

```

1/      0 :      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2/      0 :      ;
3/      0 :      ; NAME:    black-adder.asm
4/      0 :      ; EDITOR:  Kevin Cole ("The Ubuntourist") <kevin.cole@novawebdevelopment.org>
5/      0 :      ; LASTMOD: 2020.11.03 (kjc)
6/      0 :      ;
7/      0 :      ; DESCRIPTION:
8/      0 :      ;
9/      0 :      ;      Increment a memory location each time a key is pressed.
10/     0 :      ;
11/     0 :      ;      To take full advantage of the included ANSI escape sequences,
12/     0 :      ;      start minicom with the `-c on` option:
13/     0 :      ;
14/     0 :      ;      $ minicom -c on altair
15/     0 :      ;
16/     0 :      ;      or include it in the MINICOM environment variable:
17/     0 :      ;
18/     0 :      ;      $ export MINICOM="-m -c on"
19/     0 :      ;      $ minicom altair
20/     0 :      ;
21/     0 :      ;      NOTES: For help with ANSI escape sequences see:
22/     0 :      ;
23/     0 :      ;      https://en.wikipedia.org/wiki/ANSI_escape_code
24/     0 :      ;
25/     0 :      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
26/     0 :
27/     0 :      ; Code segment
28/     0 :
29/     0 :      INCLUDE      stdio ; Include standard I/O library
(1) 1/     0 :      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(1) 2/     0 :      ;
(1) 3/     0 :      ; NAME:      stdio.inc
(1) 4/     0 :      ; EDITOR:  Kevin Cole ("The Ubuntourist") <kevin.cole@novawebdevelopment.org>
(1) 5/     0 :      ; LASTMOD: 2020.11.03 (kjc)
(1) 6/     0 :      ;
(1) 7/     0 :      ; DESCRIPTION:
(1) 8/     0 :      ;
(1) 9/     0 :      ;      A rudimentary "include" for standard input and output functions
(1) 10/    0 :      ;
(1) 11/    0 :      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(1) 12/    0 :
(1) 13/    0 :      ;;;;;;;;;;;;;;
(1) 14/    0 :      ; Constants ;
(1) 15/    0 :      ;;;;;;;;;;;;;;
(1) 16/    0 :
(1) 17/    0 :      ; ASCII characters
(1) 18/    0 :

```

```

(1) 19/      0 : =150      CR: EQU 0DH ; ASCII CR (Carriage Return, a.k.a. Ctrl-M)
(1) 20/      0 : =120      LF: EQU 0AH ; ASCII LF (Line Feed a.k.a. Ctrl-J)
(1) 21/      0 : =330      ESC: EQU 1BH ; ASCII ESC (Escape, a.k.a. Ctrl-[)
(1) 22/      0 : =00      NUL: EQU 00H ; ASCII NUL (Null)
(1) 23/      0 :
(1) 24/      0 :          ; I/O
(1) 25/      0 :
(1) 26/      0 : =200      SI01S: EQU 10H ; Serial I/O communications port 1 STATUS
(1) 27/      0 : =210      SI01D: EQU 11H ; Serial I/O communications port 1 DATA
(1) 28/      0 :
(1) 29/      0 : =30      MRST: EQU 03H ; UART Master Reset
(1) 30/      0 : =10      RCVD: EQU 01H ; Character received
(1) 31/      0 : =20      SENT: EQU 002H ; Data sent. Output complete
(1) 32/      0 :
(1) 33/      0 :          ; Code segment
(1) 34/      0 :
(1) 35/ 10000 :          ORG 1000H ; Load at memory location 1000 (hex)
(1) 36/ 10000 :
(1) 37/ 10000 :          ; Initialize serial input / output
(1) 38/ 10000 :          ;
(1) 39/ 10000 : 076 003    INITIO: MVI A, MRST
(1) 40/ 10002 : 323 020      OUT SI01S ; Reset the UART
(1) 41/ 10004 : 076 025      MVI A, 15h ; Settings: No RI, No XI, RTS Low, 8N1, /16
(1) 42/ 10006 : 323 020      OUT SI01S ; Configure the UART with above settings
(1) 43/ 10010 : 311          RET ; Return
(1) 44/ 10011 :
(1) 45/ 10011 :          ; Put a character on to the serial I/O bus (stdout)
(1) 46/ 10011 :          ;
(1) 47/ 10011 : 365      PUTC: PUSH PSW ; Preserve Program Status Word
(1) 48/ 10012 : 333 020      WAIT0: IN SI01S ; Check serial I/O status bit 1 (XMIT status)
(1) 49/ 10014 : 346 002      ANI SENT ; If data not sent (i.e. XMIT not finished)...
(1) 50/ 10016 : 312 012 020      JZ WAIT0 ; ...spin wheels: continue checking status. Else...
(1) 51/ 10021 : 361      POP PSW ; ...restore Program Status Word
(1) 52/ 10022 : 323 021      OUT SI01D ; ...output byte
(1) 53/ 10024 : 311          RET ; ...return
(1) 54/ 10025 :
(1) 55/ 10025 :          ; Write a null-terminated string out to the serial port (stdout)
(1) 56/ 10025 :          ;
(1) 57/ 10025 : 012      WRITE: LDAX B ; Fetch byte
(1) 58/ 10026 : 376 000      CPI NUL ; If byte is ASCII NUL...
(1) 59/ 10030 : 310      RZ ; ...return. Else...
(1) 60/ 10031 : 315 011 020      CALL PUTC ; ...output byte
(1) 61/ 10034 : 003      INX B ; ...point to next byte
(1) 62/ 10035 : 303 025 020      JMP WRITE ; ...lather, rinse, repeat: Fetch next byte.
(1) 63/ 10040 :
(1) 64/ 10040 :          ; Get a character off of the serial I/O bus (stdin)
(1) 65/ 10040 :          ;

```

```

(1) 66/ 10040 : 365      GETC:  PUSH  PSW   ; Preserve Program Status Word
(1) 67/ 10041 : 333 020  WAITI:  IN    SI01S ; Check serial I/O status bit
(1) 68/ 10043 : 346 001      ANI    RCVD  ; If no data received...
(1) 69/ 10045 : 312 041 020  JZ     WAITI ; ...spin wheels: continue checking status. Else...
(1) 70/ 10050 : 361      POP    PSW   ; ...restore Program Status Word
(1) 71/ 10051 : 333 021      IN     SI01D ; ...read the character
(1) 72/ 10053 : 323 021      OUT    SI01D ; ...echo it
(1) 73/ 10055 : 311      RET         ; ...return
(1) 74/ 10056 :
(1) 75/ 10056 :          ; Read one line (CR- or LF-terminated string) from the serial port (stdin)
(1) 76/ 10056 :          ;
(1) 77/ 10056 : 315 040 020  READ:  CALL  GETC  ; Fetch byte
(1) 78/ 10061 : 376 015      CPI    CR   ; If byte is an ASCII CR (Carriage Return)...
(1) 79/ 10063 : 312 111 020  JZ     CRLF  ; ...add LF and return. Else...
(1) 80/ 10066 : 376 012      CPI    LF   ; ...if byte is an ASCII LF (Line Feed)...
(1) 81/ 10070 : 312 100 020  JZ     LFCR  ; ...add CR and return. Else...
(1) 82/ 10073 : 002      STAX   B       ; ...store byte in buffer
(1) 83/ 10074 : 003      INX     B       ; ...point to next empty byte
(1) 84/ 10075 : 303 056 020  JMP     READ  ; ...lather, rinse, repeat: Fetch next byte.
(1) 85/ 10100 : 076 015  LFCR:  MVI    A,CR
(1) 86/ 10102 : 315 011 020  CALL  PUTC  ; Print a CR
(1) 87/ 10105 : 076 000      MVI    A,NUL ; Null terminator
(1) 88/ 10107 : 002      STAX   B       ; Terminate the input string
(1) 89/ 10110 : 311      RET         ; Return
(1) 90/ 10111 : 076 012  CRLF:  MVI    A,LF
(1) 91/ 10113 : 315 011 020  CALL  PUTC  ; Print a LF
(1) 92/ 10116 : 076 000      MVI    A,NUL ; Null terminator
(1) 93/ 10120 : 002      STAX   B       ; Terminate the input string
(1) 94/ 10121 : 311      RET         ; Return
(1) 95/ 10122 :
30/ 10122 :
31/ 0 :          ORG    000H ; Load at memory location 000 (hex)
32/ 0 :
33/ 0 : 315 000 020  CALL  INITIO ; Initialize serialize input / output device
34/ 3 :
35/ 3 : 001 004 050  LXI    B,WORDS ; Point to instructions (WORDS)
36/ 6 : 315 025 020  CALL  WRITE ; Write WORDS to stdout (terminal)
37/ 11 :
38/ 11 : 001 312 050  LXI    B,ASK1 ; Point to first prompt (ASK1)
39/ 14 : 315 025 020  CALL  WRITE ; Write ASK1 to stdout (terminal)
40/ 17 :
41/ 17 : 001 004 040  LXI    B,BUFR ; Point to input buffer (BUFR)
42/ 22 : 315 056 020  CALL  READ  ; Read a line from stdin to BUFR
43/ 25 :
44/ 25 : 001 316 050  LXI    B,ASK2 ; Point to first prompt (ASK2)
45/ 30 : 315 025 020  CALL  WRITE ; Write ASK2 to stdout (terminal)
46/ 33 :

```

```

47/      33 : 001 004 040      LXI  B,BUFFR      ; Point to input buffer (BUFFR)
48/      36 : 315 056 020      CALL READ      ; Read a line from stdin to BUFFR
49/      41 :
50/      41 : 166              HLT                ; DEBUG
51/      42 :
52/      42 : 041 000 040      LXI  H,WATCH      ; Set location to increment
53/      45 :
54/      45 : 333 020      COUNT: IN   SI01S ; Check serial I/O status bit
55/      47 : 346 001      ANI   RCVD  ; If no data received...
56/      51 : 312 045 000      JZ    COUNT ; ...spin wheels: continue checking status. Else...
57/      54 : 333 021      IN    SI01D ; ...read the character
58/      56 : 323 021      OUT   SI01D ; ...echo it
59/      60 : 064          INR   M      ; ...increment the watched counter
60/      61 : 303 045 000      JMP   COUNT ; ...wait for next character
61/      64 :
62/      64 :                ; Data segment
63/      64 :
64/  20000 :                ORG   2000H ; Load at memory locaton 8192 (decimal)
65/  20000 :
66/  20000 : 000      WATCH: DB   000H ; Initialize to zero
67/  20001 : 000      VAL1:  DB   000H ; Data Byte at address 2000 (hex) = 0
68/  20002 : 000      VAL2:  DB   000H ; Data Byte at address 2001 (hex) = 0
69/  20003 : 000      SUM:   DB   000H ; Data Byte at address 2002 (hex) = 0
70/  20004 :
71/  20004 :      BUFFR: DS   800H ; ~ one 80x25 screens-worth of bytes at 2003 (hex)
72/  24004 :
73/  24004 : 033 133 062 112      WORDS: DB   ESC,"[2J"
74/  24010 : 015 012      DB   CR,LF
75/  24012 : 015 012 040 040      DB   CR,LF,"          "
      24016 : 040 040 040 040
      24022 : 040 040 040 040
      24026 : 040 040
76/  24030 : 040 040 040 040      DB   "          ",ESC,"[31m","BLACK ADDER",ESC,"[0m",CR,LF
      24034 : 040 040 040 040
      24040 : 040 040 040 040
      24044 : 040 040 040 040
      24050 : 033 133 063 061
      24054 : 155 102 114 101
      24060 : 103 113 040 101
      24064 : 104 104 105 122
      24070 : 033 133 060 155
      24074 : 015 012
77/  24076 : 015 012 040 040      DB   CR,LF,"          "
      24102 : 040 040 040 040
      24106 : 040 040 040 040
      24112 : 040 040
78/  24114 : 124 150 151 163      DB   "This program adds two numbers in the range"

```

```

24120 : 040 160 162 157
24124 : 147 162 141 155
24130 : 040 141 144 144
24134 : 163 040 164 167
24140 : 157 040 156 165
24144 : 155 142 145 162
24150 : 163 040 151 156
24154 : 040 164 150 145
24160 : 040 162 141 156
24164 : 147 145
79/ 24166 : 015 012 040 040      DB    CR,LF,"          "
24172 : 040 040 040 040
24176 : 040 040 040 040
24202 : 040 040
80/ 24204 : 055 061 062 067      DB    "-127 to +127. Enter values X and Y at the"
24210 : 040 164 157 040
24214 : 053 061 062 067
24220 : 056 040 105 156
24224 : 164 145 162 040
24230 : 166 141 154 165
24234 : 145 163 040 130
24240 : 040 141 156 144
24244 : 040 131 040 141
24250 : 164 040 164 150
24254 : 145
81/ 24255 : 015 012 040 040      DB    CR,LF,"          "
24261 : 040 040 040 040
24265 : 040 040 040 040
24271 : 040 040
82/ 24273 : 160 162 157 155      DB    "prompts.",CR,LF
24277 : 160 164 163 056
24303 : 015 012
83/ 24305 : 015 012      DB    CR,LF
84/ 24307 : 015 012      DB    CR,LF
85/ 24311 : 000      DB    NUL    ; NULL string terminator
86/ 24312 :
87/ 24312 : 130 072 040 000      ASK1: DB    "X: ",NUL
88/ 24316 :
89/ 24316 : 131 072 040 000      ASK2: DB    "Y: ",NUL
90/ 24322 :
91/ 24322 :      END          ; End

```

Symbol Table (* = unused):

*ARCHITECTURE :		"x86_64-unknown-linux" -	
ASK1 :	24312 C	ASK2 :	24316 C
*BIGENDIAN :	0 -	*BRANCHEXT :	0 -
BUFFR :	20004 C	*CASESENSITIVE :	0 -
*CONSTPI :	3.141592653589793 -	COUNT :	45 C
CR :	15 -	CRLF :	10111 C
*DATE :	"11/03/2020" -	ESC :	33 -
*FALSE :	0 -	*FULLPMU :	1 -
GETC :	10040 C	*HAS64 :	1 -
*HASDSP :	0 -	*HASFPU :	0 -
*HASPMMU :	0 -	*INEXTMODE :	0 -
INITIO :	10000 C	*INLWORDMODE :	0 -
*INMAXMODE :	0 -	*INSRCMODE :	0 -
*INSUPMODE :	0 -	LF :	12 -
LFCR :	10100 C	*LISTON :	1 -
*MACEXP :	7 -	*MOMCPU :	100200 -
*MOMCPUNAME :	"8080" -	MRST :	3 -
*NESTMAX :	400 -	NUL :	0 -
*PACKING :	0 -	*PADDING :	1 -
PUTC :	10011 C	RCVD :	1 -
READ :	10056 C	*RELAXED :	0 -
SENT :	2 -	SI01D :	21 -
SI01S :	20 -	*SUM :	20003 C
*TIME :	"07:38:39 PM" -	*TRUE :	1 -
*VAL1 :	20001 C	*VAL2 :	20002 C
*VERSION :	12057 -	WAITI :	10041 C
WAITO :	10012 C	WATCH :	20000 C
WORDS :	24004 C	WRITE :	10025 C
*Z80SYNTAX :	0 -		

56 symbols

32 unused symbols

Code Pages:

STANDARD (0 changed characters)

1 code page

0.00 seconds assembly time

197 lines source file

2 passes

0 errors

0 warnings