; Assignment Name :- Write X86/64 ALP to count number of positive and negative numbers from the array.

section .data

nline db 10,10

nline\_len equ $-nline

arr32 dd -11111111H, -22222222H, 33333333H, -44444444H, 55555555H

n equ 5

pmsg db 10,10,'The no. of Positive elements in 32-bit array : '

pmsg\_len equ $-pmsg

nmsg db 10,10,'The no. of Negative elements in 32-bit array : '

nmsg\_len equ $-nmsg

section .bss

p\_count resb 1

n\_count resb 1

char\_count resb 1 ; for 32-bit nos.

%macro print 2

mov eax, 4

mov ebx, 1

mov ecx, %1

mov edx, %2

int 80h

%endmacro

%macro exit 0

mov eax, 1

mov ebx, 0

int 80h

%endmacro

section .text

global \_start

\_start:

mov esi, arr32

mov edi, n

mov ebx,0; ; counter for +ve nos.

mov ecx,0; ; counter for -ve nos.

next\_num:

mov eax,[esi] ; take no. in RAX

RCL eax,1; rotate left 1 bit to check for sign bit

jc negative

positive:

inc ebx ; no carry, so no. is +ve

jmp next

negative:

inc ecx ; carry, so no. is -ve

next:

add esi,4 ; 32 bit nos i.e. 4 bytes

dec edi

jnz next\_num

mov [p\_count], ebx ; store positive count

mov [n\_count], ecx ; store negative count

print pmsg, pmsg\_len

mov eax,[p\_count] ; load value of p\_count

call disp ; display p\_count

print nmsg, nmsg\_len

mov eax,[n\_count] ; load value of n\_count

call disp ; display n\_count

print nline, nline\_len

exit

disp:

mov edi,char\_count ; load byte address of char\_count buffer

;mov ecx,1 ; number of digits

add al,30h ; calculate ASCII code

mov [edi],al ; store it in buffer

print char\_count,1 ; display result on screen

ret