**ML Assignment**

**Name:** Jagruti Ravindra Patil

**Roll No:** 2089

**Batch:** S4 Comp

**Assignment 10:**

**Code:**

section .data

msg1 db 10,10,'Enter First Number :'

msg1\_len equ $-msg1

msg2 db 10,10,'Enter Second Number :'

msg2\_len equ $-msg2

msg3 db 10,10,'Multiplication is :'

msg3\_len equ $-msg3

msg db 10,'\*\*\*MENU\*\*\*'

msg\_len equ $-msg

m1 db 10,'1. Addition Method'

m1\_len equ $-m1

m2 db 10,'2. Add and shift method'

m2\_len equ $-m2

m3 db 10,'3. Exit'

m3\_len equ $-m3

m4 db 10,'Enter choice :'

m4\_len equ $-m4

section .bss

choice resb 02

numascii resb 03

num1 resb 01

num2 resb 01

result resb 04

dispbuff resb 08

%macro dispmsg 2

mov eax, 4

mov ebx, 1

mov ecx, %1

mov edx, %2

int 80h

%endmacro

%macro accept 2

mov eax, 3

mov ebx, 0

mov ecx, %1

mov edx, %2

int 80h

%endmacro

section .text

global \_start

\_start:

menu:

dispmsg msg, msg\_len

dispmsg m1, m1\_len

dispmsg m2, m2\_len

dispmsg m3, m3\_len

dispmsg m4, m4\_len

accept choice, 02

cmp byte [choice], '1'

je SA\_method

cmp byte [choice], '2'

je addshift\_method

cmp byte [choice], '3'

je exit

exit:

mov eax, 1

mov ebx, 0

int 80h

SA\_method:

dispmsg msg1, msg1\_len

accept numascii, 3

call convert

mov [num1], bl

dispmsg msg2, msg2\_len

accept numascii, 3

call convert

xor rcx, rcx

xor rax, rax

mov al, [num1]

bk:

add rcx, rax

dec bl

jnz bk

mov [result], rcx

dispmsg msg3, msg3\_len

mov bx, [result]

call disp\_proc

jmp menu

addshift\_method:

dispmsg msg1, msg1\_len

accept numascii, 3

call convert

mov [num1], bl

dispmsg msg2, msg2\_len

accept numascii, 3

call convert

mov [num2], bl

dispmsg msg3, msg3\_len

xor rbx, rbx

xor rcx, rcx

xor rdx, rdx

xor rax, rax

mov dl, [num1]

mov bl, [num2]

mov cl, 08

z1:

shl ax, 1

rol bl, 1

jnc b1

add ax, dx

b1:

loop z1

mov bx, ax

call disp\_proc

jmp menu

convert:

mov ebx, 0

mov ecx, 2

mov esi, numascii

up1:

rol bl, 04

mov al, [esi]

cmp al, 39h

jbe skip1

sub al, 07h

skip1:

sub al, 30h

add bl, al

inc esi

loop up1

ret

disp\_proc:

mov ecx, 4

mov edi, dispbuff

dup1:

rol bx, 4

mov al, bl

and al, 0fh

cmp al, 09

jbe dskip

add al, 07h

dskip:

add al, 30h

mov [edi], al

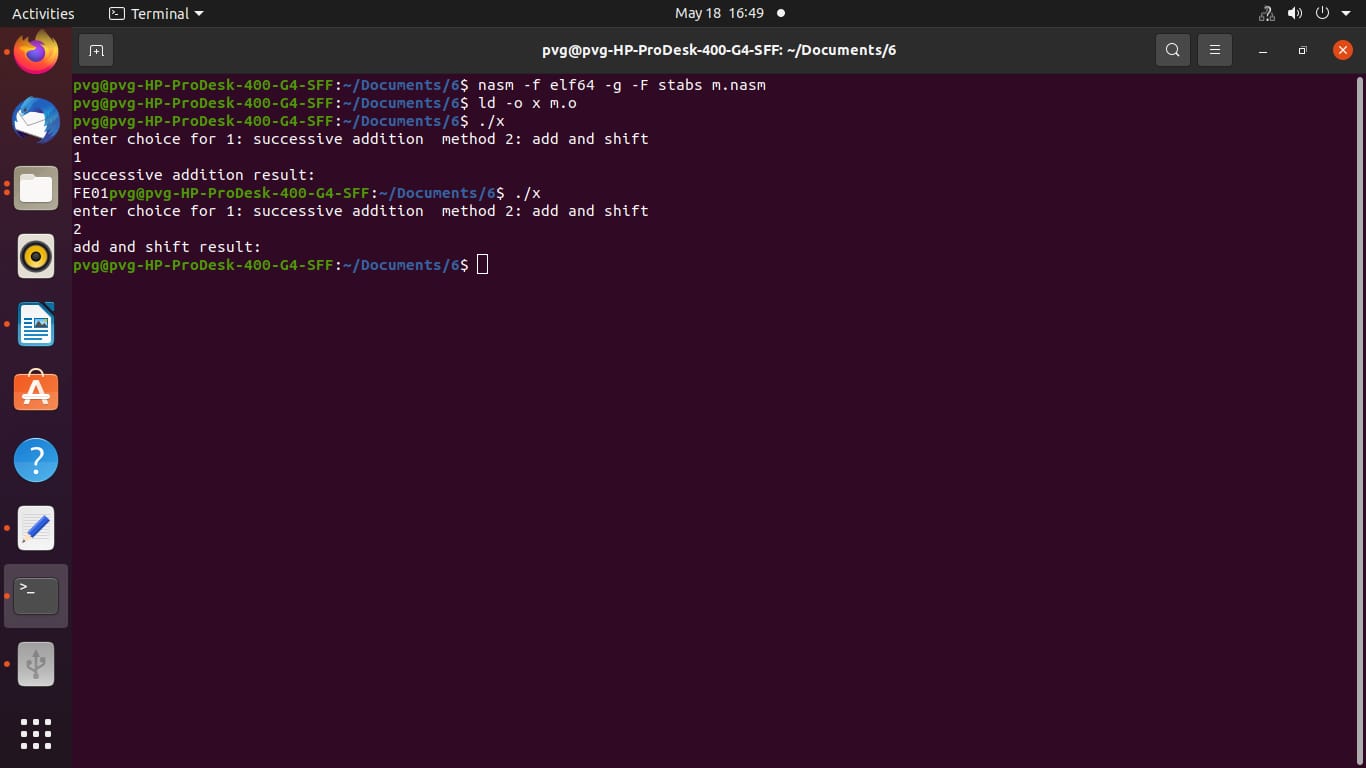
inc edi

loop dup1

dispmsg dispbuff, 4

ret

**Output:**

****