Assignment 1:

Name : Prajwal Khobragade

SYCOB138

```
;-----------------------------------------------
section .data
arr_64 dq -1111h,2222h,-3333h, 4444h, -5555h   ; initializing 64 bit array
arr_64len equ 5
pmsg db 10,10,"No of positive no : ",10   ; message display for no of positive no
pmsg_len equ $-pmsg
nmsg db 10,10,"no of negative no: ",10    ;message display for no negative no
nmsg_len  equ $-nmsg
%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 .bss
pcount resd 1
ncount resb 1
char_ans          resb      02

;-----------------------------------------------
section .text
global _start
```

```
_start:
mov rsi,arr_64     ;moving first memory location of array to source index register
mov rbx,0          ; initial no of positive no
mov rdx,0          ;initial no of negative no
mov rcx,arr_64len  ;initializing counter equal to no of elements in array


next_num:
mov rax,[rsi]      ;moving the value at memory location present in source index to accumulator
shl rax,1          ;Shifting the content of accumulator by 1
jc negative        ; jump to negative label if carry is generated


positive:
inc rbx            ;incrementing the count for positive elements
jmp next           ;jump to next lebel


negative:
inc rdx            ;incrementing the count for negative elements


next:
add rsi,8
dec rcx            ;decrementing the counter
jne next_num       ;jump to next_num label
mov [pcount],rbx   ;storing value to pcount
mov [ncount],rdx   ;storing value to ncount
print pmsg,pmsg_len
mov rax,[pcount]   ;load value to pcount to the accumulator register
call ddisp64_proc  ;Calling display procedure for pcount
```

```
print nmsg,nmsg_len
mov rax,[ncount]    ;load value to ncount to the accumulator register
call ddisp64_proc   ;Calling display procedure for pcount
Exit

;------------------------------------------------------
ddisp64_proc:
        mov         rbx,16              ; divisor=16 for hex
        mov         rcx,2               ; number of digits
        mov         rsi,char_ans+1      ; load last byte address of char_ans buffer in rsi

cnt:    mov         rdx,0               ; make rdx=0 (as in div instruction rdx:rax/rbx)
        div         rbx

        cmp         dl, 09h             ; check for remainder in rdx
        jbe     add30
        add     dl, 07h
add30:
        add         dl,30h              ; calculate ASCII code
        mov         [rsi],dl            ; store it in buffer
        dec         rsi                         ; point to one byte back

        dec         rcx                         ; decrement count
        jnz         cnt                         ; if not zero repeat

        print   char_ans,2              ; display result on screen
ret
;------------------------------------------------------------
```

Output:

```
No of positive no :
02

no of negetive no:
03
[Execution complete with exit code 0]
```