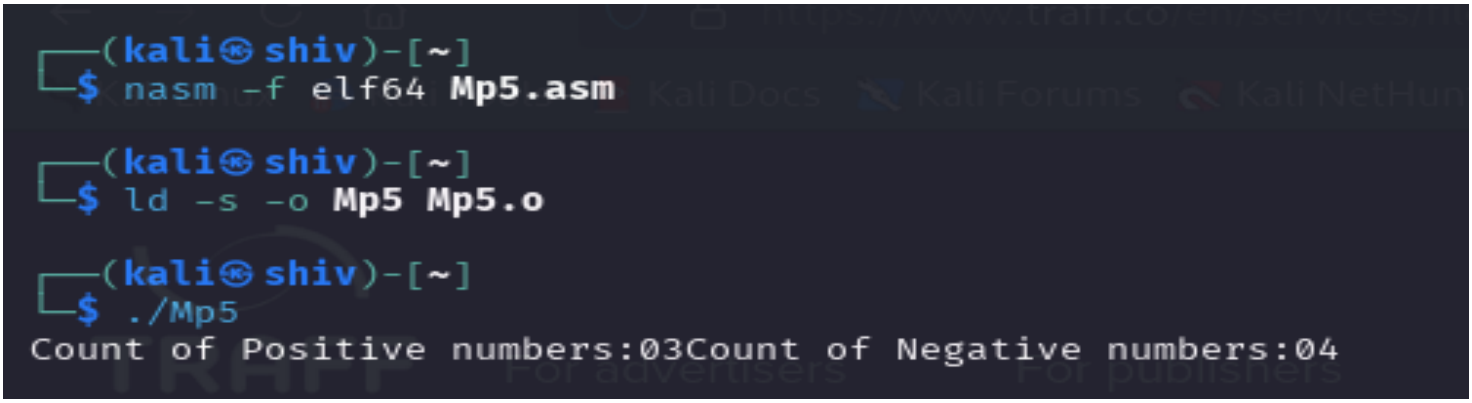
```
section .data
    msg1 db "Count of Positive numbers:"
    len1 equ $ - msg1          ; Length of msg1 string
    msg2 db "Count of Negative numbers:"
    len2 equ $ - msg2          ; Length of msg2 string
    array db 10, 12, -21, -12, -19, -34, 41 ; Array of numbers

%macro print 2
    mov rax, 01              ; System call for writing
    mov rdi, 01              ; File descriptor 1 (stdout)
    mov rsi, %1              ; Address of message to print
    mov rdx, %2              ; Length of the message
    syscall                  ; Make the system call
%endmacro

section .bss
    count resb 2             ; Counter for the loop
    pcount resb 2            ; Positive count
    ncount resb 2            ; Negative count
    totalcount resb 2        ; For storing total character output

section .text
    global _start            ; Entry point for the program

_start:
    ; Initialize counters
    mov byte [count], 07     ; Total count (7 elements in the array)
    mov byte [pcount], 00    ; Positive number counter
    mov byte [ncount], 00    ; Negative number counter
    mov rsi, array           ; Point to the start of the array

Up:
    ; Initialize AL to 00 for comparison
    mov al, 00
    add al, [rsi]            ; Add the value at the current array position
    js neg                   ; If the value is negative, jump to 'neg' label
    inc byte [pcount]        ; Increment positive count
    jmp Down                 ; Skip the negative handling

neg:
    inc byte [ncount]        ; Increment negative count

Down:
    add rsi, 01              ; Move to the next element in the array
    dec byte [count]         ; Decrement the total count
    jnz Up                   ; If count is not zero, continue the loop

    ; Print positive count
    print msg1, len1
```

```
    mov bl, [pcount]          ; Load positive count to BL
    call disp                 ; Call the display function

    ; Print newline after the positive count
    mov rsi, newline          ; Address of the newline character
    mov rdx, 1                ; Length of the newline character
    syscall                   ; Print newline

    ; Print negative count
    print msg2, len2
    mov bl, [ncount]          ; Load negative count to BL
    call disp                 ; Call the display function

    ; Exit the program
    mov rax, G0               ; Syscall number for exit
    mov rdi, 00               ; Exit status 0
    syscall                   ; Make the syscall to exit the program

; Function to display the count in hexadecimal format
disp:
    mov byte [count], 02      ; Number of iterations (2 digits for hex)
loop:
    rol bl, 04                ; Rotate left by 4 bits
    mov al, bl                ; Move the lower nibble to AL
    AND al, 0FH               ; Mask the upper nibble (keep only lower 4 bits)
    cmp al, 09                ; Compare with 9
    jbe l1                    ; If AL  <= 9, jump to 'l1'
    add al, 07h               ; Adjust for ASCII representation of A-F
l1:
    add al, 30h               ; Convert to ASCII (add '0')
    mov [totalcount], al      ; Store the character in totalcount
    print totalcount, 02      ; Print the character
    dec byte [count]          ; Decrement the loop counter
    jnz loop                  ; Continue if the loop counter is not zero

    ret                       ; Return from the function

section .data
    newline db 0<?>0A         ; Newline character (0<?>0A)
```

Output :


(kali㉿shiv)-[~]
$ nasm -f elf64 Mp5.asm
(kali㉿shiv)-[~]
$ ld -s -o Mp5 Mp5.o
(kali㉿shiv)-[~]
$ ./Mp5
Count of Positive numbers:03Count of Negative numbers:04