MICROPROCESSOR LABORATORY

Assignment No. 5

NAME :- OJUS PRAVIN JAISWAL
ROLL NO. :- SACO19108
DIVISION :- A

Assignment No. 5

Program :

```
%macro print 2
mov Rax,1
mov  Rdi,1
mov Rsi,%1
mov Rdx,%2
syscall
%endmacro

%macro exit 0
mov rax,60
mov rdi,0
syscall
%endmacro

section .data
arr dq 00h,-10h,20h,-30h,40h,-50h,60h,-70h
n equ 8
pmsg db 10,13,"The count of positive elements in the array is : ",10,13
pmsglen equ $-pmsg
nmsg db 10,13,"The count of negative element in the array is : ",10,13
nmsglen equ $-nmsg
nwline db 10,13

section .bss
```

```
pcnt resq 1
ncnt resq 1
char_answer resb 16

section .text
global _start
_start:
mov rsi,arr
mov rdi,n
mov rbx,0
mov rcx,0

up:mov rax,[rsi]
    rol rax,1
    jc negative

positive:inc rbx
            jmp next

negative:inc rcx

next:add rsi,8
       dec rdi
       jnz up

       mov [pcnt],rbx
       mov [ncnt],rcx
```

```
        print pmsg,pmsglen
        mov rax,[pcnt]
        call display

        print nmsg,nmsglen
        mov rax,[ncnt]
        call display

        print nwline,1
        exit

display:
 mov rsi,char_answer+15
 mov rcx,16

 cnt:mov rdx,0
       mov rbx,16h
       div rbx
       cmp dl,09h
       jbe add30
       add dl,07h
 add30:add dl,30h
            mov [rsi],dl
            dec rsi
            dec rcx
            jnz cnt
 print char_answer,16
ret
```

Code Input Output Run Save

```
%macro print 2
mov Rax,1
mov Rdi,1
mov Rsi,%1
mov Rdx,%2
syscall
%endmacro

%macro exit 0
mov rax,60
mov rdi,0
syscall
%endmacro

section .data
arr dq 00h,-10h,20h,-30h,40h,-50h,60h,-70h
n equ 8
pmsg db 10,13,"The count of positive elements in the array is : ",10,13
pmsglen equ $-pmsg
nmsg db 10,13,"The count of negative element in the array is : ",10,13
```

Code Input Output Run Save

```
nmsglen equ $-nmsg
nwline db 10,13

section .bss
pcnt resq 1
ncnt resq 1
char_answer resb 16

section .text
global _start
_start:

mov rsi,arr
mov rdi,n
mov rbx,0
mov rcx,0

up:mov rax,[rsi]
   rol rax,1
   jc negative
```

Code Input Output Run Save

```

positive:inc rbx
        jmp next

negative:inc rcx

next:add rsi,8
     dec rdi
     jnz up

mov [pcnt],rbx
mov [ncnt],rcx

print pmsg,pmsglen
mov rax,[pcnt]
call display

print nmsg,nmsglen
mov rax,[ncnt]
call display
```

Code Input Output Run Save

```
61
62  print nwline,1
63  exit
64
65  display:
66  mov rsi,char_answer+15
67  mov rcx,16
68  cnt:mov rdx,0
69      mov rbx,16h
70      div rbx
71      cmp dl,09h
72      jbe add30
73      add dl,07h
74  add30:add dl,30h
75        mov [rsi],dl
76        dec rsi
77        dec rcx
78        jnz cnt
79  print char_answer,16
80  ret
```

Output :