

"Heaven's Light is Our Guide"



Rajshahi University of Engineering & Technology
Department of Computer Science & Engineering

Lab Report

Course Code: CSE 2206

**Course Title: Microprocessors, Microcontrollers and
Assembly Language Sessional**

Experiment No: 09

Experiment Name: Implementation of array and addressing mode.

<p><u>Submitted By-</u></p> <p>Name : Sajidur Rahman Tarafder</p> <p>Department : CSE</p> <p>Roll No : 2003154</p> <p>Section : C</p> <p>Session:2020-21</p>	<p><u>Submitted To-</u></p> <p>Mohiuddin Ahmed</p> <p>Lecturer</p> <p>Department of CSE,</p> <p>Rajshahi University of</p> <p>Engineering and Technology,</p> <p>Rajshahi</p>
--	--

Task-1: Find the minimum and maximum element of a given array.

Source Code:

```
.model small
.stack 100h

.data
arr db 7,8,9,2,3,6,5,4,1
min db ?
max db ?
str1 db "Max Element: $"
str2 db 0ah,0dh,"Min Element: $"

.code
main proc
    mov ax,@data
    mov ds,ax

    lea si,arr
    mov cx,9
    xor ax,ax
    xor bx,bx
    mov al,[arr]
    mov bl,[arr]

    _Loop:

    cmp [si],al
    jg _Max

    cmp [si],bl
    jl _Min

    jmp continue

    _Max:

    mov al,[si]
    jmp Continue

    _Min:

    mov bl,[si]

    Continue:

    add si,1
    loop _Loop

    mov max,al
    mov min,bl

    lea dx,str1
    mov ah,9
    int 21h

    mov dl,max
    add dl,48
    mov ah,2
    int 21h
```

```
    lea dx, str2
    mov ah, 9
    int 21h

    mov dl, min
    add dl, 48
    mov ah, 2
    int 21h

    mov ax, 4c00h
    int 21h
main endp
end main
```

Output:



```
emulator screen (80x25 chars)
Max Element: 9
Min Element: 1
```

Discussion:

This assembly code aims to find the maximum and minimum elements in an array. It initializes variables for max and min, then iterates through the array elements, comparing each element to update the max and min values accordingly. After finding both values, it displays them using DOS interrupts. The code uses a simple linear comparison technique to determine the max and min elements, storing them in memory and outputting the results.

Task-2: Update all the elements of row 3 and column 3 of a given 2D array to zero(0) and display the array.

Source Code:

```
.model small
.stack 100h

.data
arr db 1,1,1,1,1
    db 1,1,1,1,1
    db 1,1,1,1,1
    db 1,1,1,1,1
    db 1,1,1,1,1
nl db 0ah,0dh,24h

.code
main proc
    mov ax, @data
    mov ds, ax

    mov bx, 0
    mov si, 2
    mov cx, 5

    _Loop1:
    mov [arr+bx+si], 0
    add bx, 5
    loop _Loop1

    xor si, si
    xor bx, bx

    mov si, 0
    mov bx, 10
    mov cx, 5

    _Loop2:
    mov [arr+bx+si], 0
    add si, 1
    loop _Loop2

    mov cx, 25
    xor si, si
    mov bl, 5
    lea si, arr

    _Loop3:
    mov ax, cx
    div bl

    cmp ah, 0
    je newline
    jmp print
```

```

newline:
    lea dx,nl
    mov ah,9
    int 21h

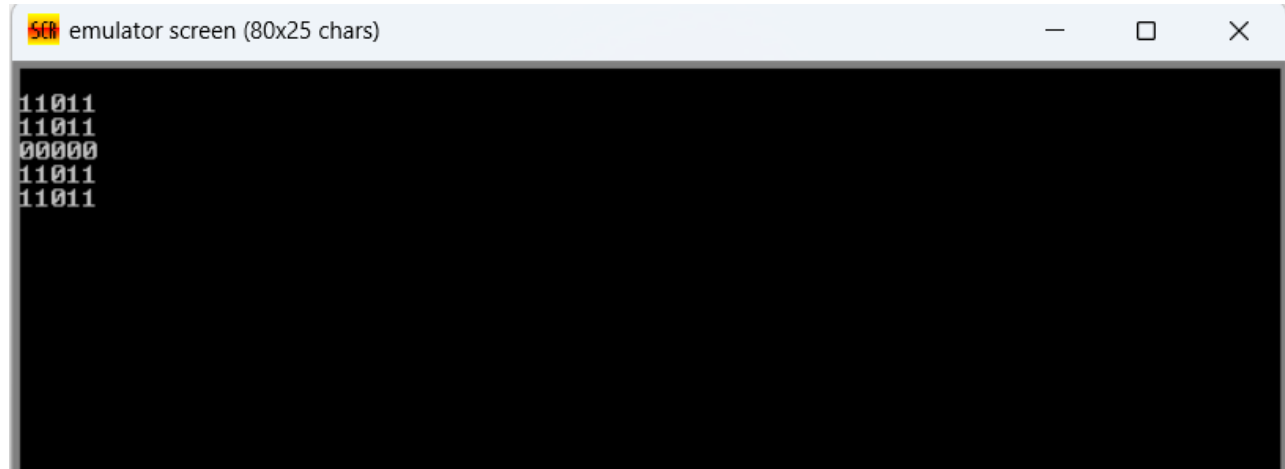
print:
    mov dl,[si]
    add dl,48
    mov ah,2
    int 21h
    add si,1

    loop _Loop3

    mov ax,4c00h
    int 21h
main endp
end main

```

Output:



```

11011
11011
00000
11011
11011

```

Discussion:

This assembly code initializes a 5x5 array with '1's, then sets every third element in each row and column to '0'. It utilizes loops and conditional checks to achieve this. Afterwards, it prints the resulting array in a grid format, converting the values to ASCII characters for display. Here based-indexed addressing mode is used to iterate through the array elements. The code uses nested loops to iterate through the array elements, checking for row and column boundaries, and prints the array with newline characters after each row. Overall, it's a program that modifies and displays a matrix-like structure in assembly language.