National University of Computer and Emerging Sciences

COAL Lab Midterm

Computer Organization and Assembly Language

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Time Allowed** | 90 Minutes | **Student Name** | |  |
| **Maximum Marks** | 100 | **Roll Number** | |  |
| **Lab Instructors** | Hazoor | | **Date** |  |

**Before you start make sure:**

1. **Fill word Format on GCR & submit PDF [Only PDF formats accepted]**
2. **PLAGIARISM WILL BE MARKED ZERO WITH NO RETAKE**

# **Activity 1: [20 Marks]**

Initialize a memory array with last 4 digits of **Your Own Roll Number** (for example, if your roll number is **16L-4195** then memory array should be initialized with **{4,1,9,5}**).

Then write a subroutine **LoadMN** which stores the matrices and using the values as described in the Code section below:

# **Activity 2: [50 Marks]**

Use the subroutine from **Activity 1**, to initialize matrices **M** and **N.**

**Theory**:

If and are two matrices of order as shown below

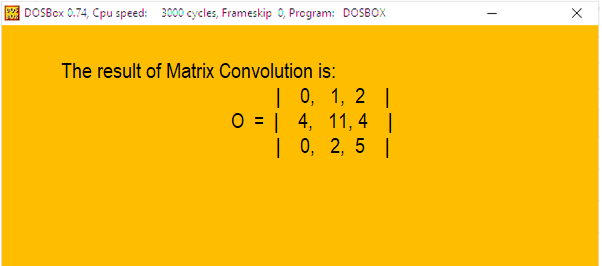
**Convolution** of two matrices can be calculated by padding first matrix and scrolling the second matrix on it across the rows and columns of A and performing dot product:

As shown above, convolution of M and N is the matrix O.

Write a subroutine, **MatConv,** which performs 2x2 matrix **convolution** using the values of M and N initialized in the **Activity 1.**

# **Activity 3: [30 Marks]**

Make a subroutine for printing the output matrix from **Activity 2** on the console as shown below



# **Code:**

**Your final program structure should look like:**

[org 0x100]

roll: db **4,1,9,5 ;** Change to last 4 digits of your roll number

a: db 0 ; store roll

b: db 0 ; store roll+1

c: db 0 ; store roll+2

d: db 0 ; store roll+3

; rows of M matrix of order 4X4

Mr1: db 0, 0, 0, 0 ;

Mr2: db 0, 0, 0, 0 ; store 0, a, b, 0

Mr3: db 0, 0, 0, 0 ; store 0, c, d, 0

Mr4: db 0, 0, 0, 0 ;

; rows of N matrix of order 2X2

Nr1: db 0, 0 ; store d, c

Nr2: db 0, 0 ; store b, a

; rows of O matrix of order 3X3

Or1: dw 0, 0, 0

Or2: dw 0, 0, 0

Or3: dw 0, 0, 0

**LoadMN**:

; Write code for LoadMN

**MatConv**:

; Write code for Matrix Convolution

**PrintMat**:

; Write code for Printing Matrix

**start**:

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

; Write Calls to the subroutines & other codes here

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

mov ax, 0x4c00

int 0x21