ASSIGNMENT NO 6

PROBLEM STATEMENT:

Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5- digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT. Display proper strings to prompt the user while accepting the input and displaying the result. (Wherever necessary, use 64-bit registers).

SOURCE CODE:

```
section .data
        msg1 db 10,10,'###### Menu for Code Conversion ######'
        db 10,'1: Hex to BCD'
        db 10,'2: BCD to Hex'
        db 10,'3: Exit'
        db 10,10,'Enter Choice:'
        msg1length  equ $-msg1

        msg2 db 10,10,'Enter 4 digit hex number::'
        msg2length equ $-msg2

        msg3 db 10,10,'BCD Equivalent:'
        msg3length  equ $-msg3

        msg4 db  10,10,'Enter 5 digit BCD number::'
        msg4length  equ $-msg4
```

```
        msg5 db 10,10,'Wrong Choice Entered....Please try again!!!',10,10
        msg5length  equ $-msg5

        msg6 db 10,10,'Hex Equivalent::'
        msg6length equ $-msg6
        cnt db  0

section .bss
        arr resb   06        ;common buffer for choice, hex and bcd input
        dispbuff resb   08
        ans resb   01



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

%macro accept 2
        mov rax,0
        mov rdi,0
        mov rsi,%1
        mov rdx,%2
```

```
        syscall
%endmacro


section .text
        global _start
_start:
menu:

        disp msg1,msg1length
        accept arr,2  ;        choice either 1,2,3 + enter

        cmp byte [arr],'1'
        jne l1
        call hex2bcd_proc

        jmp menu

l1:     cmp byte [arr],'2'
        jne l2
        call bcd2hex_proc
        jmp menu

l2:     cmp byte [arr],'3'
        je exit
        disp msg5,msg5length
        jmp menu
```

```
exit:
        mov rax,60
        mov rbx,0
        syscall



hex2bcd_proc:
        disp msg2,msg2length
        accept arr,5            ; 4 digits + enter
        call conversion
        mov rcx,0
        mov ax,bx
        mov bx,10          ;Base of Decimal No. system
l33:    mov dx,0
        div bx                  ; Divide the no by 10
        push rdx                ; Push the remainder on stack
        inc rcx
inc byte[cnt]
        cmp ax,0
        jne l33
disp msg3,msg3length
l44:    pop rdx                 ; pop the last pushed remainder from stack
        add dl,30h              ; convert it to ascii
        mov [ans],dl
disp ans,1
        dec byte[cnt]
```

```
jnz l44
        ret

bcd2hex_proc:
        disp msg4,msg4length
        accept arr,6        ; 5 digits + 1 for enter

        disp msg6,msg6length

        mov rsi,arr
        mov rcx,05
        mov rax,0
        mov ebx,0ah

l55:    mov rdx,0
        mul ebx         ; ebx * eax = edx:eax
        mov dl,[rsi]
        sub dl,30h
        add rax,rdx
        inc rsi
        dec rcx
jnz l55
        mov ebx,eax     ; store the result in ebx
        call disp32_num
        ret

conversion:
```

```
        mov bx,0
        mov ecx,04
        mov esi,arr
up1:
        rol bx,04
        mov al,[esi]
        cmp al,39h
        jbe l22
        sub al,07h
l22:    sub al,30h
        add bl,al
        inc esi
        loop up1
        ret
```

; the below procedure is to display 32 bit result in ebx why 32 bit & not 16 bit; because 5 digit bcd no ranges between 00000 to 99999 & for ;65535 ans is FFFF

; i.e if u enter the no between 00000-65535 u are getting the answer between

;0000-FFFF, but u enter i/p as 99999 urans is greater than 16 bit which is not; fitted in 16 bit register so 32 bit register is taken frresult

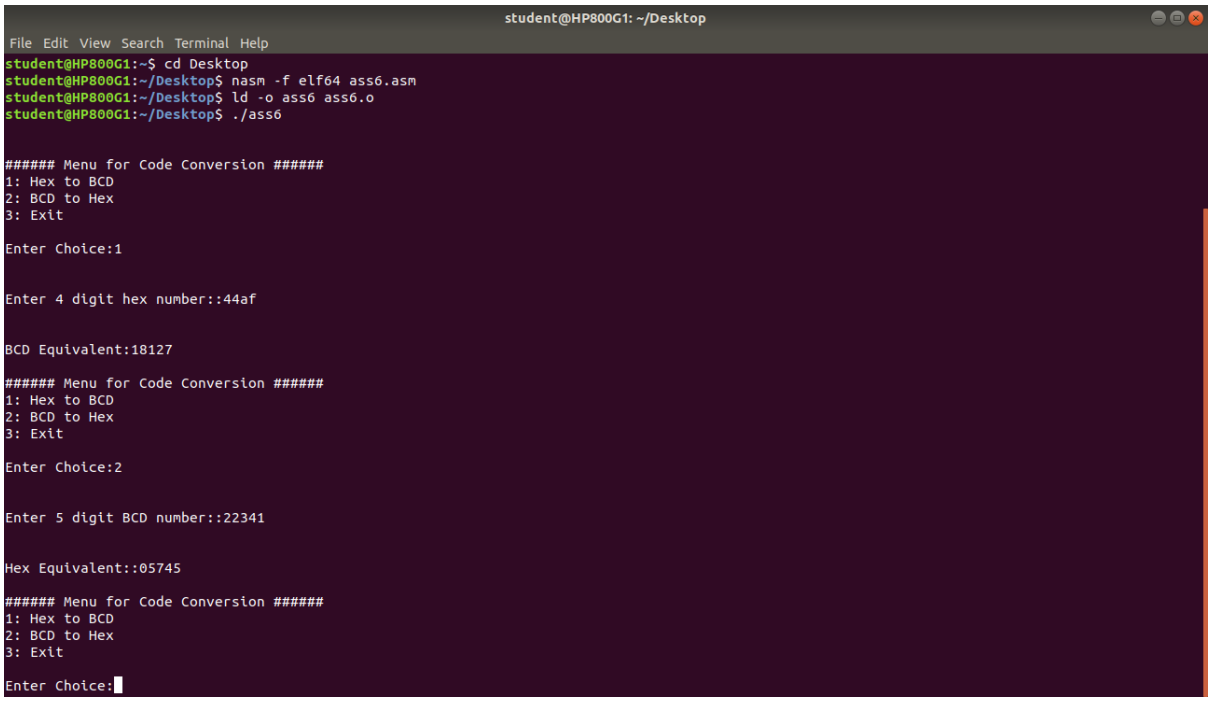
```
disp32_num:
        mov rdi,dispbuff
        mov rcx,08          ; since no is 32 bit,no of digits 8
```

```
l77:
        rol ebx,4
        mov dl,bl
        and dl,0fh
        add dl,30h
        cmp dl,39h
        jbe l66
        add dl,07h

l66:
        mov [rdi],dl
        inc rdi
        dec rcx
jnz  l77
        disp dispbuff+3,5  ;Dispays only lower 5 digits as upper three are
'0'

        ret
```

OUTPUT:


student@HP800G1: ~/Desktop
File Edit View Search Terminal Help
student@HP800G1:~$ cd Desktop
student@HP800G1:~/Desktop$ nasm -f elf64 ass6.asm
student@HP800G1:~/Desktop$ ld -o ass6 ass6.o
student@HP800G1:~/Desktop$ ./ass6
Menu for Code Conversion
1: Hex to BCD
2: BCD to Hex
3: Exit
Enter Choice:1
Enter 4 digit hex number::44af
BCD Equivalent:18127
Menu for Code Conversion
1: Hex to BCD
2: BCD to Hex
3: Exit
Enter Choice:2
Enter 5 digit BCD number::22341
Hex Equivalent::05745
Menu for Code Conversion
1: Hex to BCD
2: BCD to Hex
3: Exit
Enter Choice: