Name :- AHMED ALI ASIF

SAP ID :- 55346

Programme :- BSCS 3-1

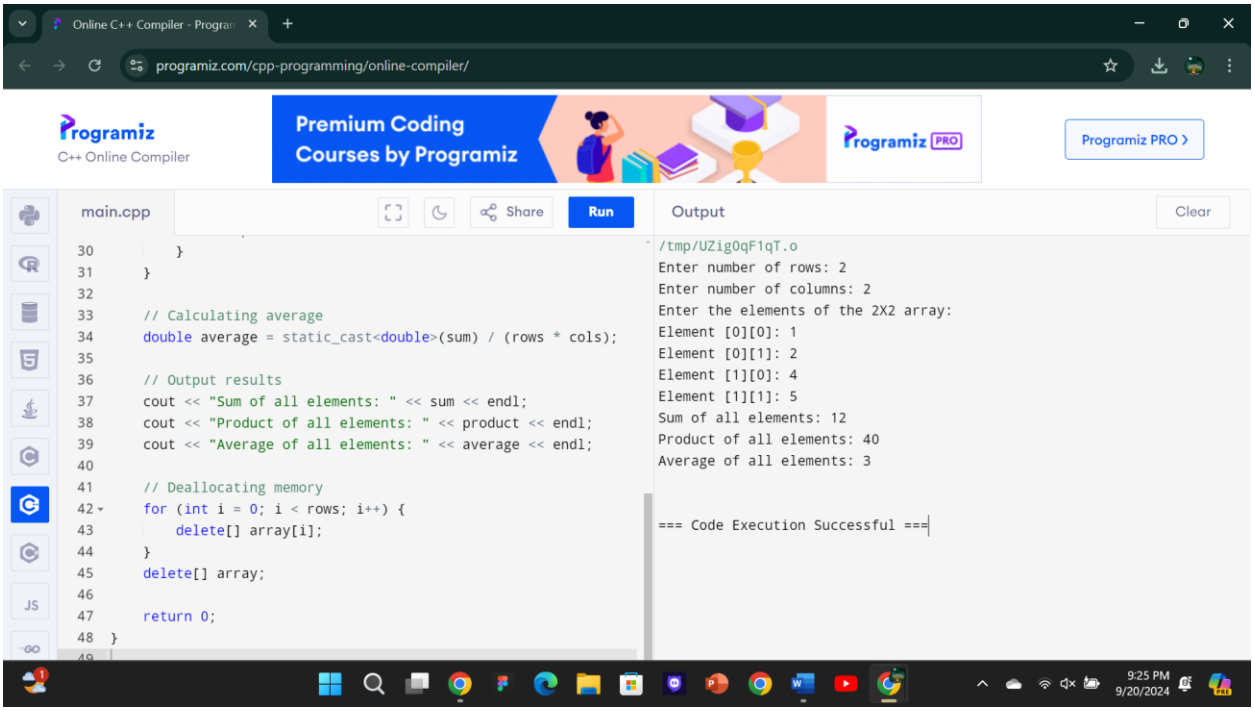
Assignment #2

Q1

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

int main() {
    int rows, cols;



    cout << "Enter number of rows: ";
    cin >> rows;
    cout << "Enter number of columns: ";
    cin >> cols;



    int** array = new int*[rows];
    for (int i = 0; i < rows; i++) {
        array[i] = new int[cols];
    }
```

```
    int sum = 0;
    int product = 1; // Initialize product


    cout << "Enter the elements of the " << rows << "X" << cols << "
array:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cout << "Element [" << i << "][" << j << "]: ";
            cin >> array[i][j];
            sum += array[i][j]; // Add element to sum
            product *= array[i][j]; // Multiply element to product
        }
    }


    double average = static_cast<double>(sum) / (rows * cols);

    cout << "Sum of all elements: " << sum << endl;
    cout << "Product of all elements: " << product << endl;
    cout << "Average of all elements: " << average << endl;


    for (int i = 0; i < rows; i++) {
        delete[] array[i];
```

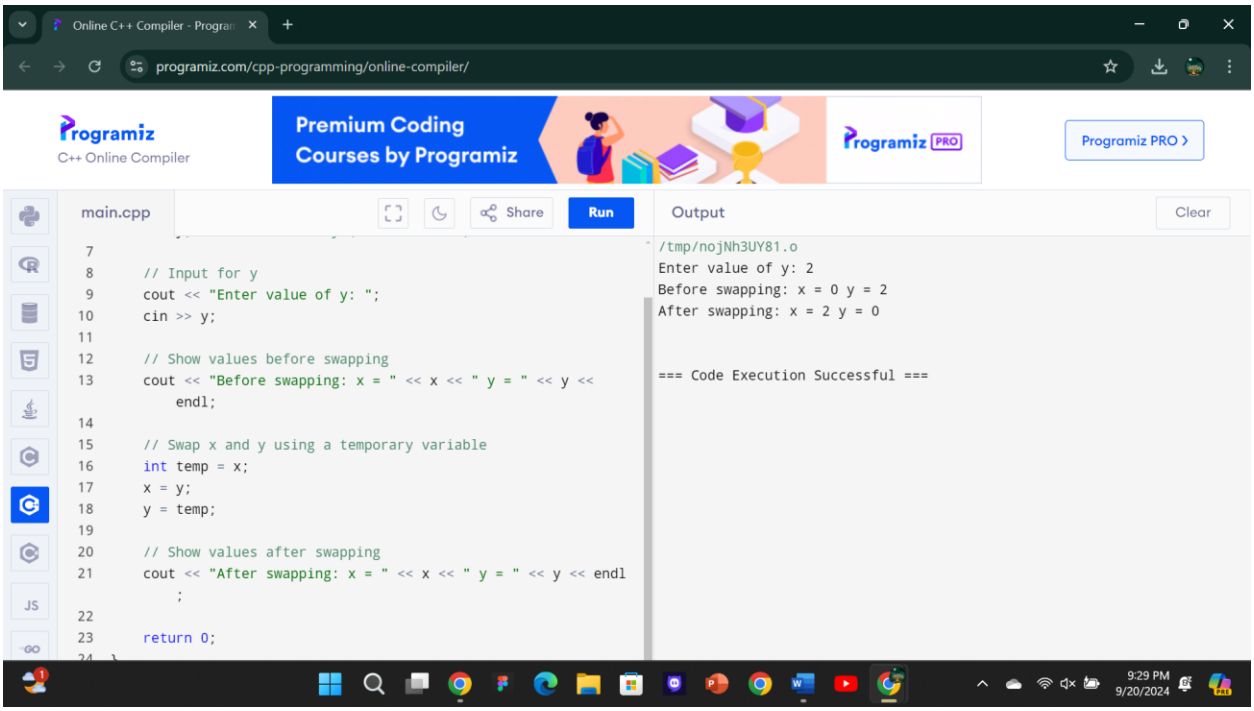
```
    }
    delete[] array;

    return 0;
}
```



Q2

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

int main() {
    int x = 0;
    int y;
```

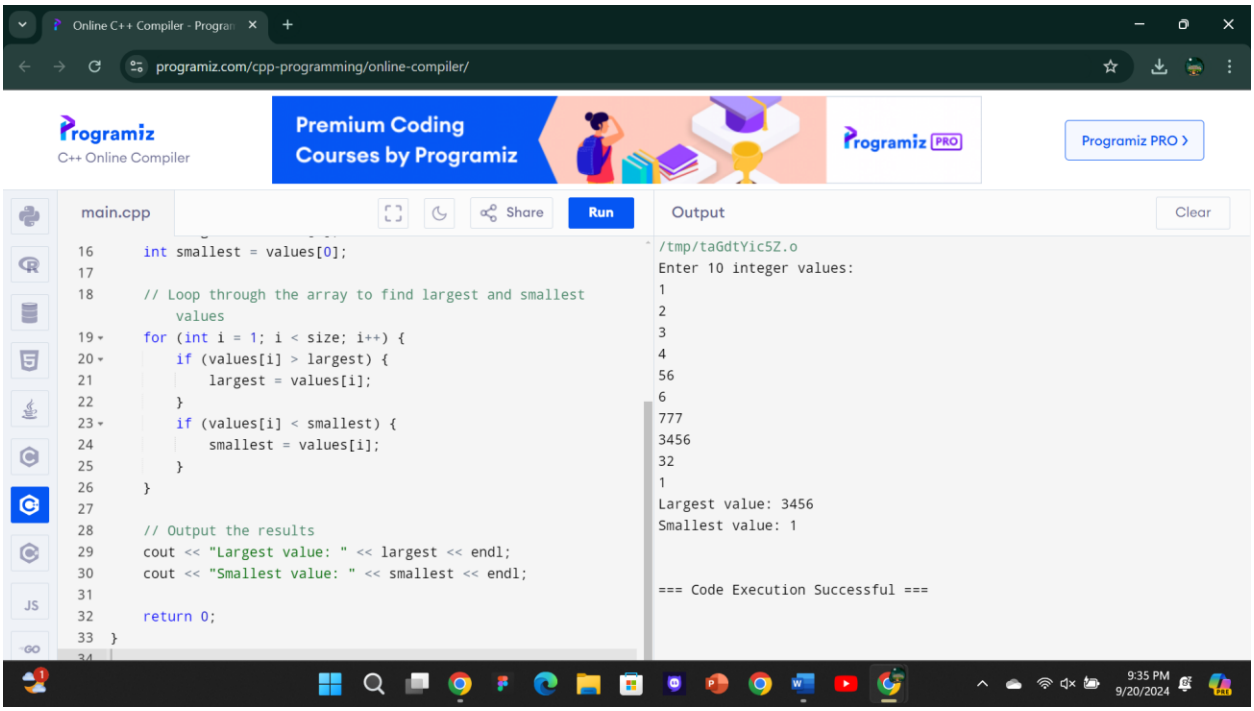
```
    // Input for y
    cout << "Enter value of y: ";
    cin >> y;


    cout << "Before swapping: x = " << x << " y = " << y <<
endl;


    int temp = x;
    x = y;
    y = temp;


    cout << "After swapping: x = " << x << " y = " << y << endl;


    return 0;
}
```



Q3

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

int main() {
    const int size = 10;
    int values[size];


    cout << "Enter 10 integer values: " << endl;
    for (int i = 0; i < size; i++) {
```

```
        cin >> values[i];
    }
    int largest = values[0];
    int smallest = values[0];
    for (int i = 1; i < size; i++) {
        if (values[i] > largest) {
            largest = values[i];
        }
        if (values[i] < smallest) {
            smallest = values[i];
        }
    }


    cout << "Largest value: " << largest << endl;
    cout << "Smallest value: " << smallest << endl;

    return 0;
}
```

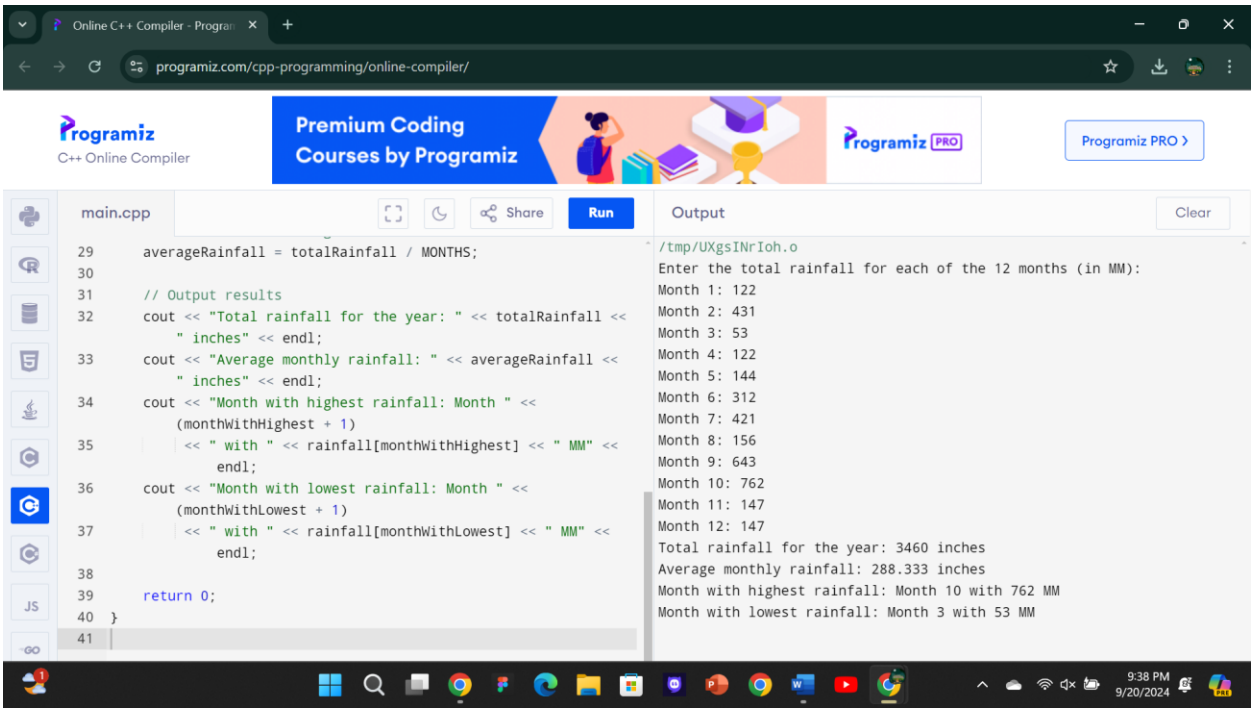


Q4

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

int main() {
    const int MONTHS = 12;
    double rainfall[MONTHS];
    double totalRainfall = 0.0;
    double averageRainfall;
    int monthWithHighest = 0;
    int monthWithLowest = 0;

    // Input rainfall for each month
```

```
    cout << "Enter the total rainfall for each of the 12 months (in MM):" << endl;
    for (int i = 0; i < MONTHS; i++) {
        cout << "Month " << (i + 1) << ": ";
        cin >> rainfall[i];
        totalRainfall += rainfall[i];

        // Compare to find highest and lowest rainfall months
        if (rainfall[i] > rainfall[monthWithHighest]) {
            monthWithHighest = i;
        }
        if (rainfall[i] < rainfall[monthWithLowest]) {
            monthWithLowest = i;
        }
    }

    // Calculate the average rainfall
    averageRainfall = totalRainfall / MONTHS;

    // Output results
    cout << "Total rainfall for the year: " << totalRainfall << " inches" << endl;
```

```
    cout << "Average monthly rainfall: " << averageRainfall <<
" inches" << endl;

    cout << "Month with highest rainfall: Month " <<
(monthWithHighest + 1)

         << " with " << rainfall[monthWithHighest] << " MM" <<
endl;

    cout << "Month with lowest rainfall: Month " <<
(monthWithLowest + 1)

         << " with " << rainfall[monthWithLowest] << " MM" <<
endl;


    return 0;

}
```



Q5

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

const int ROWS = 3;
const int COLS = 4;

int getTotal(int arr[ROWS][COLS]) {
    int total = 0;
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            total += arr[i][j];
        }
    }
    return total;
}

double getAverage(int arr[ROWS][COLS]) {
    int total = getTotal(arr);
    return (double)total / (ROWS * COLS);
}

int getRowTotal(int arr[ROWS][COLS], int row) {
```

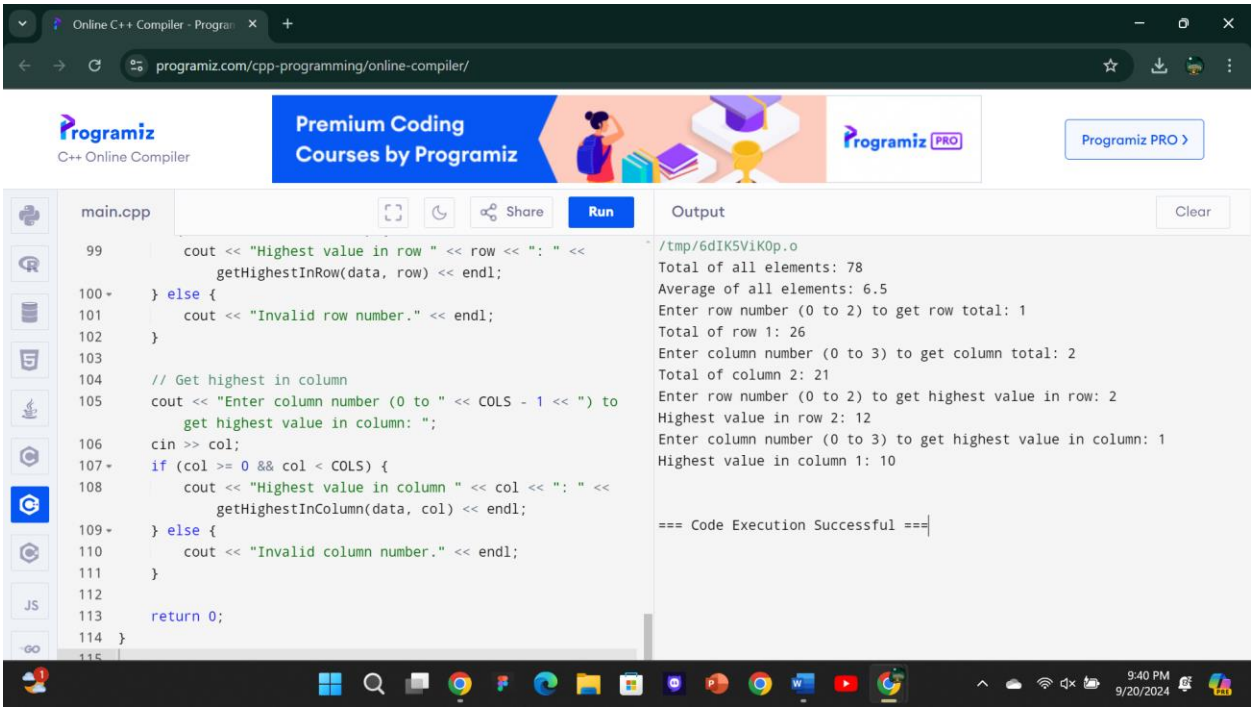
```
    int total = 0;
    for (int j = 0; j < COLS; j++) {
        total += arr[row][j];
    }
    return total;
}

int getColumnTotal(int arr[ROWS][COLS], int col) {
    int total = 0;
    for (int i = 0; i < ROWS; i++) {
        total += arr[i][col];
    }
    return total;
}

int getHighestInRow(int arr[ROWS][COLS], int row) {
    int highest = arr[row][0];
    for (int j = 1; j < COLS; j++) {
        if (arr[row][j] > highest) {
            highest = arr[row][j];
        }
    }
```

```
    return highest;
}

int getHighestInColumn(int arr[ROWS][COLS], int col) {
    int highest = arr[0][col];
    for (int i = 1; i < ROWS; i++) {
        if (arr[i][col] > highest) {
            highest = arr[i][col];
        }
    }
    return highest;
}

int main() {
    int data[ROWS][COLS] = {
        {1, 2, 3, 4},
        {5, 6, 7, 8},
        {9, 10, 11, 12}
    };
    cout << "Total of all elements: " << getTotal(data) << endl;
    cout << "Average of all elements: " << getAverage(data)
<< endl;
```

```
    int row;
    cout << "Enter row number (0 to " << ROWS - 1 << ") to get
row total: ";
    cin >> row;
    if (row >= 0 && row < ROWS) {
        cout << "Total of row " << row << ": " <<
getRowTotal(data, row) << endl;
    } else {
        cout << "Invalid row number." << endl;
    }



    int col;
    cout << "Enter column number (0 to " << COLS - 1 << ") to
get column total: ";
    cin >> col;
    if (col >= 0 && col < COLS) {
        cout << "Total of column " << col << ": " <<
getColumnTotal(data, col) << endl;
    } else {
        cout << "Invalid column number." << endl;
```

```
    }


    cout << "Enter row number (0 to " << ROWS - 1 << ") to get highest value in row: ";
    cin >> row;
    if (row >= 0 && row < ROWS) {
        cout << "Highest value in row " << row << ": " << getHighestInRow(data, row) << endl;
    } else {
        cout << "Invalid row number." << endl;
    }


   cout << "Enter column number (0 to " << COLS - 1 << ") to get highest value in column: ";
    cin >> col;
    if (col >= 0 && col < COLS) {
        cout << "Highest value in column " << col << ": " << getHighestInColumn(data, col) << endl;
    } else {
        cout << "Invalid column number." << endl;
    }
```

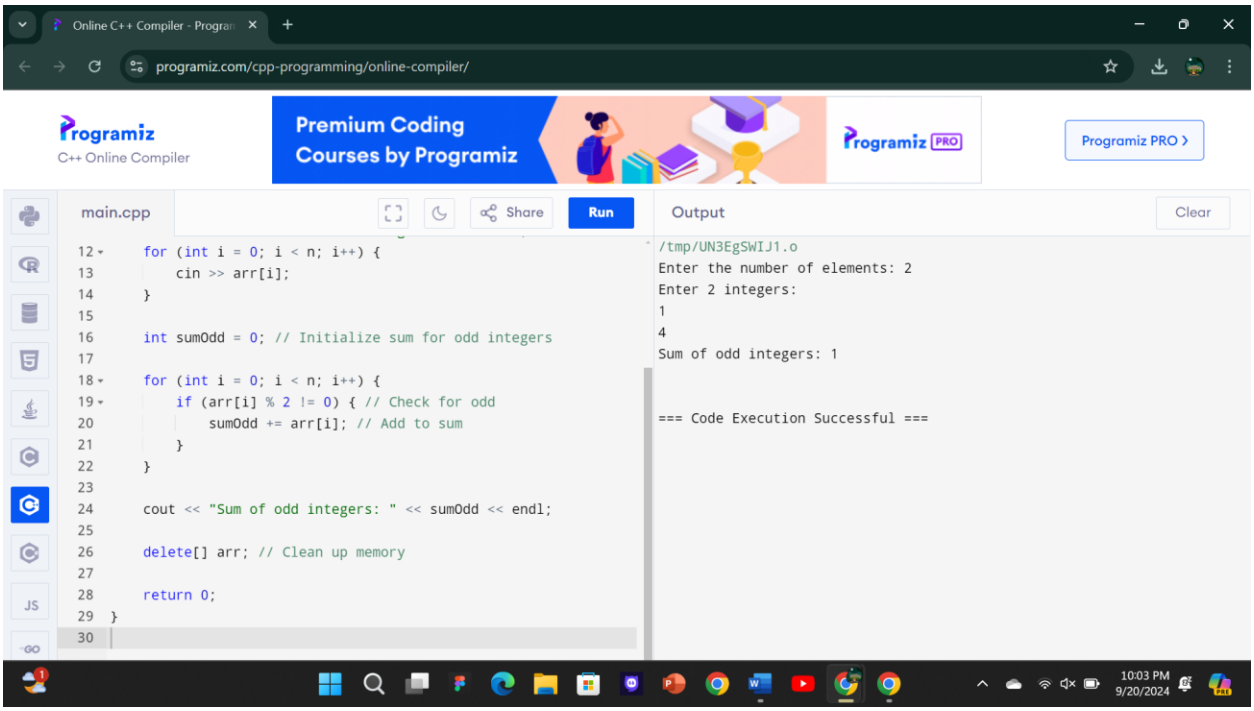
```
    return 0;
}
```



Q6

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

int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;

    int* arr = new int[n]; // Dynamic array
```

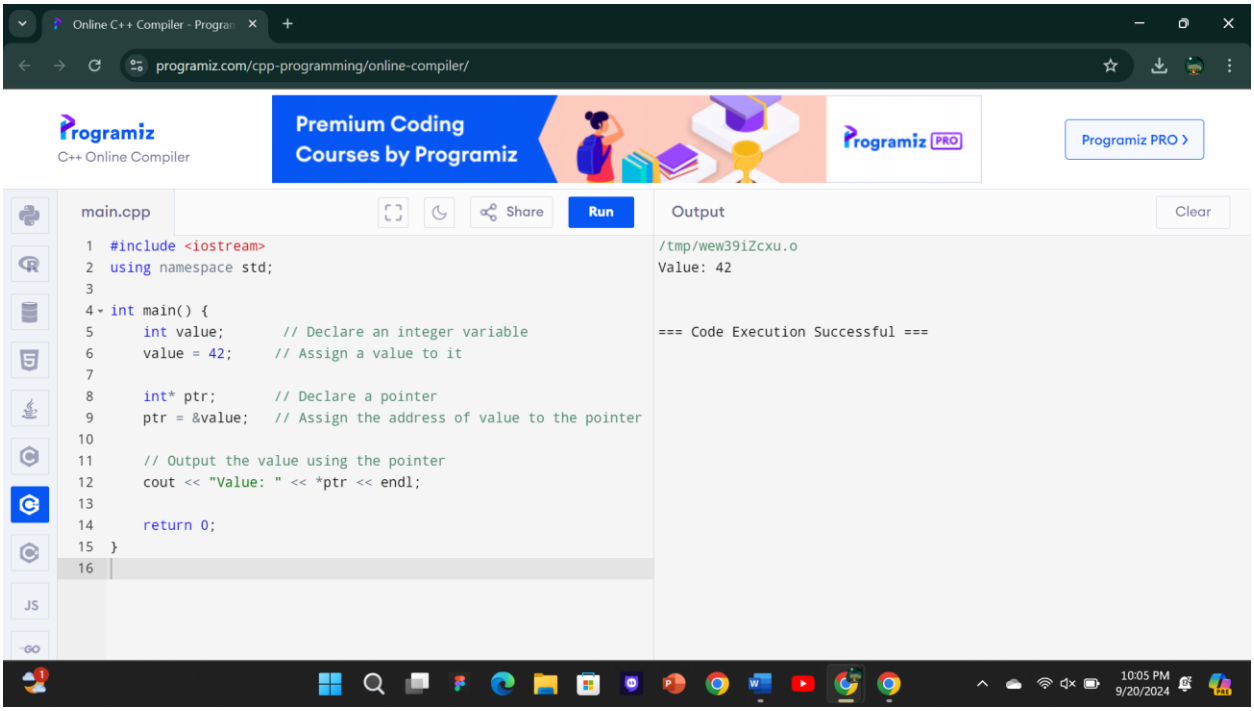
```
    cout << "Enter " << n << " integers: " << endl;
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    int sumOdd = 0; // Initialize sum for odd integers

    for (int i = 0; i < n; i++) {
        if (arr[i] % 2 != 0) {
            sumOdd += arr[i];
        }
    }

    cout << "Sum of odd integers: " << sumOdd << endl;

    delete[] arr;

    return 0;
}
```



Q7

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

int main() {
    int value;
    value = 42;

    int* ptr;
    ptr = &value;

    cout << "Value: " << *ptr << endl;
```

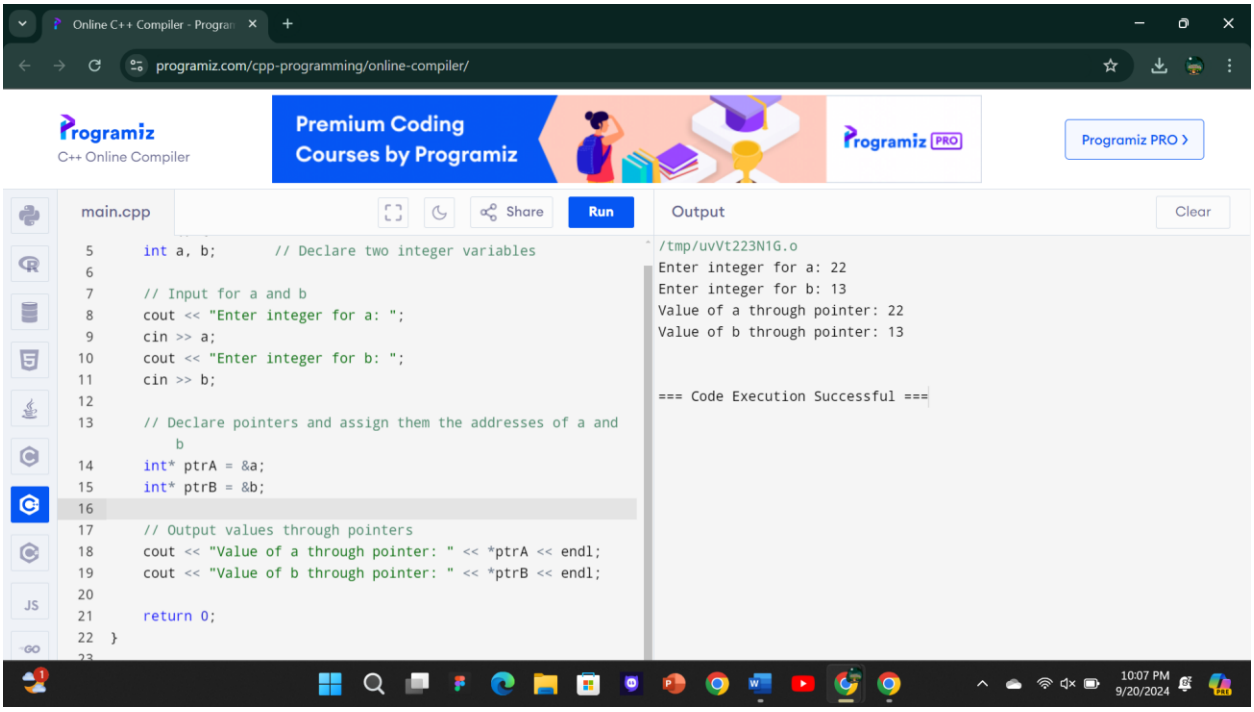
```
    return 0;
}
```



Q8

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

int main() {
    int a, b;
    cout << "Enter integer for a: ";
    cin >> a;
    cout << "Enter integer for b: ";
    cin >> b;
```

```
    int* ptrA = &a;
    int* ptrB = &b;

    cout << "Value of a through pointer: " << *ptrA << endl;
    cout << "Value of b through pointer: " << *ptrB << endl;

    return 0;
}
```



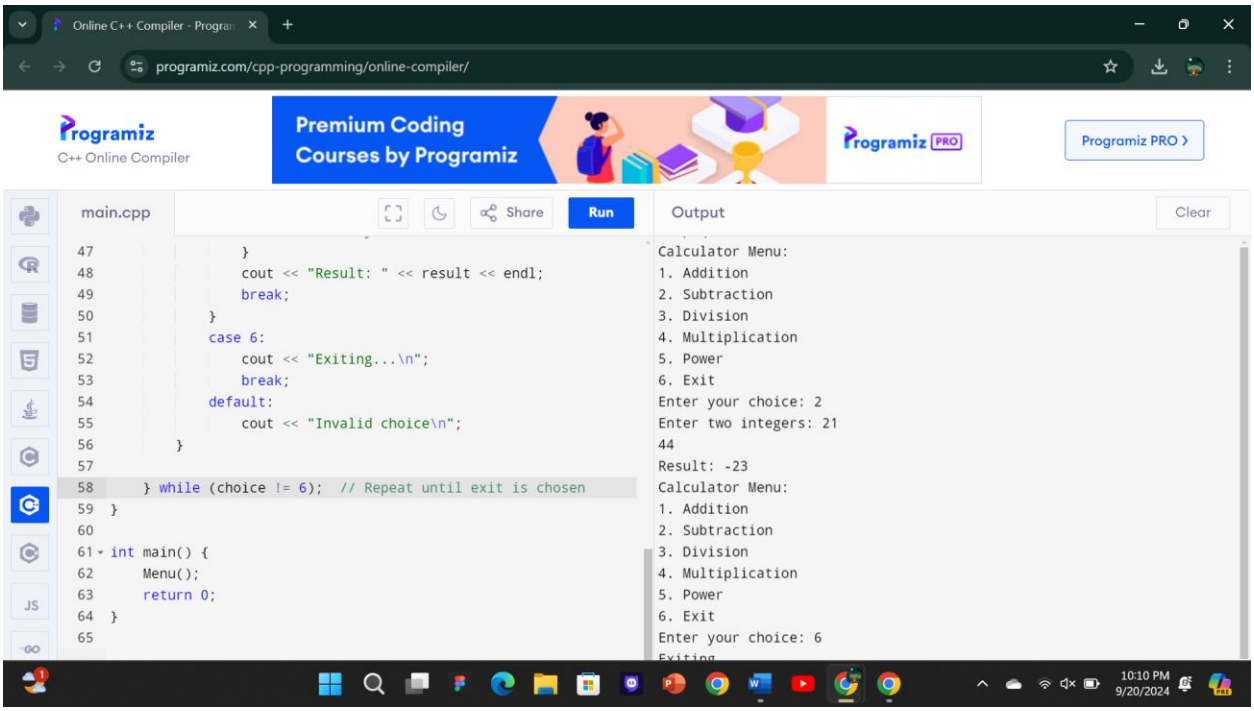
Q9

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

```
void Menu() {
    int choice;
    int a, b;

    do {

        cout << "Calculator Menu:\n";
        cout << "1. Addition\n";
        cout << "2. Subtraction\n";
        cout << "3. Division\n";
        cout << "4. Multiplication\n";
        cout << "5. Power\n";
        cout << "6. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        if (choice >= 1 && choice <= 5) {

            cout << "Enter two integers: ";
            cin >> a >> b;
        }
        switch (choice) {
```

```
            case 1:
                cout << "Result: " << (a + b) << endl;
                break;
            case 2:
                cout << "Result: " << (a - b) << endl;
                break;
            case 3:
                if (b != 0)
                    cout << "Result: " << (static_cast<double>(a) / b)
<< endl;
                else
                    cout << "Error: Division by zero\n";
                break;
            case 4:
                cout << "Result: " << (a * b) << endl;
                break;
            case 5: {
                int result = 1;
                for (int i = 0; i < b; i++) {
                    result = result * a;
                }
                cout << "Result: " << result << endl;
```

```
            break;
        }
        case 6:
            cout << "Exiting...\n";
            break;
        default:
            cout << "Invalid choice\n";
    }

  } while (choice != 6);
}

int main() {
  Menu();
  return 0;
}
```


Online C++ Compiler - Progra
programiz.com/cpp-programming/online-compiler/
Programiz
C++ Online Compiler
Premium Coding
Courses by Programiz
Programiz PRO
Programiz PRO
main.cpp
Share
Run
Output
Clear
47 }
48 cout << "Result: " << result << endl;
49 break;
50 }
51 case 6:
52 cout << "Exiting...\n";
53 break;
54 default:
55 cout << "Invalid choice\n";
56 }
57
58 } while (choice != 6); // Repeat until exit is chosen
59 }
60
61 int main() {
62 Menu();
63 return 0;
64 }
65
Calculator Menu:
1. Addition
2. Subtraction
3. Division
4. Multiplication
5. Power
6. Exit
Enter your choice: 2
Enter two integers: 21
44
Result: -23
Calculator Menu:
1. Addition
2. Subtraction
3. Division
4. Multiplication
5. Power
6. Exit
Enter your choice: 6
Exiting
10:10 PM
9/20/2024