

**CS29206: Systems Programming Lab**  
**Spring 2025**  
**Assignment 1**  
**Time: 1 hour 45 mins**

In this assignment, you will create and test a library of functions for 2-d integer matrix operations. Specifically, you will have to do the following in order. Please follow instructions carefully.

1. Create a directory named **A1<your roll no>** under your home directory (directory you are in when you log in). For example, if your roll no. is 23CS10002, the directory should be named **A123CS10002**. All instructions after this in this assignment will use this directory name, **A123CS10002**, as example, you should interpret it as your actual directory name you will use based on your own roll no.
2. Create two subdirectories under **A123CS10002**, called **include** and **lib**.
3. Define a structure datatype named **Matrix** that will store the dimensions and elements of a 2-d integer matrix. Use #define to assign the maximum size of the rows (**ROWS**) and columns (**COLS**) of the matrix. Use **ROWS** and **COLS** to allocate the matrix statically inside the structure. Put the type definition in a file called **defs.h**. You may put other things in **defs.h** later if needed for meeting the specifications given. Put **defs.h** in the subdirectory **A123CS10002/include** under your home directory. **defs.h** should not be included more than once by any file using it, directly or indirectly.
4. Implement the following C functions in a file called **matIO.c** in the directory **A123CS10002**:
  - a. **readMatrix()**: Takes a pointer to a **Matrix** type variable as parameter, and reads in its elements from the keyboard in row-major order (row-by row, starting with the first row). Returns void.
  - b. **printMatrix()**: Takes a **Matrix** type variable as parameter, and prints its elements. Returns void.

This file should include exactly one .h file named "**matIO.h**", no other header file should be included in it. Decide what you should put in **matIO.h**. Put **matIO.h** in the subdirectory **A123CS10002/include** under home.

5. Implement the following C functions in a file called **matAlg.c** in the directory **A123CS10002**:
  - a. **addMatrix()**: Takes two **Matrix** type variables A and B as parameters, and add those two matrices. Note that A and B may not be compatible for computing  $A + B$ ; in that case handle the exception (print a suitable message and return a negative integer value). Print the result matrix. Return 0.
  - b. **subMatrix()**: Takes two **Matrix** type variables A and B as parameters, and subtract B from A. Note that A and B may not be compatible for computing  $A - B$ ; in that case handle the exception (print a suitable message and return a negative integer value). Print the result matrix. Return 0.
  - c. **multMatrix()**: Takes two **Matrix** type variables A and B as parameters, and multiply the two matrices. Note that A and B may not be compatible for computing  $A \times B$ ; in that case handle the exception (print a suitable message and return a negative integer value). Print the result matrix. Return 0.

This file will include exactly one .h file named "**matAlg.h**", no other header file should be included in it. Decide what you should put in **matAlg.h**. Put **matAlg.h** in the subdirectory **A123CS10002/include** under home.

6. Create a static library **libmatrix.a** to include all functions defined in **matIO.c** and **matAlg.c**. Put **libmatrix.a** in the subdirectory **A123CS10002/lib** under your home directory.
7. Write a C program (in a file called **matops.c** in the directory **A123CS10002** under your home directory) that does the following:
  - (a) Stores the matrix dimension in structure member variable **rows** and **cols** (where, **rows<ROWS** and **cols<COLS**) and calls **readMatrix()** function to read and store the matrix elements.
  - (b) Prints the elements of the matrix using **printMatrix()**.
  - (c) Repeat steps (a) and (b) to read and print another integer matrix.
  - (d) Give a choice to the user to select one of the matrix operations (addition/subtraction/multiplication).
  - (e) Perform the user chosen matrix operation using the relevant function defined in **matAlg.c**. Print the resultant matrix.

For each of the matrix operation, if it is not possible, print a suitable message (exception handling).

8. Compile **matops.c** with the static library created. Name the executable file **matops**. Run **matops** (use the test input given) to test it.

9. Create a text file **commandlist.txt** in the directory **A123CS10002** under your home directory. List all the commands you ran from the \$ prompt of your terminal to do the above steps, in order as specified above, with one line per command. You must fire the commands when you are at the directory **A123CS10002**.
10. Tar/zip the entire **A123CS10002** directory. Submit the zip file in moodle within deadline. Example for tar using command prompt (make sure that you are in your home directory):

```
tar -cvf A123CS10002.tar ./A123CS10002/
```

Example test cases (more test cases will be used for evaluation),

```
pralay@pralay:~/asgn1$ ./matops
```

```
Enter the details of the first matrix.
```

```
Dimension of the Matrix [row col]: 3 3
```

```
Enter the Matrix elements: 1 2 3 4 5 6 7 8 9
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
Enter the details of the second matrix.
```

```
Dimension of the Matrix [row col]: 3 3
```

```
Enter the Matrix elements: 10 20 30 40 50 60 70 80 90
```

```
10 20 30
```

```
40 50 60
```

```
70 80 90
```

```
Enter your choice for matrix operation 1 for addition, 2 for subtraction, 3 for multiplication, else no operation.1
```

```
Resultant Matrix after addition
```

```
11 22 33
```

```
44 55 66
```

```
77 88 99
```

```
pralay@pralay:~/asgn1$ ./matops
```

```
Enter the details of the first matrix.
```

```
Dimension of the Matrix [row col]: 3 3
```

```
Enter the Matrix elements: 10 20 30 40 50 60 70 80 90
```

```
10 20 30
```

```
40 50 60
```

```
70 80 90
```

```
Enter the details of the second matrix.
```

```
Dimension of the Matrix [row col]: 3 3
```

```
Enter the Matrix elements: 1 2 3 4 5 6 7 8 9
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
Enter your choice for matrix operation 1 for addition, 2 for subtraction, 3 for multiplication, else no operation.2
```

```
Resultant Matrix after subtraction
```

```
9 18 27
```

```
36 45 54
```

```
63 72 81
```

```
pralay@pralay:~/asgn1$ ./matops
```

```
Enter the details of the first matrix.
```

```
Dimension of the Matrix [row col]: 3 3
```

```
Enter the Matrix elements: 2 1 3 2 4 5 6 4 5
```

```
2 1 3
```

```
2 4 5
```

```
6 4 5
```

```
Enter the details of the second matrix.
```

```
Dimension of the Matrix [row col]: 3 4
```

```
Enter the Matrix elements: 2 1 3 2 5 3 6 7 9 8 5 1
```

```
2 1 3 2
```

```
5 3 6 7
```

```
9 8 5 1
```

```
Enter your choice for matrix operation 1 for addition, 2 for subtraction, 3 for multiplication, else no operation.3
```

```
Resultant Matrix after multiplication
```

```
36 29 27 14
```

```
69 54 55 37
```

```
77 58 67 45
```

```
pralay@pralay:~/asgn1$
```