ASSIGNMENT NO 10

PROBLEM STATEMENT:

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).

SOURCE CODE:

```
section .data

        msg1 db 10,10,'***Multiplication by successive addition***'

        msg1_len equ $-msg1

        msg2 db 10,10,'Enter two digit number: '

        msg2_len equ $-msg2

        msg3 db 10,10,'Multiplication is: '

        msg3_len equ $-msg3


section .bss

        numascii resb 03

        multi1 resb 02

        resl resb 02

        resh resb 01

        dispbuff resb 04


%macro dispmsg 2

        mov eax,04
        mov ebx,01
        mov ecx,%1
        mov edx,%2
        int 80h
%endmacro

%macro accept 2
        mov eax,03
        mov ebx,00
        mov ecx,%1
        mov edx,%2
        int 80h
%endmacro

section .text
global _start
_start:
        dispmsg msg1,msg1_len
        dispmsg msg2,msg2_len
        accept numascii,03
        call packnum
        mov [multi1],bl
        dispmsg msg2,msg2_len
        accept numascii,03
        call packnum
        mov ecx,00h
mov eax,[multi1]
add1:
        add ecx,eax
        dec bl
        jnz add1                ;checks bl is 0 or not
        mov [resl],ecx

        dispmsg msg3,msg3_len
        mov ebx,[resl]
        call disp16
        mov eax,01
        mov ebx,00
        int 80h

packnum:
        mov bl,0
        mov ecx,02
        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

        disp16:
                mov ecx,4
                mov edi,dispbuff
                dub1:
                        rol bx,4
                        mov al,bl
                        and al,0fh
                        cmp al,09h
                        jbe x1
                        add al,07
                        x1:
                                add al,30h
                                mov [edi],al
                                inc edi
                                loop dub1
                                dispmsg dispbuff,4
        ret
section .data
        msg1 db 10,10,'***Multiplication by add & shift***'
        msg1_len equ $-msg1
        msg2 db 10,'Enter two digit number: '
        msg2_len equ $-msg2
        msg3 db 10,'Multiplication is: '
        msg3_len equ $-msg3

section .bss
        numascii resb 03
        multi1 resb 02
        multi2 resb 02
        resl resb 02
        dispbuff resb 04

%macro dispmsg 2
        mov eax,04
        mov ebx,01
        mov ecx,%1
        mov edx,%2
        int 80h
%endmacro

%macro accept 2
        mov eax,03
        mov ebx,00
        mov ecx,%1
        mov edx,%2
        int 80h
```

```
%endmacro

section .text
global _start
_start:
        dispmsg msg1,msg1_len
        dispmsg msg2,msg2_len
        accept numascii,03
        call packnum
        mov [multi1],bl
        dispmsg msg2,msg2_len
        accept numascii,03
        call packnum
        mov [multi2],bl
        mov al,[multi1]
        mov cl,00
        mov edx,00
        mov edx,08

        add1:
                rcr al,01
                jnc next1
                mov bh,00h
                shl bx,cl               ;shl=shift left
                add [resl],bx
                mov bl,[multi2]
```

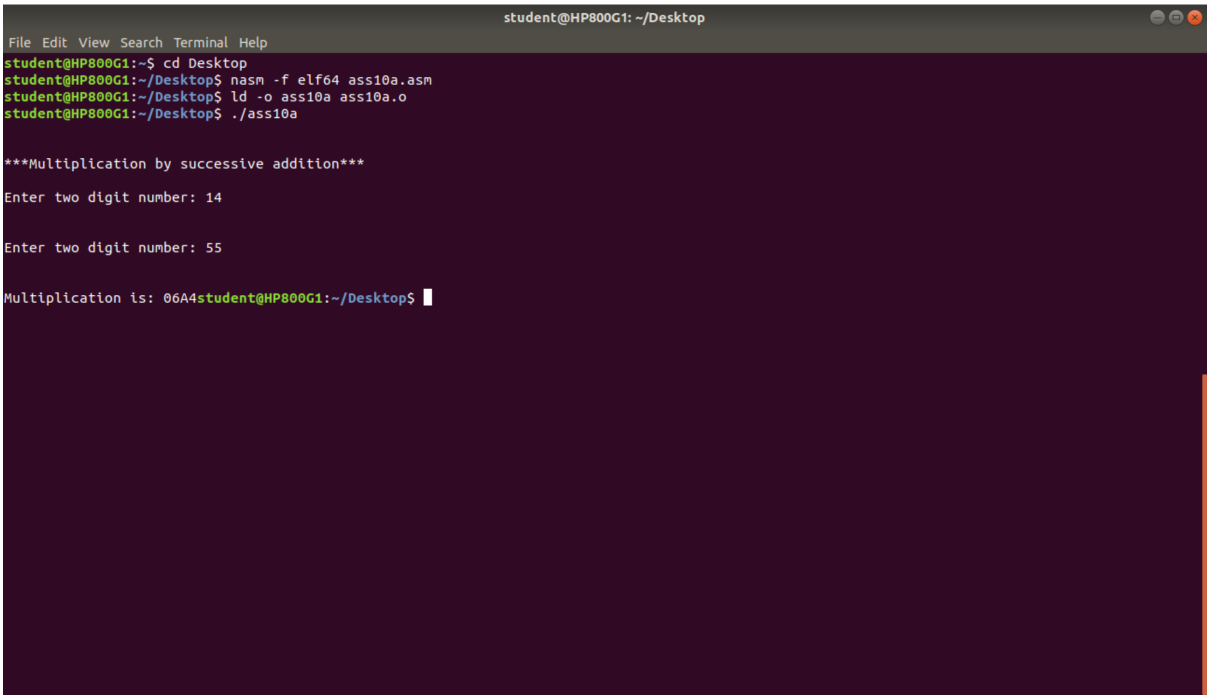
```
          next1:
                    inc cl
                    dec edx
                    jnz add1
                    dispmsg msg3,msg3_len
                    mov bx,[resl]
call disp16
mov eax,01
mov ebx,00
int 80h

packnum:
          mov bl,00
          mov ecx,02
          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


disp16:
          mov ecx,4
          mov edi,dispbuff
          dub1:
                    rol bx,4
                    mov al,bl
                    and al,0fh
                    cmp al,09h
                    jbe x1
                    add al,07
                    x1:
                              add al,30h
                              mov [edi],al
                              inc edi
                              loop dub1
                              dispmsg dispbuff,4
ret
```

OUTPUT:



student@HP800G1: ~/Desktop
File Edit View Search Terminal Help

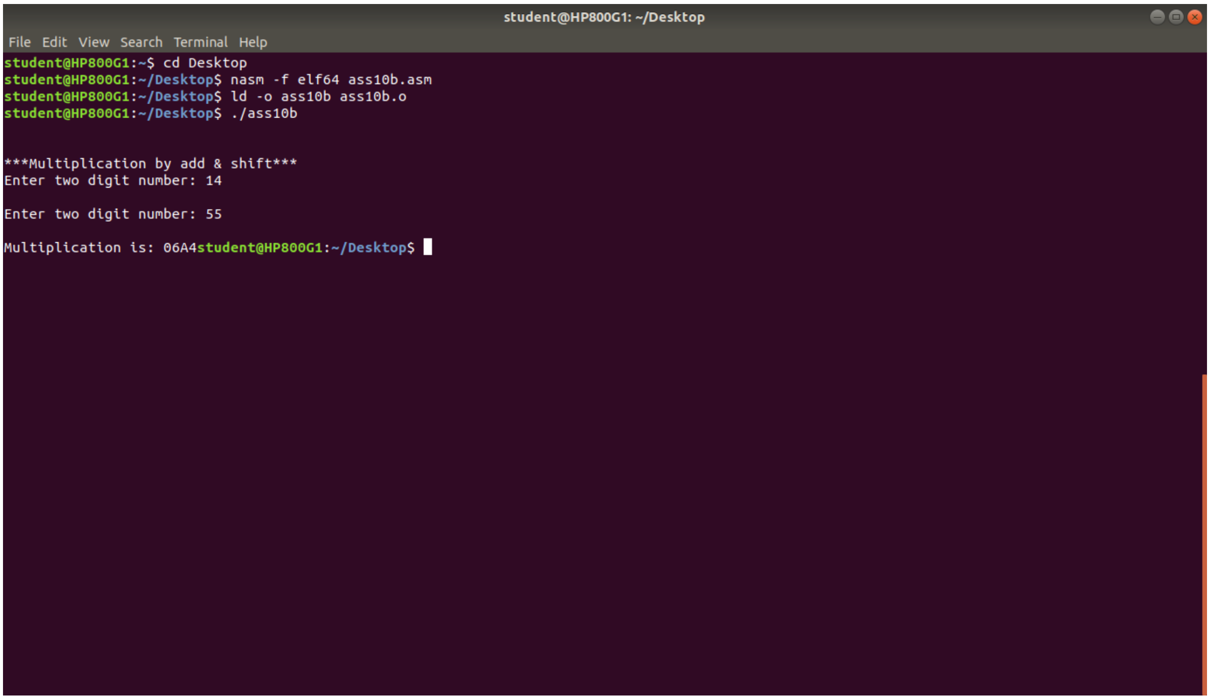
```
student@HP800G1:~$ cd Desktop
student@HP800G1:~/Desktop$ nasm -f elf64 ass10a.asm
student@HP800G1:~/Desktop$ ld -o ass10a ass10a.o
student@HP800G1:~/Desktop$ ./ass10a


***Multiplication by successive addition***

Enter two digit number: 14


Enter two digit number: 55


Multiplication is: 06A4student@HP800G1:~/Desktop$ 
```



student@HP800G1: ~/Desktop
File Edit View Search Terminal Help

```
student@HP800G1:~$ cd Desktop
student@HP800G1:~/Desktop$ nasm -f elf64 ass10b.asm
student@HP800G1:~/Desktop$ ld -o ass10b ass10b.o
student@HP800G1:~/Desktop$ ./ass10b


***Multiplication by add & shift***
Enter two digit number: 14

Enter two digit number: 55

Multiplication is: 06A4student@HP800G1:~/Desktop$ 
```