**Exercise-1**

**AIM:**

1.a.Write Assembly language program to print the numbers from 0 to 9.

**Program:**

org 100h

.data

a db 0,1,2,3,4,5,6,7,8,9

.code

mov cx,10

mov si,0

loop1:

mov al,a[si]

mov dl,al

add dl,48

mov ah,2h

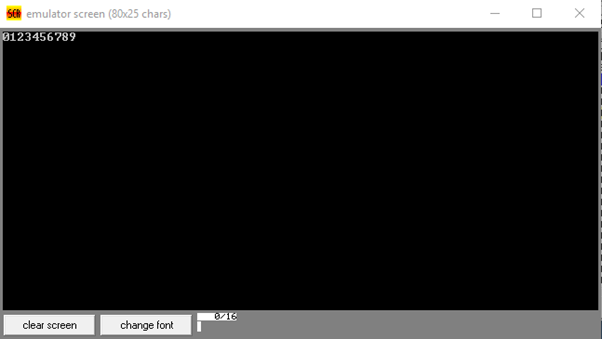
int 21h

INC si

loop loop1

Hlt

**Outcome:**



**AIM:**

1.b.Write Assembly language programs to find average of numbers stored in an array.

**Program:**

org 100h

.data

a db 1,2,3,4,5

avg db ?

.code

mov ax,@data

mov ds,ax

mov cx,5

mov si,0

mov bl,5

loop1:

ADD al,a[si]

inc si

loop loop1

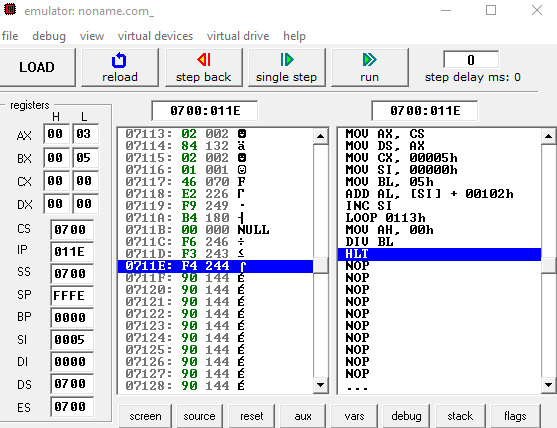
mov ah,0

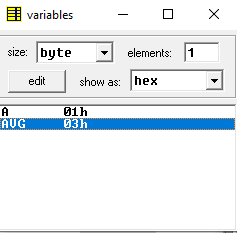
div bl

mov avg,al

Hlt

**Outcome:**





**Exercise-2**

**AIM:**

2.a. Write Assembly language programs to find the largest number in an array

**Program:**

org 100h

.data

a db 1,9,7,6

.code

mov ax,@data

mov ds,ax

mov cx,4

mov si,0

mov bh,a[si]

loop1:

INC si

cmp bh,a[si]

Jg l1

mov bh,a[si]

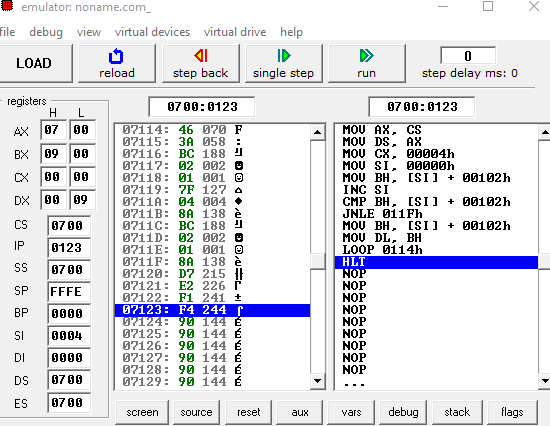
l1:

mov dl,bh

loop loop1

hlt

**Outcome:**



**AIM:**

2.b. Write Assembly language programs to sort the numbers in ascending order.

**Program:**

org 100h

.data

a db 9,8,7,6,5,4,3,2,1

.code

mov ax,@data

mov ds,ax

mov cx,9

mov si,0

bubblesort:

cmp cx,si

jz next

mov al,a[si]

mov bl,a[si+1]

cmp al,bl

ja exchange

add si,1

jmp bubblesort

exchange:

mov a[si],bl

mov a[si+1],al

add si,1

jmp bubblesort

next:

mov si,00

sub cx,1

cmp cx,0

jnz bubblesort

mov cx,9

mov si,0

loop1:

mov dl,a[si]

add dl,48

mov ah,2h

int 21h

inc si

loop loop1

**Outcome:**



**Exercise-3**

**AIM:**

3.a. Write Assembly language programs to find L.C.M of two numbers.

**Program:**

org 100h

.data

a db 9

b db 4

lcm db ?

.code

mov ax,@data

mov ds,ax

mov ah,0

mov al,a

mov bl,b

mov bh,al

div bl

cmp ah,0

je exit

jne l1

l1:

mov ah,0

mov al,bh

add al,a

mov bh,al

div bl

cmp ah,0

je exit

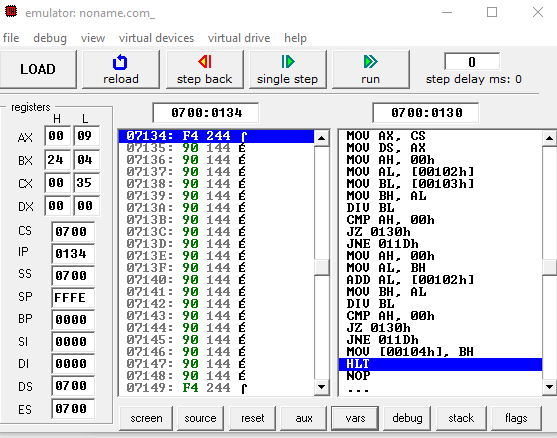
jne l1

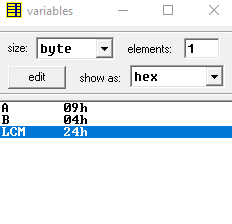
exit:

mov lcm,bh

hlt

**Outcome:**





**AIM:**

3.b. Write Assembly language programs to find G.C.D of two numbers.

**Program:**

org 100h

.data

a db 8

b db 4

hcf db ?

.code

mov ax,@data

mov ds,ax

mov al,a

mov bl,b

mov ah,0

div bl

cmp ah,0

je exit

jne l1

l1:

mov ah,0

mov al,a

dec bl

div bl

cmp ah,0

je exit

jne l1

exit:

mov hcf,bl

Hlt

**Outcome:**

