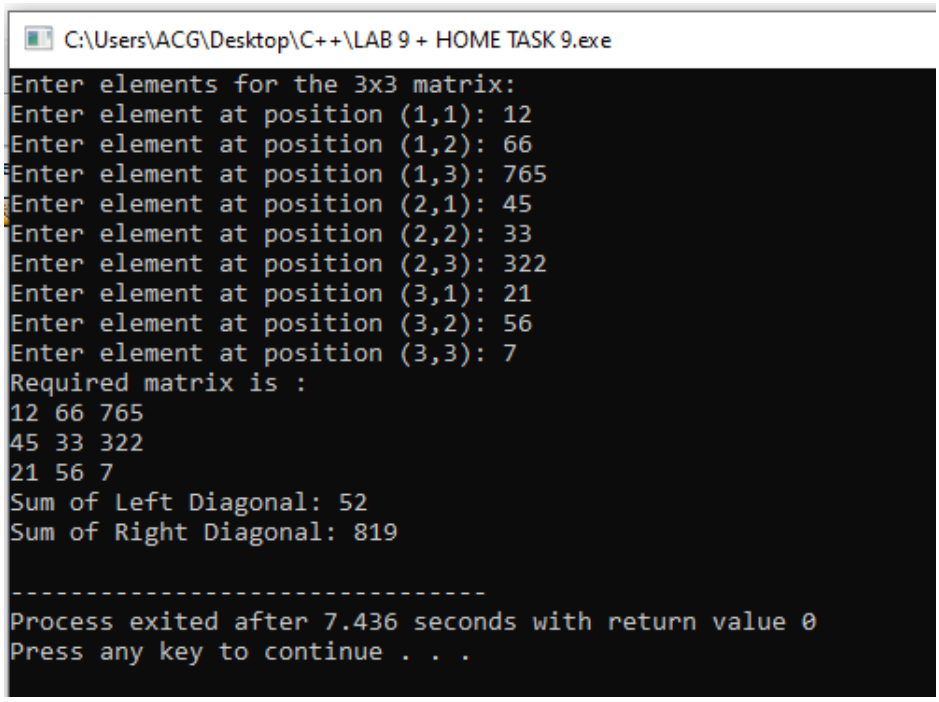
LAB 9 FOP: -

NAME	Muhammad Shehryar Ismail
CMS ID	461599
SECTION	A

TASK 1: -

```
#include <iostream>
using namespace std;
int main() {
   const int rows = 3;
   const int cols = 3;
   int matrix[rows][cols];
   cout << "Enter elements for the 3x3 matrix:" << endl;
   for (int i = 0; i < rows; ++i) {
      for (int j = 0; j < cols; ++j) {
         cout << "Enter element at position (" << i + 1 << "," << j + 1 << "): ";
         cin >> matrix[i][j];
      }
   }
   cout << "Required matrix is :"<<endl;
   for (int i = 0; i < rows; i++) {
      for (int j = 0; j < cols; j++) {
         cout << matrix[i][j] << " ";
      }
      cout << endl;
   }
   int leftDiagonalSum = 0;
   for (int i = 0; i < rows; i++) {
      leftDiagonalSum += matrix[i][i];
   }
   int rightDiagonalSum = 0;
   for (int i = 0; i < rows; i++) {
      rightDiagonalSum += matrix[i][rows - 1 - i];
   }
   cout << "Sum of Left Diagonal: " << leftDiagonalSum << endl;
   cout << "Sum of Right Diagonal: " << rightDiagonalSum << endl;
```

```
    return 0;
}
```



TASK 2: -

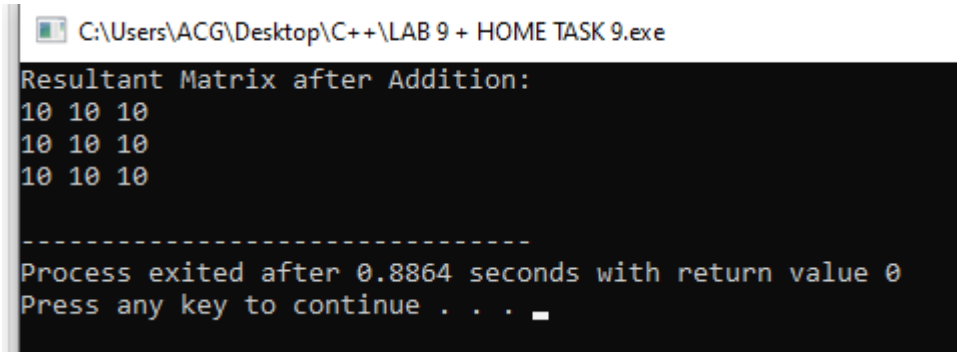
```
#include <iostream>
using namespace std;
void addMatrices(const int matrix1[3][3], const int matrix2[3][3], int result[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
}
int main() {
    const int rows = 3;
    const int cols = 3;



    int matrix1[rows][cols] = {{1, 2, 3},
```

```
                                    {4, 5, 6},
                                    {7, 8, 9}};

    int matrix2[rows][cols] = {{9, 8, 7},
                               {6, 5, 4},
                               {3, 2, 1}};
    int result[rows][cols];
    addMatrices(matrix1, matrix2, result);
    cout << "Resultant Matrix after Addition:\n";
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cout << result[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```



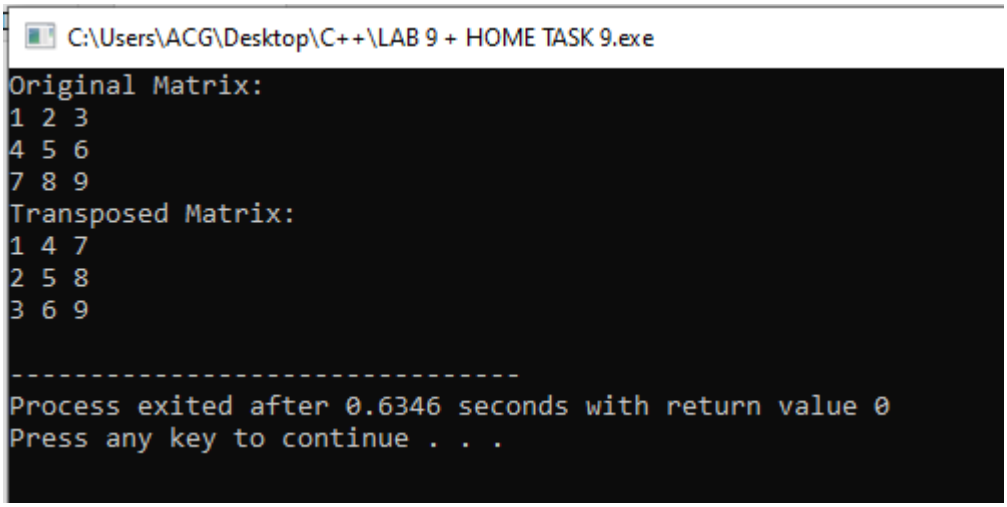
TASK 3: -

```
void transposeMatrix(const int matrix[3][3], int result[3][3]) {
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
```

```
            result[i][j] = matrix[j][i];
        }
    }
}

int main() {
    int matrix[3][3] = {{1, 2, 3},
                        {4, 5, 6},
                        {7, 8, 9}};

    int transposeMatrixResult[3][3];
    transposeMatrix(matrix, transposeMatrixResult);
    std::cout << "Original Matrix:" << std::endl;
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            std::cout << matrix[i][j] << " ";
        }
        std::cout << std::endl;
    }
    std::cout << "Transposed Matrix:" << std::endl;
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            std::cout << transposeMatrixResult[i][j] << " ";
        }
        std::cout << std::endl;
    }
    return 0;
}
```



TASK 4: -

```
#include <iostream>
using namespace std;
void inputMatrix(int matrix[3][3]) {
   std::cout << "Enter the elements of the matrix (row-wise):" << std::endl;
   for (int i = 0; i < 3; ++i) {
      for (int j = 0; j < 3; ++j) {
         std::cout << "Enter element at position (" << i + 1 << ", " << j + 1 << "): ";
         std::cin >> matrix[i][j];
      }
   }
}
void multiplyMatrices(const int matrix1[3][3], const int matrix2[3][3], int result[3][3]) {
   for (int i = 0; i < 3; ++i) {
      for (int j = 0; j < 3; ++j) {
         result[i][j] = 0;
         for (int k = 0; k < 3; ++k) {
            result[i][j] += matrix1[i][k] * matrix2[k][j];
         }
      }
   }
```

```
}

int main() {
    int matrix1[3][3], matrix2[3][3], resultMatrix[3][3];
    std::cout << "Input for Matrix 1:\n";
    inputMatrix(matrix1);

    std::cout << "\nInput for Matrix 2:\n";
    inputMatrix(matrix2);
    multiplyMatrices(matrix1, matrix2, resultMatrix);
    std::cout << "\nMatrix 1:\n";
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            std::cout << matrix1[i][j] << " ";
        }
        std::cout << std::endl;
    }

    std::cout << "\nMatrix 2:\n";
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            std::cout << matrix2[i][j] << " ";
        }
        std::cout << std::endl;
    }


    std::cout << "\nResult Matrix:\n";
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            std::cout << resultMatrix[i][j] << " ";
```

```
        }
        std::cout << std::endl;
    }
    return 0;
}
```