**DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING**

**COLLEGE OF E&ME, NUST, RAWALPINDI**

**Microprocessor and Microcontroller Based Design**

**Lab 07**

**SUBMITTED TO:**

**Dr Taimoor Zahid**

**SUBMITTED BY:**

**AMINA QADEER**

**Reg # 0000359607**

**DE-42 (C&SE)-A**

**Submission Date: 08/11/2022**

**Objectives:**

In this lab, you will be introduced to memory segmentation, and difference between physical and logical addresses of the memory. You will also deal with the different types of addressing modes and learn how to calculate the physical addresses from logical addresses. You will also write programs that will make use of loops and jump statements.

**Related Topic/Chapter in theory class:**

None

**Hardware/Software required:**

Hardware: PC

Software Tool: emu8086 v2.57

TASK 1: **Write and ALP to find the minimum number in an array  
• We have an array of 10 numbers. So we initialize the counter with 10. Also we  
initialize a pointer to point these numbers.  
• Compare first number with initial number i.e. zero. If number < minimum  
number, save number otherwise increment pointer to compare next number.  
Decrement counter, compare till all the numbers are compared. Store and  
display the minimum number in data register.**

**Solution:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

.data

array1 db 33,11,3,6,4,7,8,5,3,1

mov si,00H

.code

main proc

mov ax,@data ;initialize data segment

mov ds,ax

;access first array element

mov si,offset array1

mov cx,10

mov bl,[si]

loopA:

cmp [si],bl

jle small

;jge large

compare:

inc si

loop loopA

;display smallest number

add bl,48

mov dl,bl

mov ah,2

int 21h

jmp end

small:

mov bl,[si]

jmp compare

inc si

;large:

;mov bl,[si]

;jmp compare

;display largest number

;add bl,48

;mov dl,bl

;mov ah,2

;int 21h

end:

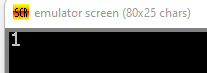
mov ah,4ch

int 21h

main endp

ret

**Output:**

****

TASK 2: **Write and ALP to find the maximum number in an array  
• We have an array of 10 numbers. So we initialize the counter with 10. Also, we  
initialize a pointer to point these numbers.  
• Compare first number with initial maximum number i.e. zero. If number >  
maximum number, save number otherwise increment pointer to compare next  
number. Decrement counter, compare till all the numbers are compared. Store  
the maximum number in MAX. Display the maximum number.**

**Solution:**

org 100h

.data

array1 db 9,9,3,6,4,7,8,5,2,1

mov si,00H

.code

main proc

mov ax,@data ;initialize data segment

mov ds,ax

;access first array element

mov si,offset array1

mov cx,10

mov bl,[si]

loopA:

cmp [si],bl

;jle small

jge large

compare:

inc si

loop loopA

;display largest number

add bl,48

mov dl,bl

mov ah,2

int 21h

jmp end

;small:

;mov bl,[si]

;jmp compare

;inc si

large:

mov bl,[si]

jmp compare

end:

mov ah,4ch

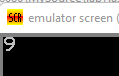
int 21h

main endp

ret

ret

**Output:**

****

TASK 3: **Write and ALP to sort the numbers in ascending order**

**Solution:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

;include 'emu8086 .inc'

org 100h

.model small

.DATA

arr db 5 dup(?)

msg1 db 10,13, "Enter 5 number in array:$"

msg2 db 10,13, "after sorting array $"

.code

main proc

mov ax,@data

mov ds,ax

lea dx,msg1

mov ah,9

int 21h

mov cx,5

mov bx,offset arr

mov ah,1

input:

int 21h

mov [bx],al

inc bx

loop input

mov cx,5

dec cx

outerloop:

mov bx,cx

mov si,0

comploop:

mov al, arr[si]

mov dl,arr[si+1]

cmp al,dl

jc noswap

mov arr[si],dl

mov arr[si+1],al

noswap:

inc si

dec bx

jnz comploop

loop outerloop

mov ah ,2

mov dl,10

int 21h

mov dl,13

int 21h

lea dx,msg2

mov ah,9

int 21h

mov cx,5

mov bx,offset arr

;this loop to display elements on the screen

outputs:

mov dl,[bx]

mov ah,2

int 21h

mov dl,32

mov ah,2

int 21h

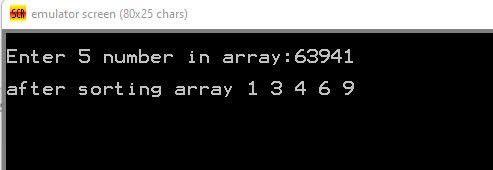
inc bx

loop outputs

main endp

ret

**Output:**

****

TASK 4: **Write and ALP to sort the numbers in descending order**

**Solution:**

org 100h

.model small

.DATA

arr db 5 dup(?)

msg1 db 10,13, "Enter 5 number in array:$"

msg2 db 10,13, "after sorting array $"

.code

main proc

mov ax,@data

mov ds,ax

lea dx,msg1

mov ah,9

int 21h

mov cx,5

mov bx,offset arr

mov ah,1

input:

int 21h

mov [bx],al

inc bx

loop input

mov cx,5

dec cx

outerloop:

mov bx,cx

mov si,0

comploop:

mov al, arr[si]

mov dl,arr[si+1]

cmp al,dl

jnc noswap

mov arr[si],dl

mov arr[si+1],al

noswap:

inc si

dec bx

jnz comploop

loop outerloop

mov ah ,2

mov dl,10

int 21h

mov dl,13

int 21h

lea dx,msg2

mov ah,9

int 21h

mov cx,5

mov bx,offset arr

;this loop to display elements on the screen

outputs:

mov dl,[bx]

mov ah,2

int 21h

mov dl,32

mov ah,2

int 21h

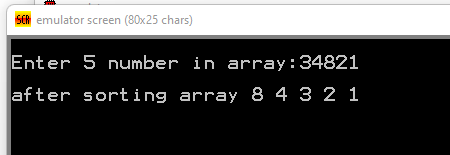
inc bx

loop outputs

main endp

ret

**Output:**

****

**Conclusion:**

In this lab, we had an introduction to Arrays and String Operation. Also, we learnt to deal with the different types of addressing modes and will also write programs that will make use of loops and jump statements. Designed algorithms for finding maximin numbers from array and sorting in ascending and descending order.