Experiment No. 5

Name – Mehatab Mahibub Sanadi
Roll No. – CO2056

---

a) HEX to BCD //

```
section .data

        hexinmsg db 10,10,'Please enter 4 digit hex number::'
        hexinmsg_len equ $-hexinmsg

        bcdopmsg db 10,10,'BCD Equivalent::'
        bcdopmsg_len equ $-bcdopmsg

section .bss
        numascii resb 06                ;common buffer for choice, hex and bcd input
        ansbuff resb 02
        dnumbuff resb 08

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

%macro accept 2
        mov eax,3
        mov ebx,0
        mov ecx,%1
        mov edx,%2
        int 80h
%endmacro

section .text
global _start
_start:

    call hex2bcd

        mov eax,1
        mov ebx,0
        int 80h

hex2bcd:
        disp hexinmsg,hexinmsg_len
        accept numascii,5
```

```
        call atoh
        mov ax,bx           ;ax=given hex number
        mov bx,10               ;Base of Decimal No. system
      mov ecx,0
h2b1:   mov dx,0
        div bx              ;bx=divisior edx=store remainder
        push rdx                ;push remainder
        inc ecx
        cmp ax,0
        jne h2b1

        mov edi,ansbuff

h2b2:   pop rdx
        add dl,30h
        mov [edi],dl
        inc edi
        loop h2b2

        disp bcdopmsg,bcdopmsg_len
        disp ansbuff,5
        ret
atoh:
        mov bx,0
        mov ecx,04
        mov esi,numascii
up1:
        rol bx,04
        mov al,[esi]
      sub al,30h
        cmp al,09h
        jbe skip1
        sub al,07h
skip1:add bl,al
        inc esi
        loop up1
        ret
```

b) BCD to HEX

```
section .data

        bcdinmsg db 10,10,'Please enter 5 digit BCD number::'
        bcdinmsg_len equ $-bcdinmsg

        hexopmsg db 10,10,'Hex Equivalent::'
        hexopmsg_len equ $-hexopmsg
```

```
section .bss
        numascii resb 06                ;common buffer for choice, hex and bcd input
        ansbuff resb 02
        dnumbuff resb 08

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

%macro accept 2
        mov eax,3
        mov ebx,0
        mov ecx,%1
        mov edx,%2
        int 80h
%endmacro

section .text
        global _start
_start:

        call bcd2hex

      mov eax,1
        mov ebx,0
        int 80h


bcd2hex:
        disp bcdinmsg,bcdinmsg_len
        accept numascii,6

        mov esi,numascii
        mov eax,0
        mov ebx,0AH
    mov ecx,5

b2h1:   mov dl,0
        mul ebx
    mov dl,[esi]
        sub dl,30h
        add eax,edx
        inc esi
    DEC ECX
    JNZ b2h1
        mov ebx,eax
```

```
        call htoa
        ret

atoh:
        mov bx,0
        mov ecx,04
        mov esi,numascii
up1:
        rol bx,04
        mov al,[esi]
    sub al,30h
        cmp al,09h
        jbe skip1
        sub al,07h
skip1:add bl,al
        inc esi
        loop up1
        ret

htoa:
        mov edi,dnumbuff        ;point esi to buffer
        mov ecx,04              ;load number of digits to display
disp1:
     rol bx,4           ;as only bx contains result so rotate number left by 4 bits
        mov dl,bl               ;move lower byte in dl
        and dl,0fh              ;mask upper digit of byte in dl
        cmp dl,09h              ;compare with 09h
        jbe next          ;if less than 39h akip adding 07 more
        add dl,07h              ;else add 07
next: add dl,30h          ;add 30h to calculate ASCII code
        mov [edi],dl            ;store ASCII code in buffer
        inc edi                 ;point to next byte
        loop disp1


disp hexopmsg,hexopmsg_len          ;run macro
disp dnumbuff,4                 ;Dispays only lower 4 digits as upper four are '0'
        ret
```