

**University of Mila**  
**Faculty of Science and Technology**  
**Department of Process Engineering**

---

## **Practical Work 06**

---

**Course: Introduction to Programming**

**Level: 1st Year ST - ENG & LMD**

**Semester 02**

**By:**

**Dr. Farouk KECITA**

**Academic Year: 2025/2026**

## Exercise 01: 1D Array Operations Using Functions

**Question:** Write a C program that uses functions to perform the following operations on a 1D array of size N:

1. Fill the array with user inputs
2. Reverse the array
3. Find the minimum element
4. Find the maximum element
5. Calculate the sum of all elements
6. Calculate the average of all elements

**Requirements:**

- Use separate functions for each operation
- Use function prototypes
- Display the original array, reversed array, min, max, sum, and average

**Expected Program Structure:**

```
1 #include <stdio.h>
2 #define MAX 100
3 // Function prototypes
4 void fillArray(int arr[], int n);
5 void displayArray(int arr[], int n);
6 void reverseArray(int arr[], int n);
7 int findMin(int arr[], int n);
8 int findMax(int arr[], int n);
9 int sumArray(int arr[], int n);
10 float averageArray(int arr[], int n);
11
12 int main() {
13     // Your code here
14     return 0;
15 }
16 // Function definitions here
```

## Exercise 02: Matrix Multiplication Using Functions

**Question:** Write a C program that multiplies two matrices A and B using functions.

**Given:**

- Matrix A: dimension  $m \times n$
- Matrix B: dimension  $n \times p$
- Result matrix C: dimension  $m \times p$

**Requirements:**

1. Write a function to read a matrix from the user
2. Write a function to display a matrix
3. Write a function to multiply two matrices (check if multiplication is possible)
4. In the main function, read dimensions and elements of A and B, then display the product

**Formula:**

$$C_{ij} = \sum_{k=1}^n A_{ik} \times B_{kj}$$

**Expected Function Prototypes:**

```

1 void readMatrix(int mat[][100], int rows, int cols, char name);
2 void displayMatrix(int mat[][100], int rows, int cols, char name);
3 void multiplyMatrices(int A[][100], int B[][100], int C[][100],
4                       int m, int n, int p);

```

**Exercise 03: Solving 2x2 Linear System Using Inverse Method**

**Question:** Write a C program to solve a system of 2 linear equations using the inverse matrix method.

**System:**

$$\begin{cases} a_{11}x + a_{12}y = b_1 \\ a_{21}x + a_{22}y = b_2 \end{cases}$$

**Matrix Form:**

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad X = \begin{pmatrix} x \\ y \end{pmatrix}, \quad B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$A \cdot X = B \quad \Rightarrow \quad X = A^{-1} \cdot B$$

**Inverse of a 2x2 Matrix:**

$$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$$

where  $\det(A) = a_{11}a_{22} - a_{12}a_{21}$

**Requirements:**

1. Read the coefficients  $a_{11}, a_{12}, a_{21}, a_{22}$  and constants  $b_1, b_2$
2. Calculate the determinant
3. If  $\det(A) \neq 0$ , calculate  $x$  and  $y$  using the inverse method
4. If  $\det(A) = 0$ , display an appropriate message
5. Verify the solution by substituting back into the original equations

## Exercise 04: Solving Linear System Using Functions

**Question:** Rewrite Exercise 03 using functions to make the code more modular and reusable.

**Requirements:** Create the following functions:

1. `float calculateDeterminant(float A[2][2]);`
  - Computes and returns the determinant of a 2x2 matrix
2. `int inverseMatrix(float A[2][2], float invA[2][2]);`
  - Computes the inverse of matrix A
  - Returns 1 if inverse exists, 0 otherwise
3. `void solveSystem(float A[2][2], float B[2], float X[2]);`
  - Solves the system  $A \cdot X = B$
  - Displays the solution or an error message
4. `void displaySystem(float A[2][2], float B[2]);`
  - Displays the system of equations in a readable format
5. `void verifySolution(float A[2][2], float X[2], float B[2]);`
  - Verifies that  $A \cdot X = B$
  - Displays the verification results

**Expected Program Structure:**

```

1 #include <stdio.h>
2 #include <math.h>
3 #define SIZE 2
4 // Function prototypes
5 float calculateDeterminant(float A[SIZE][SIZE]);
6 int inverseMatrix(float A[SIZE][SIZE], float invA[SIZE][SIZE]);
7 void solveSystem(float A[SIZE][SIZE], float B[SIZE], float X[SIZE]);
8 void displaySystem(float A[SIZE][SIZE], float B[SIZE]);
9 void verifySolution(float A[SIZE][SIZE], float X[SIZE], float
    B[SIZE]);
10 int main() {
11     float A[SIZE][SIZE], B[SIZE], X[SIZE];
12     // Read input
13     printf("Enter coefficients a11, a12, a21, a22: ");
14     scanf("%f %f %f %f", &A[0][0], &A[0][1], &A[1][0], &A[1][1]);
15     printf("Enter constants b1, b2: ");
16     scanf("%f %f", &B[0], &B[1]);
17     // Display system
18     displaySystem(A, B);
19     // Solve system
20     solveSystem(A, B, X);
21     return 0;
22 }
23
24 // Function definitions here

```