Experiment No. 4

Name – Mehatab Mahibub Sanadi
Roll No. – CO2056

```
section .data
   nline db 10, 10
   nline_len equ $-nline

   arr32 dd -11111111H, -22222222H, 33333333H, -44444444H, 55555555H
   n equ 5

   pmsg db 10, 10, 'The no. of Positive elements in 32-bit array:'
   pmsg_len equ $-pmsg

   nmsg db 10, 10, 'The no. of Negative elements in 32-bit array:'
   nmsg_len equ $-nmsg

section .bss
   p_count resb 1
   n_count resb 1
   char_count resb 1

%macro print 2
   mov eax, 4
   mov ebx, 1
   mov ecx, %1
   mov edx, %2
   int 80h
%endmacro

%macro exit 0
   mov eax, 1
   xor ebx, ebx ; Clearing ebx as it's expected to contain the exit status
   int 80h
%endmacro

section .text
   global _start

_start:
   mov esi, arr32
   mov edi, n

   xor ebx, ebx ; Initialize positive counter
   xor ecx, ecx ; Initialize negative counter

next_num:
   mov eax, [esi]
```

```
    test eax, eax ; Test the sign of the number
    js negative ; If negative, jump to negative label
    jns positive ; If positive, jump to positive label

positive:
    inc ebx ; Increment positive counter
    jmp next ; Jump to next iteration

negative:
    inc ecx ; Increment negative counter
    jmp next ; Jump to next iteration

next:
    add esi, 4 ; Move to the next element in the array
    dec edi ; Decrement loop counter
    jnz next_num ; If loop counter is not zero, repeat the loop

    mov [p_count], ebx ; Store the count of positive elements
    mov [n_count], ecx ; Store the count of negative elements

    print pmsg, pmsg_len
    mov eax, [p_count]
    call disp

    print nmsg, nmsg_len
    mov eax, [n_count]
    call disp

    print nline, nline_len
    exit ; Exit the program

disp:
    mov edi, char_count
    add al, 30h ; Convert the count to ASCII character
    mov [edi], al
    print char_count, 1 ; Print the character
    ret
```

output –

```
The no. of Positive elements in 32-bit array:2

The no. of Negative elements in 32-bit array:3
```