



**Submitted by: Sajjal Fayyaz**

**Submitted to: Lecturer Salman Irfan**

## **Task 1:**

## **Code:**

```
reversnumber.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      // Step 1: Ask user for an integer input
6      int number;
7      cout << "Please enter an integer: ";
8      cin >> number;
9
10     // Step 2: Initialize variables for reversed number and sum of digits
11     int reversedNumber = 0;
12     int digitsSum = 0;
13
14     // Step 3: Extract digits, reverse the number and calculate the sum of digits
15     int tempNumber = number; // Store the original number for future reference
16     while (tempNumber != 0) {
17         int digit = tempNumber % 10; // Get the last digit
18         reversedNumber = reversedNumber * 10 + digit; // Build the reversed number
19         digitsSum += digit; // Add the digit to the sum
20         tempNumber /= 10; // Remove the last digit from tempNumber
21     }
22
23     // Step 4: Output the reversed number and the sum of its digits
24     cout << "Reversed number: " << reversedNumber << endl;
25     cout << "Sum of digits: " << digitsSum << endl;
26
27     return 0;
}
```

*Figure 1: reverse number and sum of digits*

**Compiler:**

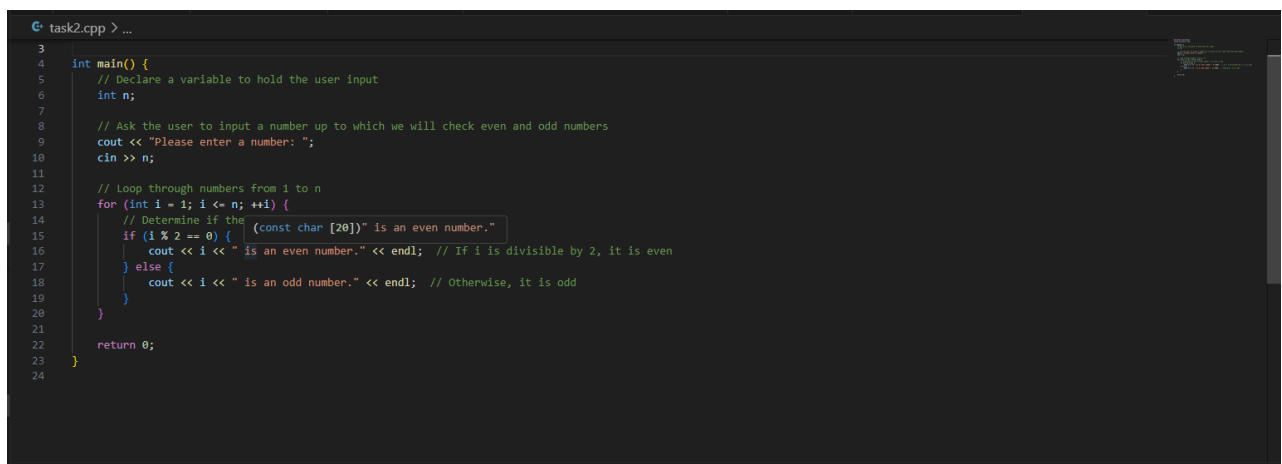


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> g++ .\reversnumber.cpp -o .\reversnumber.exe
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> .\reversnumber.exe
Please enter an integer: 965
Reversed number: 569
Sum of digits: 20
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> |
```

*Figure 2: compile a code*

## Lab task 2:

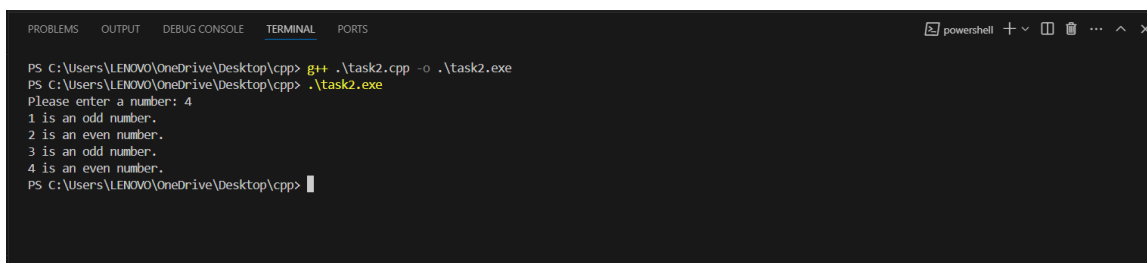
### Code:



```
G+ task2.cpp > ...
3
4 int main() {
5     // Declare a variable to hold the user input
6     int n;
7
8     // Ask the user to input a number up to which we will check even and odd numbers
9     cout << "Please enter a number: ";
10    cin >> n;
11
12    // Loop through numbers from 1 to n
13    for (int i = 1; i <= n; ++i) {
14        // Determine if the (const char [20])" is an even number."
15        if (i % 2 == 0) {
16            cout << i << " is an even number." << endl; // If i is divisible by 2, it is even
17        } else {
18            cout << i << " is an odd number." << endl; // Otherwise, it is odd
19        }
20    }
21
22    return 0;
23 }
24
```

*Figure 3: code of even and old check*

### Compiler:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> g++ .\task2.cpp -o .\task2.exe
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> .\task2.exe
Please enter a number: 4
1 is an odd number.
2 is an even number.
3 is an odd number.
4 is an even number.
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> |
```

*Figure 4: compile a code*

## Lab task 3:

### Code:

```

1 #include <iostream>
2 #include <cstdlib> // For rand() and srand()
3 #include <ctime> // For time()
4 using namespace std;
5
6 int main() {
7     srand(time(0)); // Seed random number generator with current time
8
9     // Generate a random number between 1 and 100
10    int secretNumber = rand() % 100 + 1;
11    int userGuess;
12    int attemptsCounter = 0; // Counter to track number of attempts
13
14    cout << "Welcome to the Number Guessing Game!\n";
15    cout << "I have selected a number between 1 and 100.\n";
16    cout << "Try to guess the number!\n";
17
18    // Start guessing loop
19    while (true) {
20        cout << "Enter your guess: ";
21        cin >> userGuess; // Take user's guess as input
22        attemptsCounter++; // Increment attempts counter
23
24        // Check if the guess is correct, too high, or too low
25        if (userGuess < secretNumber) {
26            cout << "Your guess is too low! Try again.\n";
27        } else if (userGuess > secretNumber) {
28            cout << "Your guess is too high! Try again.\n";
29        } else {
30            // User guessed correctly
31            cout << "Congratulations! You've guessed the number in "
32                << attemptsCounter << " attempts.\n";
33            break; // Exit the loop when the guess is correct
34        }
35    }
36
37    return 0;
38 }

```

Figure 5: code of guessing game

## Compiler:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> g++ .\task3.cpp -o .\task3.exe
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> .\task3.exe
Welcome to the Number Guessing Game!
I have selected a number between 1 and 100.
Try to guess the number!
Enter your guess: 50
Your guess is too low! Try again.
Enter your guess: 80
Your guess is too low! Try again.
Enter your guess: 90
Congratulations! You've guessed the number in 3 attempts.
PS C:\Users\LENOVO\OneDrive\Desktop\cpp>

```

Figure 6: compile a code

## Lab task 4:

### Code:

```

task4.cpp > ...
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int number, limit;
6
7     // prompt the user to input a base number for the multiplication table
8     cout << "Enter a number for the multiplication table: ";
9     cin >> number;
10
11     // Ensure the number is positive (greater than 0)
12     if (number > 0) {
13         // ask for the limit (where the multiplication should stop)
14         cout << "Enter the limit (up to where the table should be displayed): ";
15         cin >> limit;
16
17         // Display the multiplication table
18         cout << "Multiplication table for " << number << " up to " << limit << ":\n";
19         for (int i = 1; i <= limit; ++i) {
20             // Output each line of the multiplication table
21             cout << number << " x " << i << " = " << number * i << endl;
22         }
23     } else {
24         // Handle invalid input where the number is not positive
25         cout << "Error: Please enter a number greater than 0.\n";
26     }
27
28     return 0;
29 }

```

Figure 7: code of multiplication

## Compiler:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter the limit (up to where the table should be displayed): 10
Multiplication table for 6 up to 10:
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
PS C:\Users\LENOVO\OneDrive\Desktop\cpp>

```

Figure 8: compile a code

## Lab task5:

## Code:

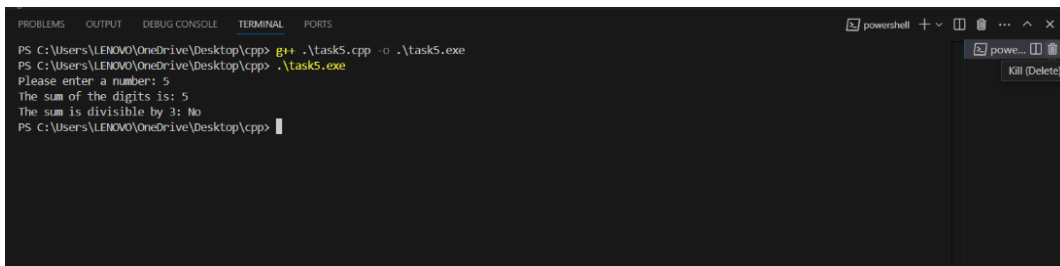
```

task5.cpp > ...
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int number;
6
7     // Ask the user to input a number
8     cout << "Please enter a number: ";
9     cin >> number;
10
11     int digitSum = 0;
12     int workingNumber = number; // Store the number for digit extraction
13
14     // Extract digits and calculate the sum of the digits
15     while (workingNumber > 0) {
16         digitSum += workingNumber % 10; // Add the last digit to the sum
17         workingNumber /= 10; // Remove the last digit
18     }
19
20     // Output the sum of the digits
21     cout << "The sum of the digits is: " << digitSum << endl;
22
23     // Check if the sum of the digits is divisible by 3
24     if (digitSum % 3 == 0) {
25         cout << "The sum is divisible by 3: yes" << endl;
26     } else {
27         cout << "The sum is divisible by 3: no" << endl;
28     }
29
30     return 0;
31 }

```

Figure 9: code of sum and divisible of 3

## Compiler:

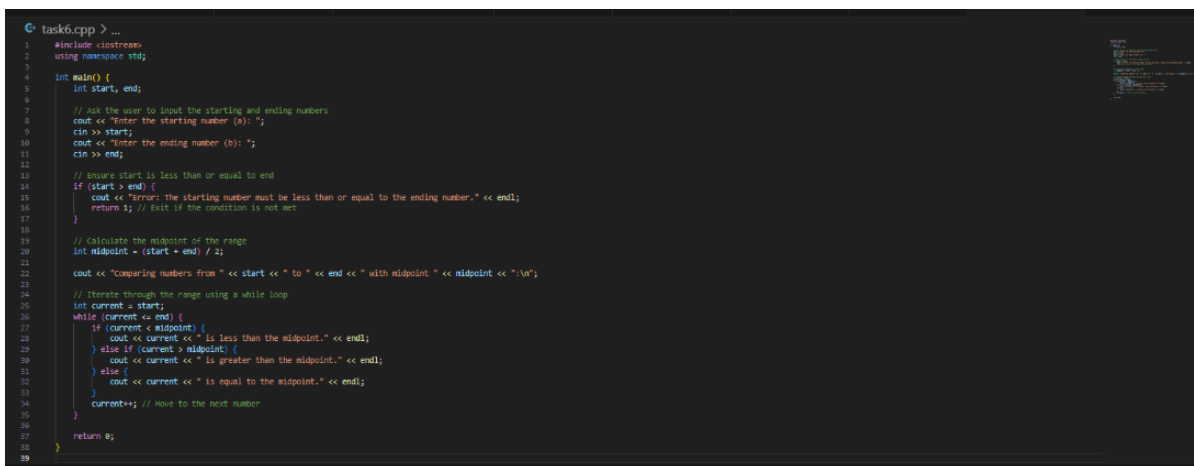


```
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> g++ .\task5.cpp -o .\task5.exe
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> .\task5.exe
Please enter a number: 5
The sum of the digits is: 5
The sum is divisible by 3: No
PS C:\Users\LENOVO\OneDrive\Desktop\cpp>
```

Figure 10: compile a code

## Lab task6:

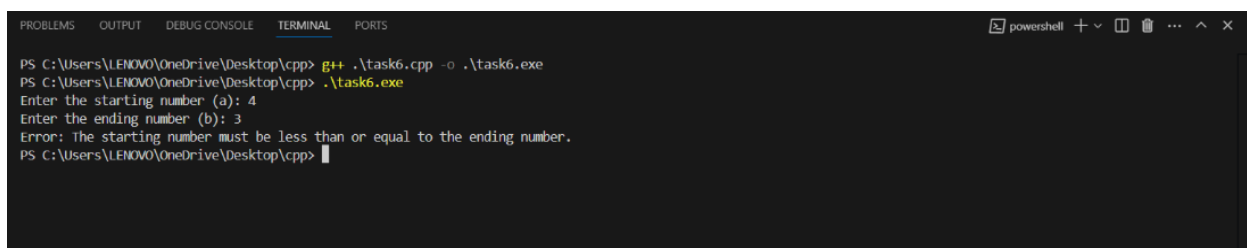
### Code:



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int start, end;
6
7     // Ask the user to input the starting and ending numbers
8     cout << "Enter the starting number (a): ";
9     cin >> start;
10    cout << "Enter the ending number (b): ";
11    cin >> end;
12
13    // Ensure start is less than or equal to end
14    if (start > end) {
15        cout << "Error: The starting number must be less than or equal to the ending number." << endl;
16        return 1; // Exit if the condition is not met
17    }
18
19    // Calculate the midpoint of the range
20    int midpoint = (start + end) / 2;
21
22    cout << "Comparing numbers from " << start << " to " << end << " with midpoint " << midpoint << "\n";
23
24    // Iterate through the range using a while loop
25    int current = start;
26    while (current <= end) {
27        if (current < midpoint) {
28            cout << current << " is less than the midpoint." << endl;
29        } else if (current > midpoint) {
30            cout << current << " is greater than the midpoint." << endl;
31        } else {
32            cout << current << " is equal to the midpoint." << endl;
33        }
34        current++; // Move to the next number
35    }
36
37    return 0;
38 }
```

Figure 11: code of number comparison

### Compiler:



```
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> g++ .\task6.cpp -o .\task6.exe
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> .\task6.exe
Enter the starting number (a): 4
Enter the ending number (b): 3
Error: The starting number must be less than or equal to the ending number.
PS C:\Users\LENOVO\OneDrive\Desktop\cpp>
```

Figure 12: compile a code

## Lab task 7:

## Code:

```
reversnumber.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      // Step 1: Ask user for an integer input
6      int number;
7      cout << "Please enter an Integer: ";
8      cin >> number;
9
10     // Step 2: Initialize variables for reversed number and sum of digits
11     int reversedNumber = 0;
12     int digitsSum = 0;
13
14     // Step 3: Extract digits, reverse the number and calculate the sum of digits
15     int tempNumber = number; // Store the original number for future reference
16     while (tempNumber != 0) {
17         int digit = tempNumber % 10; // Get the last digit
18         reversedNumber = reversedNumber * 10 + digit; // Build the reversed number
19         digitsSum += digit; // Add the digit to the sum
20         tempNumber /= 10; // Remove the last digit from tempNumber
21     }
22
23     // Step 4: Output the reversed number and the sum of its digits
24     cout << "Reversed number: " << reversedNumber << endl;
25     cout << "Sum of digits: " << digitsSum << endl;
26
27     return 0;
28 }
```

Figure 13: code of count divisible number

## Compiler:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> g++ .\task7.cpp -o .\task7.exe
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> .\task7.exe
Enter a positive number (n): 4
Numbers divisible by 3 between 1 and 4 are:
3
Total count of numbers divisible by 3: 1
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> |
```

Figure 14: compile a code

## Lab task 8:

## Code:

```

C:\task8.cpp > ...
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int largest; // Variable to store the largest number
6     int currentNumber; // Variable to store each number input by the user
7
8     // Prompt the user to enter 5 numbers
9     cout << "Please enter 5 numbers to find the largest:" << endl;
10
11     // Accept the first number to initialize 'largest' as a baseline
12     cout << "Enter number 1: ";
13     cin >> largest;
14
15     // Loop to accept 4 more numbers
16     for (int i = 2; i <= 5; ++i) {
17         cout << "Enter number " << i << ": ";
18         cin >> currentNumber;
19
20         // Compare current number with 'largest' and update if needed
21         if (currentNumber > largest) {
22             largest = currentNumber; // Update 'largest' if currentNumber is greater
23         }
24     }
25
26     // Output the largest number found
27     cout << "The largest number is: " << largest << "." << endl;
28
29     return 0;
30 }
31

```

Figure 15: code of find the largest number

## Compiler:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> g++ .\task8.cpp -o .\task8.exe
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> .\task8.exe
Please enter 5 numbers to find the largest:
Enter number 1: 8
Enter number 2: 9
Enter number 3: 2
Enter number 4: 5
Enter number 5: 11
The largest number is: 11.
PS C:\Users\LENOVO\OneDrive\Desktop\cpp>

```

Figure 16: compile a code

## Lab task 9:

### Code:

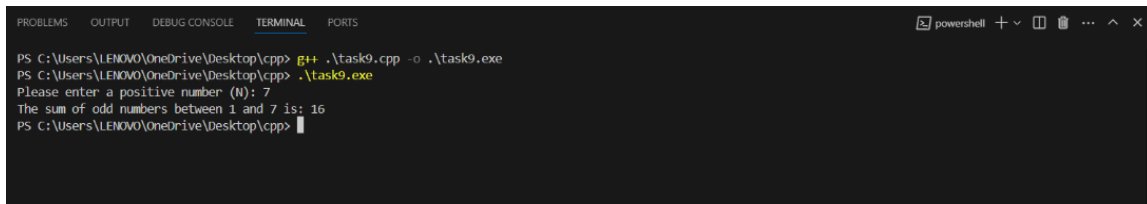
```

C:\task9.cpp > ...
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     int sum = 0;
7
8     // Prompt the user to input a number N
9     cout << "Please enter a positive number (N): ";
10    cin >> n;
11
12    // Check if the input is positive
13    if (n <= 0) {
14        cout << "Please enter a number greater than 0." << endl;
15        return 1; // Exit the program if the input is invalid
16    }
17
18    // Loop through all numbers from 1 to N
19    for (int i = 1; i <= n; ++i) {
20        // Check if the current number is odd
21        if (i % 2 != 0) {
22            sum += i; // Add the odd number to the sum
23        }
24    }
25
26    // Display the result: sum of odd numbers
27    cout << "The sum of odd numbers between 1 and " << n << " is: " << sum << endl;
28
29    return 0;
30 }
31

```

Figure 17: code of sum of number

## Compiler:

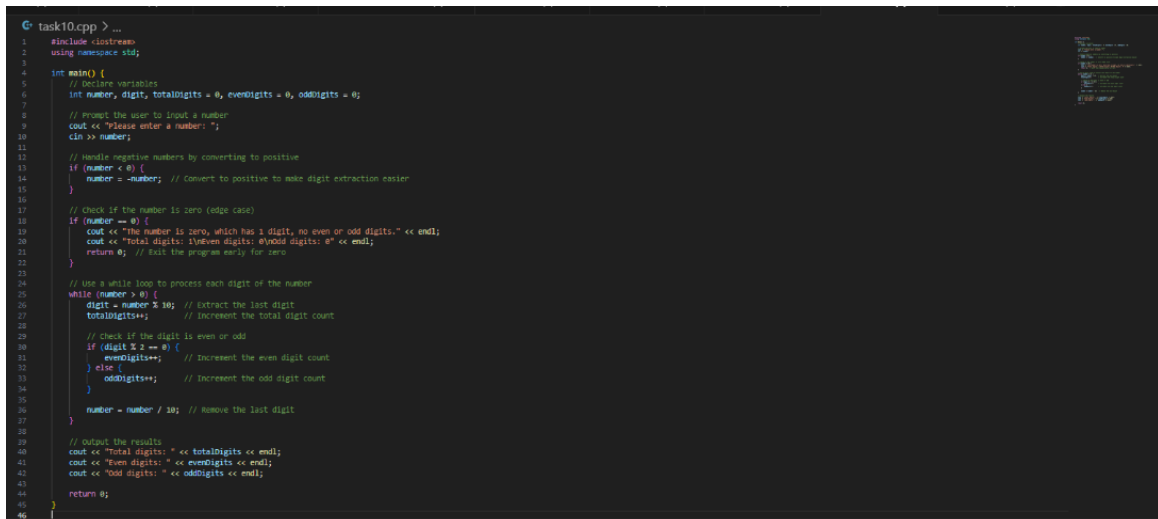


```
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> g++ -o .\task9.exe task9.cpp
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> .\task9.exe
Please enter a positive number (N): 7
The sum of odd numbers between 1 and 7 is: 16
PS C:\Users\LENOVO\OneDrive\Desktop\cpp>
```

Figure 18: compile a code

## Lab task 10:

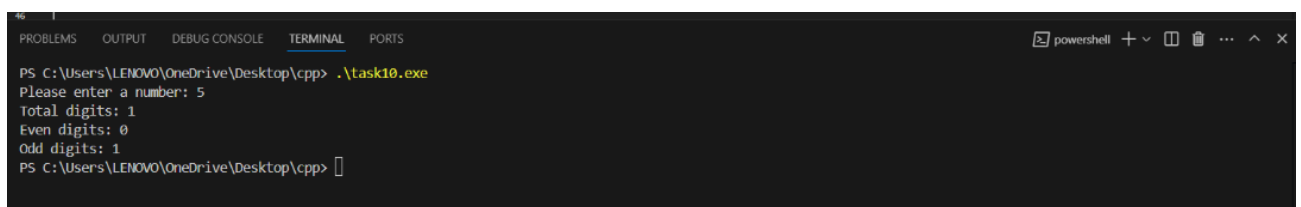
## Code:



```
1 // task10.cpp
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     // Declare variables
7     int number, digit, totalDigits = 0, evenDigits = 0, oddDigits = 0;
8
9     // Prompt the user to input a number
10    cout << "Please enter a number: ";
11    cin >> number;
12
13    // Handle negative numbers by converting to positive
14    if (number < 0) {
15        number = -number; // Convert to positive to make digit extraction easier
16    }
17
18    // Check if the number is zero (edge case)
19    if (number == 0) {
20        cout << "The number is zero, which has 1 digit, no even or odd digits." << endl;
21        cout << "Total digits: 1\nEven digits: 0\nOdd digits: 0" << endl;
22        return 0; // Exit the program early for zero
23    }
24
25    // Use a while loop to process each digit of the number
26    while (number > 0) {
27        digit = number % 10; // Extract the last digit
28        totalDigits++; // Increment the total digit count
29
30        // Check if the digit is even or odd
31        if (digit % 2 == 0) {
32            evenDigits++; // Increment the even digit count
33        } else {
34            oddDigits++; // Increment the odd digit count
35        }
36
37        number = number / 10; // Remove the last digit
38    }
39
40    // Output the results
41    cout << "Total digits: " << totalDigits << endl;
42    cout << "Even digits: " << evenDigits << endl;
43    cout << "Odd digits: " << oddDigits << endl;
44
45    return 0;
46 }
```

Figure 19: code of identify even and odd

## compiler:



```
PS C:\Users\LENOVO\OneDrive\Desktop\cpp> .\task10.exe
Please enter a number: 5
Total digits: 1
Even digits: 0
Odd digits: 1
PS C:\Users\LENOVO\OneDrive\Desktop\cpp>
```

Figure 20: compile a code