;------------------------------------------------------------------------------- ;Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal ;numbers. Use successive addition and add and shift method. (use of 64-bit ;registers is expected)

;-------------------------------------------------------------------------------

%macro input 2

mov rax,0

mov rdi,1

mov rsi,%1

mov rdx,%2

syscall

%endmacro

%macro output 2

mov rax,1

mov rdi,1

mov rsi,%1

mov rdx,%2

syscall

%endmacro

global \_start

section .bss

v1 resb 20

v2 resb 20

v3 resb 40

section .data

msg1 db 10,'Enter 1st number: '

len1 equ $-msg1

msg2 db 10,'Enter 2nd number: '

len2 equ $-msg2

msg3 db 10,'Multiplication is: '

len3 equ $-msg3

menu: db 10,'Menu',10,'1.Successive addition: ',10,'2.Add and shift: ',10,'3.Exit: ',10,'Enter a choice: '

menu\_l: equ $-menu

cnt db 0

count db 0

op db 0

section .text

\_start:

output menu,menu\_l

input op,2

mov bl,[op]

cmp bl,'1'

je succ\_add

cmp bl,'2'

je add\_shift

cmp bl,'3'

je exit

jmp \_start

succ\_add:

call accept

mov rdx,00 ;Product

cmp rbx,00h

je down\_s

up:

add rdx,rax

dec rbx

jnz up

down\_s:

mov rax,rdx

mov rsi,v3

call display

output msg3,len3

output v3,4

jmp \_start

add\_shift:

call accept

mov rdx,00 ;Product

mov cl,8

up1:

shr rbx,1

jc down

shl rax,1

jmp down1

down:

add rdx,rax

shl rax,1

down1:

dec cl

jnz up1

mov rax,rdx

mov rsi,v3

call display

output msg3,len3

output v3,4

jmp \_start

exit:

mov rax,60

mov rbx,1

syscall

accept:

output msg1,len1

input v1,3 ;1st 8 bit number

dec al

mov [cnt],al

mov rsi,v1

call convert

mov rsi,v1

mov qword[rsi],rdx

output msg2,len2

input v2,3 ;2nd 8 bit number

dec al

mov [cnt],al

mov rsi,v2

call convert

mov rsi,v2

mov qword[rsi],rdx

mov rbx,qword[rsi] ;2nd conerted number

mov rsi,v1

mov rax,qword[rsi] ;1st converted number

ret

convert:

;converting accepted number to hex number

mov rdx,0

con:

mov rax,0

mov al,byte[rsi]

cmp al,39h

jg subt

sub al,30h

jmp in

subt:

sub al,37h

in:

rol rdx,4

add rdx,rax

inc rsi

dec byte[cnt]

jnz con

ret

display:

mov cl,4

ups1: rol ax,04h

mov bx,ax

and ax,000fh

cmp ax,09h

jg dns

add al,30h

jmp dns1

dns: add al,37h

dns1: mov byte[rsi],al

mov ax,bx

inc rsi

dec cl

jnz ups1

ret

**OUTPUT :**

manu@ubuntu:~/MP$ nasm -felf64 -g -F stabs A6.nasm

manu@ubuntu:~/MP$ ld -o A6 A6.o

manu@ubuntu:~/MP$ ./A6

Menu

1.Successive addition:

2.Add and shift:

3.Exit:

Enter a choice: 1

Enter 1st number: 0A

Enter 2nd number: 00

Multiplication is: 0000

Menu

1.Successive addition:

2.Add and shift:

3.Exit:

Enter a choice: 1

Enter 1st number: 00

Enter 2nd number: 0A

Multiplication is: 0000

Menu

1.Successive addition:

2.Add and shift:

3.Exit:

Enter a choice: 2

Enter 1st number: AA

Enter 2nd number: AA

Multiplication is: 70E4

Menu

1.Successive addition:

2.Add and shift:

3.Exit:

Enter a choice: 1

Enter 1st number: BC

Enter 2nd number: 02

Multiplication is: 0178

Menu

1.Successive addition:

2.Add and shift:

3.Exit:

Enter a choice: 3