Computer Organisation & Architecture Assignment No. 3

Name: Akash Jivendra Bachhav

Batch: C3

Roll No: 199

Problem Statement:

Write an ALP for Multiplication using successive addition and Add and shift method.

Source Code:

```
%macro READ 02
mov rax,00
mov rdi,00
mov rsi,%1
mov rdx,%2
syscall
%endmacro

%macro WRITE 02
mov rax,01
mov rdi,01
mov rsi,%1
mov rdx,%2
syscall
%endmacro

Section .data
menu db 10,"1.Successive Addition",10
db "2.Shift Add",10
db "3.Exit",10
db "Enter your choice: ",10
menulen equ $-menu

msg1 db "Enter any two numbers",10
len1 equ $-msg1
msg2 db "The addition of Numbers is",10
len2 equ $-msg2
```

```
msg3 db "Invalid choice!!!",10
len3 equ $-msg3

section .bss
m resq 1
n resq 1
ans resq 1
charbuff resb 17
actl resq 1
choice resb 2
a resq 1
b resq 1
q resq 1
N resq 1

section .text
global _start
_start:
WRITE msg1,len1
READ charbuff, 17 ; 64bit= 16 digit and 1 enter
call accept
;convert ascii to hex
mov [m], rbx
READ charbuff, 17
call accept
mov [n], rbx

printmenu:
WRITE menu, menulen
READ choice, 2
cmp byte[choice],31H ;fetch 1 byte and check with choice 1
je succadd
cmp byte[choice],32H ;fetch 1 byte and check with choice 2
je shiftnadd
cmp byte[choice],33H ;fetch 1 byte and check with choice 3
je exit
WRITE msg3, len3
jmp printmenu

succadd:
mov rcx,[n] ;n will update
mov rbx,00; ;initial rbx store 0

up:
add rbx,[m]
```

```
dec rcx

jnz up
mov [ans],rbx
WRITE msg2,len2
mov rbx,[ans]
call display
jmp printmenu

shiftnadd:
mov qword[a],00H
mov rbx,[m]
mov [b],rbx
mov rbx,[n]
mov [q],rbx
mov byte[N],64

up3:
mov rbx,[q]
and rbx,01H
jz shiftaq
mov rbx,[b]
add[a],rbx
shiftaq:
shr qword[q],01H
mov rbx,[a]
and rbx,01H
jz shifta
mov rbx,01H
ror rbx,01H
or qword[q],rbx

shifta:
shr qword[a],01H
dec byte[N]
jnz up3
WRITE msg2,len2
mov rbx,[a]
call display
mov rbx,[q]
call display
jmp printmenu

exit:
mov rax,60
```

```
mov rdi,00
syscall

accept:
dec rax
mov [actl], rax
mov rbx, 00
mov rsi, charbuff

up2:
shl rbx, 04h
mov rdx, 00h
mov dl, byte[rsi]
cmp dl, 39h
jbe sub30
sub dl, 07h

sub30:
sub dl, 30h
add rbx, rdx
inc rsi
dec qword[actl]
jnz up2
ret

display:
mov rsi, charbuff
mov rcx, 16

up1:
rol rbx, 04h
mov dl, bl
and dl, 0fh
cmp dl, 09h
jbe add30
add dl, 07h

add30:
add dl, 30h
mov byte[rsi], dl
inc rsi
dec rcx
jnz up1
WRITE charbuff,16
ret
```

Output Screen: