ML Assignment

Name: Jagruti Ravindra Patil

Roll No: 2089

Batch: S4 Comp

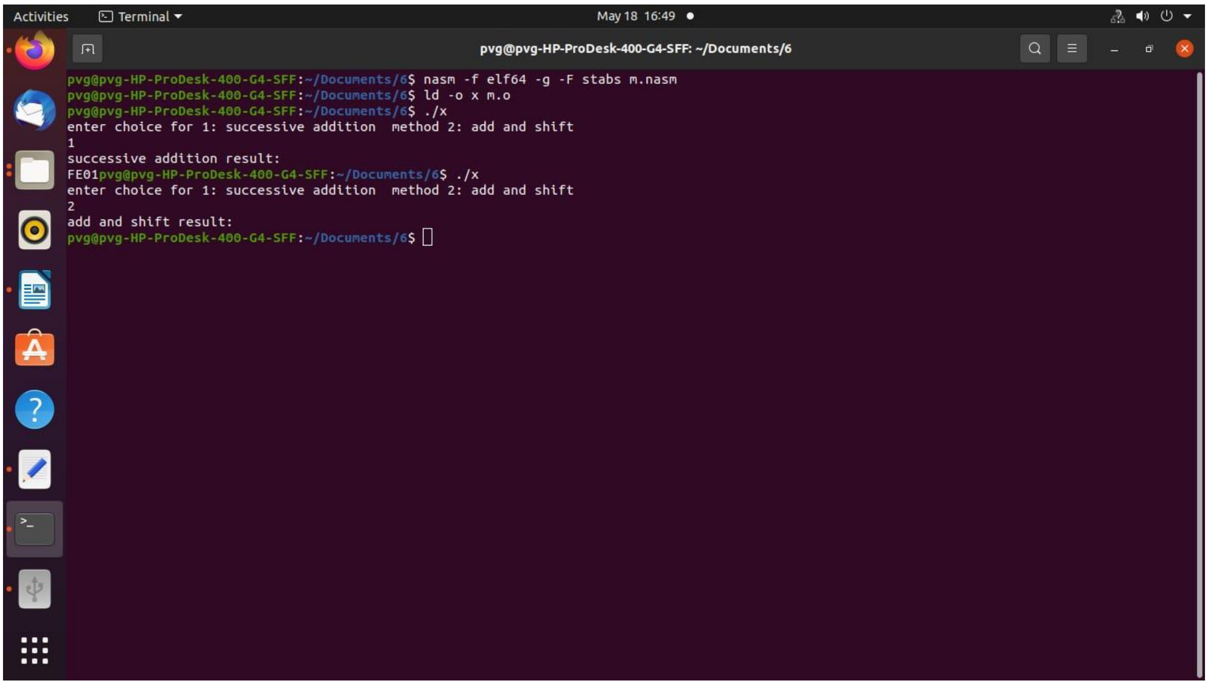
Assignment 10:

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:



```
pvg@pvg-HP-ProDesk-400-G4-SFF:~/Documents/6$ nasm -f elf64 -g -F stabs m.nasm
pvg@pvg-HP-ProDesk-400-G4-SFF:~/Documents/6$ ld -o x m.o
pvg@pvg-HP-ProDesk-400-G4-SFF:~/Documents/6$ ./x
enter choice for 1: successive addition  method 2: add and shift
1
successive addition result:
FE01pvg@pvg-HP-ProDesk-400-G4-SFF:~/Documents/6$ ./x
enter choice for 1: successive addition  method 2: add and shift
2
add and shift result:
pvg@pvg-HP-ProDesk-400-G4-SFF:~/Documents/6$
```