TITLE:- ALP Program to Conversion of 4-digit HEX to BCD and vice versa.

NAME:- CLASS:-

BATCH:- ROLL NO:-

****

```
section .data
     msg         db    10,10,"MIL Assignment 03 : Hex to BCD & BCD to Hex"
                 db    10,"----------------------------------------"
     msg_len:    equ   $-msg

     menu        db    10," --------------------Menu-----------------"
                 db    10,"1. Hex to BCD "
                 db    10,"2. BCD to Hex"
                 db    10,"3. Exit "
                 db    10
                 db    10,"Enter your choice: "
     menu_len:   equ   $-menu

     h2bmsg      db    10,"Hex to BCD "
                 db    10,"Enter 4-digit Hex number: "
     h2bmsg_len:equ    $-h2bmsg

     b2hmsg      db    10,"BCD to Hex "
                 db    10,"Enter 5-digit BCD number: "
     b2hmsg_len:equ    $-b2hmsg

     hmsg        db    10,13,"Equivalent Hex number is: "
     hmsg_len:   equ   $-hmsg

     bmsg        db    10,13,"Equivalent BCD number is: "
     bmsg_len:   equ   $-bmsg

     emsg        db    10,"You entered Invalid Data!!!",10
     emsg_len:   equ   $-emsg

;-----------------------------------------------------------------------
section .bss
     buf         resb  6
     buf_len:    equ   $-buf

     digitcount resb   1

     ans         resw  1
     char_ans    resb  4
;-----------------------------------------------------------------------

%macro  print   2
     mov   rax,1           ; Function 1 - write
     mov   rdi,1           ; To stdout
     mov   rsi,%1          ; String address
     mov   rdx,%2          ; String size
     syscall               ; invoke operating system to WRITE
%endmacro
```

```
%macro     read  2
     mov   rax,0           ; Function 0 - Read
     mov   rdi,0           ; from stdin
     mov   rsi,%1          ; buffer address
     mov   rdx,%2          ; buffer size
     syscall               ; invoke operating system to READ
%endmacro

%macro     exit  0
     mov rax, 60
     xor rdi, rdi
     syscall
%endmacro
;-----------------------------------------------------------------
section .text
     global _start
_start:
     print msg,msg_len
     print menu, menu_len

     read  buf,2
     mov   al,[buf]

c1:  cmp   al,'1'
     jne   c2
     call  hex_bcd
     jmp   _start

c2:  cmp   al,'2'
     jne   c3
     call  bcd_hex
     jmp   _start

c3:  cmp   al,'3'
     jne   err
     exit

err: print emsg,emsg_len
     jmp   _start
;-----------------------------------------------------------------
hex_bcd:
     print h2bmsg, h2bmsg_len

     call  accept_16
     mov   ax,bx
     mov   rbx,10
back:
     xor   rdx,rdx
     div   rbx

     push  dx
     inc   byte[digitcount]
```

```
      cmp   rax,0h
      jne   back

      print bmsg, bmsg_len
print_bcd:
      pop   dx
      add   dl,30h
      mov   [char_ans],dl
      print char_ans,1
      dec   byte[digitcount]
      jnz   print_bcd

      ret
;-----------------------------------------------------------------
bcd_hex:
      print b2hmsg, b2hmsg_len
      read  buf,buf_len

      mov   rsi,buf
      xor   rax,rax
      mov   rbx,10
      mov   rcx,05

back1:xor   rdx,rdx
      mul   ebx

      xor   rdx,rdx
      mov   dl,[rsi]
      sub   dl,30h
      add   rax,rdx

      inc   rsi
      dec   rcx
      jnz   back1

      mov   [ans],ax

      print bmsg, bmsg_len
      mov   ax,[ans]
      call  display_16

      ret
 ;-----------------------------------------------------------------
accept_16:
      read  buf,5      ; buflen = 4 + 1

      xor   bx,bx
      mov   rcx,4
      mov   rsi,buf
next_digit:
      shl   bx,04
      mov   al,[rsi]
```

```
      cmp        al,"0"
      jb         error
      cmp        al,"9"
      jbe        sub30

      cmp        al,"A"
      jb         error
      cmp        al,"F"
      jbe        sub37

      cmp        al,"a"
      jb         error
      cmp        al,"f"
      jbe     sub57

error:  print   emsg,emsg_len
        exit

sub57:     sub al, 20h
sub37:     sub al, 07h
sub30:     sub al, 30h

      add  bx,ax
      inc  rsi
      loop next_digit
      ret
;---------------------------------------------------------------
display_16:
      mov  rsi,char_ans+3
      mov  rcx,4

cnt:  mov  rdx,0
      mov  rbx,16
      div  rbx
      cmp  dl, 09h
      jbe  add30
      add  dl, 07h
add30:
      add  dl,30h
      mov  [rsi],dl
      dec  rsi

      dec  rcx
      jnz  cnt

      print char_ans,4
      ret
;---------------------------------------------------------------
```

**************************** OUTPUT ******************************

```
admin@oomplab:~$ nasm -f elf64 -o Pract3.o Pract3.nasm
admin@oomplab:~$ ld -o Pract3 Pract3.o
admin@oomplab:~$ ./Pract3


MIL assignment 03 : Hex to BCD & BCD to Hex
-------------------------------------------
--------------------Menu-------------------
1. Hex to BCD
2. BCD to Hex
3. Exit

Enter your choice: 1

Hex to BCD
Enter 4-digit Hex number: 00FF

Equivalent BCD number is: 255

MIL assignment 03 : Hex to BCD & BCD to Hex
-------------------------------------------
--------------------Menu-------------------
1. Hex to BCD
2. BCD to Hex
3. Exit

Enter your choice: 2

BCD to Hex
Enter 5-digit BCD number: 00255

Equivalent BCD number is: 00FF

MIL assignment 03 : Hex to BCD & BCD to Hex
-------------------------------------------
--------------------Menu-------------------
1. Hex to BCD
2. BCD to Hex
3. Exit

Enter your choice: 3
admin@oomplab:~$
```