**8-Bit Addition**

**Program :**

Data segment

    msg db 0dh,0ah,"Enter first number: $"

    msg1 db 0dh,0ah,"Enter second number: $"

    result db 0dh,0ah,"The Result of addition is: $"

Data ends

Code segment

    assume CS:Code,DS:Data

start:

    mov ax, Data

    mov DS, ax

    mov dx,offset msg ; Display contents of variable msg

    mov ah,09h

    int 21h

    mov ah,01h ; To accept 4bits input and store ASCII value into al

    int 21h

    call AsciiToHex

    mov bl, al

    mov dx,offset msg1 ; Display contents of variable msg

    mov ah,09h

    int 21h

    mov ah,01h ; To accept 4bits input and store ASCII value into al

    int 21h

    call AsciiToHex

    add bl,al;---------main addn

    mov dx,offset result ; Display contents of string result

    mov ah,09h

    int 21h

    mov cl,bl

    and bl,0f0h

    rol bl,4h; interchange nibbles

    mov  al,bl

    call HexToAscii

    mov dl, al

    mov ah,02h; display 10s place

    int 21h

    mov al,cl

    and al,0fh

    call HexToAscii

    mov dl, al; display units place

    mov ah,02h

    int 21h

    mov ah,4ch ; Terminate the program

    int 21h

        AsciiToHex proc

        cmp al,41h ; If it is greater than or equal to 41 then we also need to sub 8h along with 30

        jc skip

        sub al, 07h

        skip: sub al, 30h

        ret

        endp

        HexToAscii proc

        cmp al,0ah ; If it is greater than or equal to 0a then we also need to add 07 along with 30

        jc skip1

        add al,07h

        skip1: add al,30h

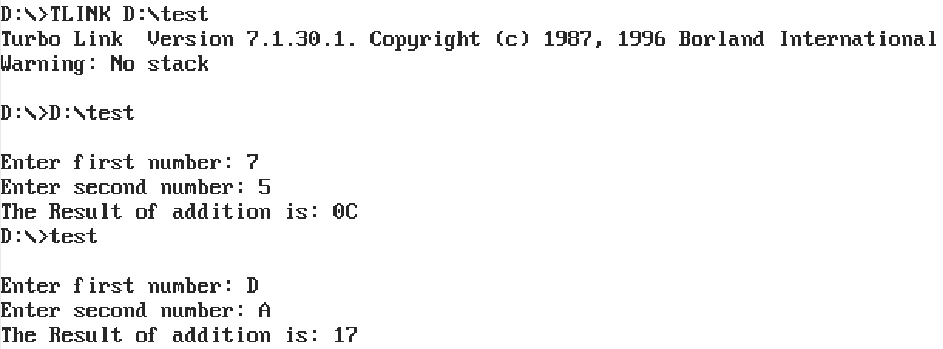
        ret

        endp

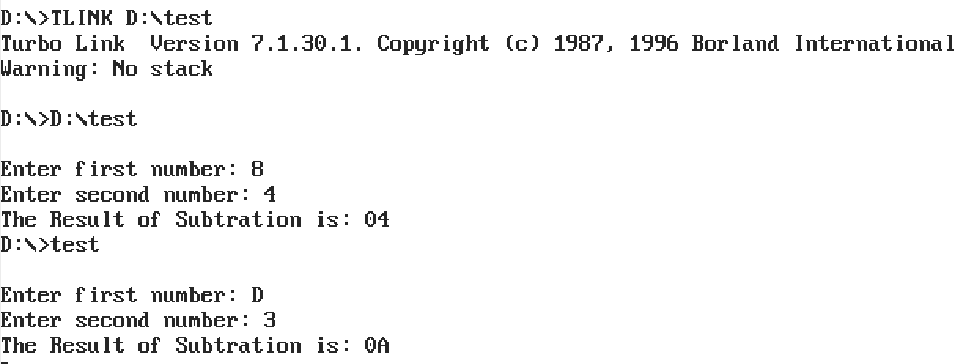
Code ends

end start

**Output :**



**Output :**



**8-Bit Subtraction**

**Program :**

Data segment

    msg db 0dh,0ah,"Enter first number: $"

    msg1 db 0dh,0ah,"Enter second number: $"

    result db 0dh,0ah,"The Result of addition is: $"

Data ends

Code segment

    assume CS:Code,DS:Data

start:

    mov ax, Data

    mov DS, ax

    mov dx,offset msg ; Display contents of variable msg

    mov ah,09h

    int 21h

    mov ah,01h ; To accept 4bits input and store ASCII value into al

    int 21h

    call AsciiToHex

    mov bl, al

    mov dx,offset msg1 ; Display contents of variable msg

    mov ah,09h

    int 21h

    mov ah,01h ; To accept 4bits input and store ASCII value into al

    int 21h

    call AsciiToHex

    sub bl,al;---------main subn

    mov dx,offset result ; Display contents of string result

    mov ah,09h

    int 21h

    mov cl,bl

    and bl,0f0h

    rol bl,4h; interchange nibbles

    mov  al,bl

    call HexToAscii

    mov dl, al

    mov ah,02h; display 10s place

    int 21h

    mov al,cl

    and al,0fh

    call HexToAscii

    mov dl, al; display units place

    mov ah,02h

    int 21h

    mov ah,4ch ; Terminate the program

    int 21h

        AsciiToHex proc

        cmp al,41h ; If it is greater than or equal to 41 then we also need to sub 8h along with 30

        jc skip

        sub al, 07h

        skip: sub al, 30h

        ret

        endp

        HexToAscii proc

        cmp al,0ah ; If it is greater than or equal to 0a then we also need to add 07 along with 30

        jc skip1

        add al,07h

        skip1: add al,30h

        ret

        endp

Code ends

end start

**16-Bit Addition**

**Program :**

Data segment

    msg db 0dh,0ah,"Enter first number: $"

    msg1 db 0dh,0ah,"Enter second number: $"

    result db 0dh,0ah,"The Result of addition is: $"

Data ends

Code segment

    assume CS:Code,DS:Data

start:

    mov ax, Data

    mov DS, ax

    mov dx,offset msg ;----- Display contents of variable msg

    mov ah,09h

    int 21h

    mov ah,01h ; To accept 4bits 10s input

    int 21h

    call AsciiToHex

    mov bl, al

    rol bl,4

    mov ah,01h ; To accept 4bits units input

    int 21h

    call AsciiToHex

    add bl,al

    mov ch,bl

    mov dx,offset msg1 ;----- Display contents of variable msg1

    mov ah,09h

    int 21h

    mov ah,01h ; To accept 4bits 10s input

    int 21h

    call AsciiToHex

    mov bl, al

    rol bl,4

    mov ah,01h ; To accept 4bits units input

    int 21h

    call AsciiToHex

    add bl,al

    add bl,ch;-------------------main addn

    mov dx,offset result ; Display contents of string result

    mov ah,09h

    int 21h

    mov cl,bl

    and bl,0f0h

    rol bl,4h; interchange nibbles

    mov  al,bl

    call HexToAscii

    mov dl, al; display 10s place

    mov ah,02h

    int 21h

    mov al,cl

    and al,0fh

    call HexToAscii

    mov dl, al; display units place

    mov ah,02h

    int 21h

    mov ah,4ch ; Terminate the program

    int 21h

        AsciiToHex proc

        cmp al,41h ; If it is greater than or equal to 41 then we also need to sub 8h along with 30

        jc skip

        sub al, 07h

        skip: sub al, 30h

        ret

        endp

        HexToAscii proc

        cmp al,0ah ; If it is greater than or equal to 0a then we also need to add 07 along with 30

        jc skip1

        add al,07h

        skip1: add al,30h

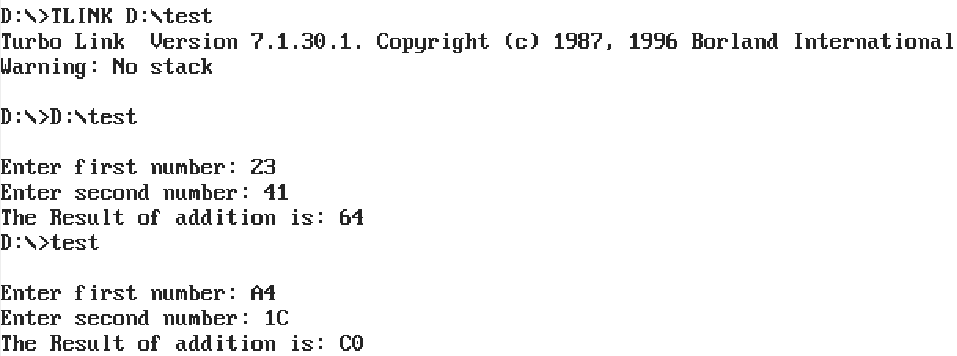
        ret

        endp

Code ends

end start

**Output :**

****

**16-Bit Subtraction**

**Program :**

Data segment

    msg db 0dh,0ah,"Enter first number: $"

    msg1 db 0dh,0ah,"Enter second number: $"

    result db 0dh,0ah,"The Result of addition is: $"

Data ends

Code segment

    assume CS:Code,DS:Data

start:

    mov ax, Data

    mov DS, ax

    mov dx,offset msg ;----- Display contents of variable msg

    mov ah,09h

    int 21h

    mov ah,01h ; To accept 4bits 10s input

    int 21h

    call AsciiToHex

    mov bl, al

    rol bl,4

    mov ah,01h ; To accept 4bits units input

    int 21h

    call AsciiToHex

    add bl,al

    mov ch,bl

    mov dx,offset msg1 ;----- Display contents of variable msg1

    mov ah,09h

    int 21h

    mov ah,01h ; To accept 4bits 10s input

    int 21h

    call AsciiToHex

    mov bl, al

    rol bl,4

    mov ah,01h ; To accept 4bits units input

    int 21h

    call AsciiToHex

    add bl,al

    call AsciiToHex

    sub ch,bl;-------------------main subn

    mov dx,offset result ; Display contents of string result

    mov ah,09h

    int 21h

    mov cl,bl

    and bl,0f0h

    rol bl,4h; interchange nibbles

    mov  al,bl

    call HexToAscii

    mov dl, al; display 10s place

    mov ah,02h

    int 21h

    mov al,cl

    and al,0fh

    call HexToAscii

    mov dl, al; display units place

    mov ah,02h

    int 21h

    mov ah,4ch ; Terminate the program

    int 21h

        AsciiToHex proc

        cmp al,41h ; If it is greater than or equal to 41 then we also need to sub 8h along with 30

        jc skip

        sub al, 07h

        skip: sub al, 30h

        ret

        endp

        HexToAscii proc

        cmp al,0ah ; If it is greater than or equal to 0a then we also need to add 07 along with 30

        jc skip1

        add al,07h

        skip1: add al,30h

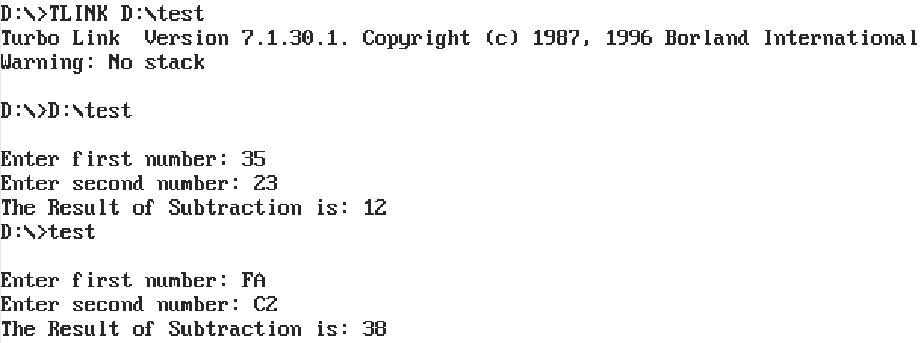
        ret

        endp

Code ends

end start

**Output :**

****