C++ Assignments | 2D Arrays - 1 | Week 6

Q1: Write a program to store 10 at every index of a 2D matrix with 5 rows and 5 columns.

2.Write a program to add two matrices and save the result in one of the given matrices.

Input 1:

1 2 3
4 5 6
7 8 9

4 5 8
0 0 8
1 2 0

Output 1:

5 7 11
4 5 14
8 10 9

Q3: Given a matrix ‘A’ of dimension n x m and 2 coordinates (l1, r1) and (l2, r2). Return the sum of the
rectangle from (l1,r1) to (l2, r2).

Input 1:

1 2 -3 4
0 0 -4 2
1 -1 2 3
-4 -5 -7 0
l1 = 1, r1 = 2 , l2 = 3 , r2 = 3

Output 1: -4

Input 2:

1 2 -3 4

0 0 -4 2
1 -1 2 3
-4 -5 -7 0
l1 = 1, r1 = 0 , l2 = 0 , r2 = 3

Output 1: 2

Q4: Write a C++ program to find the largest element of a given 2D array of integers.

Input 1:

1 3 4 6
2 4 5 7
3 5 6 8
4 6 7 9

Output 1: 9

Q5: Write a program to print the row number having the maximum sum in a given matrix.

Input 1:

1 3 5 7
3 4 7 8

1 4 12 3
Output 1: 2
Explanation : The 2nd row has the maximum sum i.e. 1+4+12+3 = 20
Q6: Write a function which accepts a 2D array of integers and its size as arguments and displays the
elements of middle row and the elements of middle column.
[Assuming the 2D Array to be a square matrix with odd dimensions i.e. 3x3, 5x5, 7x7 etc...]
Input 1:

```
1 2 3 4 5
3 4 5 6 7
7 6 5 4 3
8 7 6 5 4
1 2 37 8 0
```

Output 1:

```
3
5
7 6 5 4 3
6

37
```

ANSWER:-

```
1.#include <iostream>
using namespace std;

int main() {
   int matrix[5][5];

   for (int i = 0; i < 5; i++) {
      for (int j = 0; j < 5; j++) {
         matrix[i][j] = 10;
      }
   }

   // Display the matrix
   for (int i = 0; i < 5; i++) {
      for (int j = 0; j < 5; j++) {
         cout << matrix[i][j] << " ";
      }
      cout << endl;
   }

   return 0;
}
```

OUTPUT:-

```
10 10 10 10 10
10 10 10 10 10
10 10 10 10 10
```

```
10 10 10 10 10
10 10 10 10 10
```

```
2.#include <iostream>
using namespace std;

int main() {
    int matrix1[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
    int matrix2[3][3] = {
        {4, 5, 8},
        {0, 0, 8},
        {1, 2, 0}
    };

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            matrix1[i][j] += matrix2[i][j];
        }
    }

    // Display the result matrix
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << matrix1[i][j] << " ";
        }
        cout << endl;
    }

    return 0;
}
```

OUTPUT:-

```
5 7 11
4 5 14
8 10 9
```

```
3.#include <iostream>
using namespace std;

int sumRectangle(int matrix[][4], int l1, int r1, int l2, int r2) {
    int sum = 0;
    for (int i = l1; i <= l2; i++) {
        for (int j = r1; j <= r2; j++) {
            sum += matrix[i][j];
        }
    }
    return sum;
}

int main() {
    int matrix[4][4] = {
        {1, 2, -3, 4},
        {0, 0, -4, 2},
        {1, -1, 2, 3},
        {-4, -5, -7, 0}
    };

    int l1 = 1, r1 = 2, l2 = 3, r2 = 3;
    cout << "Sum of the rectangle is: " << sumRectangle(matrix, l1, r1, l2, r2) <<
endl;

    l1 = 1, r1 = 0, l2 = 1, r2 = 3;
    cout << "Sum of the rectangle is: " << sumRectangle(matrix, l1, r1, l2, r2) <<
endl;

    return 0;
}
```

4.

```
#include <iostream>
using namespace std;

int findLargestElement(int matrix[][4], int rows, int cols) {
    int largest = matrix[0][0];
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (matrix[i][j] > largest) {
                largest = matrix[i][j];
            }
        }
    }
    return largest;
}

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

    cout << "The largest element in the matrix is: " <<
findLargestElement(matrix, 4, 4) << endl;
    return 0;
}
```

```
5.#include <iostream>
using namespace std;

int rowWithMaxSum(int matrix[][4], int rows, int cols) {
    int maxSum = 0;
    int rowIndex = 0;
    for (int i = 0; i < rows; i++) {
        int rowSum = 0;
        for (int j = 0; j < cols; j++) {
            rowSum += matrix[i][j];
        }
        if (rowSum > maxSum) {
            maxSum = rowSum;
            rowIndex = i;
        }
    }
    return rowIndex;
}

int main() {
    int matrix[3][4] = {
        {1, 3, 5, 7},
        {3, 4, 7, 8},
        {1, 4, 12, 3}
    };

    cout << "The row with the maximum sum is: " << rowWithMaxSum(matrix, 3,
4) << endl;
    return 0;
}
```

```
6.#include <iostream>
using namespace std;

void displayMiddleRowAndColumn(int matrix[][5], int size) {
    int middle = size / 2;

    cout << "Middle row: ";
    for (int i = 0; i < size; i++) {
        cout << matrix[middle][i] << " ";
    }
    cout << endl;

    cout << "Middle column: ";
    for (int i = 0; i < size; i++) {
        cout << matrix[i][middle] << " ";
    }
    cout << endl;
}

int main() {
    int matrix[5][5] = {
        {1, 2, 3, 4, 5},
        {3, 4, 5, 6, 7},
        {7, 6, 5, 4, 3},
        {8, 7, 6, 5, 4},
        {1, 2, 37, 8, 0}
    };

    displayMiddleRowAndColumn(matrix, 5);

    return 0;
}
OUTPUT:-Middle row: 7 6 5 4 3
Middle column: 3 5 5 6 37
```