADD   A,B          ;.
C001   01 00 00        LD    BC,0000      ;...
C004   4C              LD    C,H          ;L
C005   C0              RET   NZ           ;.
C006   31 00 C0        LD    SP,C000      ;1..
```