1) Write a function (IsPrime) to check whether a number is **prime or not**.

Input:43

Output: Prime

Input:56

Output: Not Prime

2) Write a function to find the **sum of digits** of a number.

Input: 12345

Output:15

3) Print a **pattern** like this.

Input: 4

Output:

*

**

**

****

4) Check if a string is a **palindrome**, without using any built-in functions.

Input: racecar

Output: Is Palindrome

Input: water
Output: Not Palindrome

5) Create a **function** to find the **sum of elements in an array** using pointers.

- The function should take a pointer to the array and the size of the array as arguments.
- It should return the sum of all elements.

6) Write a program to **sort an array** using bubble sort and selection sort. Don't use any built-in functions.

Input: [3,5,4,11,9,2,1]

Output: [1,2,3,4,5,9,11]

7) Create a class Student with data members: name, roll number, and marks. Write functions to:

- Input details
- Display details
- Calculate percentage

Input:

Enter student name: John Doe

Enter roll number: 1

Enter marks in 3 subjects: 80 90 70

Output:

Student Details:

Name: John Doe

Roll Number: 1

Marks: 80 90 70

Percentage: 80%

8) Write a **menu-driven program** in C++ to perform the following operations using recursion:

1. Find the **factorial** of a number.
2. Find the **GCD** of two numbers.

Create appropriate functions:

- int Factorial (int n) – returns factorial of n.
- int Gcd (int a, int b) – returns GCD of a and b.

The program should display a menu to the user, take input, and perform the selected operation.

9) Create a class Book with the following data members:

- title (string)
- author (string)
- price (float)

and the following **member functions**:

- void input() — to input details of a book
- void display() — to display details of a book

Write a program to:

1. Create an **array of Book objects**.
2. Input details for all books using the input() function.
3. Display all books using the display() function.

10) Implement a **stack** using arrays in C++.

- Create functions for the following operations:
    1. push() – to insert an element into the stack
    2. pop() – to remove the top element from the stack
    3. display() – to display all elements of the stack
- Use **array-based implementation** and handle **stack overflow** and **underflow**.
- Write a **menu-driven program** to test these functions.

11) Write a program to process a 2D array of integers:

- Traverse each element and check whether it is a **palindrome**. Replace palindromes with **1** and non-palindromes with **0**.
- Print the resulting **matrix of 1s and 0s**.

  Input:

  121 23 44

  56 77 89

  33 12 9

  Output:

  1 0 1

  0 1 0

  1 0 1

12) Write a program to **find the row or column with the maximum sum** in a matrix.

Input:

Enter the number of rows: 3

Enter the number of columns: 3

Enter elements of the matrix:

1 2 3

4 5 6

7 8 9

Output:

Sum of each row:

Row 0: 6

Row 1: 15

Row 2: 24

Sum of each column:

Column 0: 12

Column 1: 15

Column 2: 18

Row 2 has the maximum sum: 24

Column 2 has the maximum sum: 18

Optional:

13) Write a **C++ program** to implement a **doubly linked list** that allows insertion at the rear (end) and deletion from both ends. It should be a menu driven program.