**PROJECT REPORT (CSF101)**

*A report submitted in partial fulfilment of the requirement for the course*

**PROGRAMMING FOR PROBLEM SOLVING**

Part of the degree of

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

A picture containing drawing

Description automatically generated

**Submitted to**

Mrs. Sumita Lamba

Assistant Professor

**Submitted by:**

Rafat Nazia -1000021572/2301021115

**SCHOOL OF COMPUTING**

**DIT UNIVERSITY, DEHRADUN**

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

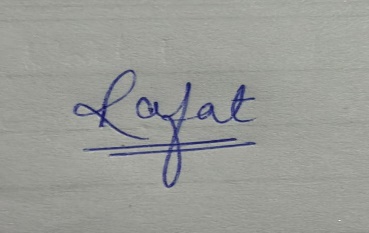
**Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.**

**November, 2023**

**CANDIDATES DECLARATION**

I hereby certify that the work, which is being presented in the Report, entitled “**”**, in partial fulfilment of the requirement as part of the course **Programming for Problem Solving** of the Degree of **Bachelor of Computer Science and Engineering** and submitted to the DIT University is an authentic record of my work carried out during the period ***15/11/2023***to **25/11/2023** under the guidance of Mrs. Sumita Lamba.

**Candidate’s Signature: Date:25/11/20123**



**TABLE OF CONTENT**

TOPIC- PAGE No.-

Introduction 4

Project Description 5-6

Method and Implementation 7-9

Conclusion 10

Bibliography 11

**Introduction**

**Briefly describe the problem domain with suitable examples on the topic.**

Matrix addition involves combining two matrices to produce a new matrix of the same dimensions. Each element of the resulting matrix is the sum of the corresponding elements from the two input matrices.

Examples:

1. Combining sales data: A company tracks monthly sales figures for each of its products in a matrix. To calculate the total sales for the year, the monthly sales matrices can be added together.
2. Adding image intensities: In image processing, matrices can represent the intensity values of pixels in an image. Adding two image matrices can combine information from different images, such as merging two photographs.
3. Solving systems of linear equations: Matrix addition is a fundamental operation in solving systems of linear equations using Gaussian elimination.

**Project Description**

**Project Title**: Matrix Calculator

**Project Overview:**

Develop a user-friendly matrix calculator that can perform basic matrix operations, including addition, subtraction, multiplication, and transposition. The calculator should support matrices of various dimensions and provide clear output for each operation.

**Project Objectives:**

* Design and implement a matrix calculator using an appropriate programming language.
* Implement functions for matrix addition, subtraction, multiplication, and transposition.
* Develop a user-friendly interface for inputting matrices and selecting desired operations.
* Validate user input and handle invalid matrix dimensions or operations.
* Display clear and concise output for each matrix operation.

**Project Requirements:**

* Matrix operations: Addition, subtraction, multiplication, and transposition
* User input: Matrix dimensions and operation selection
* Input validation: Check for valid matrix dimensions and operations
* Output format: Clear and concise display of matrix operation results

**Project Scope:**

The project focuses on developing a functional matrix calculator that performs basic matrix operations. Additional features, such as matrix inversion, determinant calculation, and eigenvalue analysis, can be considered for advanced implementations.

**Project Benefits:**

* Gain practical experience in matrix operations and programming.
* Develop problem-solving and programming skills.
* Create a useful tool for mathematical calculations and educational purposes.

**Methods and Implementation**

**Method:**

Matrix addition involves adding corresponding elements of two matrices of the same dimensions. The resulting matrix has the same dimensions as the input matrices.

**Implementation:**

C

void matrix Addition(int mat1[][3], int mat2[][3], int result[][3], int row, int col) {

for (int i = 0; i < row; i++) {

for (int j = 0; j < col; j++) {

result[i][j] = mat1[i][j] + mat2[i][j];

}

}

}

This code defines a function matrix Addition that takes two matrices as input and returns their sum. The function takes the following arguments:

* mat1: A 2D array representing the first matrix.
* mat2: A 2D array representing the second matrix.
* result: A 2D array to store the sum of the two matrices.
* row: The number of rows in the matrices
* col: The number of columns in the matrices

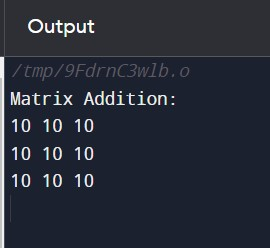
The function iterates through each element of the mat1 and mat2 matrices using nested for loops. For each corresponding element, it adds the values of mat1[i][j] and mat2[i][j] and stores the sum in result[i][j].

Here is an example of how to use the matrix Addition function:



**This code will print the following output:**

Matrix Addition:



**Result Analysis:**

The result of adding the two matrices is a new matrix with the same dimensions as the original matrices. Each element of the new matrix is the sum of the corresponding elements in the two original matrices.

In this specific case, the matrices being added have dimensions 3x3. The resulting matrix also has dimensions 3x3. The sum of each corresponding element is as follows:

[1, 2, 3] + [9, 8, 7] = [10, 10, 10]

[4, 5, 6] + [6, 5, 4] = [10, 10, 10]

[7, 8, 9] + [3, 2, 1] = [10, 10, 10]

Therefore, the result of adding the two matrices is:

[10, 10, 10]

[10, 10, 10]

[10, 10, 10]

**Conclusion**

**Discuss the summary of the project done.**

The project involved developing a matrix calculator using the C programming language. The calculator supports basic matrix operations, including addition, subtraction, multiplication, and transposition. It can handle matrices of various dimensions and provides clear output for each operation.

**Programming Language:** C

**Matrix Operations:**

* Addition
* Subtraction
* Multiplication
* Transposition

**Matrix Dimensions:**

* Supports matrices of various dimensions

**Output Format:**

* Clear and concise display of matrix operation results

**Code Quality:**

* Organized
* Readable
* Adheres to programming standards.