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** | 26th 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 matrices 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

; rows of M matrix of order 2X2

Mr1: db 0, 0 ; store a, b

Mr2: db 0, 0 ; store c, d

; 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 2X2

Or1: dw 0, 0

Or2: dw 0, 0

MatMul:

; 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 matrices **M** and **N.**

**Theory**:

If M and N are two matrices of order 2x2 as shown below

Multiplication of two matrices can be calculated by dot product of every row of with columns of as shown below:

As shown above, 1st element of can be calculated by dot product of 1st row of and 1st col of , and so on.

**Write a subroutine which performs 2x2 matrix multiplication 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

; rows of M matrix of order 2X2

Mr1: db 0, 0 ; store a, b

Mr2: db 0, 0 ; store c, d

; 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 2X2

Or1: dw 0, 0

Or2: dw 0, 0

LoadMN:

; Write code for LoadMN

MatMul:

; Write code for Matrix Multiplication of 2x2 matrices

start:

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

; Write Calls to the subroutines & other codes here

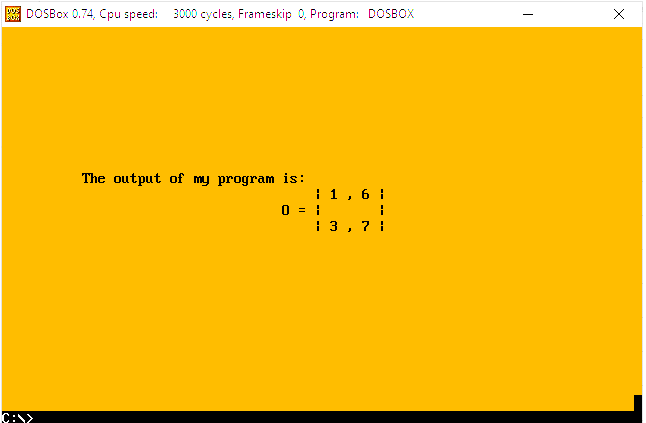
;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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:**

roll: db 7,5,4,1

a: db 0

b: db 0

c: db 0

d: db 0

;rows of matrix M

Mr1: db 0,0

Mr2: db 0,0

;rows of matrix N

Nr1: db 0,0

Nr2: db 0,0

;rows of matrix O

Or1: dw 0,0

Or2: dw 0,0

stri: db "The Output of my Program is:"

strlen: dw 28

LoadMN:

mov ax,0

mov al,[roll]

mov [a],al

mov al,[roll+1]

mov [b],al

mov al,[roll+2]

mov [c],al

mov al,[roll+3]

mov [d],al

mov ax,0

mov al,[a]

mov [Mr1],al

mov [Nr2+1],al

mov al,[b]

mov [Mr1+1],al

mov [Nr2],al

mov al,[c]

mov [Mr2],al

mov [Nr1+1],al

mov al,[d]

mov [Mr2+1],al

mov [Nr1],al

ret

MatMul:

mov ax,0

mov al,[Mr1];a

mov bl,[Nr1];d

mul bl

mov [Or1],ax

mov ax,0

mov al,[Mr1+1];b

mov bl,[Nr2];b

mul bl

add [Or1],ax

;-------------------------------------------------------------1st element of O

mov ax,0

mov al,[Mr1];a

mov bl,[Nr1+1];c

mul bl

mov [Or1+2],ax

mov ax,0

mov al,[Mr1+1];b

mov bl,[Nr2+1];a

mul bl

add [Or1+2],ax

;-------------------------------------------

mov ax,0

mov al,[Mr2];c

mov bl,[Nr1];d

mul bl

mov [Or2],ax

mov ax,0

mov al,[Mr2+1];d

mov bl,[Nr2];b

mul bl

add [Or2],ax

;-------------------------------------------------------------

mov ax,0

mov al,[Mr2];c

mov bl,[Nr1+1];c

mul bl

mov [Or2+2],ax

mov ax,0

mov al,[Mr2+1];d

mov bl,[Nr2+1];a

mul bl

add [Or2+2],ax

;-------------------------------------------

ret

Printmat:

push ax

push bx

push cx

push dx

push es

push di

mov ax,0xb800

mov es,ax

mov di,0

mov ax,0x0720

cler:

mov [es:di],ax

add di,2

cmp di,4000

jne cler

mov ax,0xb800

mov es,ax

mov ax,0

mov si,stri

mov di,0

add di,660

mov cx,[strlen]

mov ah,0x07

prin1:

mov al, [si]

mov [es:di],ax

add di,2

add si,1

loop prin1

add di,314

mov ax,0x074f

mov [es:di],ax

add di,4

mov word[es:di],0x073d

add di,4

sub di,160

mov word[es:di],0x077c

add di,4

mov ax,[Or1]

mov bx,10

nextdigit:

mov dx,0

div bx

add dl,0x30

push dx

inc cx

cmp ax,0

jnz nextdigit

nextpos:

pop dx

mov dh,0x07

mov [es:di],dx

add di,2

loop nextpos

add di,4

mov word[es:di],0x072c

add di,4

;----------------------------------------

mov ax,[Or1+2]

mov bx,10

mov cx,0

nextdigit1:

mov dx,0

div bx

add dl,0x30

push dx

inc cx

cmp ax,0

jnz nextdigit1

nextpos1:

pop dx

mov dh,0x07

mov [es:di],dx

add di,2

loop nextpos1

add di,2

mov word[es:di],0x077c

;-----------------------------------------------

add di,320

sub di,22

mov word[es:di],0x077c

add di,4

mov ax,[Or2]

mov bx,10

nextdigit2:

mov dx,0

div bx

add dl,0x30

push dx

inc cx

cmp ax,0

jnz nextdigit2

nextpos2:

pop dx

mov dh,0x07

mov [es:di],dx

add di,2

loop nextpos2

add di,6

mov word[es:di],0x072c

add di,4

;----------------------------------------

mov ax,[Or2+2]

mov bx,10

mov cx,0

nextdigit3:

mov dx,0

div bx

add dl,0x30

push dx

inc cx

cmp ax,0

jnz nextdigit3

nextpos3:

pop dx

mov dh,0x07

mov [es:di],dx

add di,2

loop nextpos3

add di,2

mov word[es:di],0x077c

;-----------------------------------------------

pop di

pop es

pop dx

pop bx

pop cx

pop ax

ret

start:

call LoadMN

call MatMul

call Printmat

mov ah,0x1

int 0x21

mov ax,0x4c00

int 0x21