/\***BATCH** : H10

**ROLL** **NO** : 23269

**PROBLEM** **STATEMENT** : Write menu driven 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 i) HEX to BCD ii) BCD to HEX iii) EXIT. Display proper strings to prompt the user while accepting the input and displaying the result. Write near procedure to complete the task.\*/

print macro msg

lea dx, msg

mov ah, 09h

int 21h

endm

.model small

.stack 100h

.data

menu db 13, 10, 'MENU : ', 13, 10, '1. Hex to BCD', 13, 10, '2. BCD to hex', 13, 10, '3. Exit ', 13, 10, 'Enter your choice. $'

msg1 db 13, 10, 'Enter 4 digit hex number : $'

msg2 db 13, 10, 'Enter 5 digit BCD number : $'

msg3 db 13, 10, 'Equivalent BCD number : $'

msg4 db 13, 10, 'Equivalent hex number : $'

msg5 db 13, 10, 'Enter valid choice$'

msg6 db 13, 10, 'Enter valid digit$'

op db 5 dup(0)

.code

mov ax, @data

mov ds, ax

main:

print menu

mov ah, 01h

int 21h

cmp al, '1'

je case1

cmp al, '2'

je case2

cmp al, '3'

je exit

print msg5

jmp main

case1:

print msg1

call input

print msg3

call hextobcd

jmp main

case2:

print msg2

call inp1

call bcdtohex

print msg4

call disp1

jmp main

exit:

mov ah, 4ch

int 21h

inp1 proc near

lea si,op

mov cl,05

c1:

mov ah,01h

int 21h

cmp al,30h

jb invalid1

cmp al,39h

jg invalid1

sub al,30h

mov [si],al

inc si

dec cl

jnz c1

c2: ret

invalid1:

print msg6

endp

input proc near

mov cx,0404h

mov ax,0000h

mov bx,0000h

in1: mov ah,01h

int 21h

cmp al,30h

jb invalid

cmp al,39h

jg a1

sub al,30h

jmp insert

a1: cmp al,41h

jb invalid

cmp al,46h

jg a2

sub al,37h

jmp insert

a2: cmp al,61h

jb invalid

cmp al,66h

jg invalid

sub al,57h

jmp insert

insert:

shl bx,cl ;move each digit to left

add bl,al

dec ch

jnz in1

ret

invalid:

print msg6

jmp main

endp

hextobcd proc near

mov ax,bx

mov bx,000ah

mov cl,00

b1: mov dx,00h

div bx

push dx

inc cx

cmp ax,00h

jne b1

b2: pop dx

add dl,30h

mov ah,02h

int 21h

dec cl

jnz b2

ret

endp

bcdtohex proc near

mov bx,0000h

mov cx,0000h

mov dx,0000h

lea si,op

mov ax,10000

mov bl,[si]

mul bx

add cx,ax

inc si

mov ax,1000

mov bl,[si]

mul bx

add cx,ax

inc si

mov ax,100

mov bl,[si]

mul bx

add cx,ax

inc si

mov ax,10

mov bl,[si]

mul bx

add cx,ax

inc si

mov ax,1

mov bl,[si]

mul bx

add cx,ax

ret

endp

disp1 proc near

mov dx,0000h

mov bx,cx

mov cx,0404h

d2: rol bx,cl

mov dx,00h

mov dl,bl

and dl,0fh

cmp dl,09h

jbe d1

add dl,07h

d1: add dl,30h

mov ah,02h

int 21h

inc si

dec ch

jnz d2

ret

endp

end

**OUTPUT :**

