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** | 24th Oct 2022 |

**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 vectors , and , using the values as described in the comment section below:

[org 0x100]

jmp start

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

; Mr1, Mr2 vectors

Mr1: db 0, 0 ; store a, b

Mr2: db 0, 0 ; store c, d

; Nr1, Nr2 vectors

Nr1: db 0, 0 ; store d, c

Nr2: db 0, 0 ; store b, a

; Or1, Or2 vectors

Or1: dw 0, 0, 0, 0

Or2: dw 0, 0, 0, 0

**LoadMN**:

\\ Write code for **LoadMN**

start:

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

; Write Calls to the subroutines & other codes here

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

mov ax, 0x4c00

int 0x21

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

Use the subroutine from **Activity 1**, to initialize vectors **Mr1, Mc1** and **Nr1, Nr2.**

**Theory**:

If and are two vectors

**Dot product** of two vectors and can be calculated as:

i.e. the product of corresponding elements and then sum both products.

If and are two vectors

**Cross product** of two vectors and can be calculated as:

i.e. the product of opposite elements and then sum both products.

**Write subroutines which perform dot product and cross product using the code below:**

[org 0x100]

jmp start

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

; Mr1, Mr2 vectors

Mr1: db 0, 0 ; store a, b

Mr2: db 0, 0 ; store c, d

; Nr1, Nr2 vectors

Nr1: db 0, 0 ; store d, c

Nr2: db 0, 0 ; store b, a

; Or1, Or2 vectors

Or1: dw 0, 0, 0, 0

Or2: dw 0, 0, 0, 0

**LoadMN**:

\\ Write code for **LoadMN**

**DotPro**:

; Write code for Dot Product of Mr1 and Nr1 vectors

**CrossPro**:

; Write code for Cross Product of Mr2 and Nr2 vectors

start:

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

; Write Calls to the subroutines & other codes here

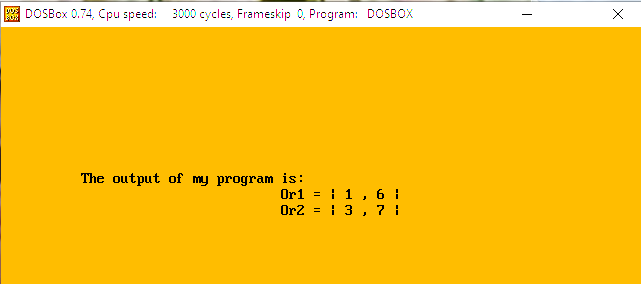
;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

mov ax, 0x4c00

int 0x21

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

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



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

[org 0x100]

jmp start

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

; Mr1, Mr2 vectors

Mr1: db 0, 0 ; store a, b

Mr2: db 0, 0 ; store c, d

; Nr1, Nr2 vectors

Nr1: db 0, 0 ; store d, c

Nr2: db 0, 0 ; store b, a

; Or1, Or2 vectors

Or1: dw 0, 0, 0, 0

Or2: dw 0, 0, 0, 0

**LoadMN**:

\\ Write code for **LoadMN**

**MatMul**:

; Write code for Dot and Cross

**PrintMat**:

; Write code for Printing Vector

start:

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

; Write Calls to the subroutines & other codes here

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

mov ax, 0x4c00

int 0x21