

Date : 6-08-2025

Name : Amit Sarkar

Id : 22235103283

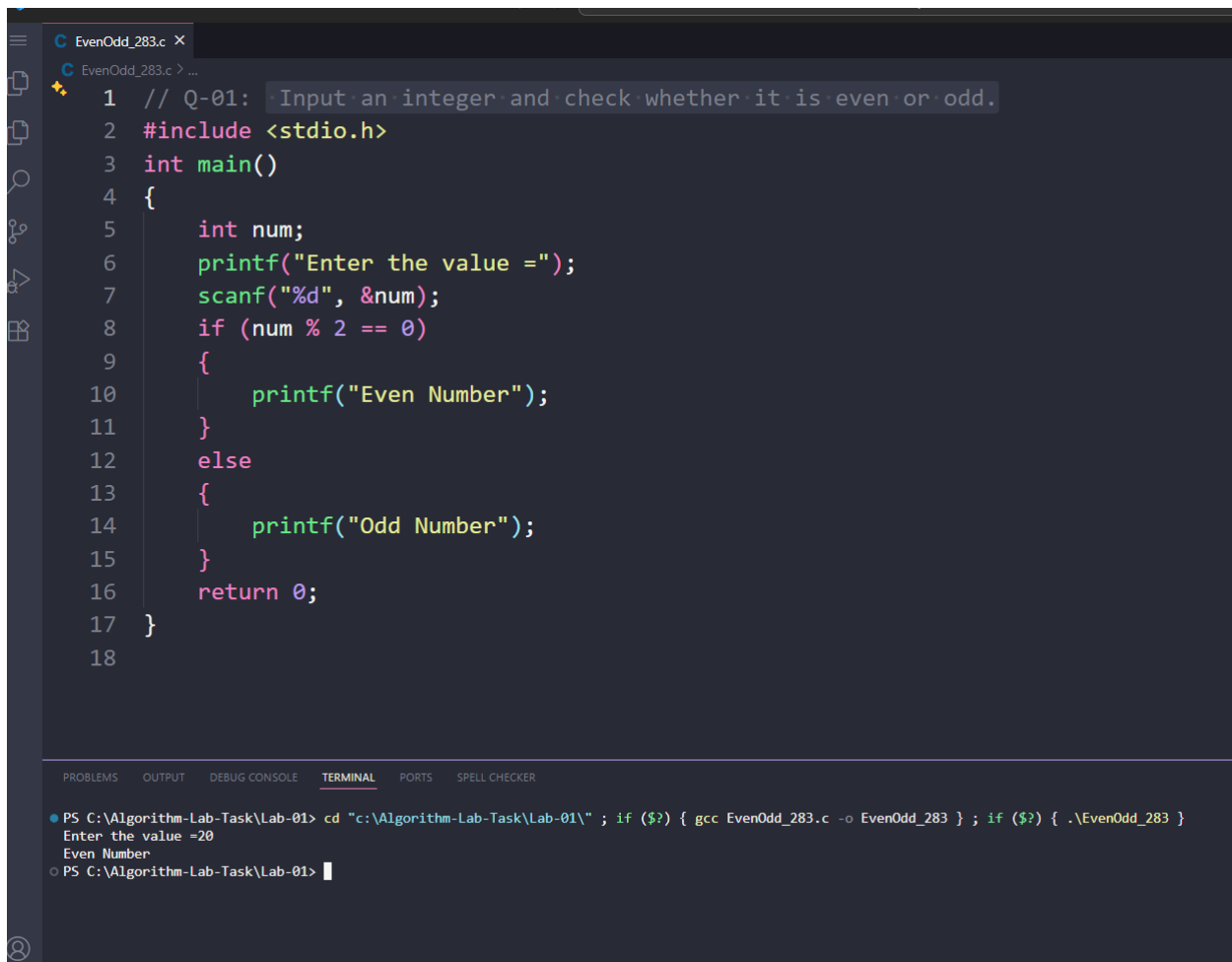
Intake : 51

Section : 06 (int-53)

### Algorithm Lab 01 (Home Task)

- Q-1 . Even or Odd : Input an integer and check whether it is even or odd.

Code with Output :



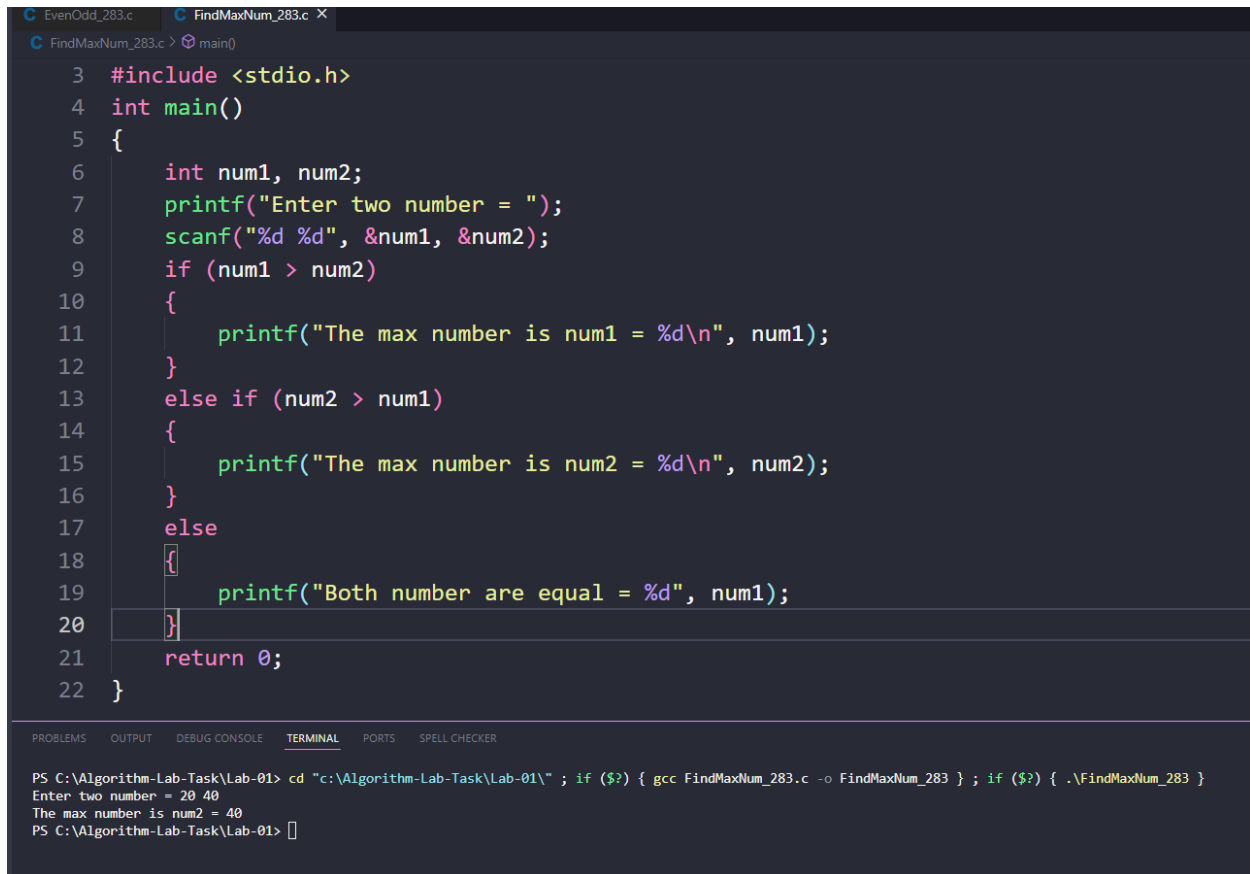
```
1 // Q-01: Input an integer and check whether it is even or odd.
2 #include <stdio.h>
3 int main()
4 {
5     int num;
6     printf("Enter the value =");
7     scanf("%d", &num);
8     if (num % 2 == 0)
9     {
10        printf("Even Number");
11    }
12    else
13    {
14        printf("Odd Number");
15    }
16    return 0;
17 }
18
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS SPELL CHECKER

```
PS C:\Algorithm-Lab-Task\Lab-01> cd "c:\Algorithm-Lab-Task\Lab-01\" ; if ($?) { gcc EvenOdd_283.c -o EvenOdd_283 } ; if ($?) { .\EvenOdd_283 }
Enter the value =20
Even Number
PS C:\Algorithm-Lab-Task\Lab-01>
```

- Q-2 . Find the Maximum of Two Numbers: Take two numbers as input and print the larger one.

Code with Output :



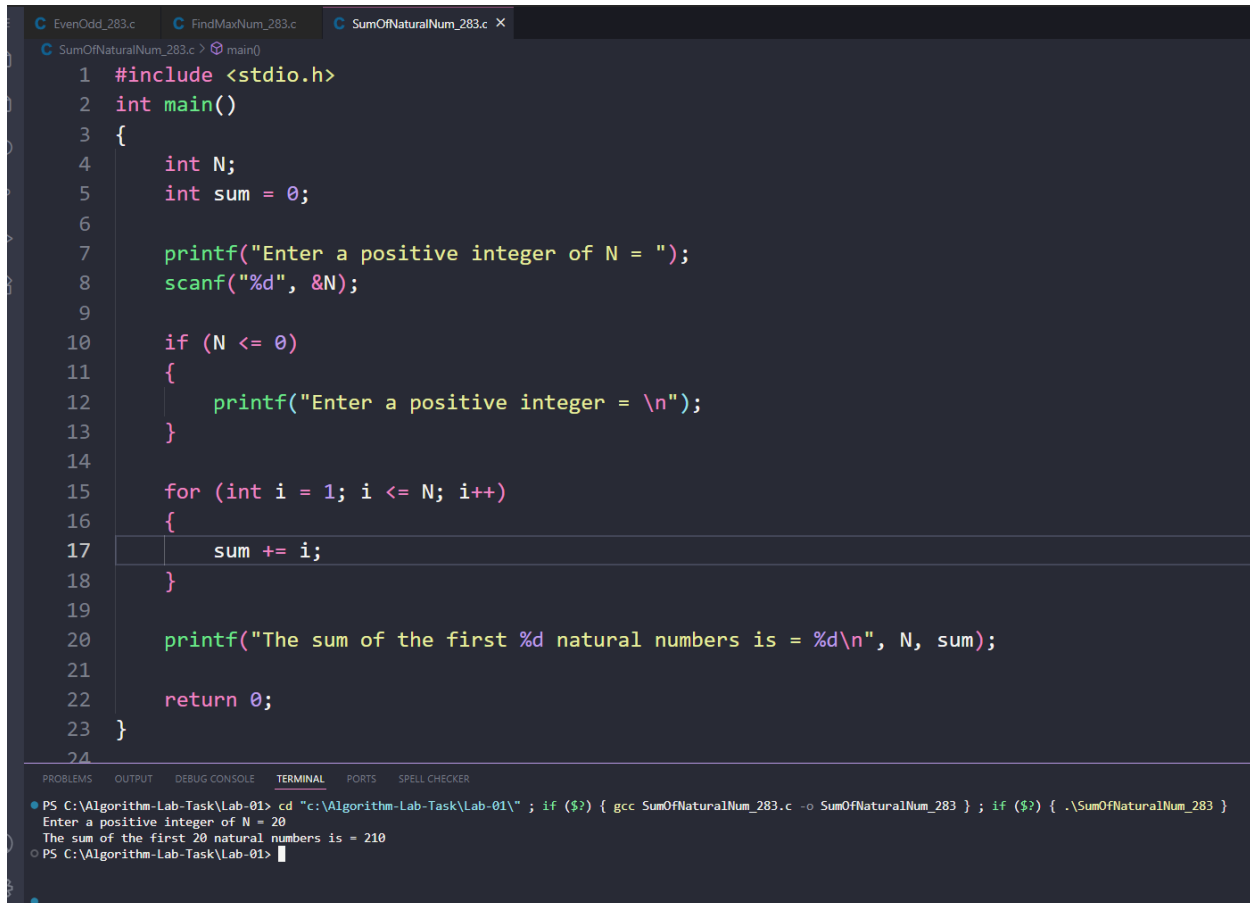
```
3  #include <stdio.h>
4  int main()
5  {
6      int num1, num2;
7      printf("Enter two number = ");
8      scanf("%d %d", &num1, &num2);
9      if (num1 > num2)
10     {
11         printf("The max number is num1 = %d\n", num1);
12     }
13     else if (num2 > num1)
14     {
15         printf("The max number is num2 = %d\n", num2);
16     }
17     else
18     {
19         printf("Both number are equal = %d", num1);
20     }
21     return 0;
22 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS SPELL CHECKER

```
PS C:\Algorithm-Lab-Task\Lab-01> cd "c:\Algorithm-Lab-Task\Lab-01" ; if ($?) { gcc FindMaxNum_283.c -o FindMaxNum_283 } ; if ($?) { .\FindMaxNum_283 }
Enter two number = 20 40
The max number is num2 = 40
PS C:\Algorithm-Lab-Task\Lab-01>
```

- Q-03 . Sum of First N Natural Numbers: Input N and print the sum from 1 to N.

Code With Output :



```
1 #include <stdio.h>
2 int main()
3 {
4     int N;
5     int sum = 0;
6
7     printf("Enter a positive integer of N = ");
8     scanf("%d", &N);
9
10    if (N <= 0)
11    {
12        printf("Enter a positive integer = \n");
13    }
14
15    for (int i = 1; i <= N; i++)
16    {
17        sum += i;
18    }
19
20    printf("The sum of the first %d natural numbers is = %d\n", N, sum);
21
22    return 0;
23 }
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

```
● PS C:\Algorithm-Lab-Task\Lab-01> cd "c:\Algorithm-Lab-Task\Lab-01\" ; if ($?) { gcc SumOfNaturalNum_283.c -o SumOfNaturalNum_283 } ; if ($?) { .\SumOfNaturalNum_283 }
Enter a positive integer of N = 20
The sum of the first 20 natural numbers is = 210
○ PS C:\Algorithm-Lab-Task\Lab-01> █
```

Q-04. Check if a Number is Prime: Input a number and check whether it is a prime number

```
EvenOdd_283.c FindMaxNum_283.c SumOfNaturalNum_283.c PrimeNum_283.c X
C PrimeNum_283.c > main()
1  #include <stdio.h>
2  #include <stdbool.h>
3
4  int main()
5  {
6      int num = 19;
7      int count = 0;
8
9      if (num <= 1)
10     {
11         printf("The number is not prime: %d\n", num);
12     }
13     else
14     {
15         for (int j = 1; j <= num; j++)
16         {
17             if (num % j == 0)
18             {
19                 count++;
20             }
21         }
22
23         if (count > 2)
24         {
25             printf("The number is not prime: %d\n", num);
26         }
27         else
28         {
29             printf("The number is prime: %d\n", num);
30         }
31     }
32
33     return 0;
34 }
```

## Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER
● PS C:\Algorithm-Lab-Task\Lab-01> cd "c:\Algorithm-Lab-Task\Lab-01\" ; if ($?) { gcc PrimeNum_283.c -o PrimeNum_283 } ; if ($?) { .\PrimeNum_283 }
The number is prime: 19
○ PS C:\Algorithm-Lab-Task\Lab-01> █
```

- Q-05. Factorial of a Number: Input a number and calculate its factorial.

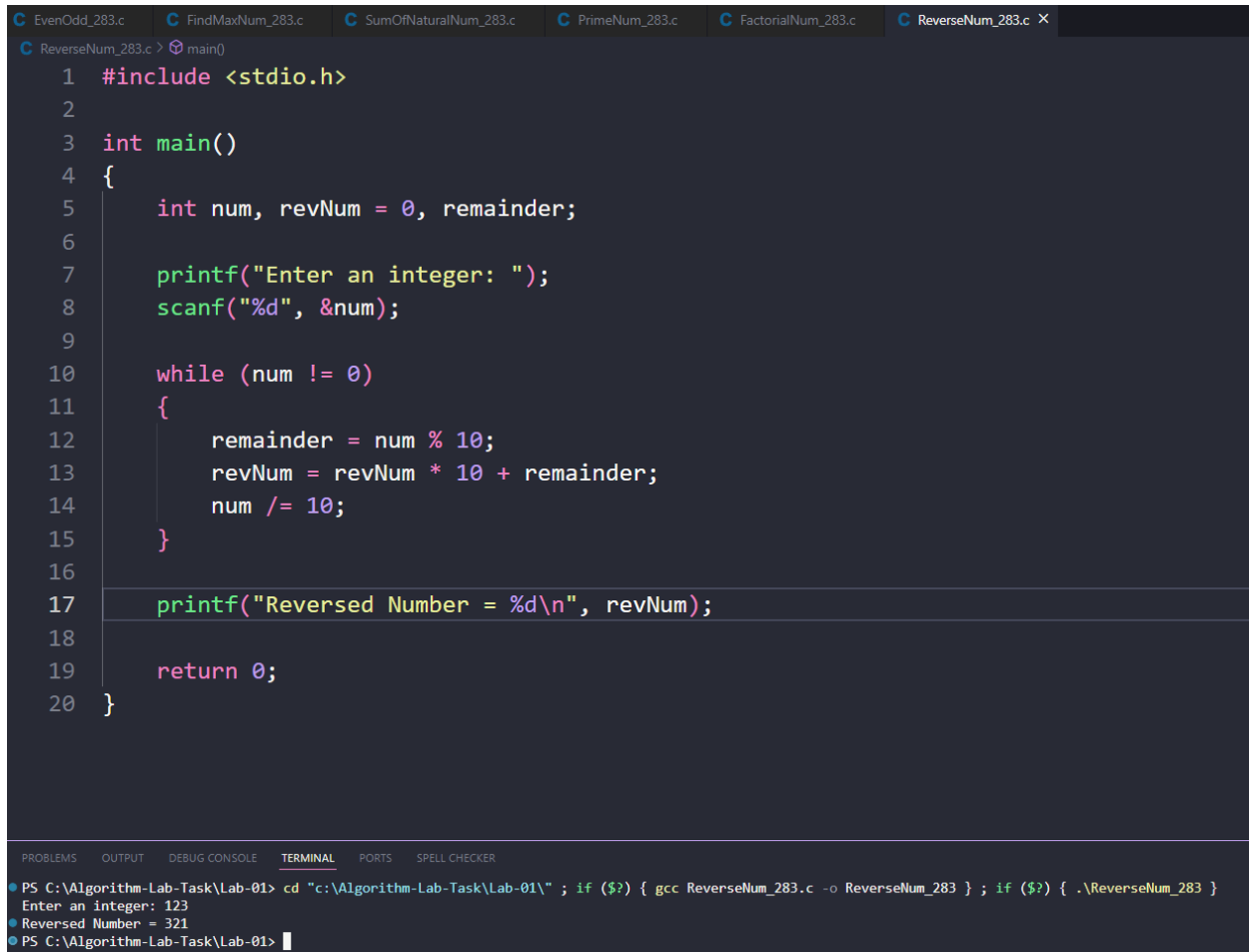
## Code with output :

```
EvenOdd_283.c FindMaxNum_283.c SumOfNaturalNum_283.c PrimeNum_283.c FactorialNum_283.c X
C FactorialNum_283.c > main()
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, factorial = 1;
6
7      printf("Enter a positive integer = ");
8      scanf("%d", &n);
9
10     if (n < 0)
11     {
12         printf("Factorial isn't defined for negative numbers : \n");
13     }
14     else
15     {
16         for (int i = 1; i <= n; i++)
17         {
18             factorial *= i;
19         }
20         printf("Factorial of %d is %d\n", n, factorial);
21     }
22
23     return 0;
24 }
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER
● PS C:\Algorithm-Lab-Task\Lab-01> cd "c:\Algorithm-Lab-Task\Lab-01\" ; if ($?) { gcc FactorialNum_283.c -o FactorialNum_283 } ; if ($?) { .\FactorialNum_283 }
Enter a positive integer = 6
● Factorial of 6 is 720
○ PS C:\Algorithm-Lab-Task\Lab-01> █
```

- Q-06. Reverse a Number: Input a number and print its reverse., Example: 123 → 321

Code With Output :



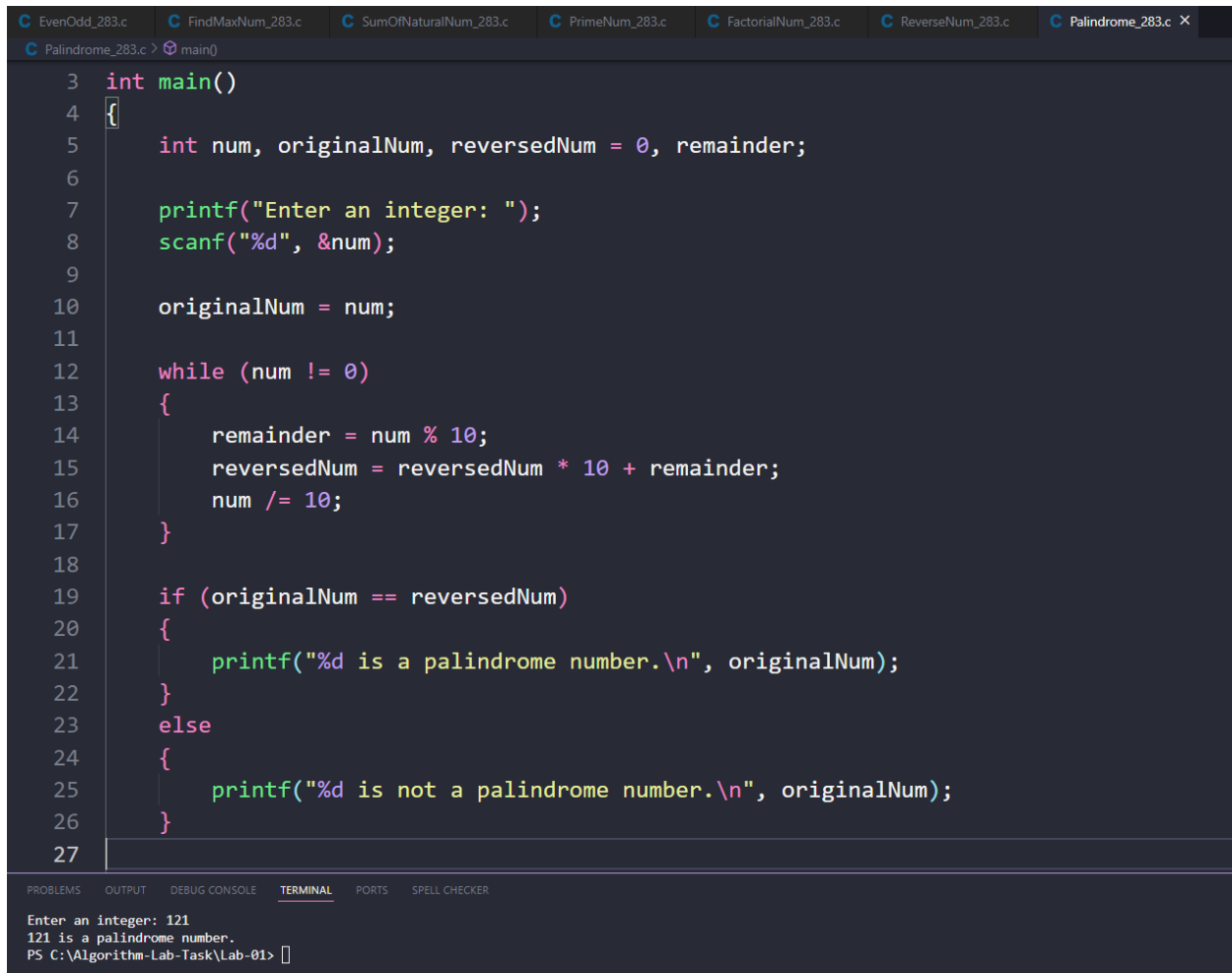
```
C EvenOdd_283.c C FindMaxNum_283.c C SumOfNaturalNum_283.c C PrimeNum_283.c C FactorialNum_283.c C ReverseNum_283.c X
C ReverseNum_283.c > main()
1  #include <stdio.h>
2
3  int main()
4  {
5      int num, revNum = 0, remainder;
6
7      printf("Enter an integer: ");
8      scanf("%d", &num);
9
10     while (num != 0)
11     {
12         remainder = num % 10;
13         revNum = revNum * 10 + remainder;
14         num /= 10;
15     }
16
17     printf("Reversed Number = %d\n", revNum);
18
19     return 0;
20 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

```
PS C:\Algorithm-Lab-Task\Lab-01> cd "c:\Algorithm-Lab-Task\Lab-01\" ; if ($?) { gcc ReverseNum_283.c -o ReverseNum_283 } ; if ($?) { .\ReverseNum_283 }
Enter an integer: 123
Reversed Number = 321
PS C:\Algorithm-Lab-Task\Lab-01>
```

- Q-07. Check if a Number is Palindrome: Input a number and check whether it reads the same forward and backward.

Code With Output :



```
3 int main()
4 {
5     int num, originalNum, reversedNum = 0, remainder;
6
7     printf("Enter an integer: ");
8     scanf("%d", &num);
9
10    originalNum = num;
11
12    while (num != 0)
13    {
14        remainder = num % 10;
15        reversedNum = reversedNum * 10 + remainder;
16        num /= 10;
17    }
18
19    if (originalNum == reversedNum)
20    {
21        printf("%d is a palindrome number.\n", originalNum);
22    }
23    else
24    {
25        printf("%d is not a palindrome number.\n", originalNum);
26    }
27 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS SPELL CHECKER

Enter an integer: 121  
121 is a palindrome number.  
PS C:\Algorithm-Lab-Task\Lab-01>

- Q-08. Print First N Fibonacci Numbers: Input N and print the first N Fibonacci numbers.

Code With Output :

```

C EvenOdd_283.c  C FindMaxNum_283.c  C SumOfNaturalNum_283.c  C PrimeNum_283.c  C FactorialNum_283.c  C ReverseNum_283.c  C Palindrome_283.c  C FibonacciNum_283.c X
C FibonacciNum_283.c > main()
3  int main()
4  {
5      int N;
6      long long x1 = 0, x2 = 1;
7      long long nextTerm = x1 + x2;
8
9      printf("Enter the number of Fibonacci numbers to print: ");
10     scanf("%d", &N);
11
12     if (N <= 0)
13     {
14         printf("Please enter a positive integer.\n");
15         return 1;
16     }
17     else if (N == 1)
18     {
19         printf("Fibonacci Series: %lld\n", x1);
20         return 0;
21     }
22     else
23     {
24         printf("Fibonacci Series: %lld, %lld", x1, x2);
25
26         for (int i = 3; i <= N; ++i)
27         {
28             printf(", %lld", nextTerm);
29             x1 = x2;
30             x2 = nextTerm;
31             nextTerm = x1 + x2;
32         }
33         printf("\n");
34     }
35
36     return 0;
37 }
```

Output :

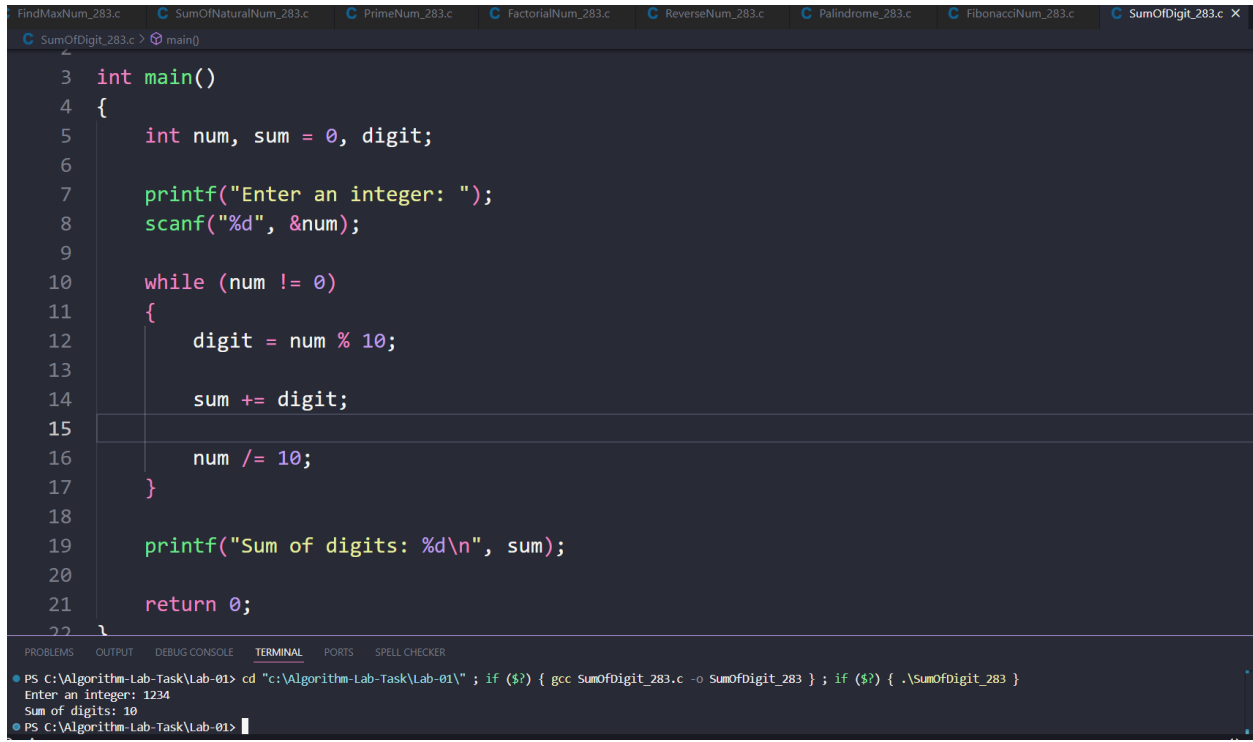
```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SPELL CHECKER
● Enter the number of Fibonacci numbers to print: 10
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34
PS C:\Algorithm-Lab-Task\Lab-01> cd "c:\Algorithm-Lab-Task\Lab-01\" ; if ($?) { gcc FibonacciNum_283.c -o FibonacciNum_283 } ; if ($?) { .\FibonacciNum_283 }
● Enter the number of Fibonacci numbers to print: 
0 0 0
```



**Q-09. Sum of Digits of a Number:** Input a number and print the sum of its digits.

**Code With Output :**



The screenshot shows a C++ IDE with a dark theme. The top bar displays several open files: `HindMaxNum_283.c`, `SumOfNaturalNum_283.c`, `PrimeNum_283.c`, `FactorialNum_283.c`, `ReverseNum_283.c`, `Palindrome_283.c`, `FibonacciNum_283.c`, and `SumOfDigit_283.c`. The `SumOfDigit_283.c` file is active, showing the following C++ code:

```
3 int main()
4 {
5     int num, sum = 0, digit;
6
7     printf("Enter an integer: ");
8     scanf("%d", &num);
9
10    while (num != 0)
11    {
12        digit = num % 10;
13
14        sum += digit;
15
16        num /= 10;
17    }
18
19    printf("Sum of digits: %d\n", sum);
20
21    return 0;
22 }
```

The bottom panel shows the terminal output for the program:

```
PS C:\Algorithm-Lab-Task\Lab-01> cd "c:\Algorithm-Lab-Task\Lab-01\" ; if ($?) { gcc SumOfDigit_283.c -o SumOfDigit_283 } ; if ($?) { .\SumOfDigit_283 }
Enter an integer: 1234
Sum of digits: 10
PS C:\Algorithm-Lab-Task\Lab-01> 
```

**Q-10. Bubble Sort:** Input an unsorted array of size N and sort it using bubble sort.

**Code With Output :**

```
EvenOdd_283.c  FindMaxNum_283.c  SumOfNaturalNum_283.c  PrimeNum_283.c  FactorialNum_283.c  Reverse
BubbleSort_283.c > printArray(int [], int)
1  #include <stdio.h>
2
3  void bubbleSort(int arr[], int size)
4  {
5      for (int step = 0; step < size - 1; ++step)
6      {
7          for (int i = 0; i < size - step - 1; ++i)
8          {
9
10             if (arr[i] > arr[i + 1])
11             {
12
13                 int temp = arr[i];
14                 arr[i] = arr[i + 1];
15                 arr[i + 1] = temp;
16             }
17         }
18     }
19 }
20
21 void printArray(int arr[], int size)
```

```

20
21 void printArray(int arr[], int size)
22 {
23     for (int i = 0; i < size; ++i)
24     {
25         printf("%d ", arr[i]);
26     }
27     printf("\n");
28 }
29
30 int main()
31 {
32     int data[] = {7, 1, 3, 2, 8};
33     int size = sizeof(data) / sizeof(data[0]);
34
35     printf("Original Array: \n");
36     printArray(data, size);
37
38     bubbleSort(data, size);
39
40     printf("Sorted Array in Ascending Order: \n");
41     printArray(data, size);
42
43     return 0;
44 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

7 1 3 2 8

● Sorted Array in Ascending Order:

1 2 3 7 8

○ PS C:\Algorithm-Lab-Task\Lab-01> |

Q-11: Linear Search: Input an array of size N and find an element using linear search.

Code With Output :

```
LinearSearch_283.c > main()
1  #include <stdio.h>
2
3  int linearSearch(int *arr, int n, int key)
4  {
5
6      for (int i = 0; i < n; i++)
7      {
8
9          if (arr[i] == key)
10         {
11             return i;
12         }
13     }
14
15     return -1;
16 }
17
18 int main()
19 {
20     int arr[] = {1, 3, 8, 5, 4};
21     int n = sizeof(arr) / sizeof(arr[0]);
22     int key = 4;
23
24     int i = linearSearch(arr, n, key);
25
26     if (i == -1)
27         printf("Key Not Found");
28     else
29         printf("Key Found at Index: %d", i);
30
31     return 0;
32 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS SPELL CHECKER

```
PS C:\Algorithm-Lab-Task\Lab-01> cd "c:\Algorithm-Lab-Task\Lab-01" ; if ($?) { gcc LinearSearch_283.c -o LinearSearch_283 } ; if ($?) { .\LinearSearch_283 }
Key Found at Index: 4
PS C:\Algorithm-Lab-Task\Lab-01> 
```

**Q-12. Binary Search:** Input a sorted array and search for an element using binary search.

Code With Output :

```
EvenOdd_283.c FindMaxNum_283.c SumOfNaturalNum_283.c PrimeNum_283.c FactorialNum_283.c ReverseNum_283.c Palindrome_283.c FibonacciNum_283.c Su
BinarySearch_283.c > main()
1  #include <stdio.h>
2
3  int binarySearch(int arr[], int left, int right, int key)
4  {
5      while (left <= right)
6      {
7          int mid = left + (right - left) / 2;
8
9          if (arr[mid] == key)
10         {
11             return mid;
12         }
13
14         if (arr[mid] < key)
15         {
16             left = mid + 1;
17         }
18         else
19         {
20             right = mid - 1;
21         }
22     }
23
24     return -1;
25 }
26
27 int main()
28 {
29     int arr[] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91};
30     int n = sizeof(arr) / sizeof(arr[0]);
31     int key = 12;
```

```
EvenOdd_283.c FindMaxNum_283.c SumOfNaturalNum_283.c PrimeNum_283.c FactorialNum_283.c ReverseNum_283.c Palindrome_283.c
BinarySearch_283.c > main()
27 int main()
31     int key = 12;
32
33     printf("Array: ");
34     for (int i = 0; i < n; i++)
35     {
36         printf("%d ", arr[i]);
37     }
38     printf("\n");
39
40     int result = binarySearch(arr, 0, n - 1, key);
41
42     if (result == -1)
43     {
44         printf("Element %d is not present in the array.\n", key);
45     }
46     else
47     {
48         printf("Element %d is present at index %d.\n", key, result);
49     }
50
51     key = 25;
52     result = binarySearch(arr, 0, n - 1, key);
53     if (result == -1)
54     {
55         printf("Element %d is not present in the array.\n", key);
56     }
57     else
58     {
59         printf("Element %d is present at index %d.\n", key, result);
60     }
61
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS SPELL CHECKER

Array: 2 5 8 12 16 23 38 56 72 91  
● Element 12 is present at index 3.  
● Element 25 is not present in the array.  
PS C:\Algorithm-Lab-Task\Lab-01>

**Q-13. Selection Sort:** Sort an unsorted array using selection sort., Example: 64 25 12 22 11 → 11 12 22 25 64

Code With Output :

```
FindMaxNum_283.c SumOfNaturalNum_283.c PrimeNum_283.c FactorialNum_283.c ReverseNum_283.c Palindrome_283.c FibonacciNum_283.c
SelectionSort_283.c > printArray(int [], int)
1  #include <stdio.h>
2
3  void swap(int *a, int *b)
4  {
5      int temp = *a;
6      *a = *b;
7      *b = temp;
8  }
9
10 void selectionSort(int arr[], int n)
11 {
12     int i, j, min_idx;
13
14     for (i = 0; i < n - 1; i++)
15     {
16         min_idx = i;
17         for (j = i + 1; j < n; j++)
18         {
19             if (arr[j] < arr[min_idx])
20             {
21                 min_idx = j;
22             }
23         }
24
25         if (min_idx != i)
26         {
27             swap(&arr[min_idx], &arr[i]);
28         }
29     }
30 }
31
```

```

10 void selectionSort(int arr[], int n)
31
32 void printArray(int arr[], int size)
33 {
34     for (int i = 0; i < size; i++)
35     {
36         printf("%d ", arr[i]);
37     }
38     printf("\n");
39 }
40
41 int main()
42 {
43     int arr[] = {64, 25, 12, 22, 11};
44     int n = sizeof(arr) / sizeof(arr[0]);
45
46     printf("Original array: \n");
47     printArray(arr, n);
48
49     selectionSort(arr, n);
50
51     printf("Sorted array: \n");
52     printArray(arr, n);
53
54     return 0;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER

64 25 12 22 11

● Sorted array:

11 12 22 25 64

○ PS C:\Algorithm-Lab-Task\Lab-01>