



## **Programming Fundamental(CS111)**

### **Assignment # 03**

[CLO-2, Taxonomy Level-C3, PLO-2]

## **Solution**

**Course:** BSCS-1

**Semester:** 1<sup>st</sup> (Fall 2023)

**Due Date:** 27/10/2023

**Total Marks:** 30

---

### **Instructions**

1. *Plagiarism, copy & past material will lead to the cancellation of your assignment.*
  2. *Write your Name, Reg# on the first page (title page) of your submission.*
  3. *No late submission*
- 

1.

```
//this program is about array operations
#include <iostream>
using namespace std;

int main(){
    const int ARRAY_SIZE = 10;
    int arr[ARRAY_SIZE];

    // initialize the array with user-input values
    cout << "Enter " << ARRAY_SIZE << " integer values:\n";
    for (int i = 0; i < ARRAY_SIZE; ++i) {
        cout << "Enter value at index " << i << ": ";
        cin >> arr[i];
    }

    //Display array elements
    cout << "Array elements:\n";
    for (int i = 0; i < ARRAY_SIZE; ++i)
        cout << arr[i] << " ";

    //Calculate the sum of array elements
    int sum = 0;
    for (int i = 0; i < ARRAY_SIZE; ++i) {
        sum += arr[i];
    }
    return sum;
```

```
//find the minimum and maximum values in the array
minValue = maxValue = arr[0];
for (int i = 1; i < ARRAY_SIZE; ++i) {
    if (arr[i] < minValue) {
        minValue = arr[i];
    } else if (arr[i] > maxValue) {
        maxValue = arr[i];
    }
}

//sort the array in ascending order

for (int i = 0; i < ARRAY_SIZE; ++i) {
    for (int j = 0; j < ARRAY_SIZE ; ++j) {
        // Swap elements if they are in the wrong order
        if (arr[j] > arr[j + 1]) {
            int temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
        }
    }
}

return 0;
}
```

2.

```
//This program deal with matrix multiplication

#include <iostream>
using namespace std;
const int MATRIX_SIZE = 3;
int main() {
    int matrix[MATRIX_SIZE][MATRIX_SIZE];
    int scalar;
    // Initialize the matrix with user-input values
    cout << "Enter values for a 3x3 matrix:\n";
    for (int i = 0; i < MATRIX_SIZE; ++i) {
        for (int j = 0; j < MATRIX_SIZE; ++j) {
            cout << "Enter value at position (" << i + 1 << ", " << j + 1 << "): ";
            cin >> matrix[i][j];
        }
    }
    // Display the matrix
    cout << "\nMatrix:\n";
    for (int i = 0; i < MATRIX_SIZE; ++i) {
        for (int j = 0; j < MATRIX_SIZE; ++j) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }

    // Calculate and display the transpose of the matrix
    cout << "\nTranspose of the matrix:\n";
    for (int i = 0; i < MATRIX_SIZE; ++i) {
        for (int j = 0; j < MATRIX_SIZE; ++j) {
            cout << matrix[j][i] << " ";
        }
        cout << endl;
    }

    // Find and display the sum of the main diagonal elements
    int sum = 0;
    for (int i = 0; i < MATRIX_SIZE; ++i) {
        sum += matrix[i][i];
    }
    cout << "\nSum of the main diagonal elements: " << sum << endl;

    // Multiply the matrix by a scalar (user-input) and display the result
    cout << "\nEnter a scalar to multiply the matrix by: ";
    cin >> scalar;
    cout << "\nMatrix multiplied by scalar " << scalar << ":\n";
    for (int i = 0; i < MATRIX_SIZE; ++i) {
        for (int j = 0; j < MATRIX_SIZE; ++j) {
            cout << matrix[i][j] * scalar << " ";
        }
        cout << endl;
    }
}
```

3.

```
// 2D arrays / matrix
#include <iostream>
using namespace std;
const int MATRIX_SIZE = 3;
int main() {
    int matrix[MATRIX_SIZE][MATRIX_SIZE];

    // Initialize the matrix with user-input values
    cout << "Enter values for a 3x3 matrix:\n";
    for (int i = 0; i < MATRIX_SIZE; ++i) {
        for (int j = 0; j < MATRIX_SIZE; ++j) {
            cout << "Enter value at position (" << i + 1 << ", " << j + 1 << "): ";
            cin >> matrix[i][j];
        }
    }

    // Display the matrix
    cout << "\nMatrix:\n";
    for (int i = 0; i < MATRIX_SIZE; ++i) {
        for (int j = 0; j < MATRIX_SIZE; ++j) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }

    // Calculate and display the sum of each row and each column
    cout << "\nSum of each row:\n";
    for (int i = 0; i < MATRIX_SIZE; ++i) {
        int rowSum = 0;
        for (int j = 0; j < MATRIX_SIZE; ++j) {
            rowSum += matrix[i][j];
        }
        cout << "Row " << i + 1 << ": " << rowSum << endl;
    }

    cout << "\nSum of each column:\n";
    for (int j = 0; j < MATRIX_SIZE; ++j) {
        int colSum = 0;
        for (int i = 0; i < MATRIX_SIZE; ++i) {
            colSum += matrix[i][j];
        }
        cout << "Column " << j + 1 << ": " << colSum << endl;
    }
}
```

```

// Check and display whether the matrix is symmetric or not
bool symmetric = true;
for (int i = 0; i < MATRIX_SIZE; ++i) {
    for (int j = 0; j < MATRIX_SIZE; ++j) {
        if (matrix[i][j] != matrix[j][i]) {
            symmetric = false;
            break;
        }
    }
    if (!symmetric) {
        break;
    }
}

if (symmetric) {
    cout << "\nThe matrix is symmetric.\n";
} else {

    cout << "\nThe matrix is not symmetric.\n";
}

return 0;
}

```