



NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

CS-114-FUNDAMENTAL OF PROGRAMING LAB MANUAL # 9

➤ **NAME:** Muhammad Uzair

➤ **CLASS:** ME 15

➤ **SECTION:** B

➤ **CMS ID:** 466092

TASK # 1

CODE:

```
#include <iostream>

using namespace std;

// Function to calculate the sum of the left diagonal elements of a 3x3 matrix
int leftDiagonalSum(int matrix[3][3]) {
    int sum = 0;
    for (int i = 0; i < 3; ++i) {
        sum += matrix[i][i];
    }
    return sum;
}

// Function to calculate the sum of the right diagonal elements of a 3x3 matrix
int rightDiagonalSum(int matrix[3][3]) {
    int sum = 0;
    for (int i = 0; i < 3; ++i) {
        sum += matrix[i][2 - i];
    }
    return sum;
}

// Function to display a 3x3 matrix
void displayMatrix(int matrix[3][3]) {
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cout << matrix[i][j] << " ";
        }
    }
}
```

```

    }

    cout << endl;
}

}

int main() {
    int matrix[3][3];

    cout << "Enter the elements of the 3x3 matrix:" << endl;
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cin >> matrix[i][j];
        }
    }

    cout << "Matrix:" << endl;
    displayMatrix(matrix);

    int leftDiagonal = leftDiagonalSum(matrix);
    int rightDiagonal = rightDiagonalSum(matrix);

    cout << "Sum of the left diagonal: " << leftDiagonal << endl;
    cout << "Sum of the right diagonal: " << rightDiagonal << endl;

    return 0;
}

```

RESULT:

```
Enter the elements of the 3x3 matrix:
2
3
4
5
6
8
7
3
2
Matrix:
2 3 4
5 6 8
7 3 2
Sum of the left diagonal: 10
Sum of the right diagonal: 17

-----
Process exited after 8.4 seconds with return value 0
Press any key to continue . . .
```

TASK # 2

CODE:

```
#include <iostream>
```

```
using namespace std;
```

```
void addMatrices(int matrix1[3][3], 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];
```

```
        }
```

```
    }
```

```
}
```

```
void displayMatrix(int matrix[3][3]) {
```

```
    for (int i = 0; i < 3; ++i) {
```

```
        for (int j = 0; j < 3; ++j) {  
            cout << matrix[i][j] << " ";  
        }  
        cout << endl;  
    }  
}
```

```
int main() {  
    int matrix1[3][3], matrix2[3][3], result[3][3];  
  
    cout << "Enter the elements of the first 3x3 matrix:" << endl;  
    for (int i = 0; i < 3; ++i) {  
        for (int j = 0; j < 3; ++j) {  
            cin >> matrix1[i][j];  
        }  
    }  
  
    cout << "Enter the elements of the second 3x3 matrix:" << endl;  
    for (int i = 0; i < 3; ++i) {  
        for (int j = 0; j < 3; ++j) {  
            cin >> matrix2[i][j];  
        }  
    }  
  
    addMatrices(matrix1, matrix2, result);  
  
    cout << "Result of matrix addition:" << endl;  
    displayMatrix(result);  
  
    return 0;
```

```
}
```

RESULT:

```
Enter the elements of the first 3x3 matrix:
4
5
6
7
8
2
3

4
5
Enter the elements of the second 3x3 matrix:
1
2
3
9
8
6
7
4
5
Result of matrix addition:
5 7 9
16 16 8
10 8 10

-----
Process exited after 12.68 seconds with return value 0
Press any key to continue . . .
```

TASK # 3

CODE:

```
#include <iostream>

using namespace std;

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

```
        for (int j = 0; j < 3; ++j) {  
            result[j][i] = matrix[i][j];  
        }  
    }  
}
```

```
void displayMatrix(int matrix[3][3]) {  
    for (int i = 0; i < 3; ++i) {  
        for (int j = 0; j < 3; ++j) {  
            cout << matrix[i][j] << " ";  
        }  
        cout << endl;  
    }  
}
```

```
int main() {  
    int matrix[3][3], result[3][3];  
  
    cout << "Enter the elements of the 3x3 matrix:" << endl;  
    for (int i = 0; i < 3; ++i) {  
        for (int j = 0; j < 3; ++j) {  
            cin >> matrix[i][j];  
        }  
    }
```

```
    transposeMatrix(matrix, result);
```

```
    cout << "Transpose of the matrix:" << endl;  
    displayMatrix(result);
```

```
    return 0;
}
```

RESULT:

```
Enter the elements of the 3x3 matrix:
2
3
4
5
6
8
7
21
2
Transpose of the matrix:
2 5 7
3 6 21
4 8 2

-----
Process exited after 8.162 seconds with return value 0
Press any key to continue . . .
```

TASK # 4

CODE:

```
#include <iostream>

using namespace std;

// Function to multiply two 3x3 matrices
void multiplyMatrices(int firstMatrix[3][3], int secondMatrix[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] += firstMatrix[i][k] * secondMatrix[k][j];
        }
    }
}

```

// Function to display a 3x3 matrix

```

void displayMatrix(int matrix[3][3]) {
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
}

```

```

int main() {
    int firstMatrix[3][3], secondMatrix[3][3], result[3][3];

    cout << "Enter the elements of the first 3x3 matrix:" << endl;
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            cin >> firstMatrix[i][j];
        }
    }
}

```

```

    cout << "Enter the elements of the second 3x3 matrix:" << endl;
    for (int i = 0; i < 3; ++i) {

```

```

        for (int j = 0; j < 3; ++j) {
            cin >> secondMatrix[i][j];
        }
    }

    multiplyMatrices(firstMatrix, secondMatrix, result);

    cout << "Result of matrix multiplication:" << endl;
    displayMatrix(result);

    return 0;
}

```

RESULT:

```

Enter the elements of the first 3x3 matrix:
4
5
6
9
2
6
2
7
5
Enter the elements of the second 3x3 matrix:
1
2
3
4
5
9
5
4
6
Result of matrix multiplication:
54 57 93
47 52 81
55 59 99

-----
Process exited after 22.05 seconds with return value 0
Press any key to continue . . .

```

TASK # 5

CODE:

```
#include <iostream>

using namespace std;

void printtable(int multiplier, int limit){
    if(multiplier>limit){
        return ;
    }
    cout<<" 15 x "<<multiplier<<" = "<<(15*multiplier)<<endl;
    printtable (multiplier+1,limit);

}

int main(){
    int limit=10;

    cout<<"The multiplication table of 15 is: \n";
    printtable (1,limit );
}
```

RESULT:

The multiplication table of 15 is:

$$15 \times 1 = 15$$

$$15 \times 2 = 30$$

$$15 \times 3 = 45$$

$$15 \times 4 = 60$$

$$15 \times 5 = 75$$

$$15 \times 6 = 90$$

$$15 \times 7 = 105$$

$$15 \times 8 = 120$$

$$15 \times 9 = 135$$

$$15 \times 10 = 150$$

Process exited after 0.1856 seconds with return value 0
Press any key to continue . . . ■