Name: **Tushar Nankani** Roll No: **1902112** Batch: **C23**

**Microprocessor: Experiment 6**

**Aim**: Assembly program to find minimum and maximum of a given array.

**Theory:**

A naive solution is to compare each array element for minimum and maximum elements by considering a single item at a time. The time complexity of this solution would be linear.

Another solution can be to first sort the array and pick out the first element and the last element which will be the minimum element and maximum elements respectively.

**ALGORITHM:**

**For max**

1.Start2.Initialize data segment through AX register in the DS register.

3.Initialize the SI to 2000h

4.Initialize total elements of array as a count in CX(e.g 0005h)

5.Preserve the above count in c temporary variable.

6.Display the message as “Enter an array elements”

7.Read first digit in AL register through keyboard(e.g. AL=31h)

8.Call Input procedure to make anumber from ASCII hexadecimal to a normal hexadecimal number.AL=01h

9.Move AL contents to BL

10.Rotate BL contents by 4 in left direction.

11.Read second digit in AL register through keyboard (e.g AL=32h)

12.Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number.AL=02h

13.Add BL and AL contents (BL<-BL+AL)

14.Store the BL (current accepted number) to location pointed by SI

15.Increment SI by 1 to point to next location for the next number

16.Repeat step no. 7 to 15 till CX count reaches to 0.

17.Initialize SI again to 2000h and CX also with total number of elements.

18.Initialize AL with first element pointed by SI for the next comparison

Lab ManualMicroprocessorMs.Urvi Kore27

19.Compare number pointed by SI from an array with AL register

20.If carry is not generated (i.e. if number in AL < number pointed by SI) then goto step no. 22 else goto step no. 21

21.Make a unconditional jump to step no. 23

22.Move number pointed by SI to AL

23.Incremented SI by 1

24.Decrement CX by 1

25.Compare CX with 0000h (i.e. Repeat step no.19 to 25 till all numbers of array are not covered for the comparison)

26.If Zero flag is not set then jump to step no.19

27.Finally maximum number will be available in AL register.

28.Display the contents of AL register.

29.Stop.

**For min**

1.Start

2.Initialize data segment through AX register in the DS register.

3.Initialize the SI to 5000h

4.Initialize total elements of array as a count in CX(e.g 0005h)

5.Preserve the above count in c temporary variable.

6.Display the message as “Enter an array elements”

7.Read first digit in AL register through keyboard(e.g. AL=31h)

8.Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimalnumber.AL=01h

9.Move AL contents to BL

10.Rotate BL contents by 4 in left direction.

11.Read second digit in AL register through keyboard (e.g AL=32h)

12.Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number.AL=02h

13.Add BL and ALcontents (BL<-BL+AL)

14.Store the BL (current accepted number) to location pointed by SI

15.Increment SI by 1 to point to next location for the next number

16.Repeat step no. 7 to 15 till CX count reaches to 0.

17.Initialize SI again to 5000h and CX also with total number of elements.

18.Initialize AL with first element pointed by SI for the next comparison

19.Compare number pointed by SI from an array with AL register

20.If carry is generated (i.e. if number in AL > number pointed by SI) then goto step no. 22 else goto step no. 21

21.Make a unconditional jump to step no. 23

22.Move number pointed by SI to AL

23.Incremented SI by 1

24.Decrement CX by 1

25.Compare CX with 0000h (i.e. Repeat step no.19 to 25 till all numbers of array are not covered for the comparison)

26.If Zero flag isnot set then jump to step no.19

27.Finally minimum number will be available in AL register.

28.Display the contents of AL register.

29.Stop.

Code:

1. For max

Data Segment

msg db 0dh,0ah,"Please enter the length of the array: $"

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

newl db 0dh,0ah," $"

res db 0dh,0ah,"The maximum is: $"

len db ?

max db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov ax,Data

mov DS,ax

mov dx,offset msg

mov ah,09h

int 21h

call Accept

mov len,bl

mov cl,bl

mov ch,00h

mov di,1000h

back: mov dx,offset msg1

mov ah,09h

int 21h

call Accept

mov [di],bl

inc di

loop back

mov di,1000h

mov cl,len

mov ch,00h

mov dx,offset newl

mov ah,09h

int 21h

mov al,[di]

mov max,al

chk: mov bl,max

mov al,[di]

cmp bl,al

jnc a

mov max,al

jmp b

a: mov max,bl

b: inc di

loop chk

mov dx,offset res

mov ah,09h

int 21h

mov bl,max

call DispNum

mov ah,4ch

int 21h

Accept proc

mov ah,01h

int 21h

call AsciiToHex

rol al,4

mov bl,al

mov ah,01h

int 21h

call AsciiToHex

add bl,al

ret

endp

DispNum proc

mov dl,bl

and dl,0f0h

ror dl,4

call HexToAscii

mov ah,02h

int 21h

mov dl,bl

and dl,0fh

call HexToAscii

mov ah,02h

int 21h

endp

AsciiToHex proc

cmp al,41h

jc sk

sub al,07h

sk: sub al,30h

ret

endp

HexToAscii proc

cmp dl,0ah

jc sk2

add dl,07h

sk2: add dl,30h

ret

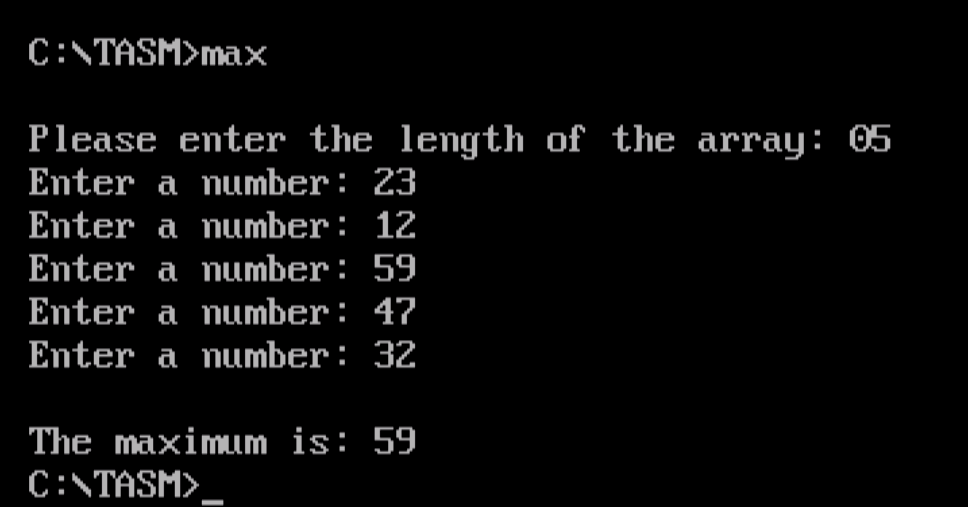
endp

Code ends

end Start

OUTPUT:

1. For max



2. For min

Data Segment

msg db 0dh,0ah,"Please enter the length of the array: $"

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

newl db 0dh,0ah," $"

res db 0dh,0ah,"The minimum is: $"

len db ?

min db ?

Data ends

Code Segment

assume CS:Code,DS:Data

Start:

mov ax,Data

mov DS,ax

mov dx,offset msg

mov ah,09h

int 21h

call Accept

mov len,bl

mov cl,bl

mov ch,00h

mov di,1000h

back: mov dx,offset msg1

mov ah,09h

int 21h

call Accept

mov [di],bl

inc di

loop back

mov di,1000h

mov cl,len

mov ch,00h

mov dx,offset newl

mov ah,09h

int 21h

mov al,[di]

mov min,al

chk: mov bl,min

mov al,[di]

cmp bl,al

jc a

mov min,al

jmp b

a: mov min,bl

b: inc di

loop chk

mov dx,offset res

mov ah,09h

int 21h

mov bl,min

call DispNum

mov ah,4ch

int 21h

Accept proc

mov ah,01h

int 21h

call AsciiToHex

rol al,4

mov bl,al

mov ah,01h

int 21h

call AsciiToHex

add bl,al

ret

endp

DispNum proc

mov dl,bl

and dl,0f0h

ror dl,4

call HexToAscii

mov ah,02h

int 21h

mov dl,bl

and dl,0fh

call HexToAscii

mov ah,02h

int 21h

endp

AsciiToHex proc

cmp al,41h

jc sk

sub al,07h

sk: sub al,30h

ret

endp

HexToAscii proc

cmp dl,0ah

jc sk2

add dl,07h

sk2: add dl,30h

ret

endp

Code ends

end Start

**OUTPUT:**

For min

