**Program :**

Data segment

    msg  db 10,13,"  Sorted array in descending order : $"

    msg1  db 10,13,"  $"

    arr   db 25h, 10h, 45h, 30h, 50h, 75h

Data ends

Code segment

           assume DS:Data,CS:Code

    start:

           mov    ax,Data

           mov    ds,ax

       mov    cx,5

    outerLoop:

           lea    SI,arr

           mov    bl,[SI]

           inc    SI

           mov    bh,5

    innerLoop:

           dec    bh

           cmp    bl,[SI]

           jnc    skipSwap

           mov    dl,bl

           mov    bl,[SI]

           dec    SI

           mov    [SI],bl

           inc    SI

           mov    [SI],dl

    skipSwap:

           mov    bl,[SI]

           inc    SI

           cmp    bh,00h

           jnz    innerLoop

           loop   outerLoop

       lea    SI,arr

       mov    cx,6

       lea    dx,msg

       mov    ah,09h

       int    21h

    output:

           lea    dx,msg1

           mov    ah,09h

           int    21h

           mov    bl,00h

           mov    bl,[SI]

           and    bl,0f0h

           ror    bl,04h

           call   hexToAscii

           mov    bl,[SI]

           and    bl,0fh

           call   hexToAscii

           inc    SI

           loop   output

       mov    ah,4ch

       int    21h

hexToAscii proc

           cmp    bl,0ah

           jc     skip2

           add    bl,07h

    skip2: add    bl,30h

           mov    dl,bl

           mov    ah,02h

           int    21h

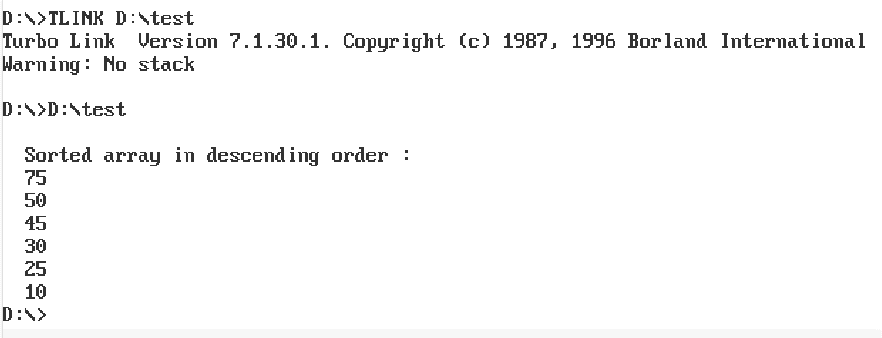
           ret

           endp

Code ends

end start

**Output :**

****

**Program :**

Data segment

    msg db 0dh,0ah,"Minimum element of array is: $"

    arr db 20h, 50h, 30h, 10h, 40h

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 cx, 04h

    lea SI, arr

    mov bl,[SI]

    inc SI

    iterate :

        cmp bl, [SI]

        jc skip

        mov bl, [SI]

        skip :

        inc SI

        loop iterate

    mov al,bl

    and al,0F0h

    ror al,4h

    call HexToAscii

    mov al,bl

    and al,0Fh

    call HexToAscii

    mov ah,4ch ; Terminate the program

    int 21h

        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

        mov dl, al; display

        mov ah,02h

        int 21h

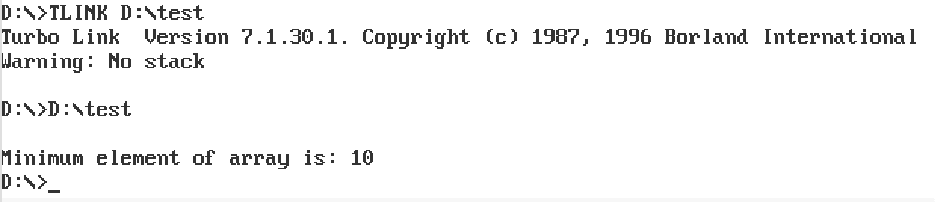
        ret

        endp

Code ends

end start

**Output :**



**Program :**

#include<iostream.h>

#include<stdlib.h>

int main(){

int size = 0;

 while(size != 4){

    cout<<"-----CURSOR-----"<<endl;

    cout<<"1. Increment\n2. Decrement\n3. Disable\n4. Exit"<<endl;

    cout<<"Enter choice: ";

    int choice;

    cin>>choice;

    switch(choice){

    case 1:{

        size ++;

        asm{

            mov cx, size;

            mov ah, 01h;

            int 10h;

        }

        break;

    }

    case 2:{

        size --;

        asm{

            mov cx, size;

            mov ah, 01h;

            int 10h;

        }

        break;

    }

    case 3:{

        asm{

            mov cl, 20h

            mov ah, 01h;

            int 10h;

        }

        break;

    }

    case 4: return 0;

    }

 }

 return 0;

}

**Output :**

