

Matrix Operations

```
#include <stdio.h>

// Function to accept and print a matrix
void acceptAndPrintMatrix(int matrix[][10], int rows, int cols) {
    printf("Enter the elements of the matrix:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }
    printf("Matrix:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}

// Function to add two matrices
void addMatrices(int matrix1[][10], int matrix2[][10], int result[][10], int rows, int cols) {
    printf("\nAddition of matrices:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
            printf("%d ", result[i][j]);
        }
        printf("\n");
    }
}

// Function to print diagonal elements of a matrix
void printDiagonalElements(int matrix[][10], int rows, int cols) {
    printf("\nDiagonal elements:\n");
    for (int i = 0; i < rows && i < cols; i++) {
        printf("%d ", matrix[i][i]);
    }
    printf("\n");
}

// Function to print transpose of a matrix
void printTranspose(int matrix[][10], int rows, int cols) {
    printf("\nTranspose of matrix:\n");
    for (int i = 0; i < cols; i++) {
        for (int j = 0; j < rows; j++) {
            printf("%d ", matrix[j][i]);
        }
        printf("\n");
    }
}
```

Matrix Operations

```
int main() {
    int rows, cols;

    printf("Enter the number of rows and columns of the matrices: ");
    scanf("%d %d", &rows, &cols);

    int matrix1[rows][cols], matrix2[rows][cols], result[rows][cols];

    printf("\nEnter elements of the first matrix:\n");
    acceptAndPrintMatrix(matrix1, rows, cols);

    printf("\nEnter elements of the second matrix:\n");
    acceptAndPrintMatrix(matrix2, rows, cols);

    addMatrices(matrix1, matrix2, result, rows, cols);

    printDiagonalElements(matrix1, rows, cols);

    printTranspose(matrix1, rows, cols);

    return 0;
}
```