Experiment-1
Matrix Operations

(Duration: 105 mins)

Purpose: Matrices are a useful way to represent, manipulate and study linear maps between finite dimensional vector space. This experiment aims to equip you with a simple set of functions that help you in your future work on linear systems.

Problem Statement

You are asked to write three functions to perform basic matrix operations: matrix addition, matrix multiplication, matrix transpose. You need to verify your functions by comparing the results with the examples in the laboratory procedure given in the last section.
The running program in bash shell should look in Figure 1 (on the last page). An ideal program should:

- Get the array dimensions of operand matrices from the user **(10pts)**
- Let the user choose which operation will take place **(10pts)**
- Prompt an error message when dimensions of the arrays do not match (column size and row size must match for matrix multiplication) **(10pts)**
- Demonstrate working functions (**All functions should be written based on prototypes below**)**(70pts)**
- Write a matrix addition function that has the following prototype. **(20pts)**

```
// Function prototype for matrix addition
void matAdd(int nRow , // Number of rows
            int nCol, // Number of columns
            double *A, // Memory adress for the first operand matrix
            double *B) // Memory adress for the second operand matrix
```

- Write a matrix multiplication function that has the following prototype. **(30pts)**

```
// Function prototype for matrix multiplication
void matMul(int nRow1, // Number of rows - the first operand matrix
            int nCol1, // Number of columns - the first operand matrix
            int nRow2, // Number of rows - the second operand matrix
            int nCol2, // Number of columns - the second operand matrix
            double A[nRow1][nCol1], //The first operand matrix
            double B[nRow2][nCol2]) //The second operand matrix
```

- Write a matrix transpose function that has the following prototype. **(20pts)**

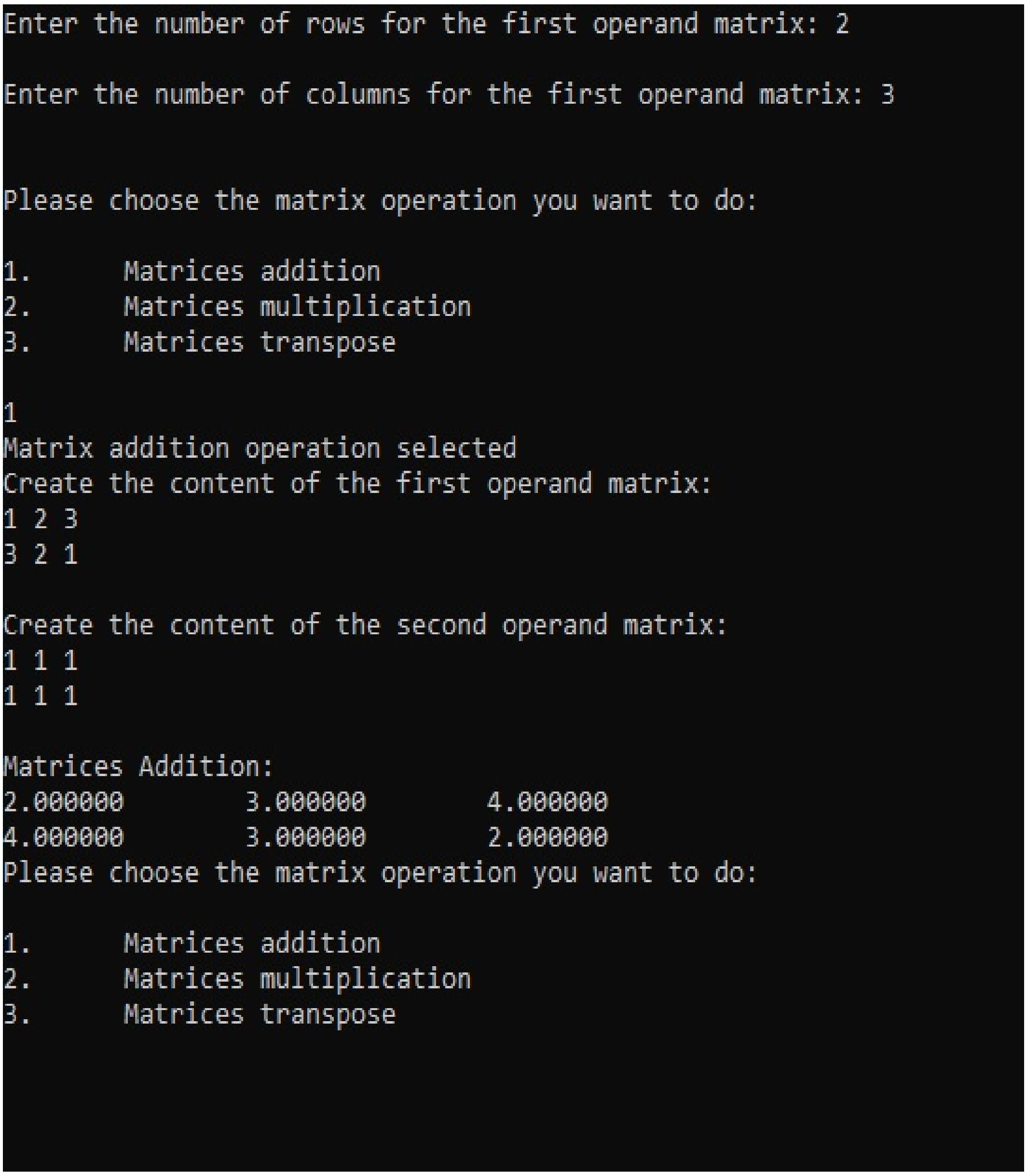
```
// Function prototype for matrix multiplication
void matTra(int nRow, // Number of rows
            int nCol, // Number of columns
            double A[nRow][nCol]) // The operand matrix
```

- Call addition, multiplication and transpose operations to solve the problems below.

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} 11 & 23 \\ 9 & 17 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 2 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 3 & 4 & 1 \end{bmatrix}$$



```
Enter the number of rows for the first operand matrix: 2

Enter the number of columns for the first operand matrix: 3


Please choose the matrix operation you want to do:

1.      Matrices addition
2.      Matrices multiplication
3.      Matrices transpose

1
Matrix addition operation selected
Create the content of the first operand matrix:
1 2 3
3 2 1

Create the content of the second operand matrix:
1 1 1
1 1 1

Matrices Addition:
2.000000        3.000000        4.000000
4.000000        3.000000        2.000000
Please choose the matrix operation you want to do:

1.      Matrices addition
2.      Matrices multiplication
3.      Matrices transpose
```

Figure 1: Snapshot from bash shell