# Name: Shrinivas Hatyalikar

# Div : CS-B

# Roll No: 24

**Aim :-** Write 64 bit ALP programs for counting the number of negative and positive numbers in hexadecimal array and display it on output console

**Appartus:**

* Core 2 duo/i3/i5/i7 - 64bit processor
* OS – ubuntu 32bit/64bit OS
* Assembler used –nasm (the netwide assembler)
* Editor Used – gedit

**Theory :**

**Procedures**

**Display64\_proc**

Display 64 procedure used to convert hexadecimal numbers into there ascii and display

Operation:

First divide number by 16 for converting the value into hexadecimal and check whether number is greater than 9 or not if greater than 9 then jump to add30 procedure to add 30 to get the ascii number and print it in console

**Algorithm :- counting the negative and positive number in array**

1. Start the program.

2. In data section initialize the array globally and store the size of an array in variable n.

3. Declare the variable with store the count for positive and negative numbers.

4. Write the macros for print, read and exit operation.

5. In text section, move the contents of array to rsi register.

6. Store the value of n in rcx register.

7. Initialize positive number count and negative number count to zero.

8. Perform msb test of array element and find if msb is 1 or 0 of the number.

9. If it is 1 then it is negative number, store the negative count in rdx and increment the negative number count.

10. If it is 0 then it is positive number and store the positive number count in rbx and increment the negative number count.

11. Decrement the array element count from ecx until 0, jump to step 8.

12. Write a display procedure to display the count of positive and negatiove numbers.

13. Display the positive number message and positive number count.

14. Display the negative number message and negative number count.

15. Exit.

**Program:**

section .data

nline db 10,10

nline\_len equ $-nline

arr64 dq -11111111H, 22222222H, -33333333H, 44444444H, 55555555H

n equ 5

pmsg db 10,10,"The no of positive element in an array: "

pmsg\_len equ $-pmsg

nmsg db 10,10,"The no of negative element in an array: "

nmsg\_len equ $-nmsg

Section .bss

p\_count resq 1

n\_count resq 1

char\_ans resb 2

d1 resb 2

%macro Print 2

mov rax, 1

mov rdi, 1

mov rsi, %1

mov rdx, %2

syscall

%endmacro

%macro Read 2

mov rax, 0

mov rdi, 0

mov rsi, %1

mov rdx, %2

syscall

%endmacro

%macro Exit 0

mov rax, 60

mov rdi, 0

syscall

%endmacro

section .text

Global \_start

\_start:

mov rsi,arr64

mov rcx, n

mov rbx, 0

mov rdx, 0

next\_num:

mov rax, [rsi]

rol rax,1

jc negative

positive:

inc rbx

jmp next

negative:

inc rdx

next:

add rsi, 8

dec rcx

jnz next\_num

mov [p\_count],rbx

mov [n\_count],rdx

Print pmsg, pmsg\_len

mov rax,[p\_count]

call display64\_proc

Print nmsg, nmsg\_len

mov rax,[n\_count]

call display64\_proc

Print nline, nline\_len

Exit

display64\_proc:

mov rbx,16

mov rcx,2

mov rsi,char\_ans+1

cnt: mov rdx,0

div rbx

cmp dl, 09H

jbe add30

add dl,07H

add30:

add dl,30h

mov [rsi],dl

dec rsi

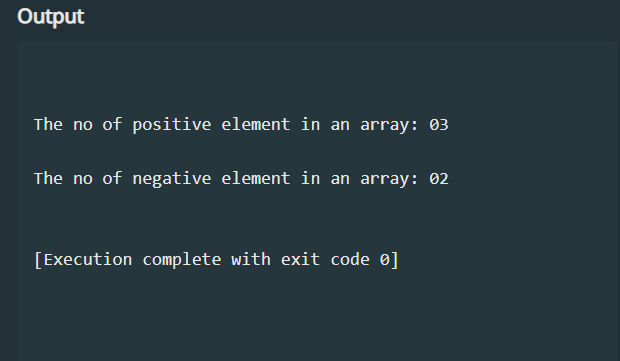
dec rcx

jnz cnt

Print char\_ans, 2

ret

**Output:**

****

**Conclusion:**

* We have calculated the number of positive and negative elements in hexadecimal array.
* The messages are printed using the display64 procedure written.
* The p\_count maintains the count of positive numbers and n\_count maintains the count of negative numbers.
* The count is displayed on the console.