# Program No.10

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

section .data

msg db 'Enter two digit Number::',0xa
msg_len equ $-msg
res db 10,'Multiplication of elements is::'
res_len equ $-res
choice db 'Enter your Choice:',0xa
       db'1.Successive Addition',0xa
       db '2.Add and Shift method',0xa
       db '3.Exit',0xa
choice_len equ $-choice

section .bss
num resb 03
num1 resb 01
result resb 04
cho resb 2


section .text

global _start
_start:

 xor rax,rax
 xor rbx,rbx
 xor rcx,rcx
 xor rdx,rdx
 mov byte[result],0
 mov byte[num],0
 mov byte[num1],0

        mov rax,1
 mov rdi,1
 mov rsi,choice
 mov rdx,choice_len
 syscall
```

```
        mov rax,0                       ;; read choice
 mov rdi,0
 mov rsi,cho
 mov rdx,2
 syscall



 cmp byte[cho],31h              ;; comparing choice
 je a

 cmp byte[cho],32h
 je b

        jmp exit

 a:  call Succe_addition

 jmp _start

 b:  call Add_shift

 jmp _start

exit:
 mov rax,60
 mov rdi,0
 syscall

convert:                                        ;; ASCII to
Hex conversion
 xor rbx,rbx
 xor rcx,rcx
 xor rax,rax

 mov rcx,02
 mov rsi,num
 up1:
 rol bl,04

 mov al,[rsi]
 cmp al,39h
 jbe p1
 sub al,07h
```

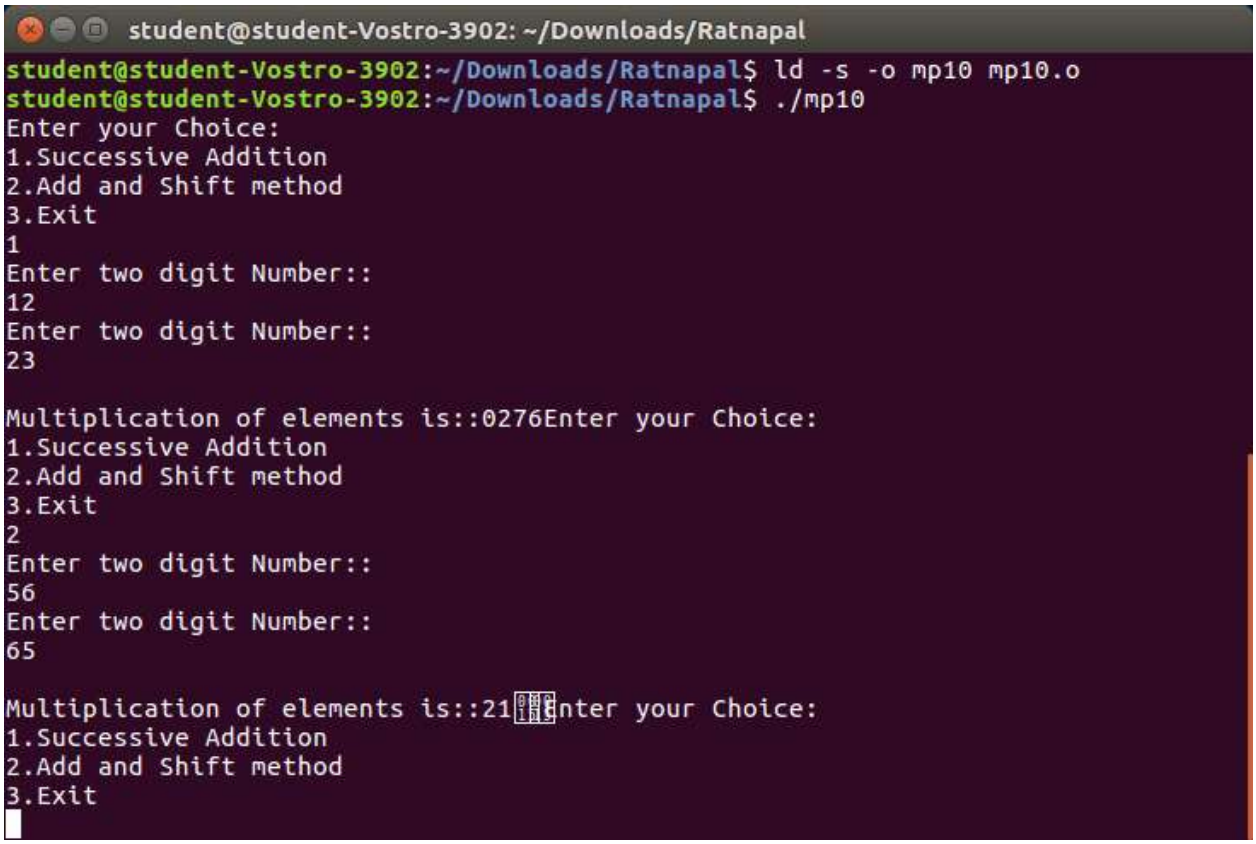
```
 jmp p2
 p1: sub al,30h
 p2: add bl,al
 inc rsi
 loop up1
ret

display:                          ;; Hex to ASCII conversion
 mov rcx,4
 mov rdi,result
 dup1:
 rol bx,4
 mov al,bl
 and al,0fh
 cmp al,09h
 jbe p3
 add al,07h
 jmp p4
 p3: add al,30h
 p4:mov [rdi],al
 inc rdi
 loop dup1

        mov rax,1
 mov rdi,1
 mov rsi,result
 mov rdx,4
 syscall


ret


Succe_addition:

        mov rax,1
 mov rdi,1
 mov rsi,msg
 mov rdx,msg_len
 syscall



        mov rax,0
 mov rdi,0
 mov rsi,num
 mov rdx,3
```

```
 syscall


 call convert
 mov [num1],bl

        mov rax,1
 mov rdi,1
 mov rsi,msg
 mov rdx,msg_len
 syscall


        mov rax,0
 mov rdi,0
 mov rsi,num
 mov rdx,3
 syscall


 call convert
 xor rcx,rcx
 xor rax,rax
 mov rax,[num1]

 repet:
 add rcx,rax
 dec bl
 jnz repet

 mov [result],rcx

        mov rax,1
 mov rdi,1
 mov rsi,res
 mov rdx,res_len
 syscall


 mov rbx,[result]

 call display
ret
```

```
Add_shift:

        mov rax,1
 mov rdi,1
 mov rsi,msg
 mov rdx,msg_len
 syscall



        mov rax,0
 mov rdi,0
 mov rsi,num
 mov rdx,3
 syscall



 call convert
 mov [num1],bl


        mov rax,1
 mov rdi,1
 mov rsi,msg
 mov rdx,msg_len
 syscall



        mov rax,0
 mov rdi,0
 mov rsi,num
 mov rdx,3
 syscall



 call convert

 mov [num],bl

 xor rbx,rbx
 xor rcx,rcx
 xor rdx,rdx
```

```
 xor rax,rax
 mov dl,08
 mov al,[num1]
 mov bl,[num]

 p11:
        shr bx,01
 jnc p
 add cx,ax
 p:
        shl ax,01
 dec dl
 jnz p11

 mov [result],rcx


        mov rax,1
 mov rdi,1
 mov rsi,res
 mov rdx,res_len
 syscall

 ;dispmsg res,res_len

 mov rbx,[result]
 call display

ret
```

# Output:


student@student-Vostro-3902: ~/Downloads/Ratnapal
student@student-Vostro-3902:~/Downloads/Ratnapal$ ld -s -o mp10 mp10.o
student@student-Vostro-3902:~/Downloads/Ratnapal$ ./mp10
Enter your Choice:
1.Successive Addition
2.Add and Shift method
3.Exit
1
Enter two digit Number::
12
Enter two digit Number::
23
Multiplication of elements is::0276Enter your Choice:
1.Successive Addition
2.Add and Shift method
3.Exit
2
Enter two digit Number::
56
Enter two digit Number::
65
Multiplication of elements is::21nter your Choice:
1.Successive Addition
2.Add and Shift method
3.Exit