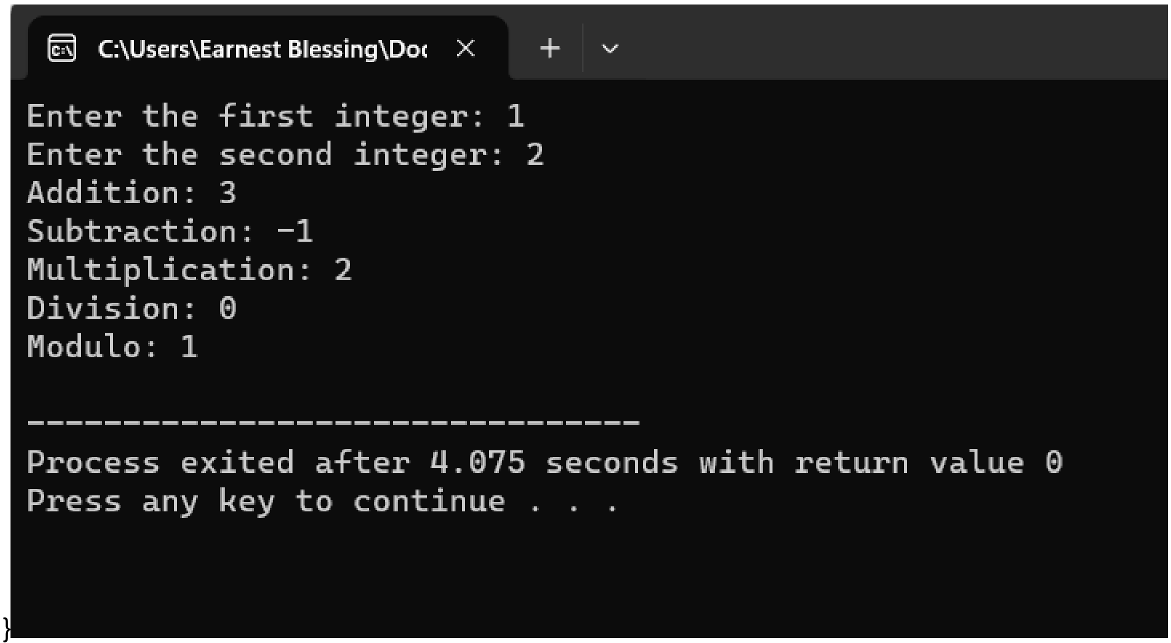
1. Write a program to read in two integers and perform the following operations on them: addition, subtraction, multiplication, division, and modulo.

```
#include <iostream>

int main() {
    // Declare variables to store input values
    int firstNumber, secondNumber;

    // Read in two integers
    std::cout << "Enter the first integer: ";
    std::cin >> firstNumber;

    std::cout << "Enter the second integer: ";
    std::cin >> secondNumber;

    // Perform operations
    int additionResult = firstNumber + secondNumber;
    int subtractionResult = firstNumber - secondNumber;

    // Check for division by zero before performing division
    if (secondNumber != 0) {
        int multiplicationResult = firstNumber * secondNumber;
        int divisionResult = firstNumber / secondNumber;
        int moduloResult = firstNumber % secondNumber;

        // Display results
        std::cout << "Addition: " << additionResult << std::endl;
        std::cout << "Subtraction: " << subtractionResult << std::endl;
        std::cout << "Multiplication: " << multiplicationResult << std::endl;
        std::cout << "Division: " << divisionResult << std::endl;
        std::cout << "Modulo: " << moduloResult << std::endl;
```

```
  } else {
    std::cout << "Error: Division by zero is not allowed." << std::endl;
  }

  return 0;
```

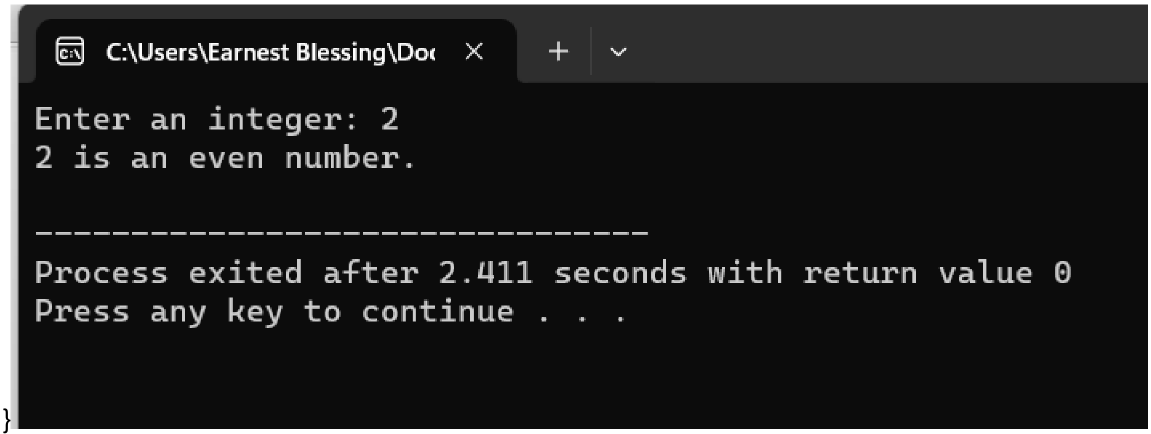


```
Enter the first integer: 1
Enter the second integer: 2
Addition: 3
Subtraction: -1
Multiplication: 2
Division: 0
Modulo: 1

--------------------------------
Process exited after 4.075 seconds with return value 0
Press any key to continue . . .
```

```
}
```

2. Program to determine the integer is odd or even

```
#include <iostream>


int main() {
  int number;
  std::cout << "Enter an integer: ";
  std::cin >> number;
  if (number % 2 == 0) {
    std::cout << number << " is an even number." << std::endl;
  } else {
    std::cout << number << " is an odd number." << std::endl;
  }
  return 0;
```



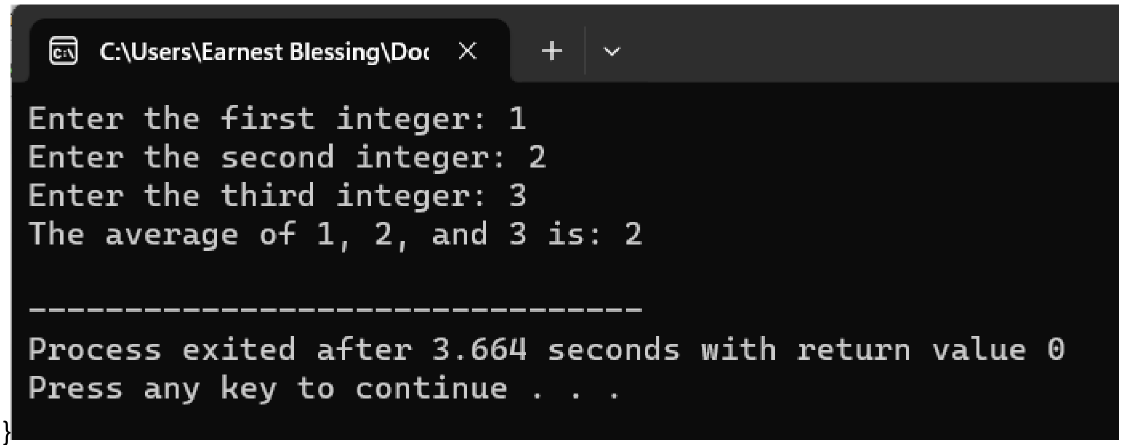
```
Enter an integer: 2
2 is an even number.

--------------------------------
Process exited after 2.411 seconds with return value 0
Press any key to continue . . .
```

}

3. Program to compute the average of three integers

```
#include <iostream>


int main() {
   // Declare variables to store user input
   int num1, num2, num3;


   // Read three integers from the user
   std::cout << "Enter the first integer: ";
   std::cin >> num1;


   std::cout << "Enter the second integer: ";
   std::cin >> num2;


   std::cout << "Enter the third integer: ";
   std::cin >> num3;


   // Calculate the average
   double average = static_cast<double>(num1 + num2 + num3) / 3;


   // Display the result
   std::cout << "The average of " << num1 << ", " << num2 << ", and " << num3 << " is: " << average << 
std::endl;
```

```
    return 0;
```

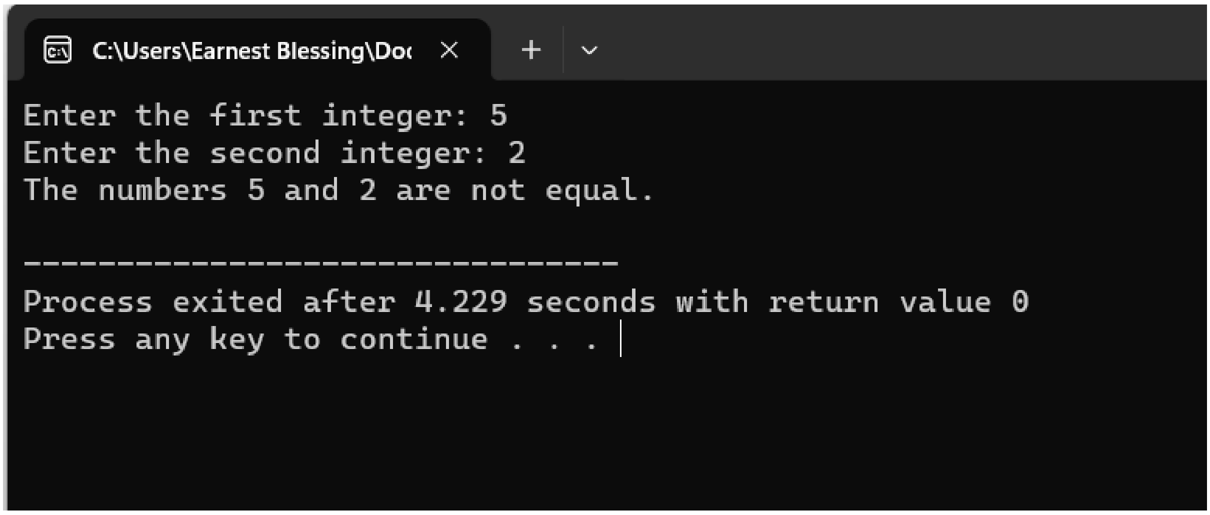


```
Enter the first integer: 1
Enter the second integer: 2
Enter the third integer: 3
The average of 1, 2, and 3 is: 2

--------------------------------
Process exited after 3.664 seconds with return value 0
Press any key to continue . . .
```

```
}
```

4. Program to check two numbers are equal or not

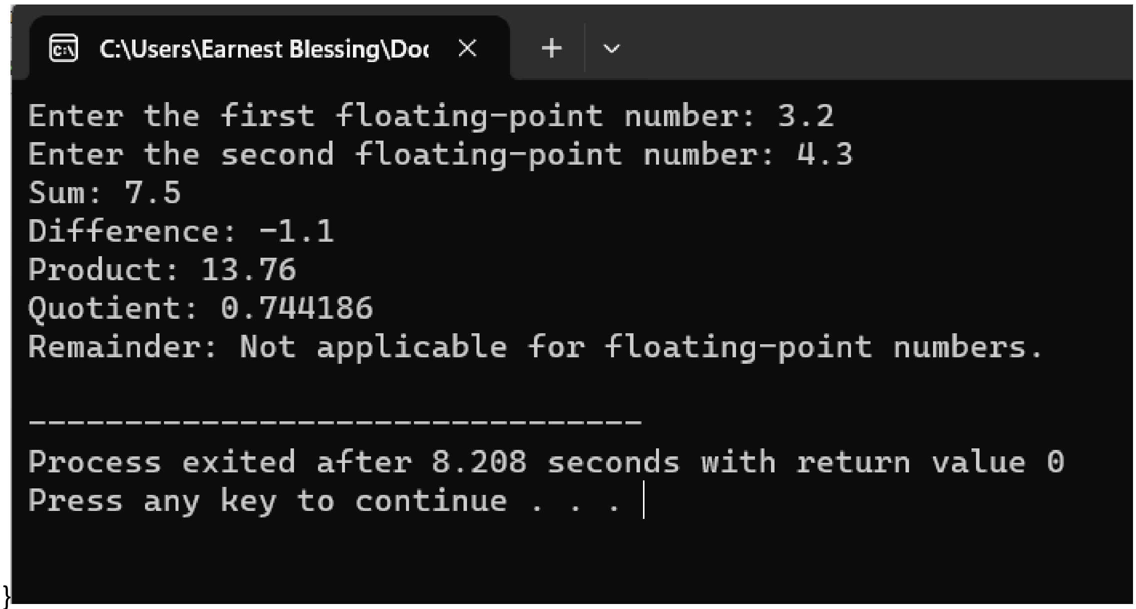
```
#include <iostream>


int main() {
    // Declare variables to store user input
    int num1, num2;


    // Read two integers from the user
    std::cout << "Enter the first integer: ";
    std::cin >> num1;


    std::cout << "Enter the second integer: ";
    std::cin >> num2;


    // Check if the numbers are equal
    if (num1 == num2) {
        std::cout << "The numbers " << num1 << " and " << num2 << " are equal." << std::endl;
    } else {
        std::cout << "The numbers " << num1 << " and " << num2 << " are not equal." << std::endl;
    }
    return 0;
}
```



```
Enter the first integer: 5
Enter the second integer: 2
The numbers 5 and 2 are not equal.

--------------------------------
Process exited after 4.229 seconds with return value 0
Press any key to continue . . .
```

5. Write a program to read in two Floating numbers and perform the following operations on them: addition, subtraction, multiplication, division, and modulo.

```
#include <iostream>


int main() {
    // Declare variables to store user input
    double num1, num2;


    // Read two floating-point numbers from the user
    std::cout << "Enter the first floating-point number: ";
    std::cin >> num1;


    std::cout << "Enter the second floating-point number: ";
    std::cin >> num2;


    // Perform operations and display results
    std::cout << "Sum: " << num1 + num2 << std::endl;
    std::cout << "Difference: " << num1 - num2 << std::endl;
    std::cout << "Product: " << num1 * num2 << std::endl;
    // Check if the second number is not zero before performing division and modulo
    if (num2 != 0) {
        std::cout << "Quotient: " << num1 / num2 << std::endl;
        std::cout << "Remainder: Not applicable for floating-point numbers." << std::endl;
    } else {
        std::cout << "Cannot perform division because the second number is zero." << std::endl;
    }

    return 0;
```

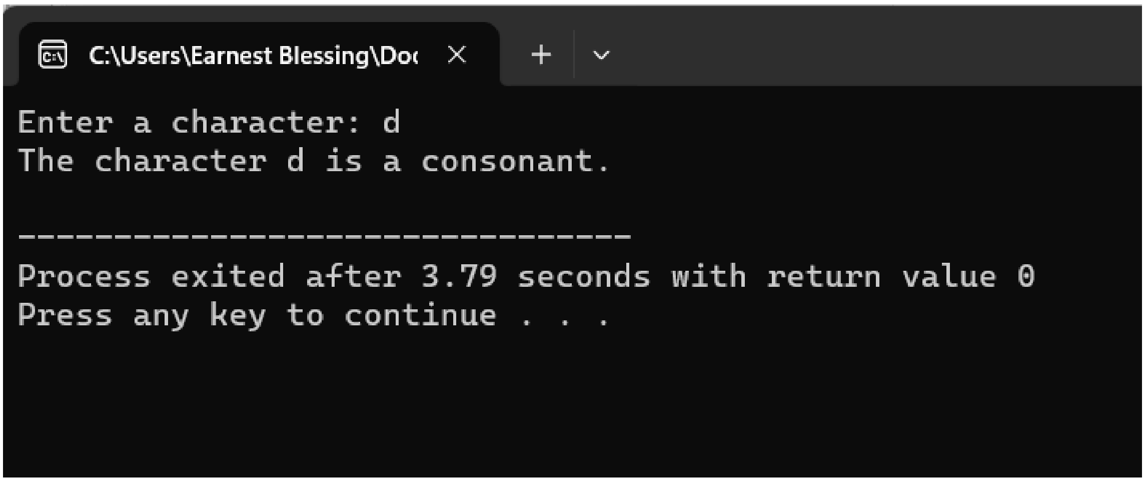


```
Enter the first floating-point number: 3.2
Enter the second floating-point number: 4.3
Sum: 7.5
Difference: -1.1
Product: 13.76
Quotient: 0.744186
Remainder: Not applicable for floating-point numbers.

--------------------------------
Process exited after 8.208 seconds with return value 0
Press any key to continue . . .
```

```
}
```

6. Program to check the character is a vowel or consonant

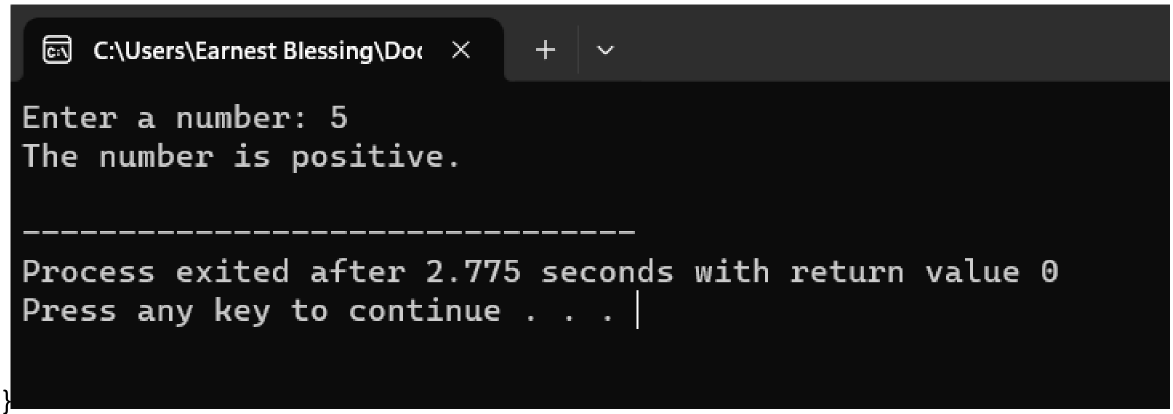
```
#include <iostream>

int main() {
    // Declare a variable to store user input
    char ch;

    // Read a character from the user
    std::cout << "Enter a character: ";
    std::cin >> ch;

    // Check if the character is a vowel
    if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
        ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
        std::cout << "The character " << ch << " is a vowel." << std::endl;
    } else {
        std::cout << "The character " << ch << " is a consonant." << std::endl;
    }

    return 0;
}
```



```
Enter a character: d
The character d is a consonant.

--------------------------------
Process exited after 3.79 seconds with return value 0
Press any key to continue . . .
```

7. Program to check the number is positive, negative or zero

```
#include <iostream>

int main() {
    // Declare variable
    double number;

    // Read a number from the user
    std::cout << "Enter a number: ";
    std::cin >> number;
    // Check whether the number is positive, negative, or zero

    if (number > 0) {

        std::cout << "The number is positive." << std::endl;

    } else if (number < 0) {

        std::cout << "The number is negative." << std::endl;

    } else {

        std::cout << "The number is zero." << std::endl;

    }


    return 0;
```

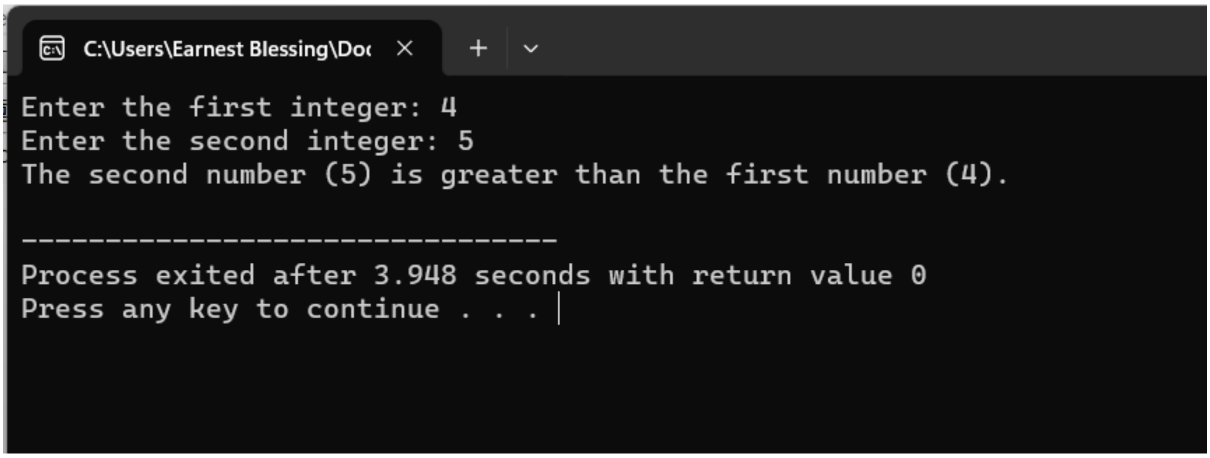


```
Enter a number: 5
The number is positive.

--------------------------------
Process exited after 2.775 seconds with return value 0
Press any key to continue . . .
```

```
}
```

8. Program to determine which number is greater among two integers

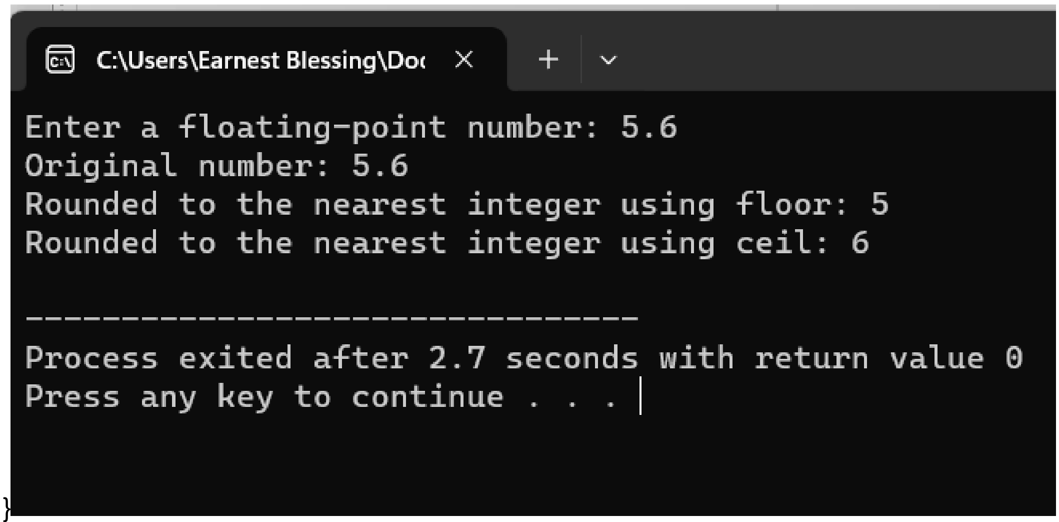
```
#include <iostream>


int main() {

    // Declare variables

    int firstNumber, secondNumber;


    // Read two numbers from the user

    std::cout << "Enter the first integer: ";

    std::cin >> firstNumber;


    std::cout << "Enter the second integer: ";
    std::cin >> secondNumber;


    // Determine the greater number
    if (firstNumber > secondNumber) {
        std::cout << "The first number (" << firstNumber << ") is greater than the second number (" <<
secondNumber << ")." << std::endl;
    } else if (secondNumber > firstNumber) {
        std::cout << "The second number (" << secondNumber << ") is greater than the first number ("
<< firstNumber << ")." << std::endl;
    } else {
        std::cout << "Both numbers are equal." << std::endl;
    }


    return 0;
}
```



```
Enter the first integer: 4
Enter the second integer: 5
The second number (5) is greater than the first number (4).

--------------------------------
Process exited after 3.948 seconds with return value 0
Press any key to continue . . .
```

9. Program to read a floating-number and round it to the nearest integer using the floor an ceil functions.

```
#include <iostream>

#include <cmath>


int main() {

    // Declare variables

    double floatingNumber;
    // Read a floating-point number from the user
    std::cout << "Enter a floating-point number: ";
    std::cin >> floatingNumber;

    // Round the number using floor and ceil functions
    int roundedFloor = static_cast<int>(std::floor(floatingNumber));
    int roundedCeil = static_cast<int>(std::ceil(floatingNumber));

    // Display the results
    std::cout << "Original number: " << floatingNumber << std::endl;
    std::cout << "Rounded to the nearest integer using floor: " << roundedFloor << std::endl;
    std::cout << "Rounded to the nearest integer using ceil: " << roundedCeil << std::endl;

    return 0;
```

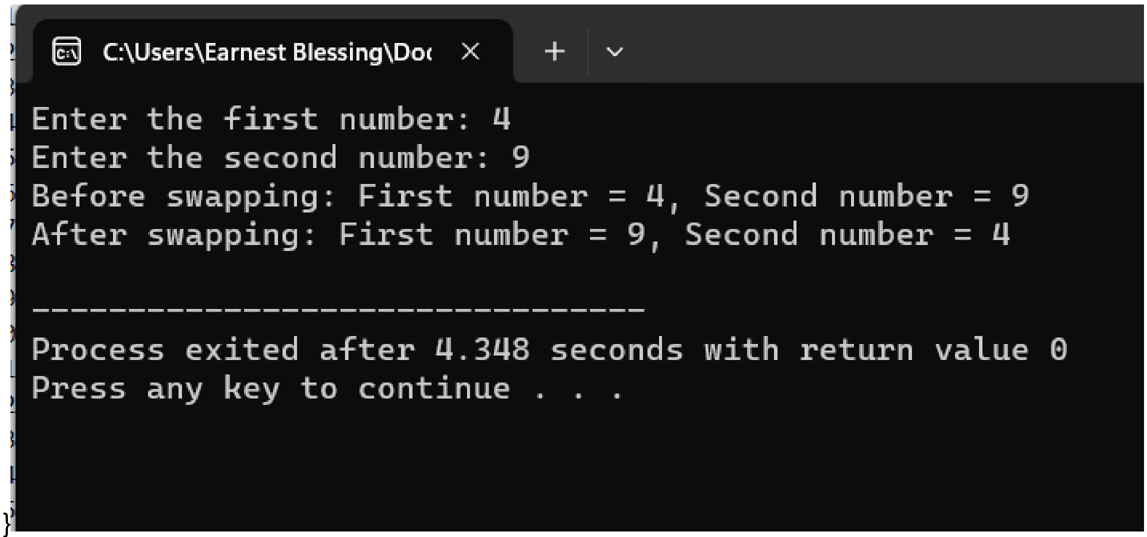


```
Enter a floating-point number: 5.6
Original number: 5.6
Rounded to the nearest integer using floor: 5
Rounded to the nearest integer using ceil: 6

--------------------------------
Process exited after 2.7 seconds with return value 0
Press any key to continue . . .
```

```
}
```

10. Program to swap two numbers using bitwise XOR operator

```
#include <iostream>

int main() {
    // Declare variables
    int firstNumber, secondNumber;
    // Read two numbers from the user
    std::cout << "Enter the first number: ";
    std::cin >> firstNumber;

    std::cout << "Enter the second number: ";
    std::cin >> secondNumber;

    // Display the numbers before swapping
    std::cout << "Before swapping: First number = " << firstNumber << ", Second number = " << secondNumber << std::endl;

    // Swap the numbers using bitwise XOR
    firstNumber = firstNumber ^ secondNumber;
    secondNumber = firstNumber ^ secondNumber;
    firstNumber = firstNumber ^ secondNumber;

    // Display the numbers after swapping
    std::cout << "After swapping: First number = " << firstNumber << ", Second number = " << secondNumber << std::endl;

    return 0;
```

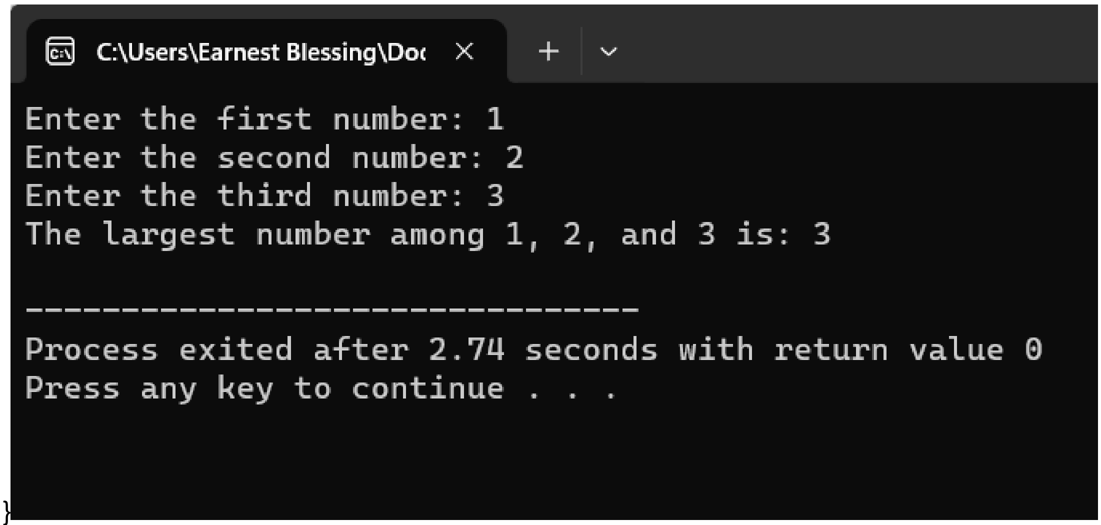


```
Enter the first number: 4
Enter the second number: 9
Before swapping: First number = 4, Second number = 9
After swapping: First number = 9, Second number = 4

--------------------------------
Process exited after 4.348 seconds with return value 0
Press any key to continue . . .
```

```
}
```

11. Largest among three numbers using ternary conditional operator

```
#include <iostream>

int main() {
    // Declare variables
    int num1, num2, num3, largest;

    // Read three numbers from the user
    std::cout << "Enter the first number: ";
    std::cin >> num1;

    std::cout << "Enter the second number: ";
    std::cin >> num2;

    std::cout << "Enter the third number: ";
    std::cin >> num3;

    // Use ternary conditional operator to find the largest number
    largest = (num1 > num2) ? ((num1 > num3) ? num1 : num3) : ((num2 > num3) ? num2 : num3);

    // Display the largest number
    std::cout << "The largest number among " << num1 << ", " << num2 << ", and " << num3 << " is: " << largest << std::endl;

    return 0;
```



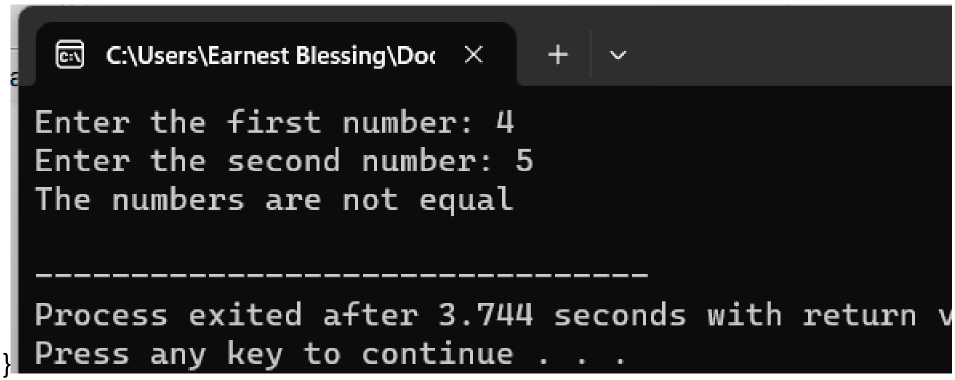
```
Enter the first number: 1
Enter the second number: 2
Enter the third number: 3
The largest number among 1, 2, and 3 is: 3

--------------------------------
Process exited after 2.74 seconds with return value 0
Press any key to continue . . .
```

}

12. Program to check two numbers are equal or not using ternary conditional operator

```
#include <iostream>

int main() {
    // Declare variables
    int num1, num2;

    // Read two numbers from the user
    std::cout << "Enter the first number: ";
    std::cin >> num1;

    std::cout << "Enter the second number: ";
    std::cin >> num2;

    // Use ternary conditional operator to check equality
    std::cout << "The numbers are " << ((num1 == num2) ? "equal" : "not equal") << std::endl;

    return 0;
```

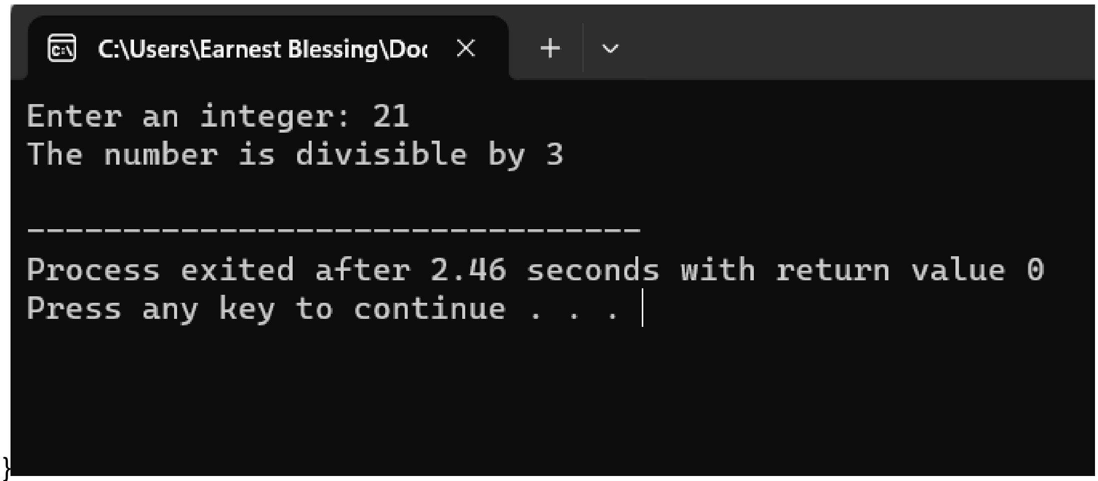
}



```
Enter the first number: 4
Enter the second number: 5
The numbers are not equal

--------------------------------
Process exited after 3.744 seconds with return v
Press any key to continue . . .
```

13. Program to check the integer is divisible by 3 or not using ternary conditional operator

```
#include <iostream>


int main() {
   // Declare variable
   int number;


   // Read an integer from the user
   std::cout << "Enter an integer: ";
   std::cin >> number;


   // Use ternary conditional operator to check divisibility by 3
   std::cout << "The number is " << ((number % 3 == 0) ? "divisible by 3" : "not divisible by 3") << std::endl;


   return 0;
```

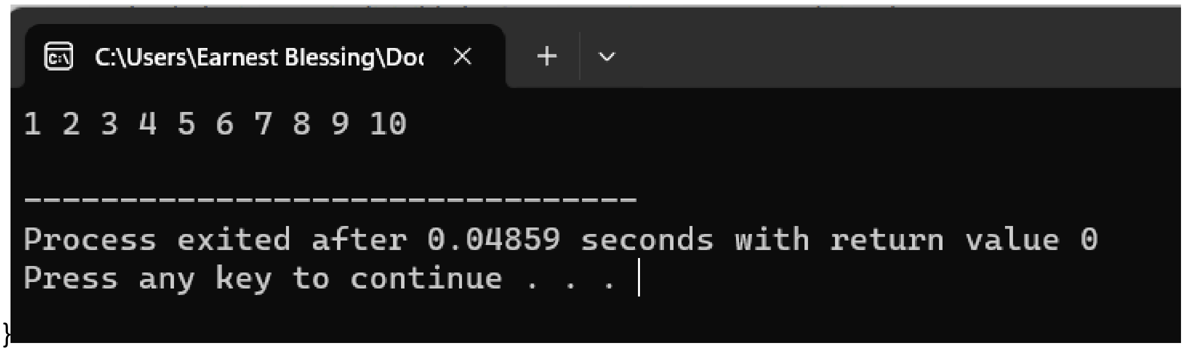


```
Enter an integer: 21
The number is divisible by 3

--------------------------------
Process exited after 2.46 seconds with return value 0
Press any key to continue . . .
```

}

14. Program to print numbers from 1 to 10 using for loop

```
#include <iostream>


int main() {
   // Use a for loop to print numbers from 1 to 10
   for (int i = 1; i <= 10; ++i) {
      std::cout << i << " ";
   }


   // Add a newline at the end for better formatting
   std::cout << std::endl;


   return 0;
```

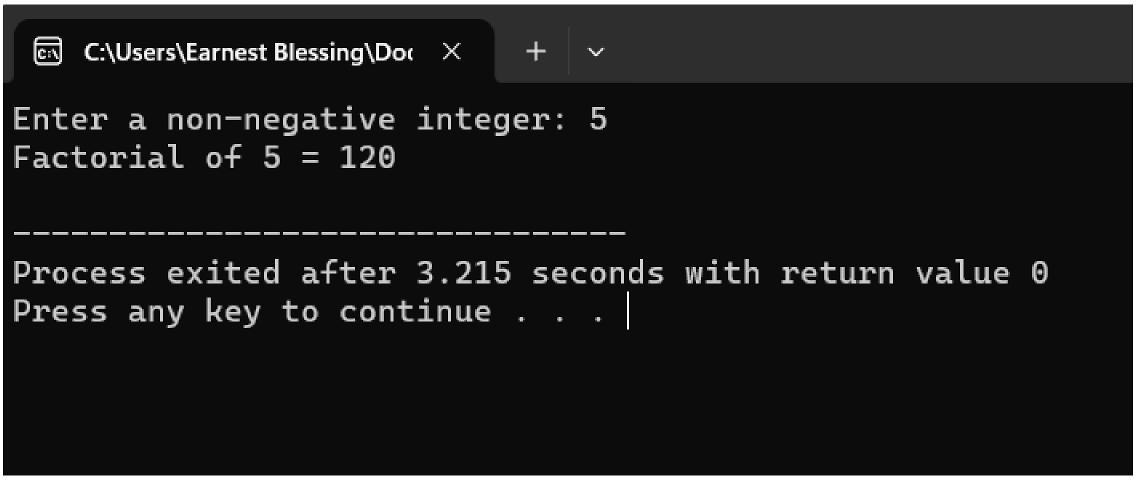


```
1 2 3 4 5 6 7 8 9 10

--------------------------------
Process exited after 0.04859 seconds with return value 0
Press any key to continue . . .
```

}

15. Factorial of a number using for loop

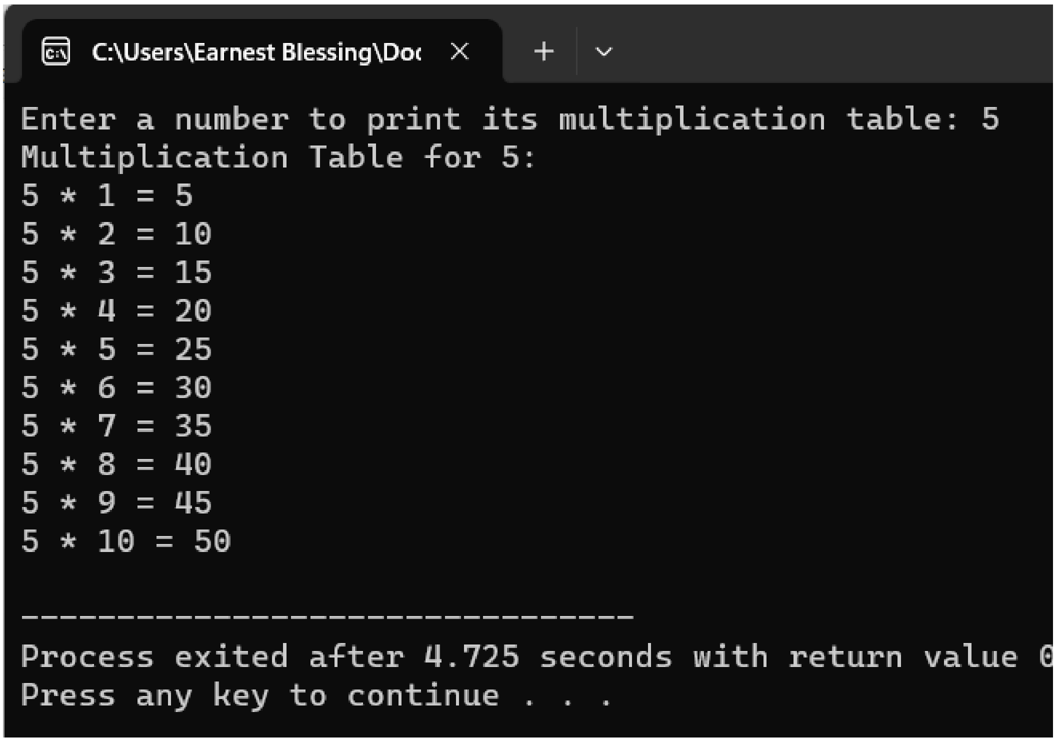
```cpp
#include <iostream>

int main() {
    // Declare variables
    int number;
    unsigned long long factorial = 1;  // Use unsigned long long to handle larger factorials

    // Read a number from the user
    std::cout << "Enter a non-negative integer: ";
    std::cin >> number;

    // Check if the number is non-negative
    if (number < 0) {
        std::cout << "Factorial is not defined for negative numbers." << std::endl;
    } else {
        // Calculate the factorial using a for loop
        for (int i = 1; i <= number; ++i) {
            factorial *= i;
        }

        // Display the factorial
        std::cout << "Factorial of " << number << " = " << factorial << std::endl;
    }

    return 0;
}
```



```
Enter a non-negative integer: 5
Factorial of 5 = 120

--------------------------------
Process exited after 3.215 seconds with return value 0
Press any key to continue . . .
```

16.

```
#include <iostream>

int main() {
  // Declare a variable to store the number for which the multiplication table will be printed
  int number;

  // Get the number from the user
  std::cout << "Enter a number to print its multiplication table: ";
  std::cin >> number;

  // Print the multiplication table using a for loop
  std::cout << "Multiplication Table for " << number << ":" << std::endl;
  for (int i = 1; i <= 10; ++i) {
    std::cout << number << " * " << i << " = " << (number * i) << std::endl;
  }

  return 0;
}
```

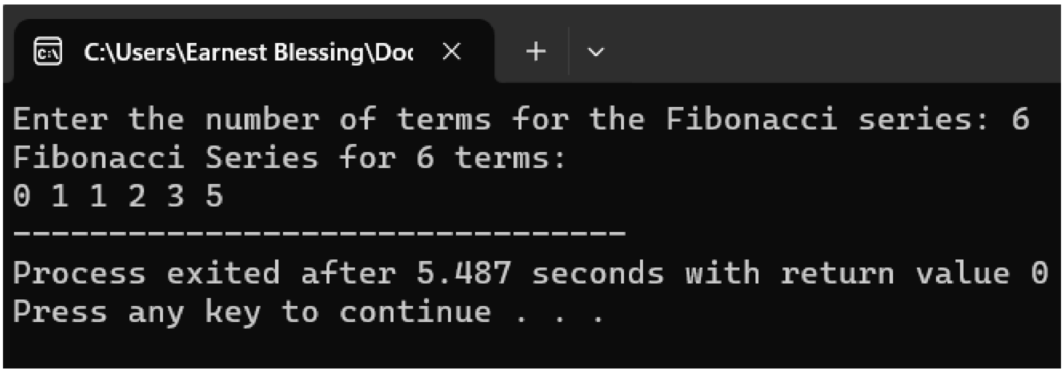


```
Enter a number to print its multiplication table: 5
Multiplication Table for 5:
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50

--------------------------------
Process exited after 4.725 seconds with return value 0
Press any key to continue . . .
```

17.

```
#include <iostream>

int main() {
   // Declare variables to store the Fibonacci series terms
   int n;

   // Get the number of terms in the Fibonacci series from the user
   std::cout << "Enter the number of terms for the Fibonacci series: ";
   std::cin >> n;

   // Initialize the first two terms of the Fibonacci series
   int firstTerm = 0, secondTerm = 1;

   // Print the Fibonacci series using a for loop
   std::cout << "Fibonacci Series for " << n << " terms:" << std::endl;
```

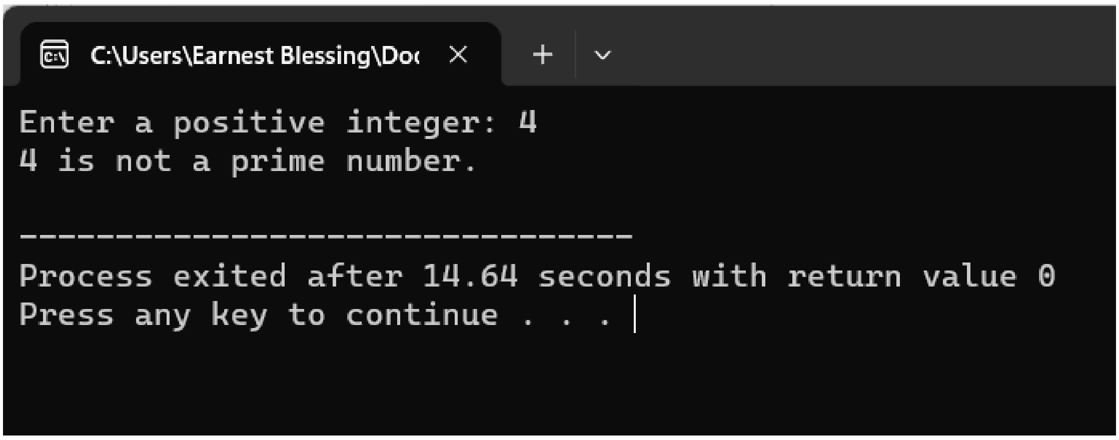
```
    for (int i = 0; i < n; ++i) {
        std::cout << firstTerm << " ";

        // Calculate the next term in the series
        int nextTerm = firstTerm + secondTerm;

        // Update firstTerm and secondTerm for the next iteration
        firstTerm = secondTerm;
        secondTerm = nextTerm;
    }

    return 0;
}
```



```
Enter the number of terms for the Fibonacci series: 6
Fibonacci Series for 6 terms:
0 1 1 2 3 5
--------------------------------
Process exited after 5.487 seconds with return value 0
Press any key to continue . . .
```

18.

```
#include <iostream>

int main() {
    // Declare variables
    int number;
    bool isPrime = true;
    // Get the number from the user
    std::cout << "Enter a positive integer: ";
    std::cin >> number;

    // Check if the number is greater than 1
    if (number > 1) {
        // Check for factors using a for loop
        for (int i = 2; i <= number / 2; ++i) {
            if (number % i == 0) {
                isPrime = false;
                break; // No need to check further if a factor is found
            }
        }
    } else {
        isPrime = false; // Numbers less than or equal to 1 are not prime
    }

    // Display the result
    if (isPrime) {
        std::cout << number << " is a prime number." << std::endl;
    } else {
        std::cout << number << " is not a prime number." << std::endl;
    }

    return 0;
}
```

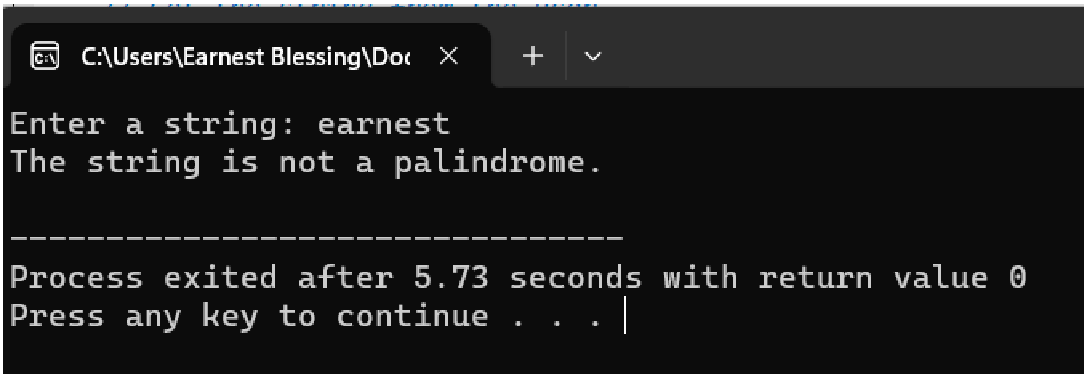


```
Enter a positive integer: 4
4 is not a prime number.

--------------------------------
Process exited after 14.64 seconds with return value 0
Press any key to continue . . .
```

19.

```
#include <iostream>
#include <cctype> // For using std::tolower

int main() {
   // Declare variables
   std::string inputString;

   // Get the string from the user
   std::cout << "Enter a string: ";
   std::getline(std::cin, inputString);

   // Remove spaces and convert the string to lowercase
   std::string processedString;
   for (char character : inputString) {
      if (!std::isspace(character)) {
         processedString += std::tolower(character);
      }
   }

   // Check if the processed string is a palindrome using a while loop
```

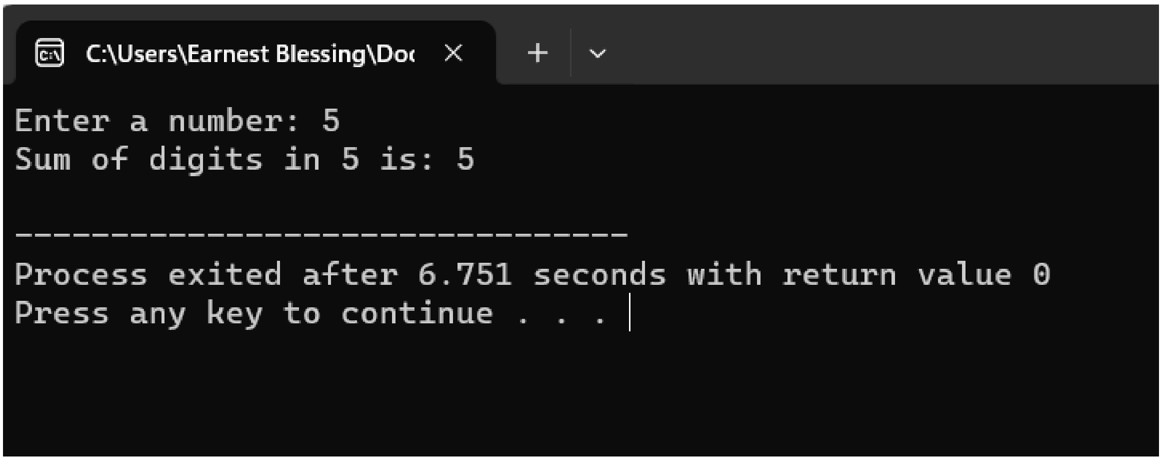
```
    bool isPalindrome = true;
    int start = 0;
    int end = processedString.length() - 1;

    while (start < end) {
        if (processedString[start] != processedString[end]) {
            isPalindrome = false;
            break;
        }
        ++start;
        --end;
    }

    // Display the result
    if (isPalindrome) {
        std::cout << "The string is a palindrome." << std::endl;
    } else {
        std::cout << "The string is not a palindrome." << std::endl;
    }

    return 0;
}
```



```
Enter a string: earnest
The string is not a palindrome.

--------------------------------
Process exited after 5.73 seconds with return value 0
Press any key to continue . . .
```

20.

```
#include <iostream>

int main() {
    // Declare variables
    int number, originalNumber;
    int sum = 0;

    // Get the number from the user
    std::cout << "Enter a number: ";
    std::cin >> number;

    // Save the original number for later comparison
    originalNumber = number;

    // Calculate the sum of digits using a while loop
    while (number != 0) {
        // Extract the last digit
        int digit = number % 10;

        // Add the digit to the sum
        sum += digit;

        // Remove the last digit
        number /= 10;
    }

    // Display the result
    std::cout << "Sum of digits in " << originalNumber << " is: " << sum << std::endl;

    return 0;
}
```



```
Enter a number: 5
Sum of digits in 5 is: 5

--------------------------------
Process exited after 6.751 seconds with return value 0
Press any key to continue . . .
```

21.