COA – Lab Assignment 4

Name: Bhavin Patil

Roll No- 78

GR No : 12120056

Div : CS- D

Batch : B3

Problem Statement : Write 8086 ALP to count positive number and negative number from the array of signed number stored in memory.

Instructions –

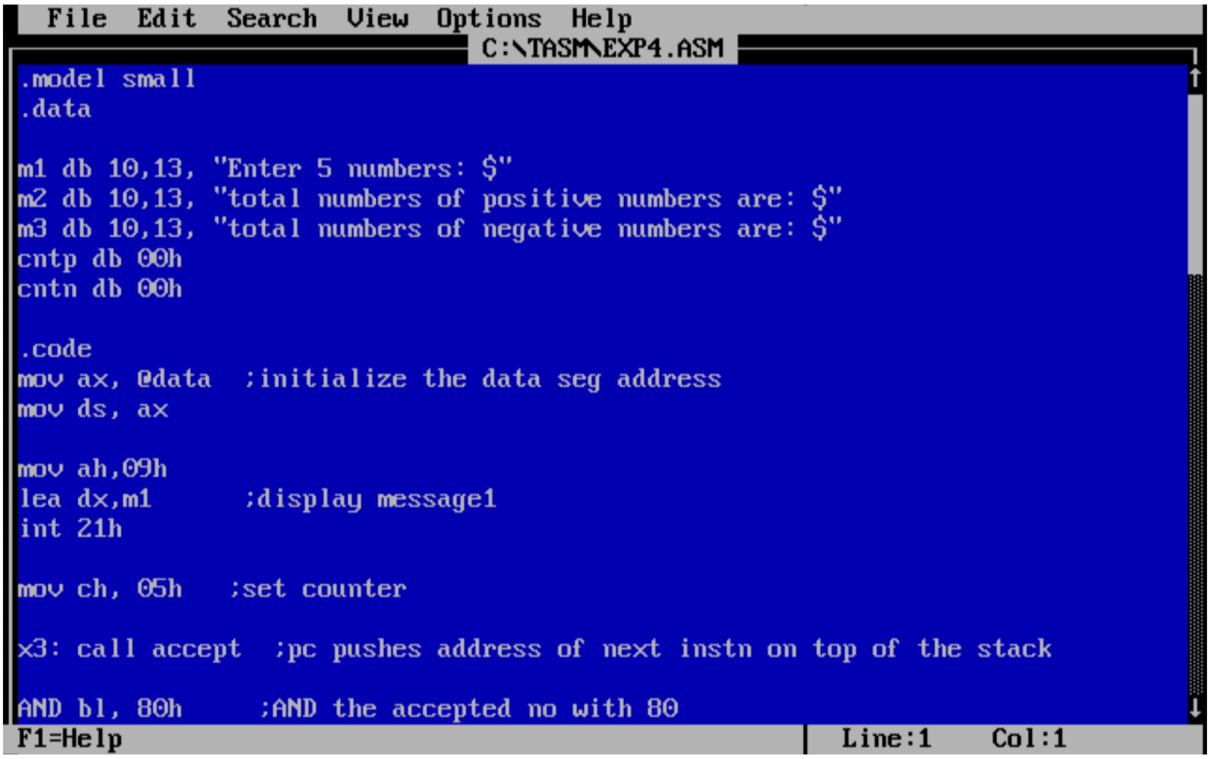
- MOV: This instruction is used to move data from one location to another.
  Syntax – mov destination, source
- LEA (Load Effective Address): It loads the specified register with the offset of a memory location.
- JNZ: (conditional jump) The program sequence is transferred to a particular level or a 16-bit address if Z=0 (or zero flag is 0)
- DIV: The DIV (unsigned divide) instruction performs 8-bit, 16-bit, and 32-bit division on unsigned integers.
- CALL: CALL instruction is used to call a subroutine. Subroutines are often used to perform tasks that need to be performed frequently.
- JLE: It performs a signed comparison jump after a cmp if the destination operand is less than or equal to the source operand.
- CMP: The CMP instruction compares two operands. It is generally used in conditional execution. This instruction basically subtracts one operand from the other for comparing whether the operands are equal or not.
- SHL: It is a bitwise rotation, also known as a circular shift, is a bitwise operation that shifts all bits of its operand.

Commands –

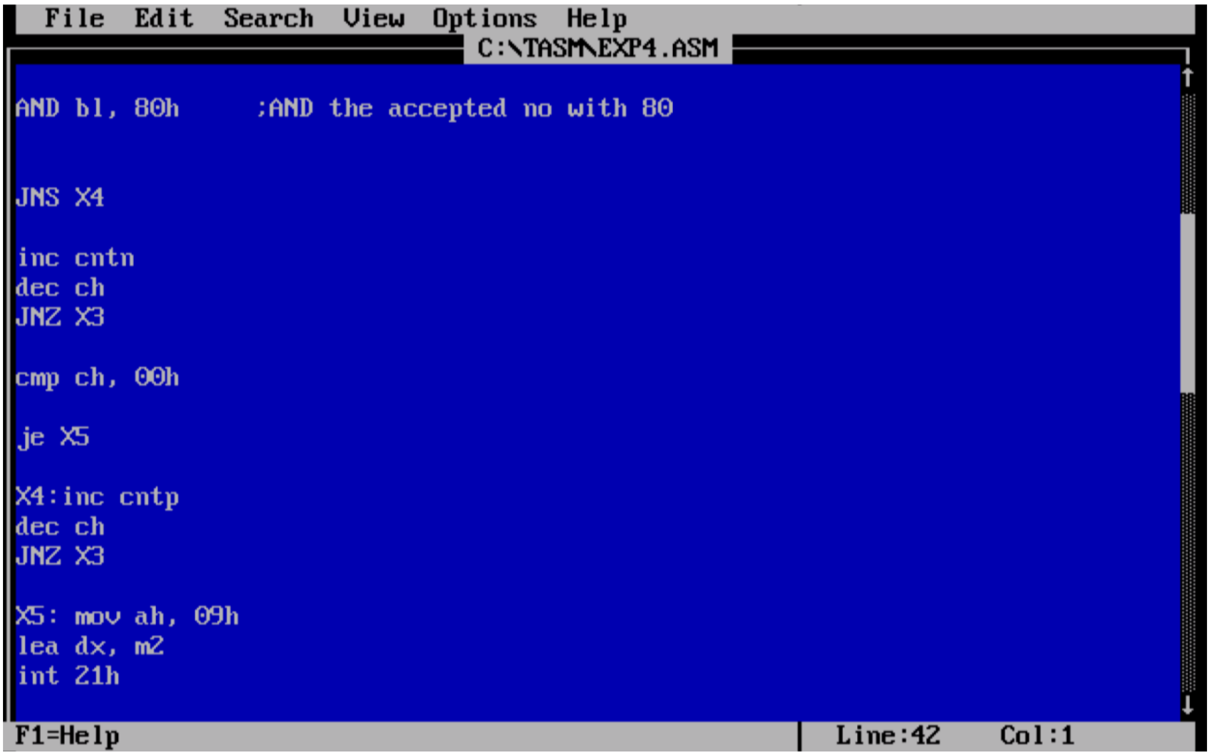
1. **01h :** It is used to read character from standard input, with echo, result is stored in AL.
2. **02h :** It is used to display single character
3. **09h**: Displays the string until “$” is reached.
4. **Int 21h**: Interrupt used to exit the program.
5. **.data**: This Command is used only when we want to store in Data Segment, basically, it is the memory access of the Data Segment. Whatever we want to print must be written here. Also, the variables are declared here.
6. **10, 13**: They work as Escape Sequence Character
7. **$**: It states the end of a Statement
8. **Db (Define Byte)**: It acts as an Assembler Directive
9. **. code**: Full Logical Program is written here
10. **Tasm –** Used for Compilation
11. **tlink**– Perform linking operation

Screenshots of Source Code and Output:

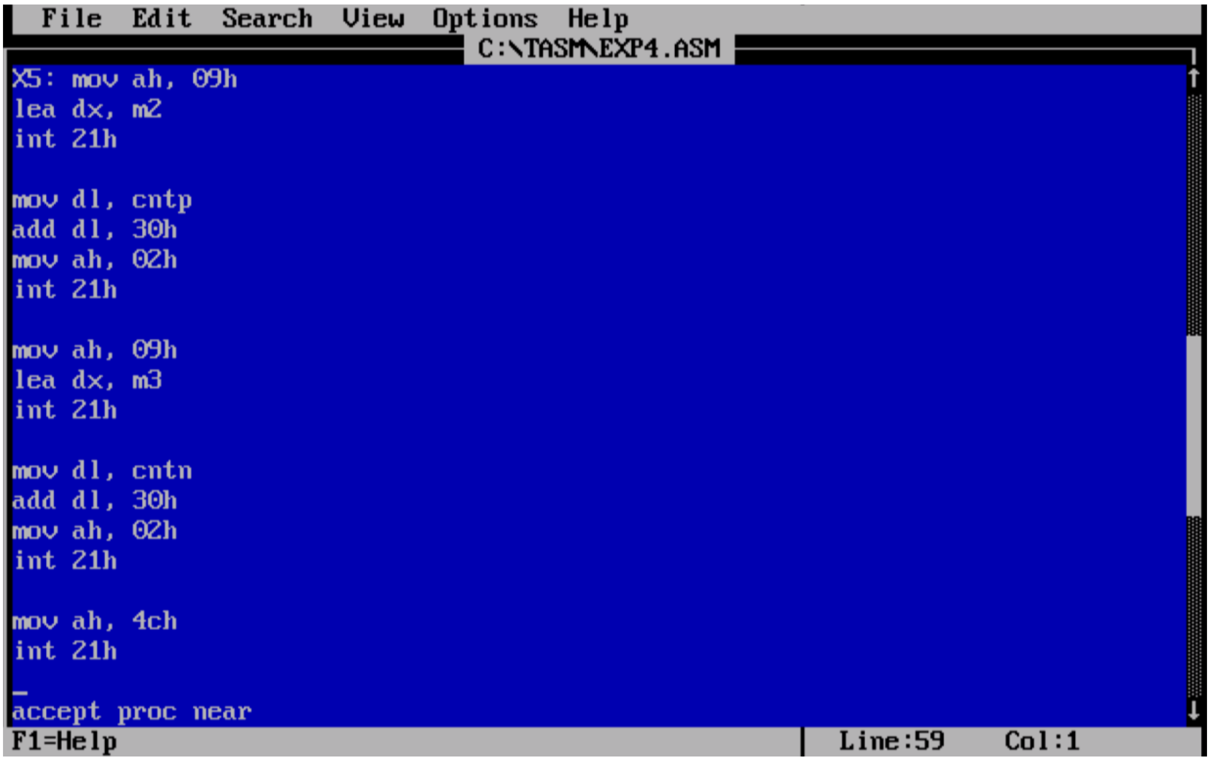
Source Code –



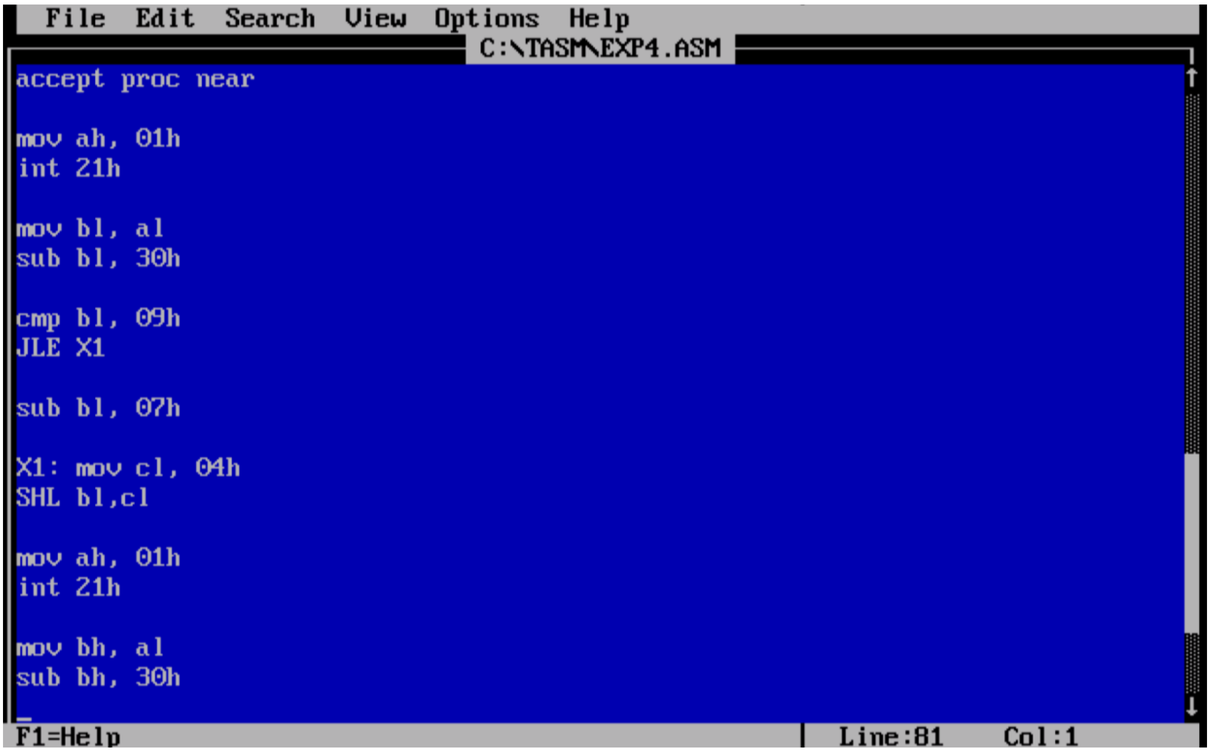
```
.model small
.data

m1 db 10,13, "Enter 5 numbers: $"
m2 db 10,13, "total numbers of positive numbers are: $"
m3 db 10,13, "total numbers of negative numbers are: $"
cntp db 00h
cntn db 00h

.code
mov ax, @data  ;initialize the data seg address
mov ds, ax

mov ah,09h
lea dx,m1       ;display message1
int 21h

mov ch, 05h   ;set counter

x3: call accept  ;pc pushes address of next instn on top of the stack

AND bl, 80h      ;AND the accepted no with 80
```



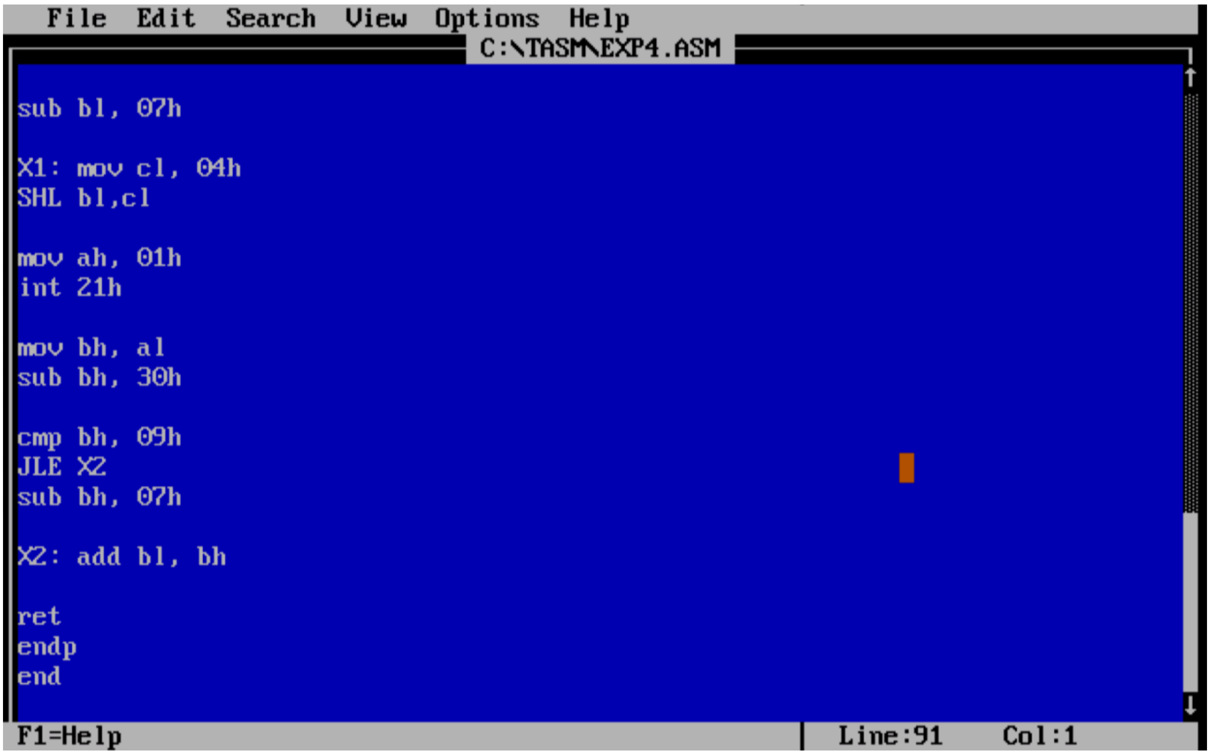
```
AND bl, 80h      ;AND the accepted no with 80


JNS X4

inc cntn
dec ch
JNZ X3

cmp ch, 00h

je X5

X4:inc cntp
dec ch
JNZ X3

X5: mov ah, 09h
lea dx, m2
int 21h
```



```
X5: mov ah, 09h
lea dx, m2
int 21h

mov dl, cntp
add dl, 30h
mov ah, 02h
int 21h

mov ah, 09h
lea dx, m3
int 21h

mov dl, cntn
add dl, 30h
mov ah, 02h
int 21h

mov ah, 4ch
int 21h

accept proc near
```

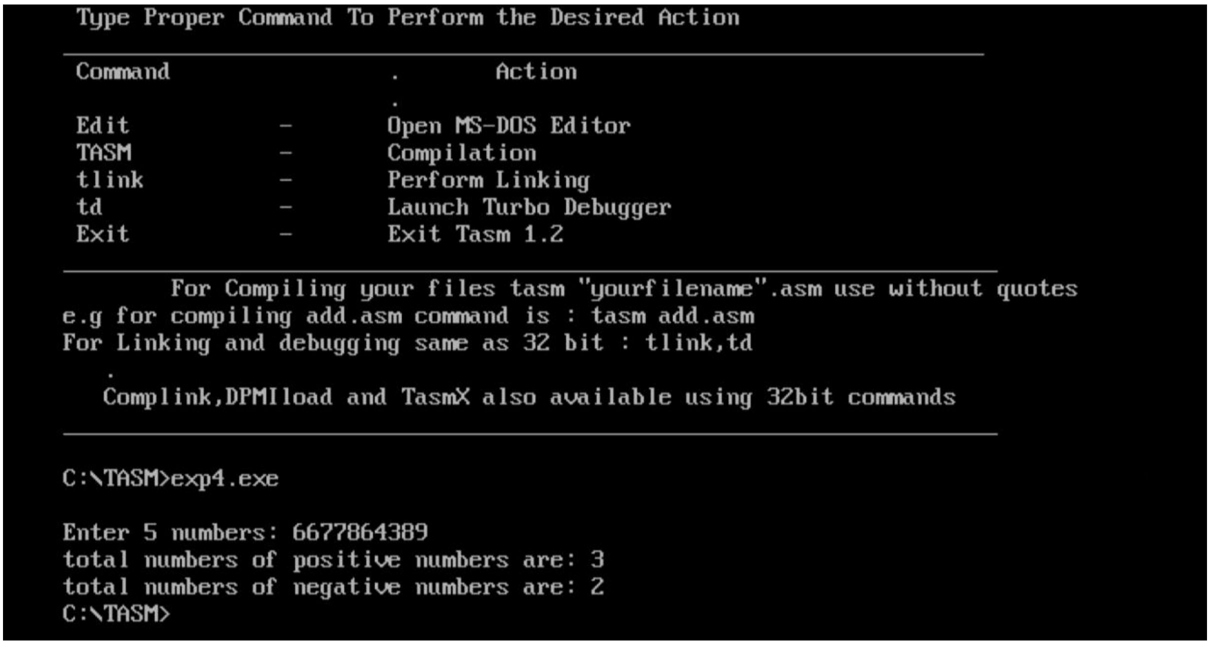


```
accept proc near

mov ah, 01h
int 21h

mov bl, al
sub bl, 30h

cmp bl, 09h
JLE X1

sub bl, 07h

X1: mov cl, 04h
SHL bl,cl

mov ah, 01h
int 21h

mov bh, al
sub bh, 30h
```



```

sub bl, 07h

X1: mov cl, 04h
SHL bl,cl

mov ah, 01h
int 21h

mov bh, al
sub bh, 30h

cmp bh, 09h
JLE X2
sub bh, 07h

X2: add bl, bh

ret
endp
end
```

Output:



```
     Type Proper Command To Perform the Desired Action

    _____________________________________________________________________
     Command                 .       Action
                             .
     Edit           -        Open MS-DOS Editor
     TASM           -        Compilation
     tlink          -        Perform Linking
     td             -        Launch Turbo Debugger
     Exit           -        Exit Tasm 1.2
    ______________________________________________________________________
            For Compiling your files tasm "yourfilename".asm use without quotes
    e.g for compiling add.asm command is : tasm add.asm
    For Linking and debugging same as 32 bit : tlink,td
       .
       Complink,DPMIload and TasmX also available using 32bit commands
    ______________________________________________________________________

    C:\TASM>exp4.exe

    Enter 5 numbers: 6677864389
    total numbers of positive numbers are: 3
    total numbers of negative numbers are: 2
    C:\TASM>
```