

## Program No. 6

; 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).

```
global _start
```

```
_start:
```

```
section .text
```

```
; macro for system call for write
```

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

```
; macro for system call for read
```

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

```
;-----First Choice Hex to BCD-----
ch1:
```

```
    ; accept numbers
    disp msg1,len1
    accept num,02
    call convert
```

```

    mov [no.1],al
    accept num,03
    call convert
    mov [no.2],al
    disp msg2,len2

; Form ax as input
    mov ah,[no.1]
    mov al,[no.2]

;Point esi to predefined array in .data
    mov esi,array1

; Hex to BCD conversion
15:
    mov dx,0000h
    mov bx,[esi]
    div bx
    mov [rem],dx
    mov [t1],al
    push rsi
    call disp_proc
    pop rsi
    inc esi
    inc esi
    mov ax,[rem]
    dec byte[cnt]
jnz 15

    disp msg,len

;To exit program.
ch3:
    mov rax,60
    mov rdi,0
    syscall

;CONVERT

procedure
convert:
    mov esi,num
    mov al,[esi]
    cmp al,39h
    jle l1

```

```

        sub al,07h
11: sub al,30h
    rol al,04h ;to swap number

    mov bl,al

    inc esi
    mov al,[esi]
    cmp al,39h
    jle 12
        sub al,07h
12: sub al,30h

    add al,bl
    mov [t1],al
ret

```

;CONVERT2 procedure

```

convert2:
    mov al,[num]
    cmp al,39h
    jle 18
    sub al,07h
18: sub al,30h
ret

```

;DISPLAY procedure

disp\_proc:

;for unt's place

```

    mov al,[t1]
    cmp al,09h
    jle 14
    add al,07h
14: add al,30h
    mov [t2],al
    disp t2,1
ret

```

;DISPLAY@ procedure

display2:

```

    mov rsi,charans+3
    mov rcx,04h

112: mov rdx,0
    mov rbx,10h
    div rbx

```

```

    cmp dl,09h
    jle l3
    add dl,07h
    l3:add dl,30h
    mov [rsi],dl
    dec rsi
    dec rcx
jnz l12

```

```

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

```

```
ret
```

#### section .data

```

msg: db "",10
len: equ $-msg
msg1: db "Enter Hex number : ",10
len1: equ $-msg1
msg2: db "BCD equivalent is : ",10
len2: equ $-msg2
msg3: db "#####MENU#####",10
      db "1.Hex to BCD.",10
      db "2.BCD to Hex.",10
      db "3.Exit.",10
len3: equ $-msg3
msg4: db "Enter your choice : ",10
len4: equ $-msg4
msg5: db "Enter BCD number : ",10
len5: equ $-msg5
msg6: db "Hex equivalent is : ",10
len6: equ $-msg6

array1 dw 2710h,03E8h,0064h,000Ah,0001h
cnt db 5
cnt2 db 5

```

#### section .bss

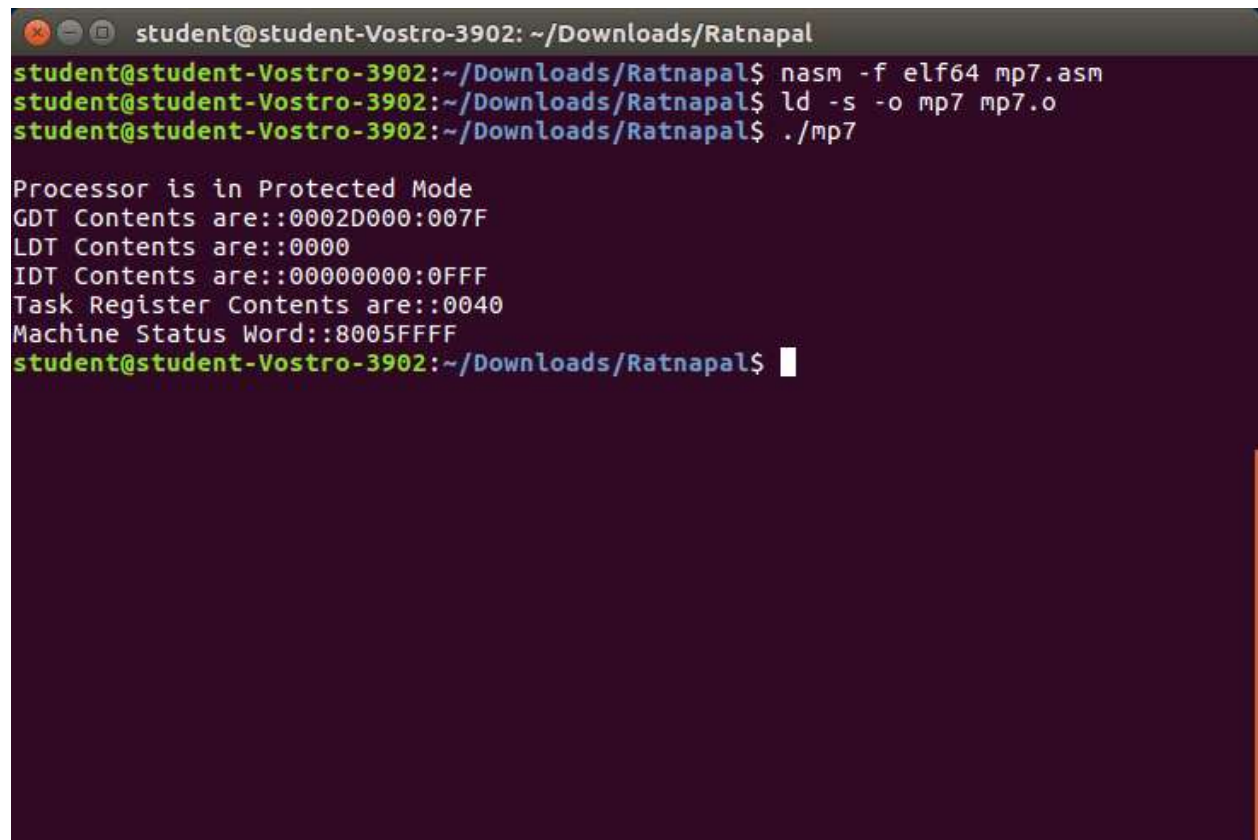
```

num resb 03
no.1 resb 02
no.2 resb 02
t1 resb 03
t2 resb 03
t3 resb 03

```

```
rem resw 02
result resw 03
choice resb 03
charans resb 08
```

Output :

A terminal window with a dark purple background and light green text. The title bar shows 'student@student-Vostro-3902: ~/Downloads/Ratnapal'. The terminal contains the following text:

```
student@student-Vostro-3902:~/Downloads/Ratnapal$ nasm -f elf64 mp7.asm
student@student-Vostro-3902:~/Downloads/Ratnapal$ ld -s -o mp7 mp7.o
student@student-Vostro-3902:~/Downloads/Ratnapal$ ./mp7

Processor is in Protected Mode
GDT Contents are::0002D000:007F
LDT Contents are::0000
IDT Contents are::00000000:0FFF
Task Register Contents are::0040
Machine Status Word::8005FFFF
student@student-Vostro-3902:~/Downloads/Ratnapal$
```