19/10/20 Lab 5

Q-> Program for sorting array using bubble sort method in ascending order.

```
.model small
.data
        n   db  5
        a   db  05, 07, 04, 03, 06
.code
        mov   ax, @data
        mov   ds , ax
        mov   cl, n
        dec   cl
outloop :  mov  ch, cl
           mov  si , 00h
    inloop :  mov  al, a[si]
              inc  si
              cmp  al, a [0]
              jc noexch
              xchg  al, a[si]
              mov  a[si-1], al
    noexch :  dec  ch
              jnz  inloop
              dec  cl
              jnz  outloop
              mov  ah, 4ch
              int  21h
     End
```

Q→ Program to sort an array and its values options are accepted from user.

```
.model small
display macro msg
        lea dx, msg
        mov ah, 09h
        int 21h
endm
.data
list db 02h, 01h, 34h, 0f4h, 09h, 05h
number equ $-list
msg1 db 0dh, 0ah, "1> ascending $"
msg2 db 0dh, 0ah, "2> descending $"
msg3 db 0dh, 0ah, "3> exit $"
msg4 db 0dh, 0ah, "Enter choice $"
msg5 db 0dh, 0ah, "Invalid choice $"
.code
    start: mov ax, @data
           mov ds, ax
           lea si, list
           mov ch, number - 1
           display msg1
           display msg2
           display msg3
           display msg4
        mov ah, 01h
        int 21h
        sub alh, 30h
        cmp al, 01h
        je ascsort
        cmp al, 02h
        je dessort
        cmp al, 03h
        je final
        display msg5
        jmp final
ascsort: mov bl, 00h
         mov si, offset list
         mov cl, 00h
         mov bh, ch
         sub bh, bl
npass: cmp cl, bh
         jnc next
         mov al, [si]
         mov bp, 01h
         cmp al, ds:[bp][si]
         jc nope
         xchg al, [si+1]
         xchg si, al
nope : inc cl
```

```
            inc si
            jmp nxpass
next :      inc bl
            cmp bl, ch
            jc again
            jmp final
dessort :   mov bl, 00h
again1 :    mov si, offset list
            mov cl, 00h
            mov bh, ch
            sub bh, bl
nxpass1 :   cmp cl, bh
            jnc next
            mov al, [si]
            mov bp, 01h
            cmp al, ds:[bp][si]
            jnc nope1
            xchg al, [si+1]
            xchg [si], al
nope1 :     inc cl
            inc si
            jmp nxpass1
next1 :     inc bl
            cmp bl, ch
            jc again1
final :     mov ah, 4ch
            int 21h
end start
```