

National University of Sciences and Technology (NUST)
Department of Mechanical Engineering (SMME)



Fundamentals of Programming (FOP)

[Lab Tasks](#)

[Lab Manual 9](#)

By

Muhammad Owais

461359

Teacher: Sir Muhammad Affan

Lab Task-1:

```
1  #include<iostream>
2  using namespace std;
3  int main() {
4      int matrix[3][3] = {{1, 4, 5},
5                          {9, 5, 7},
6                          {3, 2, 0}};
7      int LDSum = 0;
8      for (int i = 0; i < 3; ++i)
9          LDSum += matrix[i][i];
10     int RDSum = 0;
11     for (int i = 0; i < 3; ++i)
12         RDSum += matrix[i][2 - i];
13     cout<<"Left Diagonal Sum: "<<LDSum<<endl;
14     cout<<"Right Diagonal Sum: "<<RDSum<<endl;
15     return 0;
16 }
```

```
17  D:\University\CS\Untitled1.exe
18
19  Left Diagonal Sum: 6
20  Right Diagonal Sum: 13
21
22  -----
23  Process exited after 0.04607 seconds with return value 0
24  Press any key to continue . . .
25
26
```

Lab Task-2:

```
1  #include<iostream>
2  using namespace std;
3  void addMatrices(int arr1[3][3], int arr2[3][3], int result[3][3]) {
4      for (int i = 0; i < 3; ++i) {
5          for (int j = 0; j < 3; ++j) {
6              result[i][j] = arr1[i][j] + arr2[i][j];
7          }
8      }
9  }
10 int main() {
11     int matrix1[3][3] = {{7, 6, 3},
12                          {4, 8, 0},
13                          {7, 2, 1}};
14
15     int matrix2[3][3] = {{7, 5, 0},
16                          {4, 9, 1},
17                          {3, 4, 6}};
18     int resultMatrix[3][3];
19     addMatrices(matrix1, matrix2, resultMatrix);
20     cout << "Addition of two 2D arrays: " << endl;
21     for (int i = 0; i < 3; ++i) {
22         for (int j = 0; j < 3; ++j) {
23             std::cout << resultMatrix[i][j] << " ";
24         }
25         std::cout << std::endl;
26     }
27     return 0;
28 }
```

```
29  D:\University\CS\Untitled1.exe
30
31  Addition of two 2D arrays:
32  14 11 3
33  8 17 1
34  10 6 7
35
36  -----
37  Process exited after 0.04641 seconds with return value 0
38  Press any key to continue . . .
39
```

Lab Task-3:

```
1  #include<iostream>
2  using namespace std;
3  void TMatrix(int arr[3][3], int result[3][3]) {
4      for (int i = 0; i < 3; ++i) {
5          for (int j = 0; j < 3; ++j) {
6              result[i][j] = arr[j][i];
7          }
8      }
9  }
10 int main() {
11     int matrix[3][3] = {{1, 2, 3},
12                         {4, 5, 6},
13                         {7, 8, 9}};
14     int TMatrixResult[3][3];
15     TMatrix(matrix, TMatrixResult);
16     cout << "Original Matrix: " << endl;
17     for (int i = 0; i < 3; ++i) {
18         for (int j = 0; j < 3; ++j) {
19             std::cout << matrix[i][j] << " ";
20         }
21         cout << endl;
22     }
23     cout << "Transposed Matrix: " << endl;
24     for (int i = 0; i < 3; ++i) {
25         for (int j = 0; j < 3; ++j) {
26             cout << TMatrixResult[i][j] << " ";
27         }
28         cout << endl;
29     }
30     return 0;
31 }
```

```
32  D:\University\CS\Untitled1.exe
33
34  Original Matrix:
35  1 2 3
36  4 5 6
37  7 8 9
38  Transposed Matrix:
39  1 4 7
40  2 5 8
41  3 6 9
42
```

Lab Task-4:

```
1  #include<iostream>
2  using namespace std;
3  void multiplyMatrix(int mat1[3][3], int mat2[3][3], int result[3][3]) {
4      for (int i = 0; i < 3; ++i) {
5          for (int j = 0; j < 3; ++j) {
6              result[i][j] = 0;
7              for (int k = 0; k < 3; ++k) {
8                  result[i][j] += mat1[i][k] * mat2[k][j];
9              }
10         }
11     }
12 }
13
14 int main() {
15     int matrix1[3][3] = {{1, 2, 3},
16                          {4, 5, 6},
17                          {7, 8, 9}};
18
19     int matrix2[3][3] = {{9, 8, 7},
20                          {6, 5, 4},
21                          {3, 2, 1}};
22     int resultMatrix[3][3];
23     multiplyMatrix(matrix1, matrix2, resultMatrix);
24     cout << "Resultant Matrix after Multiplication: "<<endl;
25     for (int i = 0; i < 3; ++i) {
26         for (int j = 0; j < 3; ++j) {
27             std::cout << resultMatrix[i][j] << " ";
28         }
29         cout<<endl;
30     }
31
32     return 0;
33 }
```

```
34
35 D:\University\CS\Untitled1.exe
36 Resultant Matrix after Multiplication:
37 30 24 18
38 84 69 54
39 138 114 90
40
41 -----
42 Process exited after 0.05838 seconds with return value 0
Press any key to continue . . .
```

Lab Task-5:

```
1  #include <iostream>
2  using namespace std;
3  void printTable(int multiplier, int multiplicand = 1) {
4      if (multiplicand <= 10) {
5          cout << multiplier << " x " << multiplicand << " = " << multiplier * multiplicand << endl;
6          printTable(multiplier, multiplicand + 1);
7      }
8  }
9  int main() {
10     int num = 15;
11     cout << "Multiplication Table of " << num << endl;
12     printTable(num);
13     return 0;
14 }
```

```
15
16 D:\University\CS\Untitled1.exe
17 Multiplication Table of 15
18 15 x 1 = 15
19 15 x 2 = 30
20 15 x 3 = 45
21 15 x 4 = 60
22 15 x 5 = 75
23 15 x 6 = 90
24 15 x 7 = 105
25 15 x 8 = 120
26 15 x 9 = 135
27 15 x 10 = 150
28
29 -----
Process exited after 0.04509 seconds with return value 0
Press any key to continue . . .
```