Assignment No. 1

1. WAP to count positive and negative numbers from a series of eight signed numbers (2435H, 0C00H etc.).

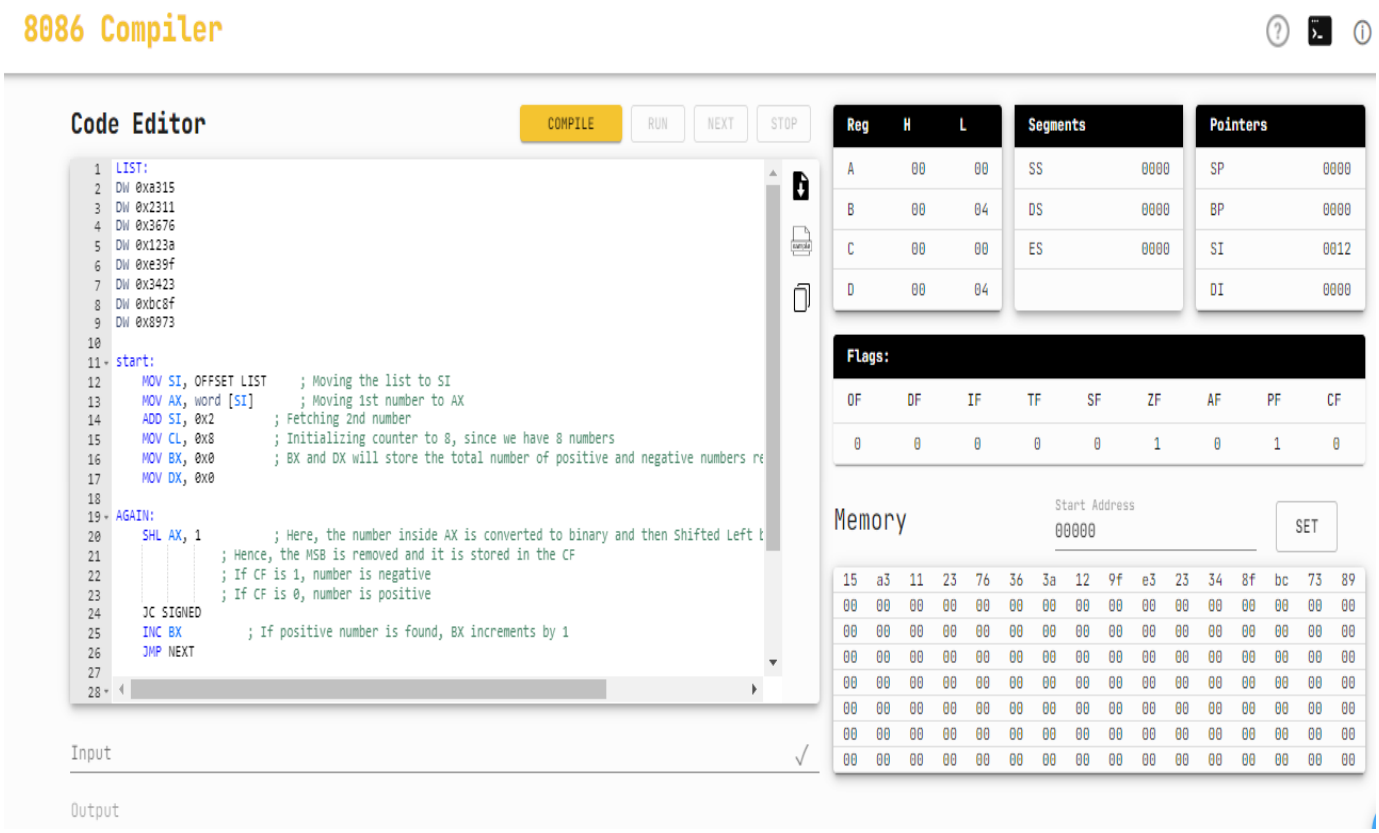
Program:

```
LIST:
DW 0xa315
DW 0x2311
DW 0x3676
DW 0x123a
DW 0xe39f
DW 0x3423
DW 0xbc8f
DW 0x8973
start:
    MOV SI, OFFSET LIST      ; Moving the list to SI
    MOV AX, word [SI]        ; Moving 1st number to AX
    ADD SI, 0x2              ; Fetching 2nd number
    MOV CL, 0x8              ; Initializing counter to 8, since we have 8 numbers
    MOV BX, 0x0              ; BX and DX will store the total number of
                             positive and negative numbers respectively in the
                             given list
  MOV DX, 0x0
```

```
AGAIN:
SHL AX, 1          ; Here, the number inside AX is converted to binary
                     and then Shifted Left by 1 bit
                   ; Hence, the MSB is removed and it is stored in
the CF
                   ; If CF is 1, number is negative
                   ; If CF is 0, number is positive
JC SIGNED
INC BX             ; If positive number is found, BX increments by 1
JMP NEXT

SIGNED:
INC DX             ; If negative number is found, DX increments by 1
NEXT:
ADD SI, 0x2
MOV AX, word [SI]
DEC CL             ; Decrementing counter i.e CL after checking each
                     number
JNZ AGAIN          ; Process continues till CL becomes zero
```

Output:



2.WAP to find out average of a given string (list of numbers) of data bytes neglecting fractions.

Program:

LIST:

DW 0xab18

DW 0x0324

DW 0x03fa

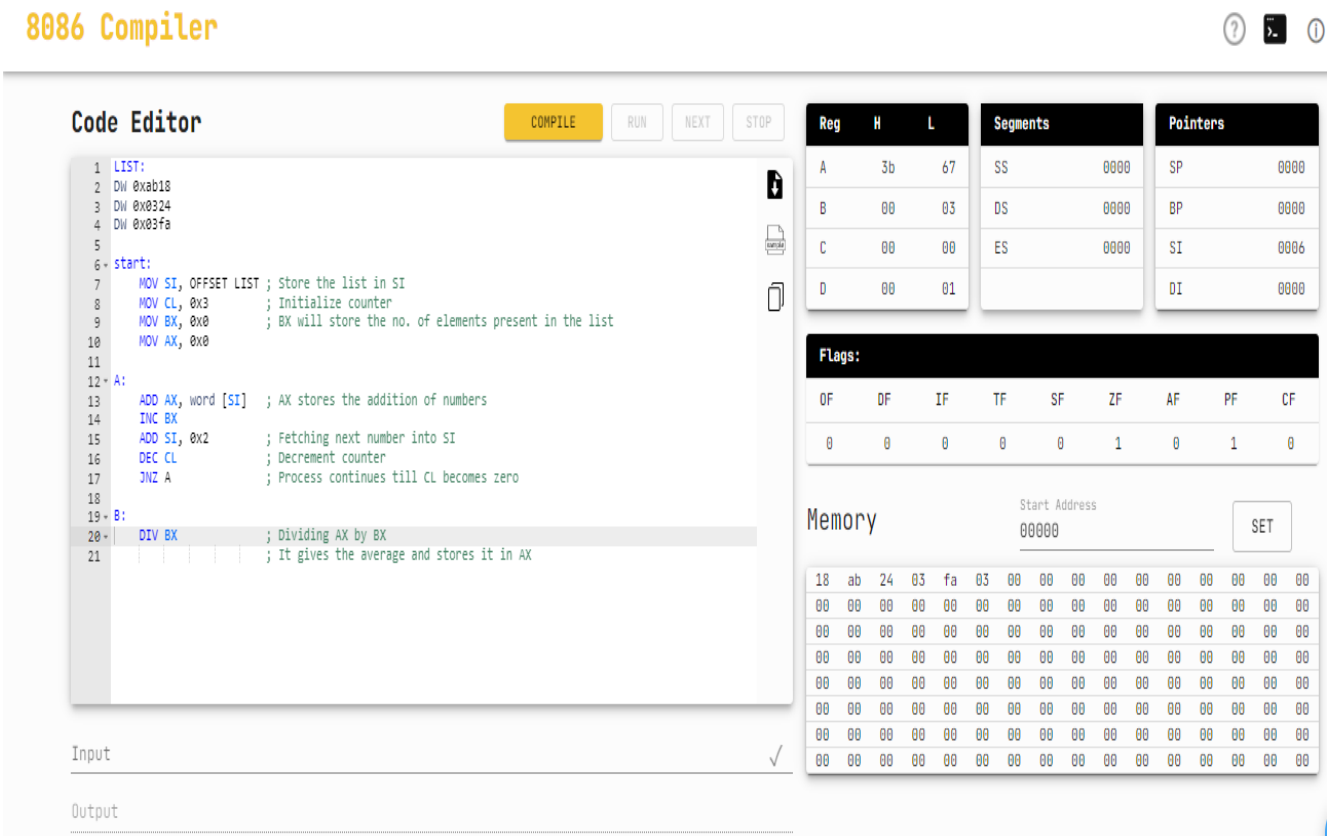
start:

```
    MOV SI, OFFSET LIST ; Store the list in SI
    MOV CL, 0x3         ; Initialize counter
    MOV BX, 0x0         ; BX will store the no. of elements present in the list
    MOV AX, 0x0

A:
    ADD AX, word [SI]   ; AX stores the addition of numbers
    INC BX
    ADD SI, 0x2         ; Fetching next number into SI
    DEC CL              ; Decrement counter
    JNZ A               ; Process continues till CL becomes zero

B:
    DIV BX              ; Dividing AX by BX
                        ; It gives the average and stores it in AX
```

Output:


8086 Compiler
Code Editor
COMPILE
RUN
NEXT
STOP
LIST:
DW 0xab18
DW 0x0324
DW 0x03fa
start:
MOV SI, OFFSET LIST ; Store the list in SI
MOV CL, 0x3 ; Initialize counter
MOV BX, 0x0 ; BX will store the no. of elements present in the list
MOV AX, 0x0
A:
ADD AX, word [SI] ; AX stores the addition of numbers
INC BX
ADD SI, 0x2 ; Fetching next number into SI
DEC CL ; Decrement counter
JNZ A ; Process continues till CL becomes zero
B:
DIV BX ; Dividing AX by BX
; It gives the average and stores it in AX
Input
Output
Reg H L
A 3b 67
B 00 03
C 00 00
D 00 01
Segments
SS 0000
DS 0000
ES 0000
Pointers
SP 0000
BP 0000
SI 0006
DI 0000
Flags:
OF DF IF TF SF ZF AF PF CF
0 0 0 0 0 1 0 1 0
Memory
Start Address
00000
SET
18 ab 24 03 fa 03 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00