Title : Write 64-bit ALP to accept array of numbers from user and perform addition of array elements and display the result on screen.

Code:

%macro scall 4

mov rax,%1

mov rdi,%2

mov rsi,%3

mov rdx,%4

syscall

%endmacro

section .data

m1 db "Enter no. of array elements:",10d,13d

l1 equ $-m1

m2 db "Enter number:",10d,13d

l2 equ $-m2

m3 db "Array contents are:",10d,13d

l3 equ $-m3

m4 db "Addition is:",10d,13d

l4 equ $-m4

m5 db " "

l5 equ $-m5

section .bss

num resb 20

cnt resb 20

cnt1 resb 20

cnt2 resb 20

array resb 200

char\_ans resb 16

section .text

global \_start

\_start:

scall 1,1,m1,l1

scall 0,0,num,3

call accept\_proc

mov [cnt],bx

mov [cnt1],bx

mov [cnt2],bx

mov rbp,array

up1:

scall 1,1,m2,l2

scall 0,0,num,3

call accept\_proc

mov [rbp],bx

add rbp,2

dec word[cnt]

jnz up1

scall 1,1,m3,l3

mov rbx,array

mov ax,[cnt1]

mov [cnt],ax

up2:

mov ax,[rbx]

call display\_proc

add rbx,2

dec byte[cnt]

jnz up2

scall 1,1,m4,l4

back1:

mov ax,[cnt2]

mov [cnt],ax

mov ax,00h

mov bx,00h

mov rbx,array

back2:

add ax,[rbx]

add bx,2

dec byte[cnt]

jnz back2

call display\_proc

mov rax,60

mov rdi,0

syscall

accept\_proc:

mov rsi,num

mov rbx,0

mov rax,0

mov rcx,2

back:

rol rbx,04

mov al,[rsi]

cmp al,39h

jbe next

sub al,07h

next:

sub al,30h

add bx,ax

inc rsi

dec rcx

jnz back

ret

display\_proc:

mov rbp,char\_ans

mov rcx,2

up3:

rol al,04

mov dl,al

and dl,0Fh

cmp dl,09h

jbe next1

add dl,07h

next1:

add dl,30h

mov [rbp],dl

inc rbp

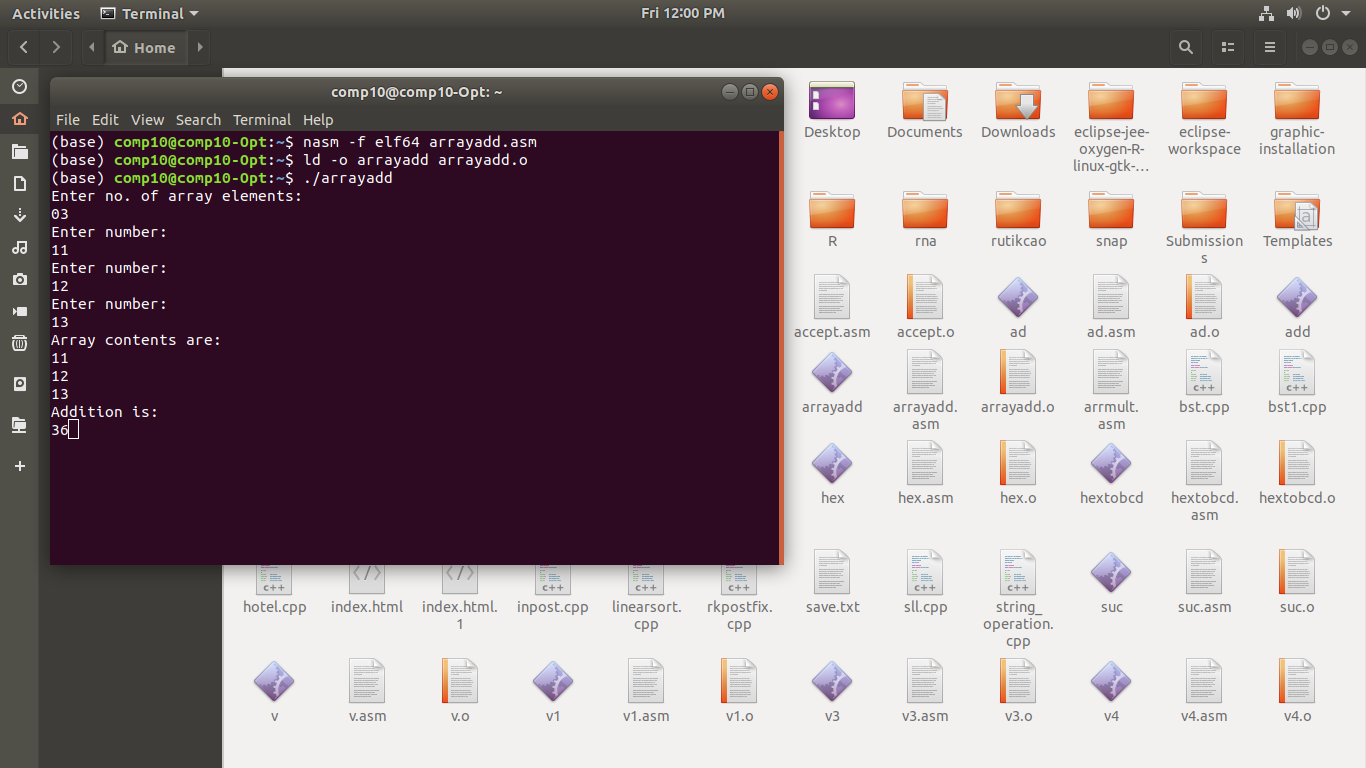
dec rcx

jnz up3

scall 1,1,char\_ans,3

scall 0,0,m5,l5

ret

Output: