**Question 1**

**Code:**

[org 0x0100]

mov ax,0

mov cx,0

mov dx,0

mov bx,0

mov cx,6

outerloop:

mov bx,0

innerloop:

mov ax,[rollno +bx ]

cmp ax, [rollno + bx+2]

jbe noswap

mov dx , [rollno + bx+2]

mov [rollno + bx+2],ax

mov [rollno + bx],dx

noswap:

add bx,2

cmp bx,10

jne innerloop

sub cx,1

jnz outerloop

xor cx,cx

mov bx,[rollno+2] ; to show the second smallest

mov cx,[rollno+8] ; to show the second larger

mov dx,[rollno+10] ; to show the larger

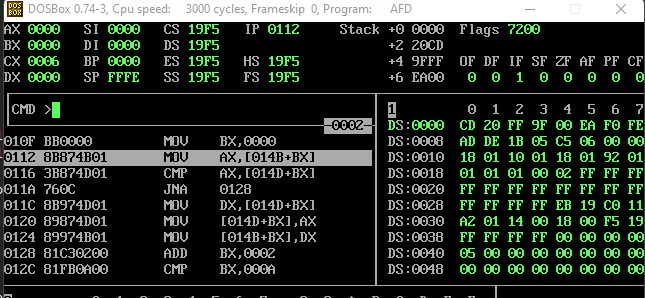
mov ax,0x4c00

int 0x21

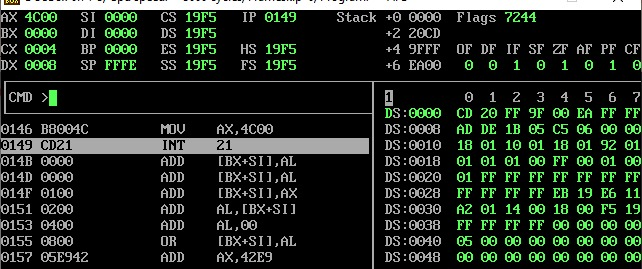
rollno: dw 2,1,8,0,0,4

**Screenshot:**

**Starting:**



**Ending**



**Question 2**

**Code**

[org 0x0100]

mov ax,0 ; intialize all to 0.

mov cx,0

mov dx,0

mov bx,0

mov cx,6 ; inialize for counter /loop

outerloop:

mov bx,10

innerloop:

mov ax,[rollno +bx ]

cmp ax, [rollno + bx-2]

jbe noswap

mov dx , [rollno + bx-2]

mov [rollno + bx-2],ax

mov [rollno + bx],dx

noswap:

sub bx,2

cmp bx,0

jne innerloop

sub cx,1

jnz outerloop

xor cx,cx

mov ax,0

mov cx,6

mov bx,0

sort:

mov ax,[rollno + bx]

mov [descend +bx ],ax

add bx,2

sub cx,1

jnz sort

mov ax,[descend]

mov ax,0x4c00

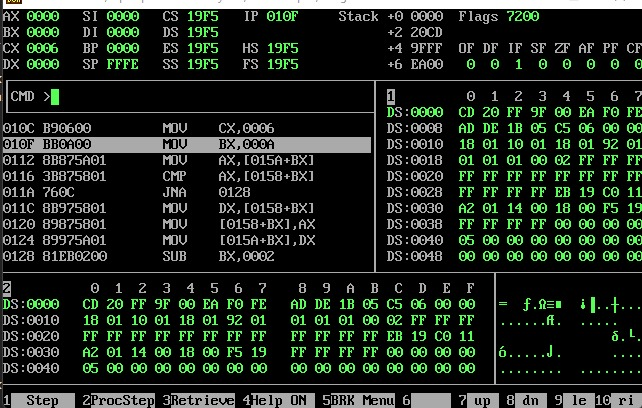
int 0x21 ;exit

rollno: dw 2,1,8,0,0,4

descend : dw 0,0,0,0,0,0

**Screenshot**

**Starting**



**Ending**

