EXPERIMENT No.5

Name: Sandya Kashid

Roll No.: 12131

Subject : Microprocessor

section .data

msg1 db "Count of Positive numbers:", 10 ; 10 is line feed

len1 equ $ - msg1

msg2 db "Count of Negative numbers:", 10

len2 equ $ - msg2

array db 10, 12, -21, -12, -19, -34, 41 ; Example array with both positive and negative numbers

array\_len equ 7 ; Array length

%macro print 2

mov rax, 1 ; syscall number for write

mov rdi, 1 ; file descriptor (1 is stdout)

mov rsi, %1 ; address of string to print

mov rdx, %2 ; length of the string

syscall

%endmacro

section .bss

count resb 1

pcount resb 1

ncount resb 1

totalcount resb 2

section .text

global \_start

\_start:

; Initialize counts and pointer

mov byte [count], array\_len ; Total number of elements in the array

mov byte [pcount], 0 ; Initialize positive count to 0

mov byte [ncount], 0 ; Initialize negative count to 0

mov rsi, array ; Address of the array

Up:

; Check if there are more elements to process

mov al, [count]

cmp al, 0

je Done

; Load current element

mov al, [rsi]

js neg ; Jump to neg if value is negative (less than 0)

; Increment positive count

inc byte [pcount]

jmp Down

neg:

; Increment negative count

inc byte [ncount]

Down:

; Move to the next element in the array

inc rsi

dec byte [count]

jnz Up ; Repeat if there are more elements

Done:

; Print the positive count

print msg1, len1

mov al, [pcount]

call disp

; Print the negative count

print msg2, len2

mov al, [ncount]

call disp

; Exit the program

mov rax, 60 ; syscall number for exit

mov rdi, 0 ; exit code 0

syscall

disp:

; This function converts the value in AL (byte) to ASCII and prints it

; Move the count to totalcount for display

movzx rbx, al ; Move AL to RBX (we use RBX to process)

mov byte [totalcount], 0 ; Clear totalcount

mov byte [totalcount + 1], 0

; Convert to ASCII and print

add rbx, '0' ; Convert to ASCII

mov byte [totalcount], bl

print totalcount, 2 ; Print the result

ret

OUTPUT:

Count of Positive numbers: 7

Count of Negative numbers: 0