.model small

.data ;data segment

menu db 10,13,'1.Hex to BCD'

db 10,13,'2.BCD to Hex'

db 10,13,'3.Exit'

db 10,13,'Enter Your Choice: $'

wrgdb 10,13,'Wrong Choice$'

msg1 db 10,13,'Enter Hex: $'

msg2 db 10,13,'Equivalent BCD is: $'

msg3 db 10,13,'Enter BCD: $'

msg4 db 10,13,'Equivalent Hex is: $'

arrdb 25 dup('$')

count db 0

ansdw 0

.code ;code segment

movax,@data ;data initialization

movds,ax

again:

mov ah,09h

lea dx,menu

int 21h

mov ah,01h ;accept choice

int 21h

cmp al,'1' ;comapre the accepted value

jne b2

call hex\_bcd ;call hex\_bcd function

jmp again ;go back to menu

b2:

cmp al,'2'

jne b3

call bcd\_hex ;call bcd\_hex function

jmp again

b3:

cmp al,'3'

jne b4

jmp exit ;jump to exit

b4:

mov ah,09h

lea dx,wrg ;wrong choice message

int 21h

jmp again

proc hex\_bcd

mov ah,09h

lea dx,msg1 ;display message

int 21h

mov bx,0 ;initialize bx with 0

mov ch,04 ;move value 4 to ch

a4: mov ah,01 ;To accept single digit from user

int 21h

movdl,al

cmp dl,39h ;compare dl with 39

jbe a3 ;jump to a3 if below or equal

cmp dl,39h ;compare dl with 39

jg a6 ;jump to a6 if greater

a3: sub dl,30h ;subtract dl by 30

jmp a5 ;jump to a5

a6: sub dl,37h ;subtract dl by 37

jmp a5 ;jump to a5

a5: rol bx,04 ;rotate bx by 4

add bx,dx ;add contents bx and dx

decch ;decrement ch

jnz a4 ;jump if not zero to label a4

movax,bx

mov bx,10 ;move 10 to bx

mov dx,00 ;set dx to 0

back:divbx ;divide ax by bx

push dx ;push dx into stack

inc count ;increment count

cmp ax,0h

jne back

mov ah,09h

lea dx,msg2 ;display message

int 21h

print\_bcd:

pop dx ;pop dx from stack

add dl,30h

mov [arr],dl

mov ah,09h

lea dx,arr

int 21h

dec count

jnzprint\_bcd

ret

hex\_bcdendp

proc bcd\_hex

mov ah,09h

lea dx,msg3 ;print msg3

int 21h

mov bx,0000h

mov cl,05 ;set counter to 5

up3:mov ax,000Ah ;mov 10 to ax

mulbx ;multiply ax and bx

movbx,ax

mov ah,01h ;accept a character

int 21h

sub al,30h

mov ah,00

add bx,ax ;add bx and ax

dec cl

jnz up3

movans,bx ;movbx to ans

mov ah,09h

lea dx,msg4

int 21h

movax,ans

mov ch,04 ;initialize ch with 4 print four numbers

movbx,ax ;move the result in bx

l2: rol bx,04 ;rotate bx left 4 times

movdl,bl ;move contents of bl to dl

and dl,0fh ;logical and dl with 0fh

cmp dl,09 ;compare dl with 09

jbe l1 ;jump to l1 if equal

add dl,07h

l1: add dl,30h ;For converting hex to ascii

mov ah,02 ;To print single digit

int 21h

decch

jnz l2

ret

bcd\_hexendp

exit:mov ah,4ch ;exit procedure

int 21h

end

