***Program:***

include Irvine32.inc

.data

msg1 db "enter the hex value ",0

.code

main proc

call inp\_p

call con\_bin

call crlf

call readchar

exit

main endp

inp\_p proc

mov edx,offset msg1

call writestring

call Readhex

ret

inp\_p endp

con\_bin proc

mov ecx,32

mov edx,0

mov ebx,eax

l1:

rol ebx,1

jnc Show0

Show1:

mov al,'1'

jmp Cont

Show0:

mov al,'0'

Cont:

call writechar

inc edx

cmp edx,4

jnz rest

mov edx,0

mov al,' '

call writechar

rest:

loop l1:

ret

con\_bin endp

end main

:(TASK 02):

***Program:***

include irvine32.inc

.data

msg1 db "enter the binary number ",0

msg2 db "invalid input enter again",0

msg3 db "input taken successfully",0

inp db 32 dup('#')

.code

main proc

call inp\_proc

call con\_hex

call writehex

call readchar

exit

main endp

inp\_proc proc

mov edx,offset msg1

call writestring

mov esi,offset inp

mov ecx,32

mov edx,0

input\_loop:

call readchar

call writechar

cmp al,48

jz ADD0

cmp al,49

jnz goback

ADD1:

mov bl,1

mov [esi],1

inc esi

inc edx

jmp rest

ADD0:

mov bl.0

mov [esi],bl

inc esi

inc edx

jmp rest

goback:

call crlf

mov edx,offset msg2

call writestring

call crlf

jmp input\_proc

rest:

cmp edx,4

jnz dontprintspace

printspace:

mov al,' '

call writechar

mov edx,0

dontprintspace:

loop input\_loop

call crlf

mov edx,offset msg3

call writestring

call readchar

call crlf

ret

inp\_proc endp

con\_hex proc

mov ecx,32

mov eax,0

mov esi,0

l1:

movzx edx,byte ptr[inp+esi]

rcr dl,1

rcl eax,1

inc esi

rest:

loop l1

ret

con\_hex endp

end main

:(TASK 03):

***Program:***

include irvine32.inc

.data

msg1 db "enter the hex number ",0

msg2 db "invalid input enter again",0

msg3 db "decimal equivalent is ",0

inp db ?

.code

main proc

call inp\_proc

call con\_dec

call out\_proc

call readchar

exit

main endp

inp\_proc proc

input\_proc:

mov edx,offset msg1

call writestring

call readchar

mov inp,al

cmp bl,48

jb check2nd

check1st:

cmp bl,57

jbe storecharacter

check2nd:

and bl,11011111b

cmp bl,'A'

jb reinput

cmp bl,'F'

ja reinput

mov al,bl

storecharacter

mov inp,bl

jmp rest

reinput:

call writechar

call crlf

mov edx,offset msg2

call writestring

call crlf

jmp input\_proc

rest:

call writechar

call crlf

ret

inp\_proc endp

con\_dec proc

mov bl,inp

cmp bl,57

ja case2

case1:

cub bl,30h

jmp rest

case2:

sub bl,37h

rest:

ret

con\_dec endp

out\_proc proc

mov edx,offset msg3

call writestring

movzx eax,bl

call writedec

ret

out\_proc endp

end main