SUBJECT: MICROPROCESSOR LAB (MPL) **NAME:PRIYANKA SALUNKE**	
CLASS: SE COMP A	**ROLL NO.: F19111151**
SEMESTER: SEM-II	**YEAR: 2020-21**
DATE OF PERFORMANCE:	**DATE OF SUBMISSION:**
EXAMINED:	

Assignment No-05

Title:-Count no. of positive and negative numbers

Assignment Name: - Write an ALP to count no. of positive and negative numbers from the array.

Objective-

- To understand the assembly language program
- To understand 64 bit interrupt.

Outcome-

- Students will be able to write code for how to count positive and negative number from array
- Students will be able to understand different assembly language instruction.

Prerequisite -

System call of Unix for Assembly language Program.

Hardware Requirement-

Desktop PC

Software Requirement-

Ubuntu 14.04,

Assembler: NASM version 2.10.07
Linker: ld

Introduction:-

Theory:

Algorithm:

1. Start
2. Initialise section .data
3. Define variable for array,pcount,ncount
4. Count Positive and negative number using JS command.
5. Display counts
6. Terminate program using system call
6. Stop

Conclusion:- Hence we implemented an ALP to count positive and negative number from array and display count.

Questions:-

Q.1.Explain BT,JS,loop instruction with Example?
Q.2 Explain Paging in 80386?
Q.3 Draw control registers of 80386

Program

```
%macro print 2
mov rax,1
mov rdi,1
mov rsi,%1
mov rdx,%2
syscall
%endmacro

section .data

m0 db "Counting +ve and -ve elements of an array.",10
l0 equ $-m0

m1 db "Positive nos. are : "
l1 equ $-m1

m2 db "Negative nos. are :  "
l2 equ $-m2

array db -1h,2h,-3h,4h,-5h,-6h,-7h
pcount db 0
ncount db 0
newline db 0xa

section .bss

dispbuff resb 2


section .text
global _start
_start:
        print m0,l0

        mov rsi,array
        mov rcx,07

again:
        mov al, [rsi]
        cmp al,00h
        js next1
```

```
                inc byte[pcount]
        jmp pskip

next1:  inc byte[ncount]

pskip:  add rsi,1

                loop again


print m1,l1

mov bl,[pcount]
call disp_result
print newline,1

print m2,l2
mov bl,[ncount]
call disp_result
print newline,1

mov rax,60                                      ;terminate program
xor rdi,rdi
Syscall


;procedure to convert hex number to its equivalent ASCII
disp_result:
        mov rdi,dispbuff
        mov rcx,02

        dispup1:
          rol bl,4
          mov dl,bl
          and dl,0fh
          add dl,30h
          cmp dl,39h
          jbe dispskip1
          add dl,07h

        dispskip1:
           mov [rdi],dl
           inc rdi
           loop dispup1
           print dispbuff,2
ret
```

Output

Code Input Output Run Download Update

```
Counting +ve and -ve elements of an array.
Positive nos. are : 02
Negative nos. are :  05

[Program exited with exit code 0]
```

Questions and Answers:

PRACTICAL ASSIGNMENT

1. Explain BT, JS, loop instructions with examples

Ans BT (Bit test):
This instruction tests the status of the specified bit in the instruction. The status of that bit is copied to carry flag.
eg:
BT, EAX 05 This instruction copies the bit 5 of the EAX register to carry flag

JS (Jump if sign or jump if negative):
In this instruction sign flag (SF) is set
eg:
Finding even/odd number is possible using JS instruction. If it is set, the number is negative else the number is positive

Loop:
This instruction is used to repeat a series of instructions some number of times.
The number of times the instruction sequence is to be repeated is located into CX (ECX). Each time loop executes, CX is decremented by 1
If CX ≠ 0, execution will jump to destination specied label
If CX = 0, execution will go to the next instruction after loop

eg.:

```
MOV SI , offset ARRAY
MOV AL , 00H
MOV CX , 10H , Counter loaded
AI : ADD AL [SI];
INC SI
LOOP AI
```

2. Explain paging in 80386?

Ans Paging is one of the memory management techniques used for virtual memory multitasking operating system.
The segmentation scheme may divide the physical memory into a variable size segments of the program, but the pages do not have any logical relations with the program and paging divides the memory into a fixed size pages.
The pages are just have just fixed size position of the program module or data.
The advantage of paging scheme is that the complete segment of a task need not be in the physical memory at any time.
Only a few pages of the segment, which are required currently for the execution need to be available in the physical memory. Thus the memory requirement of the test is substantially reduced relinquishing the available memory for other tasks.
Whenever the other pages of task are required for

execution, they may be fetched from the secondary storage.
The paging is a mechanism provider an effective technique to manage the physical memory for multitasking systems.

3 Draw the control register of 80386?