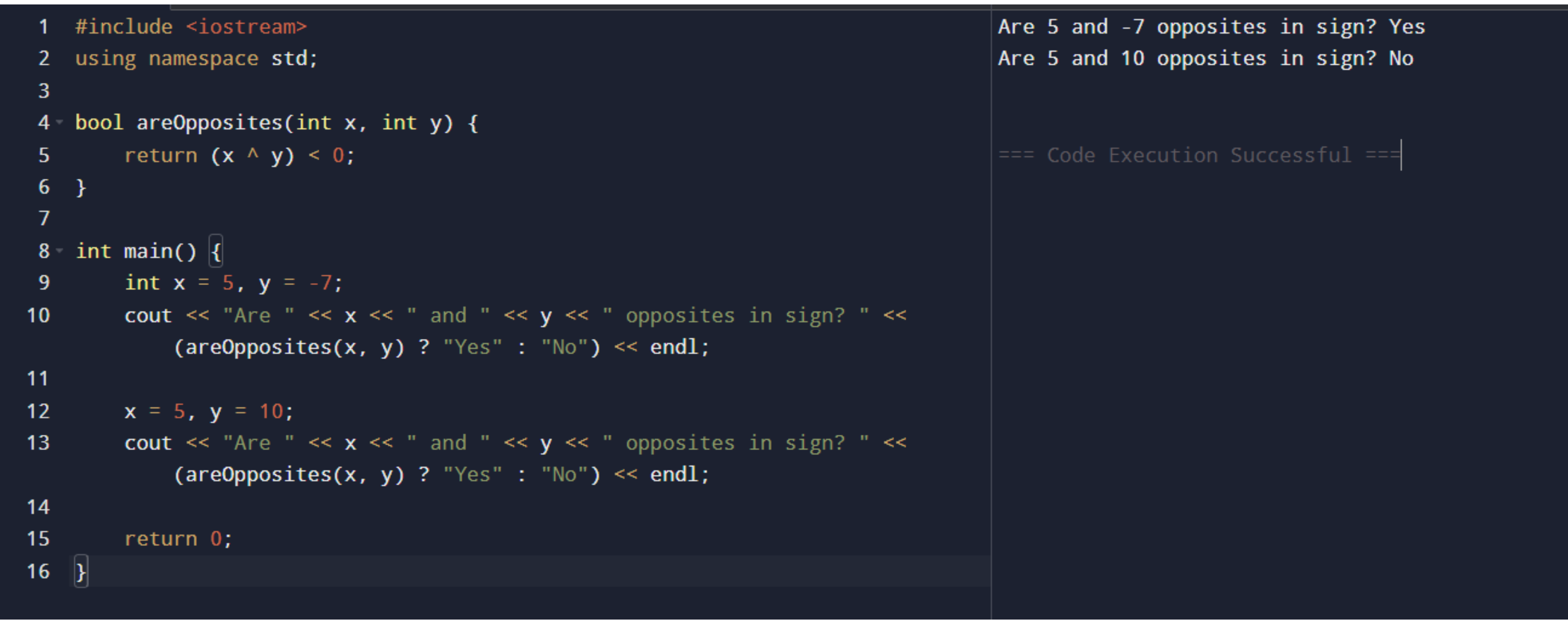
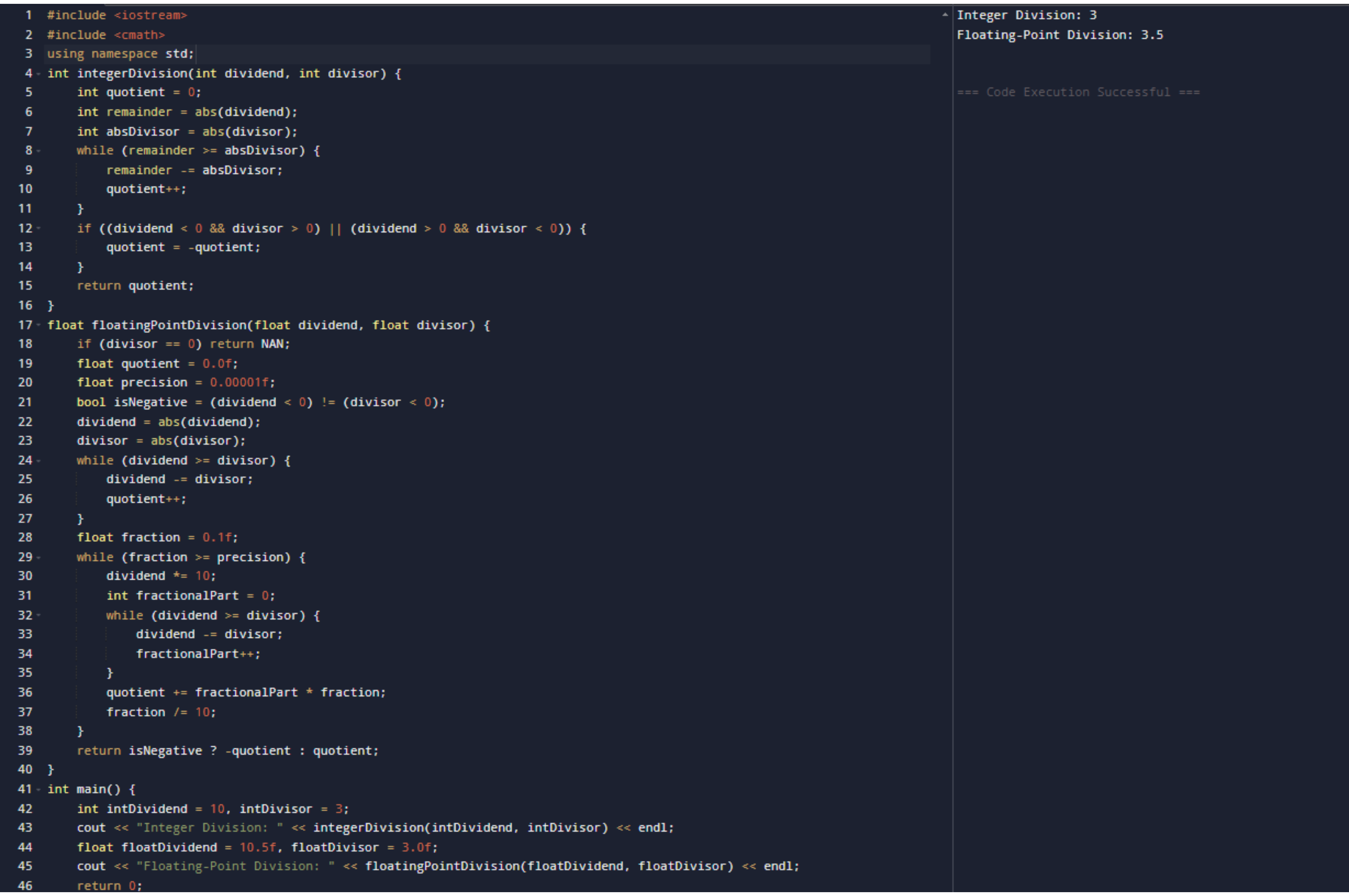


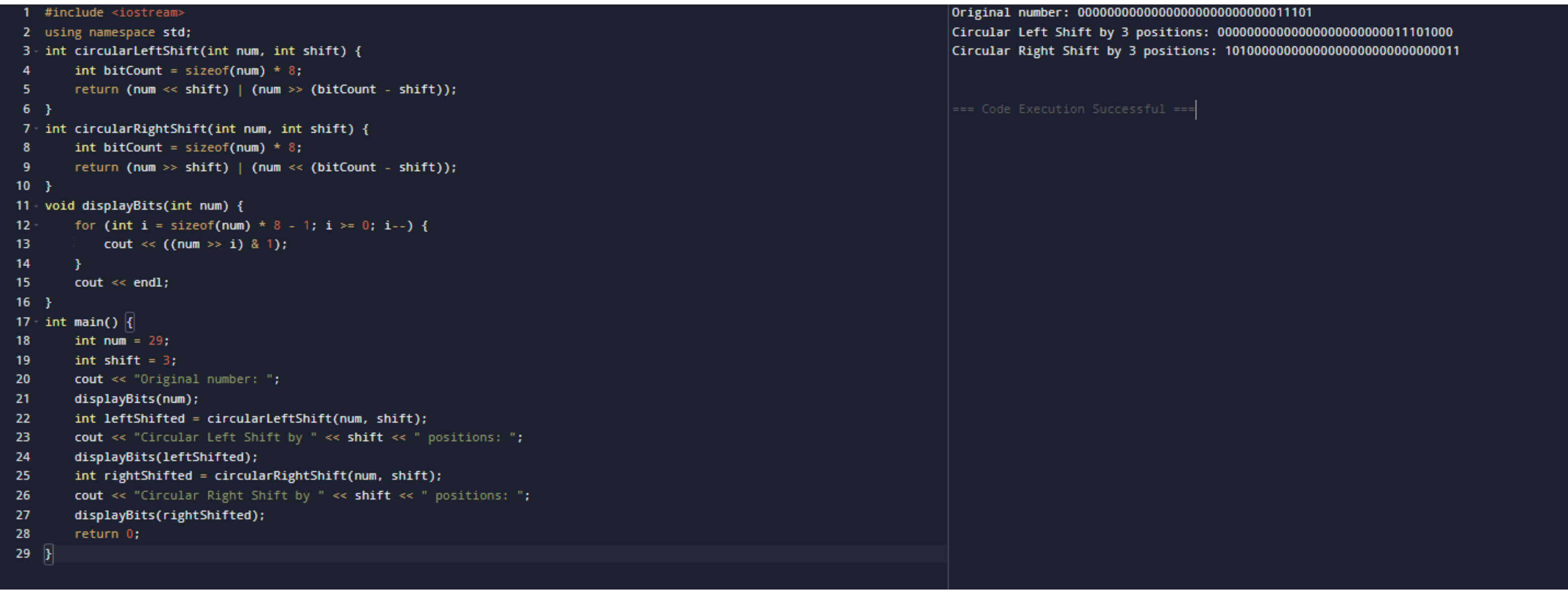
```cpp
#include <iostream>
using namespace std;

bool areOpposites(int x, int y) {
    return (x ^ y) < 0;
}

int main() {
    int x = 5, y = -7;
    cout << "Are " << x << " and " << y << " opposites in sign? " <<
        (areOpposites(x, y) ? "Yes" : "No") << endl;

    x = 5, y = 10;
    cout << "Are " << x << " and " << y << " opposites in sign? " <<
        (areOpposites(x, y) ? "Yes" : "No") << endl;

    return 0;
}
```

```
Are 5 and -7 opposites in sign? Yes
Are 5 and 10 opposites in sign? No


=== Code Execution Successful ===
```



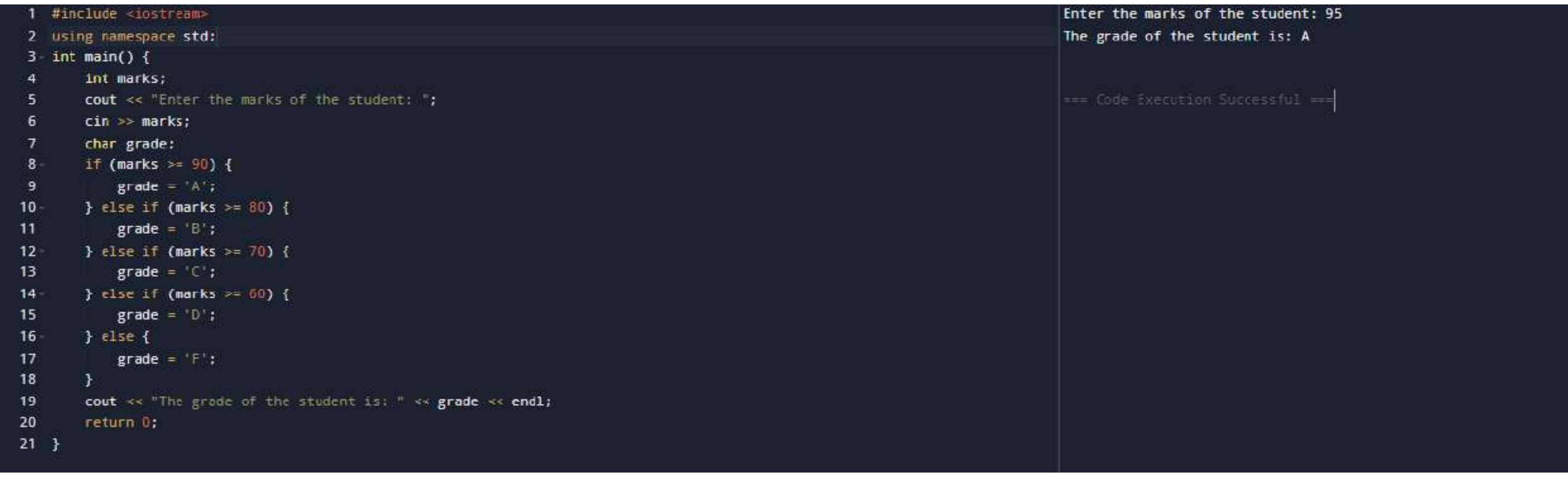
```cpp
#include <iostream>
#include <cmath>
using namespace std;
int integerDivision(int dividend, int divisor) {
    int quotient = 0;
    int remainder = abs(dividend);
    int absDivisor = abs(divisor);
    while (remainder >= absDivisor) {
        remainder -= absDivisor;
        quotient++;
    }
    if ((dividend < 0 && divisor > 0) || (dividend > 0 && divisor < 0)) {
        quotient = -quotient;
    }
    return quotient;
}
float floatingPointDivision(float dividend, float divisor) {
    if (divisor == 0) return NAN;
    float quotient = 0.0f;
    float precision = 0.00001f;
    bool isNegative = (dividend < 0) != (divisor < 0);
    dividend = abs(dividend);
    divisor = abs(divisor);
    while (dividend >= divisor) {
        dividend -= divisor;
        quotient++;
    }
    float fraction = 0.1f;
    while (fraction >= precision) {
        dividend *= 10;
        int fractionalPart = 0;
        while (dividend >= divisor) {
            dividend -= divisor;
            fractionalPart++;
        }
        quotient += fractionalPart * fraction;
        fraction /= 10;
    }
    return isNegative ? -quotient : quotient;
}
int main() {
    int intDividend = 10, intDivisor = 3;
    cout << "Integer Division: " << integerDivision(intDividend, intDivisor) << endl;
    float floatDividend = 10.5f, floatDivisor = 3.0f;
    cout << "Floating-Point Division: " << floatingPointDivision(floatDividend, floatDivisor) << endl;
    return 0;
```

```
Integer Division: 3
Floating-Point Division: 3.5


=== Code Execution Successful ===
```



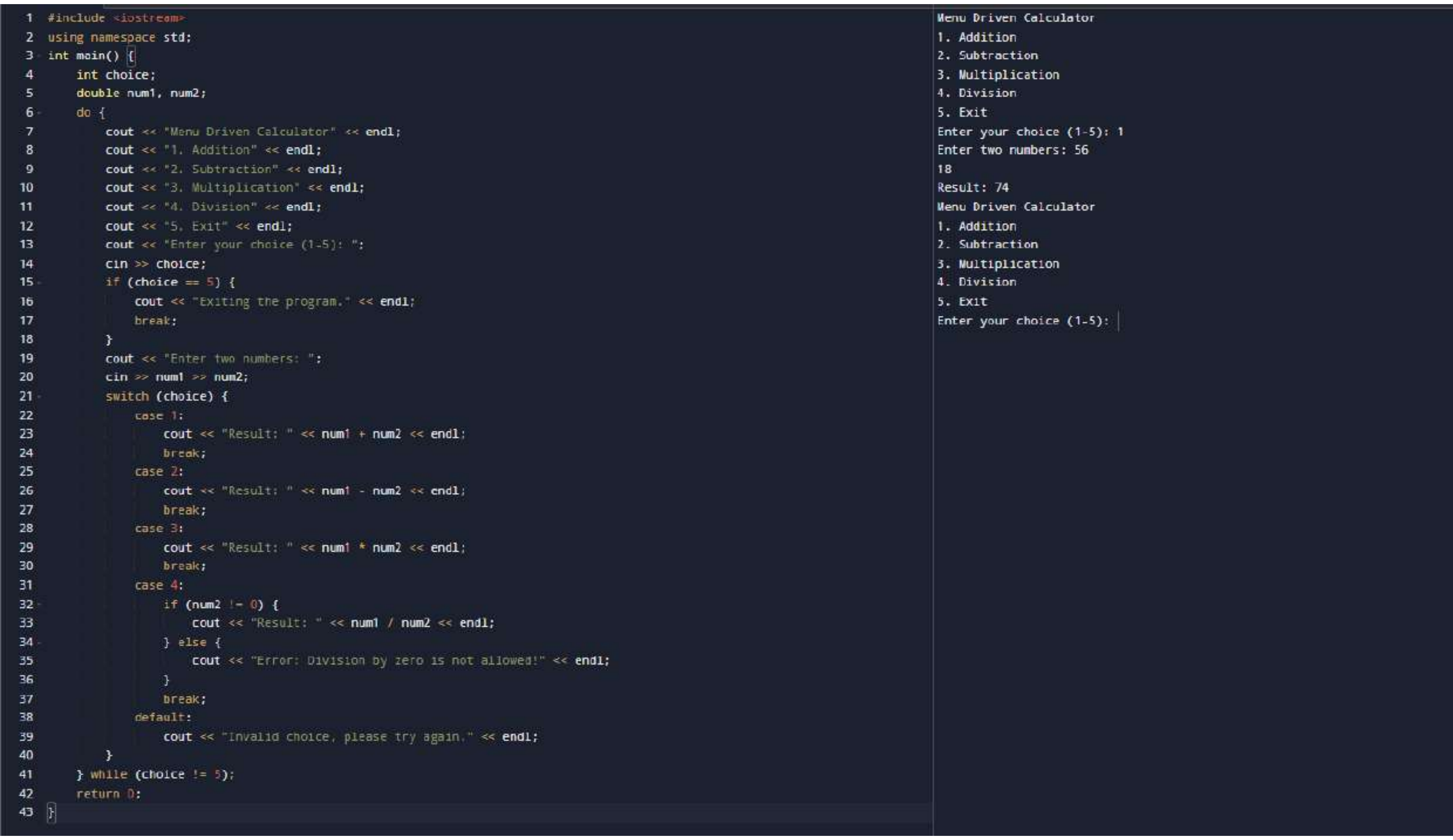
```cpp
#include <iostream>
using namespace std;
int circularLeftShift(int num, int shift) {
    int bitCount = sizeof(num) * 8;
    return (num << shift) | (num >> (bitCount - shift));
}
int circularRightShift(int num, int shift) {
    int bitCount = sizeof(num) * 8;
    return (num >> shift) | (num << (bitCount - shift));
}
void displayBits(int num) {
    for (int i = sizeof(num) * 8 - 1; i >= 0; i--) {
        cout << ((num >> i) & 1);
    }
    cout << endl;
}
int main() {
    int num = 29;
    int shift = 3;
    cout << "Original number: ";
    displayBits(num);
    int leftShifted = circularLeftShift(num, shift);
    cout << "Circular Left Shift by " << shift << " positions: ";
    displayBits(leftShifted);
    int rightShifted = circularRightShift(num, shift);
    cout << "Circular Right Shift by " << shift << " positions: ";
    displayBits(rightShifted);
    return 0;
}
```

```
Original number: 00000000000000000000000000011101
Circular Left Shift by 3 positions: 00000000000000000000000011101000
Circular Right Shift by 3 positions: 10100000000000000000000000000011


=== Code Execution Successful ===
```



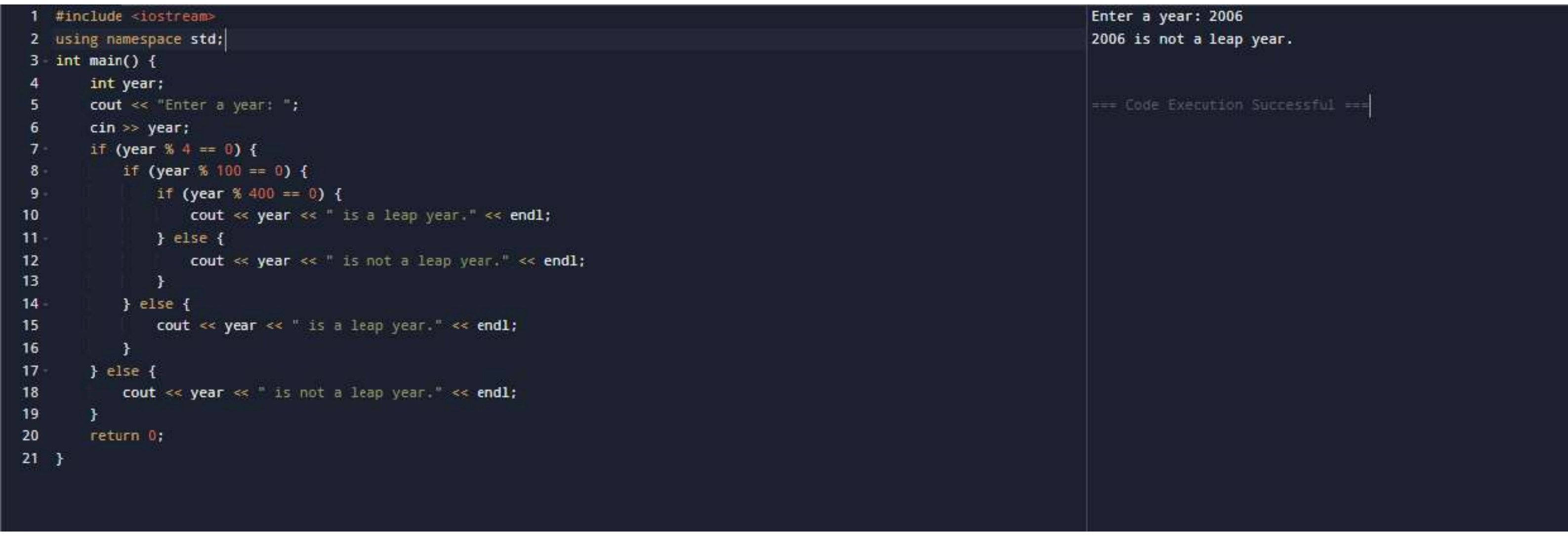
```cpp
#include <iostream>
using namespace std:
int main() {
    int marks;
    cout << "Enter the marks of the student: ";
    cin >> marks;
    char grade:
    if (marks >= 90) {
        grade = 'A';
    } else if (marks >= 80) {
        grade = 'B';
    } else if (marks >= 70) {
        grade = 'C';
    } else if (marks >= 60) {
        grade = 'D';
    } else {
        grade = 'F';
    }
    cout << "The grade of the student is: " << grade << endl;
    return 0;
}
```

```
Enter the marks of the student: 95
The grade of the student is: A


=== Code Execution Successful ===
```



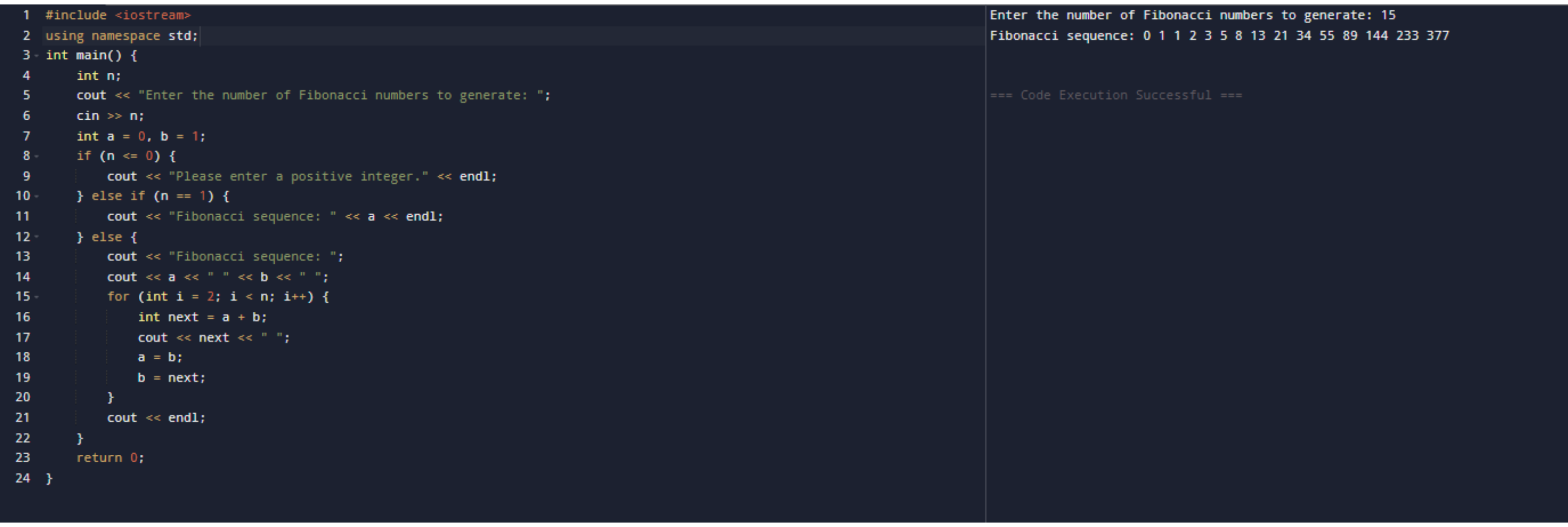
```cpp
#include <iostream>
using namespace std;
int main() {
    int choice;
    double num1, num2;
    do {
        cout << "Menu Driven Calculator" << endl;
        cout << "1. Addition" << endl;
        cout << "2. Subtraction" << endl;
        cout << "3. Multiplication" << endl;
        cout << "4. Division" << endl;
        cout << "5. Exit" << endl;
        cout << "Enter your choice (1-5): ";
        cin >> choice;
        if (choice == 5) {
            cout << "Exiting the program." << endl;
            break;
        }
        cout << "Enter two numbers: ";
        cin >> num1 >> num2;
        switch (choice) {
            case 1:
                cout << "Result: " << num1 + num2 << endl;
                break;
            case 2:
                cout << "Result: " << num1 - num2 << endl;
                break;
            case 3:
                cout << "Result: " << num1 * num2 << endl;
                break;
            case 4:
                if (num2 != 0) {
                    cout << "Result: " << num1 / num2 << endl;
                } else {
                    cout << "Error: Division by zero is not allowed!" << endl;
                }
                break;
            default:
                cout << "Invalid choice, please try again." << endl;
        }
    } while (choice != 5);
    return 0;
}
```

```
Menu Driven Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice (1-5): 1
Enter two numbers: 56
18
Result: 74
Menu Driven Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice (1-5):
```



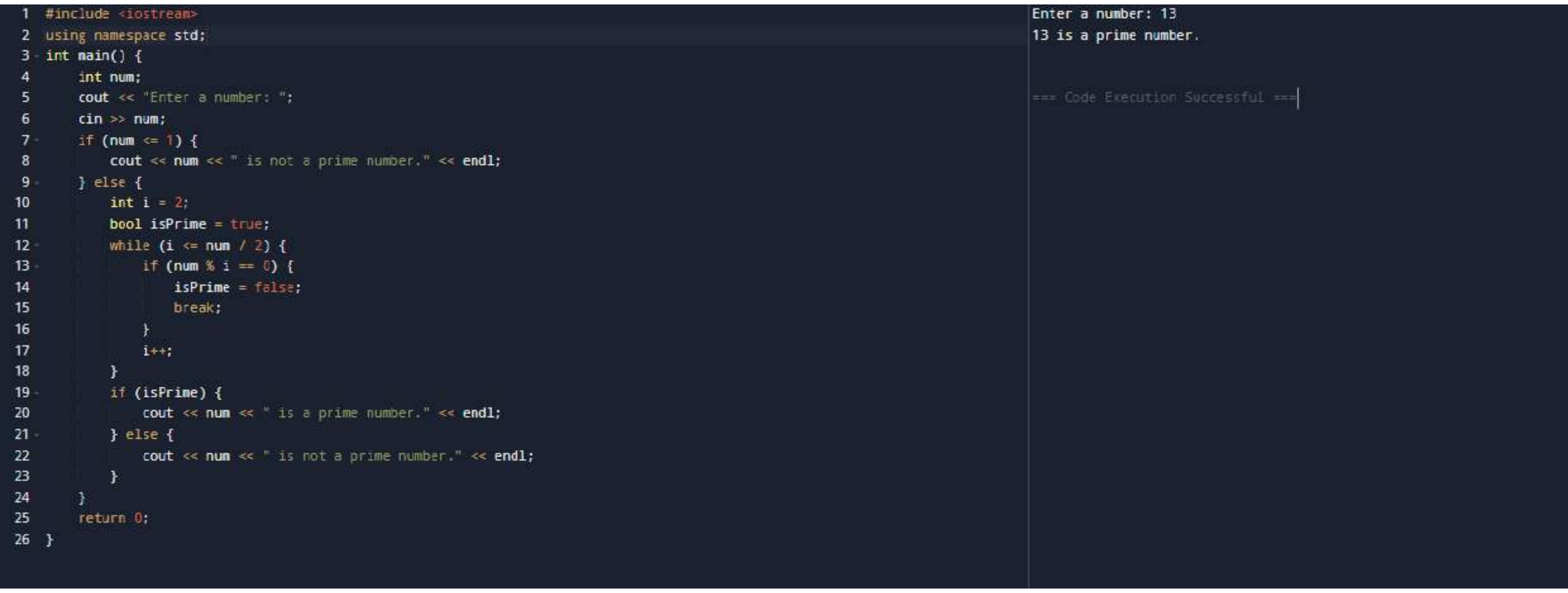
```cpp
#include <iostream>
using namespace std;
int main() {
    int year;
    cout << "Enter a year: ";
    cin >> year;
    if (year % 4 == 0) {
        if (year % 100 == 0) {
            if (year % 400 == 0) {
                cout << year << " is a leap year." << endl;
            } else {
                cout << year << " is not a leap year." << endl;
            }
        } else {
            cout << year << " is a leap year." << endl;
        }
    } else {
        cout << year << " is not a leap year." << endl;
    }
    return 0;
}
```

```
Enter a year: 2006
2006 is not a leap year.


=== Code Execution Successful ===
```



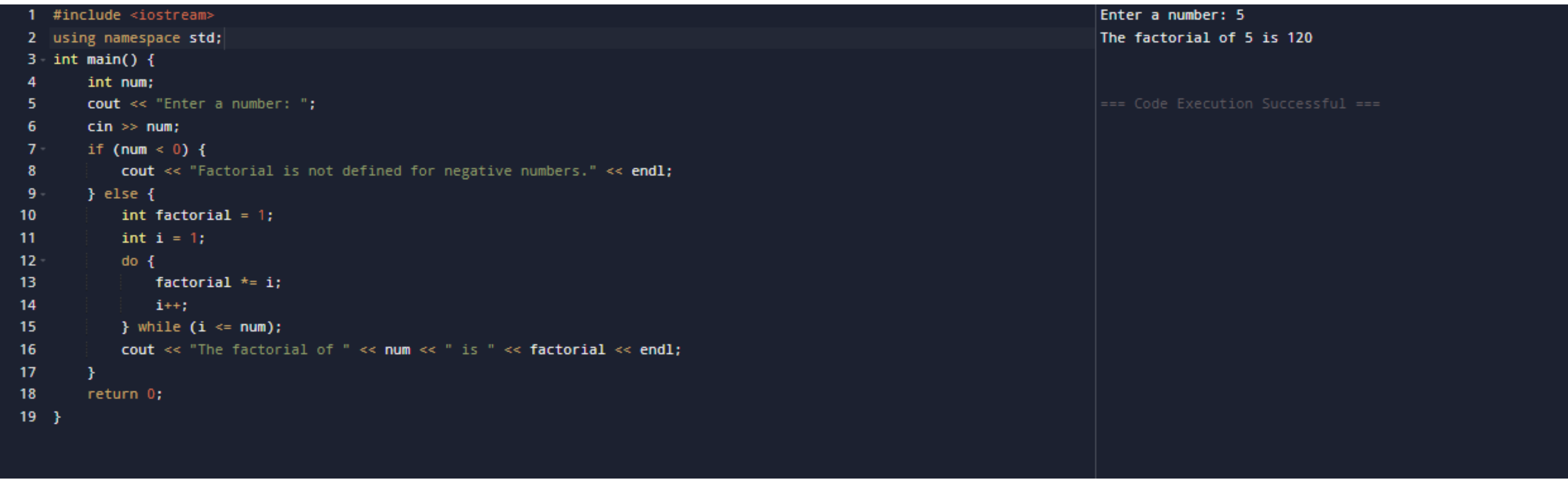
```cpp
#include <iostream>
using namespace std;
int main() {
    int n;
    cout << "Enter the number of Fibonacci numbers to generate: ";
    cin >> n;
    int a = 0, b = 1;
    if (n <= 0) {
        cout << "Please enter a positive integer." << endl;
    } else if (n == 1) {
        cout << "Fibonacci sequence: " << a << endl;
    } else {
        cout << "Fibonacci sequence: ";
        cout << a << " " << b << " ";
        for (int i = 2; i < n; i++) {
            int next = a + b;
            cout << next << " ";
            a = b;
            b = next;
        }
        cout << endl;
    }
    return 0;
}
```

```
Enter the number of Fibonacci numbers to generate: 15
Fibonacci sequence: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377


=== Code Execution Successful ===
```



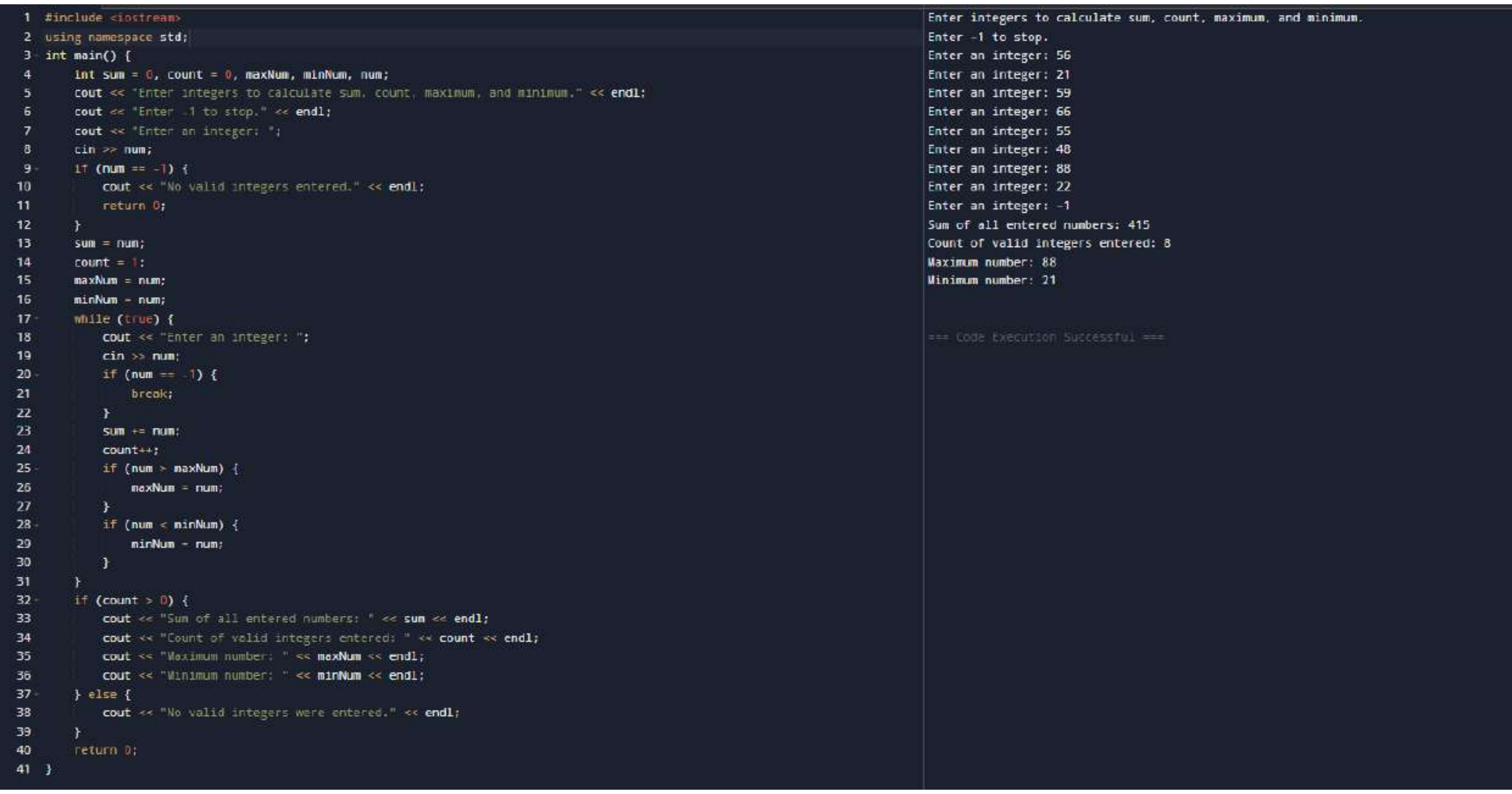
```
#include <iostream>
using namespace std;
int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    if (num <= 1) {
        cout << num << " is not a prime number." << endl;
    } else {
        int i = 2;
        bool isPrime = true;
        while (i <= num / 2) {
            if (num % i == 0) {
                isPrime = false;
                break;
            }
            i++;
        }
        if (isPrime) {
            cout << num << " is a prime number." << endl;
        } else {
            cout << num << " is not a prime number." << endl;
        }
    }
    return 0;
}
```

```
Enter a number: 13
13 is a prime number.


=== Code Execution Successful ===
```



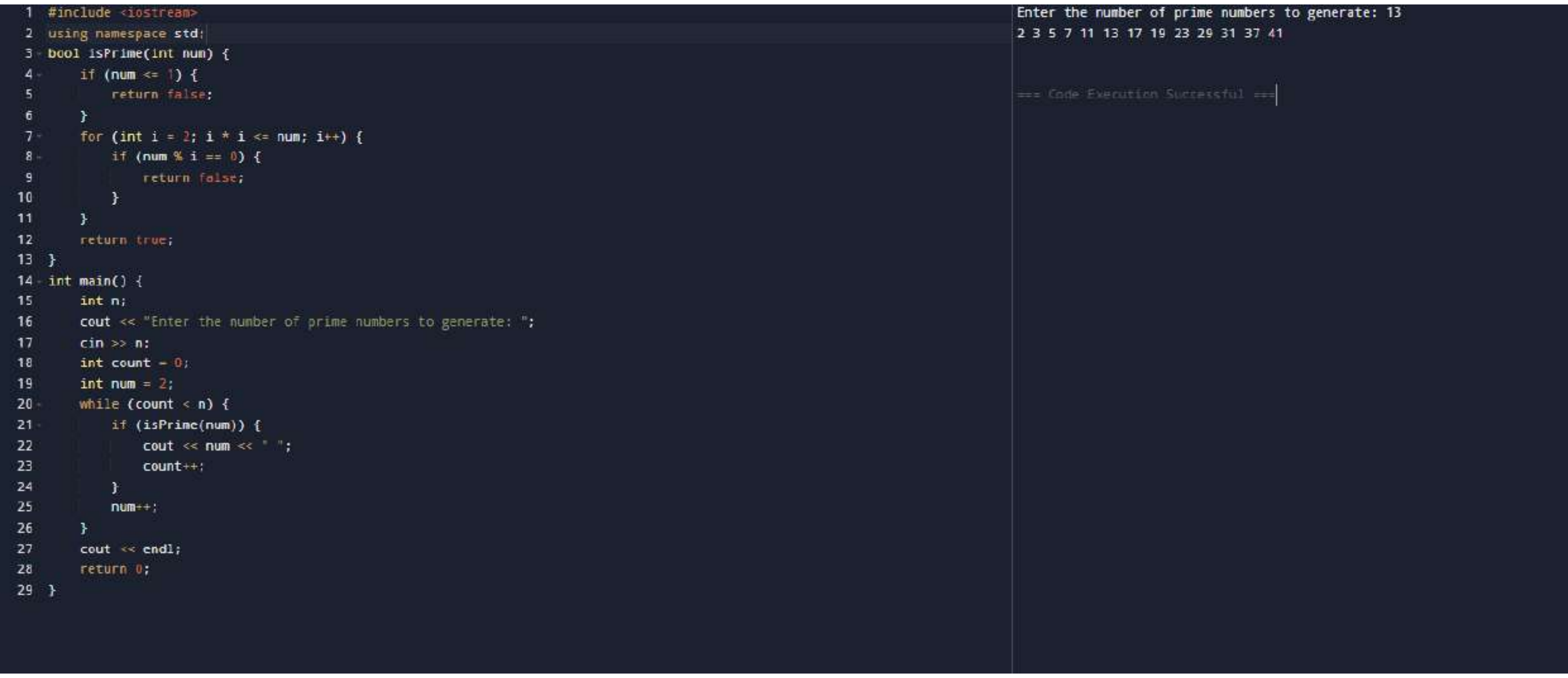
```cpp
#include <iostream>
using namespace std;
int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    if (num < 0) {
        cout << "Factorial is not defined for negative numbers." << endl;
    } else {
        int factorial = 1;
        int i = 1;
        do {
            factorial *= i;
            i++;
        } while (i <= num);
        cout << "The factorial of " << num << " is " << factorial << endl;
    }
    return 0;
}
```

```
Enter a number: 5
The factorial of 5 is 120


=== Code Execution Successful ===
```



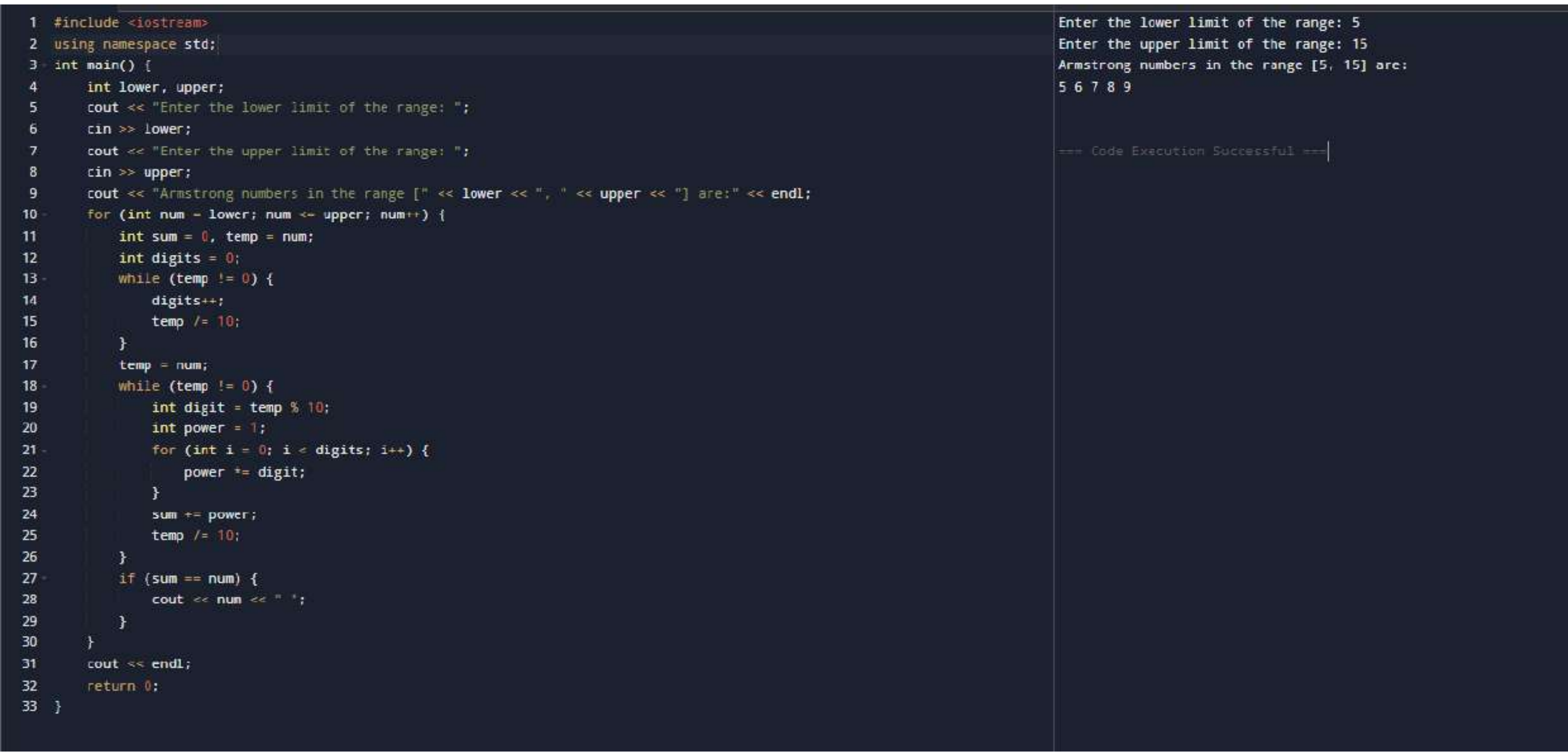
```cpp
#include <iostream>
using namespace std;
int main() {
    int sum = 0, count = 0, maxNum, minNum, num;
    cout << "Enter integers to calculate sum, count, maximum, and minimum." << endl;
    cout << "Enter -1 to stop." << endl;
    cout << "Enter an integer: ";
    cin >> num;
    if (num == -1) {
        cout << "No valid integers entered." << endl;
        return 0;
    }
    sum = num;
    count = 1;
    maxNum = num;
    minNum = num;
    while (true) {
        cout << "Enter an integer: ";
        cin >> num;
        if (num == -1) {
            break;
        }
        sum += num;
        count++;
        if (num > maxNum) {
            maxNum = num;
        }
        if (num < minNum) {
            minNum = num;
        }
    }
    if (count > 0) {
        cout << "Sum of all entered numbers: " << sum << endl;
        cout << "Count of valid integers entered: " << count << endl;
        cout << "Maximum number: " << maxNum << endl;
        cout << "Minimum number: " << minNum << endl;
    } else {
        cout << "No valid integers were entered." << endl;
    }
    return 0;
}
```

```
Enter integers to calculate sum, count, maximum, and minimum.
Enter -1 to stop.
Enter an integer: 56
Enter an integer: 21
Enter an integer: 59
Enter an integer: 66
Enter an integer: 55
Enter an integer: 48
Enter an integer: 88
Enter an integer: 22
Enter an integer: -1
Sum of all entered numbers: 415
Count of valid integers entered: 8
Maximum number: 88
Minimum number: 21


=== Code Execution Successful ===
```



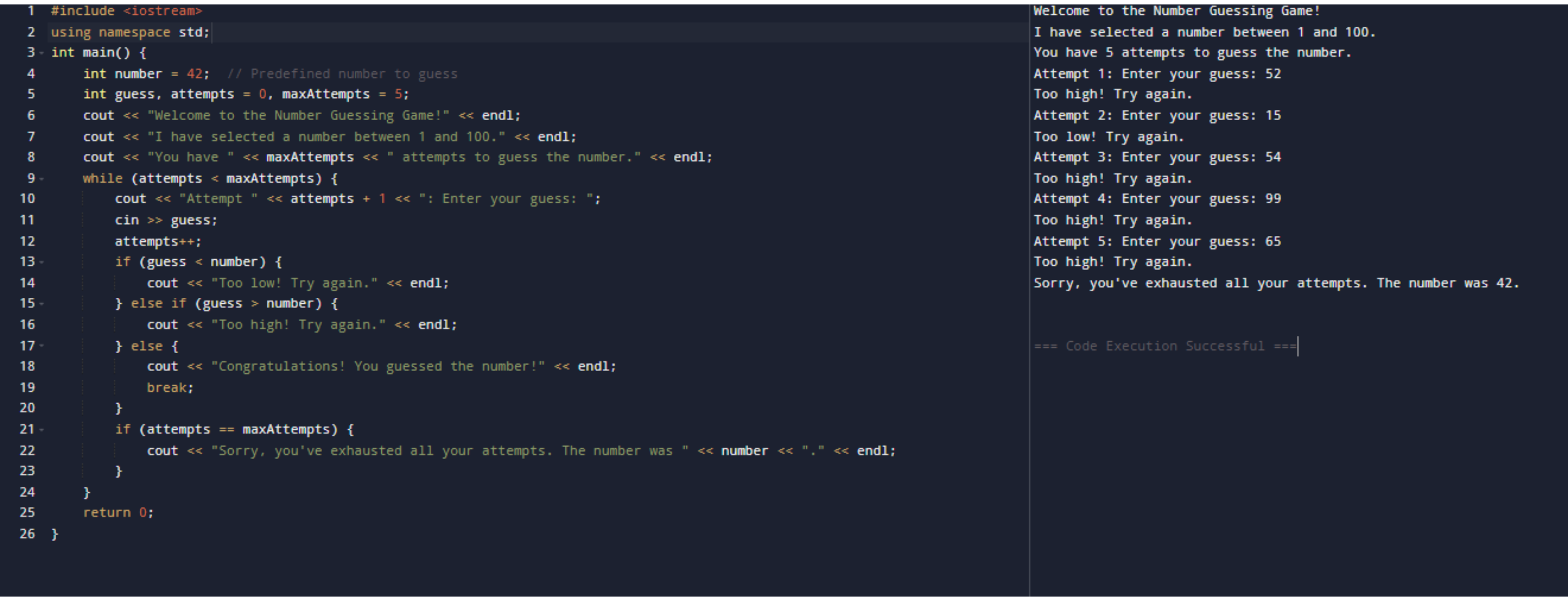
```cpp
#include <iostream>
using namespace std;
bool isPrime(int num) {
    if (num <= 1) {
        return false;
    }
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}
int main() {
    int n;
    cout << "Enter the number of prime numbers to generate: ";
    cin >> n;
    int count = 0;
    int num = 2;
    while (count < n) {
        if (isPrime(num)) {
            cout << num << " ";
            count++;
        }
        num++;
    }
    cout << endl;
    return 0;
}
```

```
Enter the number of prime numbers to generate: 13
2 3 5 7 11 13 17 19 23 29 31 37 41


=== Code Execution Successful ===
```



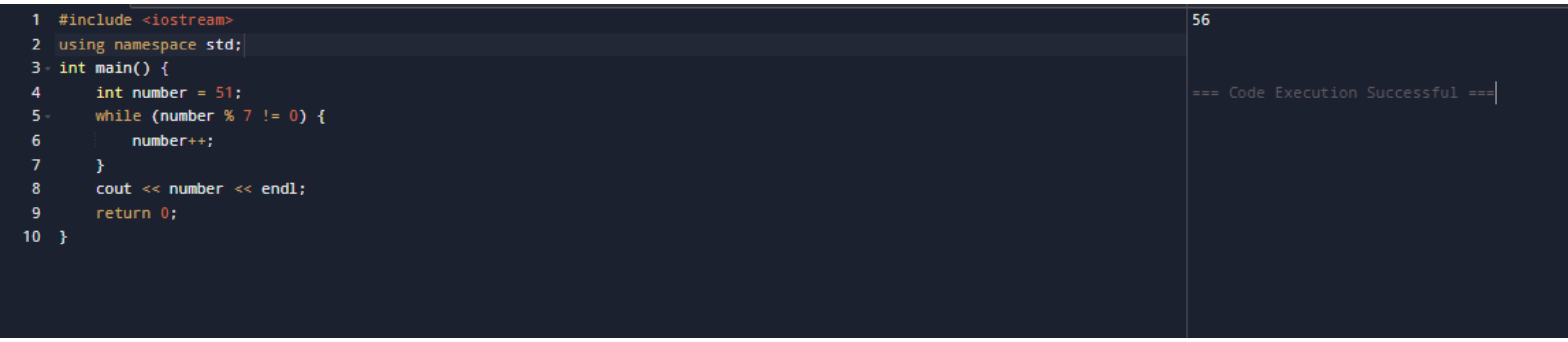
```cpp
#include <iostream>
using namespace std;
int main() {
    int lower, upper;
    cout << "Enter the lower limit of the range: ";
    cin >> lower;
    cout << "Enter the upper limit of the range: ";
    cin >> upper;
    cout << "Armstrong numbers in the range [" << lower << ", " << upper << "] are:" << endl;
    for (int num = lower; num <= upper; num++) {
        int sum = 0, temp = num;
        int digits = 0;
        while (temp != 0) {
            digits++;
            temp /= 10;
        }
        temp = num;
        while (temp != 0) {
            int digit = temp % 10;
            int power = 1;
            for (int i = 0; i < digits; i++) {
                power *= digit;
            }
            sum += power;
            temp /= 10;
        }
        if (sum == num) {
            cout << num << " ";
        }
    }
    cout << endl;
    return 0;
}
```

```
Enter the lower limit of the range: 5
Enter the upper limit of the range: 15
Armstrong numbers in the range [5, 15] are:
5 6 7 8 9


=== Code Execution Successful ===
```



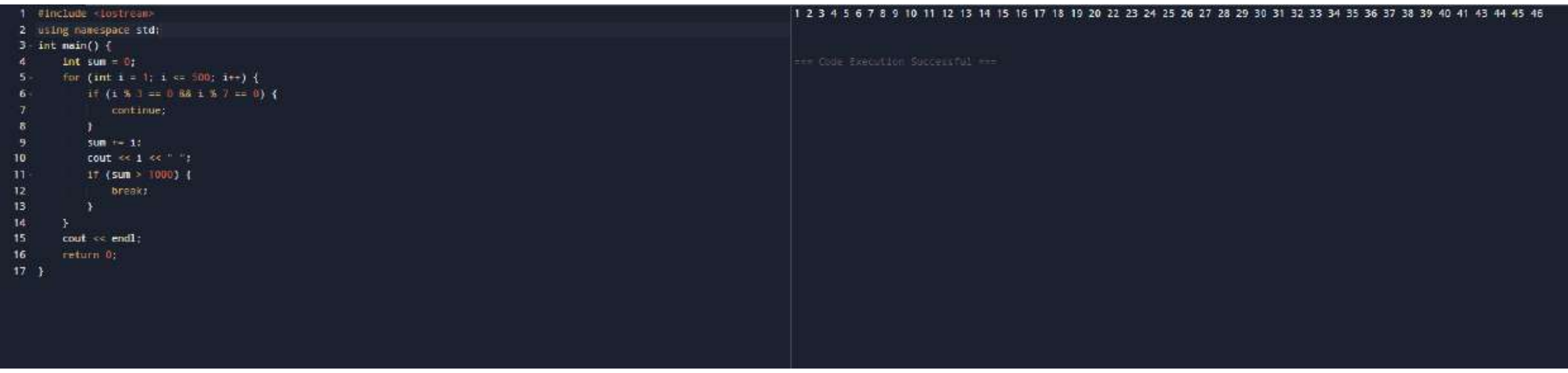
```cpp
#include <iostream>
using namespace std;
int main() {
    int number = 42;  // Predefined number to guess
    int guess, attempts = 0, maxAttempts = 5;
    cout << "Welcome to the Number Guessing Game!" << endl;
    cout << "I have selected a number between 1 and 100." << endl;
    cout << "You have " << maxAttempts << " attempts to guess the number." << endl;
    while (attempts < maxAttempts) {
        cout << "Attempt " << attempts + 1 << ": Enter your guess: ";
        cin >> guess;
        attempts++;
        if (guess < number) {
            cout << "Too low! Try again." << endl;
        } else if (guess > number) {
            cout << "Too high! Try again." << endl;
        } else {
            cout << "Congratulations! You guessed the number!" << endl;
            break;
        }
        if (attempts == maxAttempts) {
            cout << "Sorry, you've exhausted all your attempts. The number was " << number << "." << endl;
        }
    }
    return 0;
}
```

```
Welcome to the Number Guessing Game!
I have selected a number between 1 and 100.
You have 5 attempts to guess the number.
Attempt 1: Enter your guess: 52
Too high! Try again.
Attempt 2: Enter your guess: 15
Too low! Try again.
Attempt 3: Enter your guess: 54
Too high! Try again.
Attempt 4: Enter your guess: 99
Too high! Try again.
Attempt 5: Enter your guess: 65
Too high! Try again.
Sorry, you've exhausted all your attempts. The number was 42.


=== Code Execution Successful ===
```



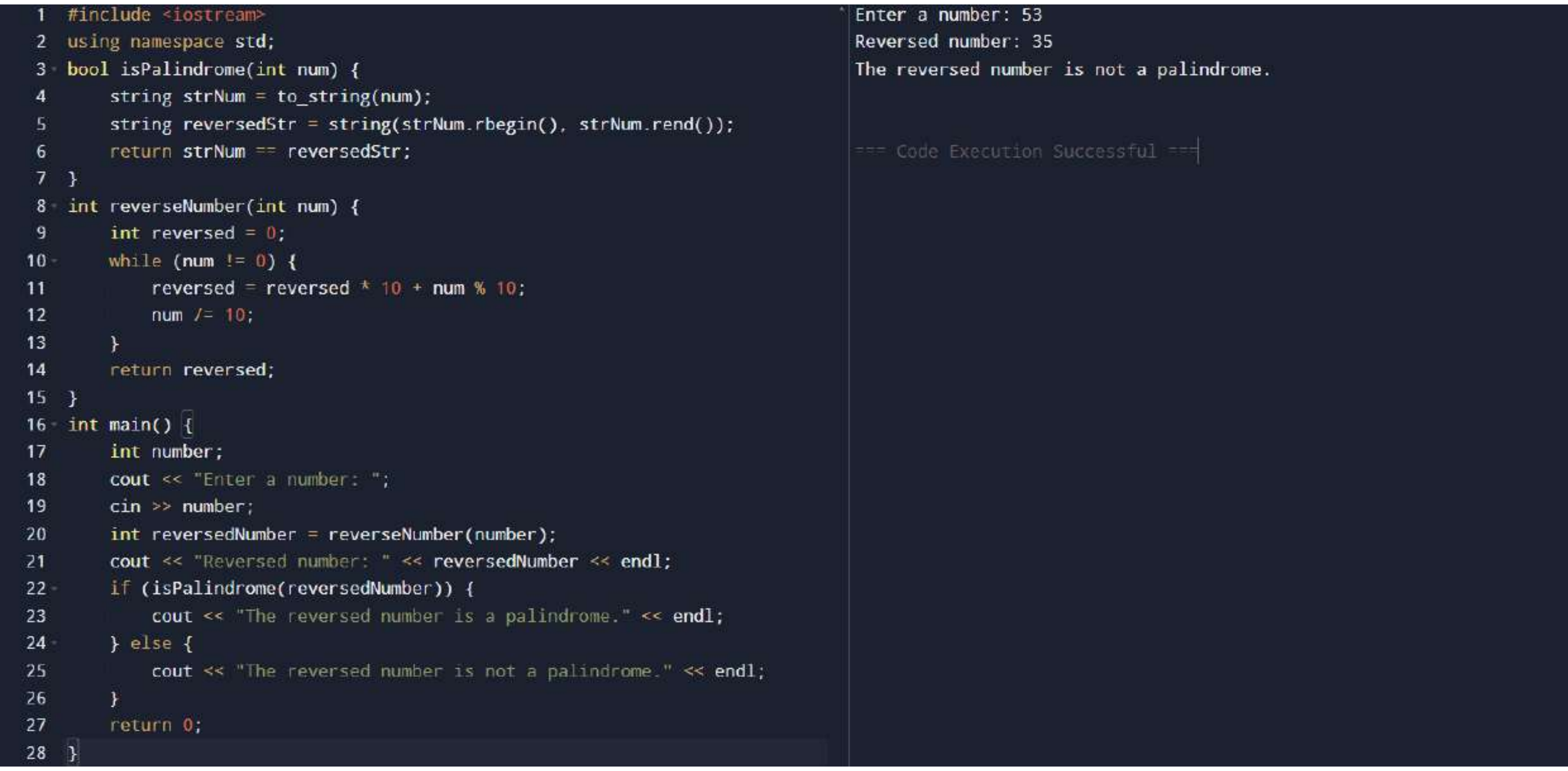
```cpp
#include <iostream>
using namespace std;
int main() {
    int number = 51;
    while (number % 7 != 0) {
        number++;
    }
    cout << number << endl;
    return 0;
}
```

```
56


=== Code Execution Successful ===
```



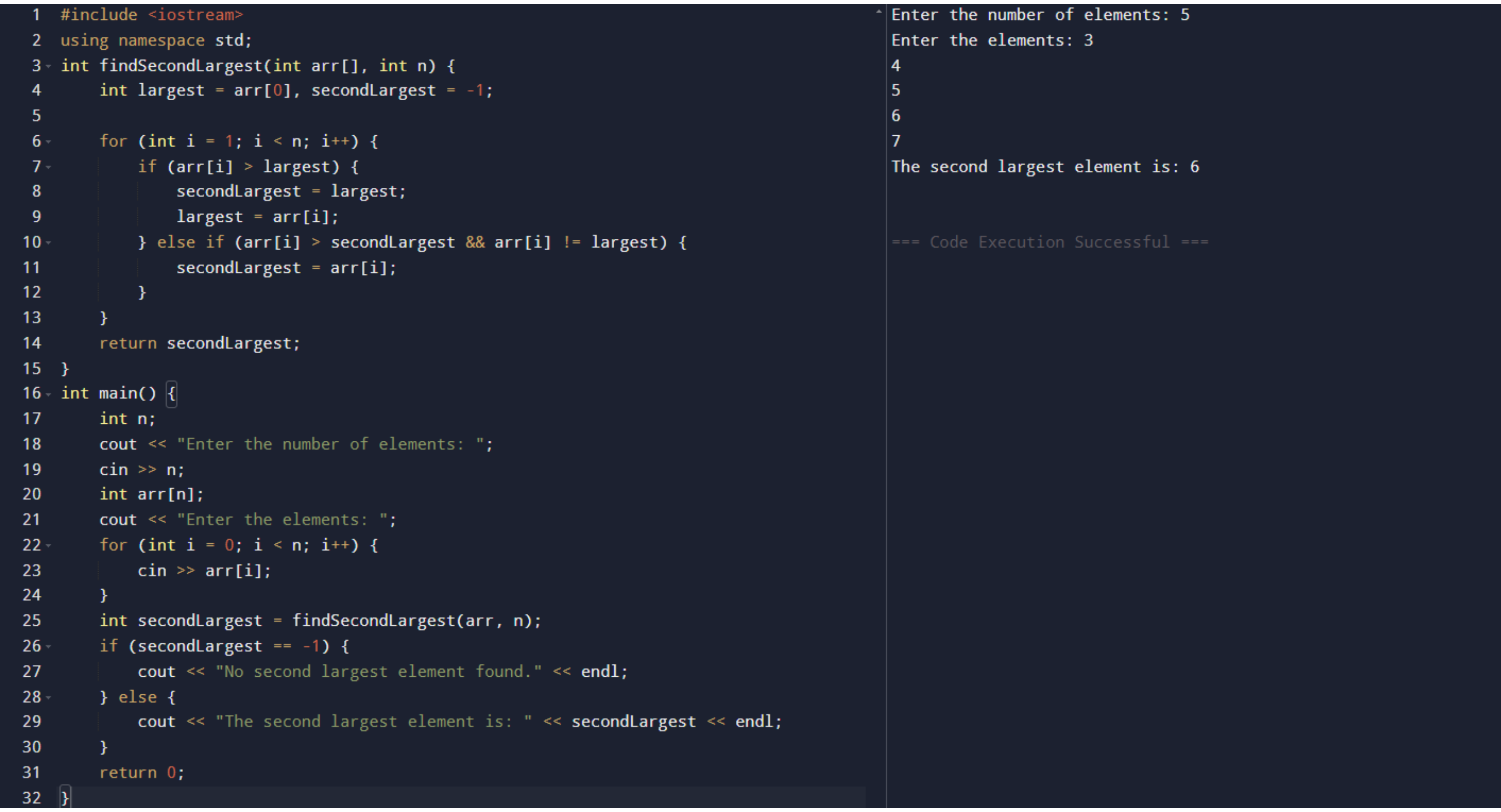
```cpp
#include <iostream>
using namespace std;
int main() {
    int sum = 0;
    for (int i = 1; i <= 500; i++) {
        if (i % 3 == 0 && i % 7 == 0) {
            continue;
        }
        sum += i;
        cout << i << " ";
        if (sum > 1000) {
            break;
        }
    }
    cout << endl;
    return 0;
}
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 43 44 45 46


=== Code Execution Successful ===
```



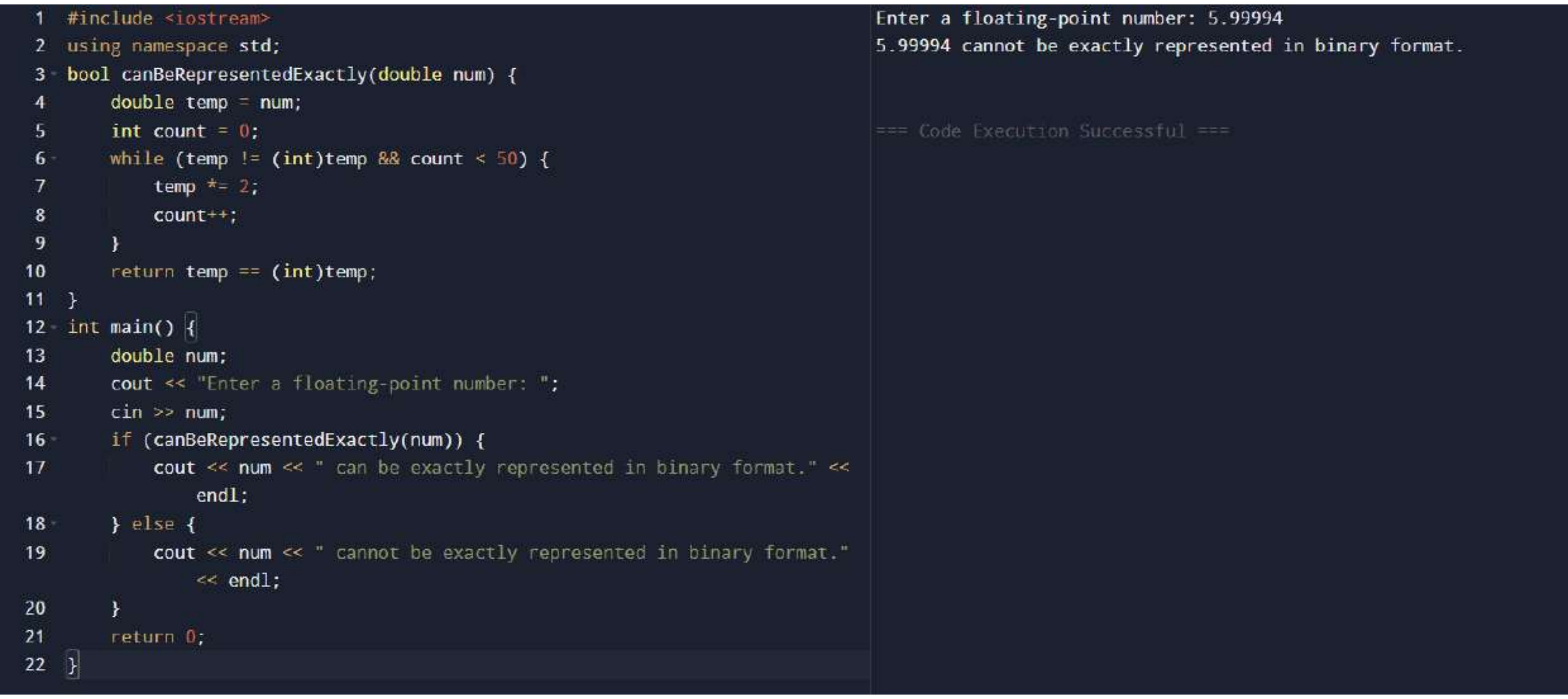
```cpp
#include <iostream>
using namespace std;
bool isPalindrome(int num) {
    string strNum = to_string(num);
    string reversedStr = string(strNum.rbegin(), strNum.rend());
    return strNum == reversedStr;
}
int reverseNumber(int num) {
    int reversed = 0;
    while (num != 0) {
        reversed = reversed * 10 + num % 10;
        num /= 10;
    }
    return reversed;
}
int main() {
    int number;
    cout << "Enter a number: ";
    cin >> number;
    int reversedNumber = reverseNumber(number);
    cout << "Reversed number: " << reversedNumber << endl;
    if (isPalindrome(reversedNumber)) {
        cout << "The reversed number is a palindrome." << endl;
    } else {
        cout << "The reversed number is not a palindrome." << endl;
    }
    return 0;
}
```

```
Enter a number: 53
Reversed number: 35
The reversed number is not a palindrome.


=== Code Execution Successful ===
```



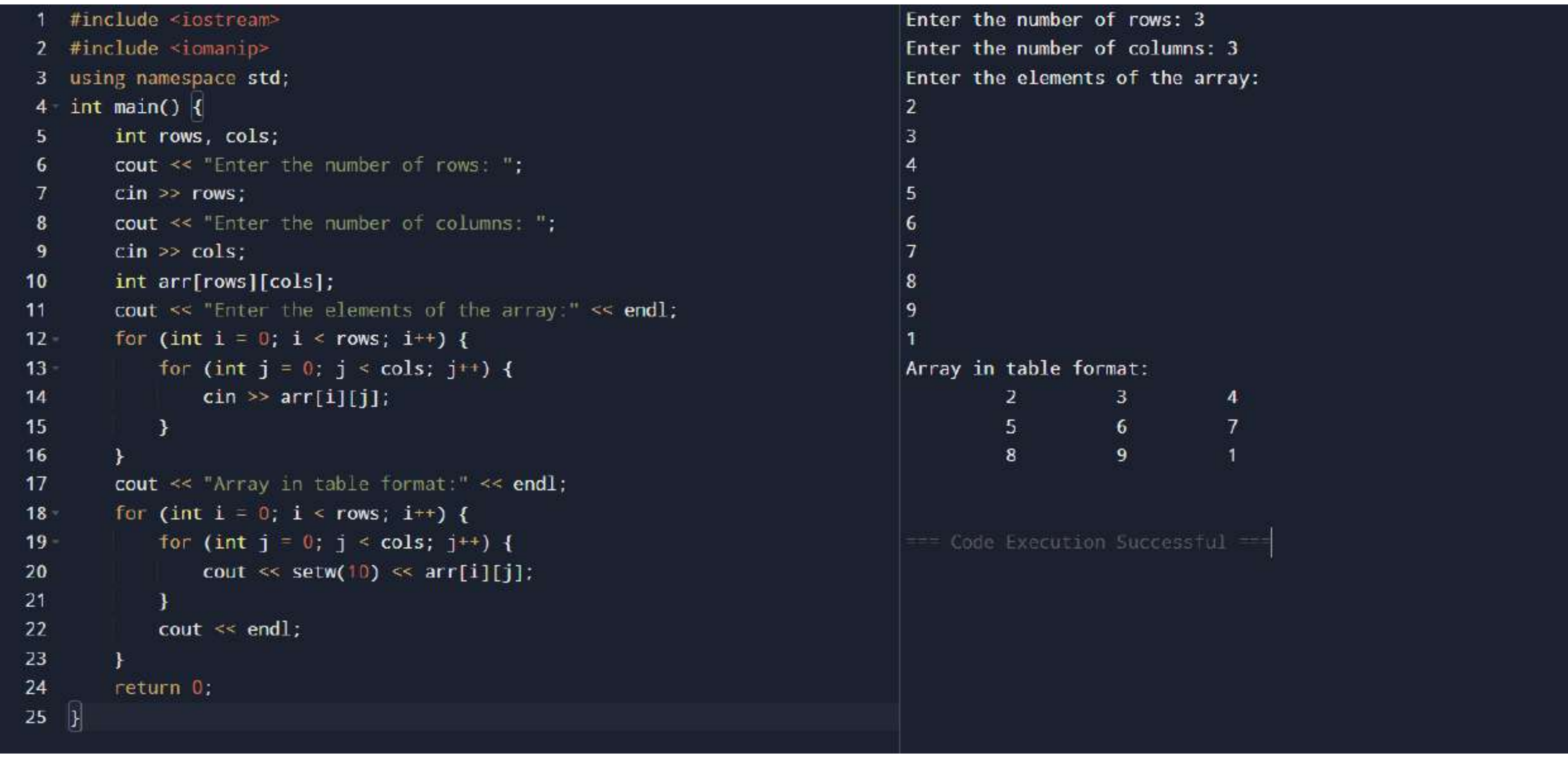
```cpp
#include <iostream>
using namespace std;
int findSecondLargest(int arr[], int n) {
    int largest = arr[0], secondLargest = -1;

    for (int i = 1; i < n; i++) {
        if (arr[i] > largest) {
            secondLargest = largest;
            largest = arr[i];
        } else if (arr[i] > secondLargest && arr[i] != largest) {
            secondLargest = arr[i];
        }
    }
    return secondLargest;
}
int main() {
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;
    int arr[n];
    cout << "Enter the elements: ";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    int secondLargest = findSecondLargest(arr, n);
    if (secondLargest == -1) {
        cout << "No second largest element found." << endl;
    } else {
        cout << "The second largest element is: " << secondLargest << endl;
    }
    return 0;
}
```

```
Enter the number of elements: 5
Enter the elements: 3
4
5
6
7
The second largest element is: 6


=== Code Execution Successful ===
```



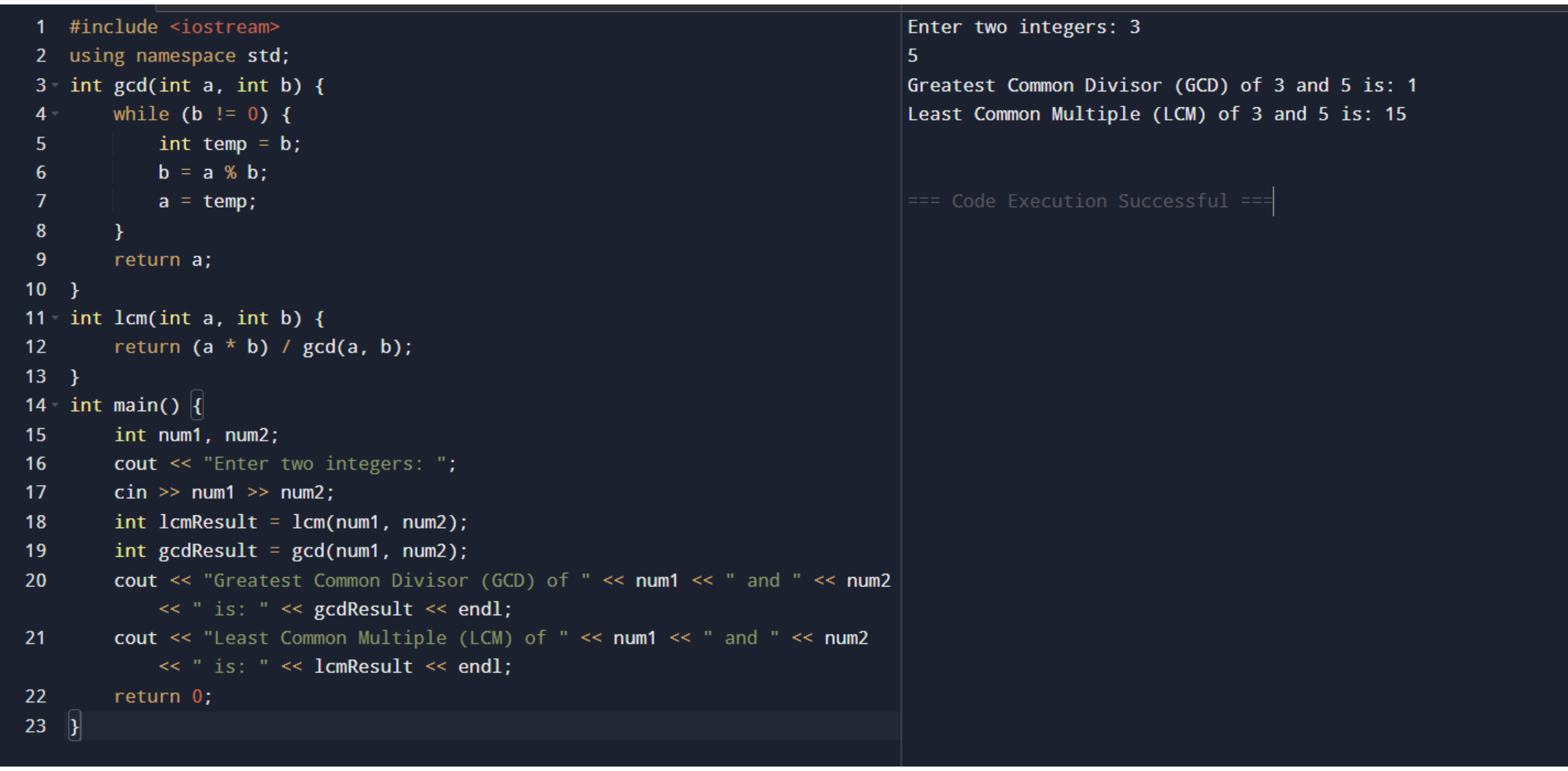
```cpp
#include <iostream>
using namespace std;
bool canBeRepresentedExactly(double num) {
    double temp = num;
    int count = 0;
    while (temp != (int)temp && count < 50) {
        temp *= 2;
        count++;
    }
    return temp == (int)temp;
}
int main() {
    double num;
    cout << "Enter a floating-point number: ";
    cin >> num;
    if (canBeRepresentedExactly(num)) {
        cout << num << " can be exactly represented in binary format." << endl;
    } else {
        cout << num << " cannot be exactly represented in binary format." << endl;
    }
    return 0;
}
```

```
Enter a floating-point number: 5.99994
5.99994 cannot be exactly represented in binary format.


=== Code Execution Successful ===
```



```cpp
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    int rows, cols;
    cout << "Enter the number of rows: ";
    cin >> rows;
    cout << "Enter the number of columns: ";
    cin >> cols;
    int arr[rows][cols];
    cout << "Enter the elements of the array:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cin >> arr[i][j];
        }
    }
    cout << "Array in table format:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cout << setw(10) << arr[i][j];
        }
        cout << endl;
    }
    return 0;
}
```

```
Enter the number of rows: 3
Enter the number of columns: 3
Enter the elements of the array:
2
3
4
5
6
7
8
9
1
Array in table format:
         2         3         4
         5         6         7
         8         9         1


=== Code Execution Successful ===
```



```cpp
#include <iostream>
using namespace std;
int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}
int lcm(int a, int b) {
    return (a * b) / gcd(a, b);
}
int main() {
    int num1, num2;
    cout << "Enter two integers: ";
    cin >> num1 >> num2;
    int lcmResult = lcm(num1, num2);
    int gcdResult = gcd(num1, num2);
    cout << "Greatest Common Divisor (GCD) of " << num1 << " and " << num2
        << " is: " << gcdResult << endl;
    cout << "Least Common Multiple (LCM) of " << num1 << " and " << num2
        << " is: " << lcmResult << endl;
    return 0;
}
```

```
Enter two integers: 3
5
Greatest Common Divisor (GCD) of 3 and 5 is: 1
Least Common Multiple (LCM) of 3 and 5 is: 15


=== Code Execution Successful ===
```