

### 1. Determining Even/Odd Numbers

**Difficulty:** Easy

**Topics:** Basic Programming

**Description:** Write a program to check whether a number is even or odd.

**Example:**

Input: `number = 4`

Output: `Even`

Explanation: Since 4 is divisible by 2, it is an even number.

### 2. Checking for Prime Numbers

**Difficulty:** Easy

**Topics:** Basic Programming, Number Theory

**Description:** Write a program to determine if a number is prime.

**Example:**

Input: `number = 7`

Output: `Prime`

Explanation: 7 has no divisors other than 1 and itself, so it is a prime number.

### 3. Validating Leap Years

**Difficulty:** Easy

**Topics:** Basic Programming, Date Handling

**Description:** Write a program to check if a given year is a leap year.

**Example:**

Input: `year = 2020`

Output: `Leap Year`

Explanation: 2020 is divisible by 4 but not by 100, or it is divisible by 400, so it is a leap year.

### 4. Calculating Armstrong Numbers

**Difficulty:** Easy

**Topics:** Basic Programming, Number Theory

**Description:** Write a program to check if a number is an Armstrong number.

**Example:**

Input: `number = 153`

Output: `Armstrong Number`

Explanation: 153 is an Armstrong number because  $1^3 + 5^3 + 3^3 = 153$ .

### 5. Generating the Fibonacci Series

**Difficulty:** Easy

**Topics:** Basic Programming, Sequences

**Description:** Write a program to generate the Fibonacci series up to a given number.

**Example:**

Input: `limit = 10`

Output: `[0, 1, 1, 2, 3, 5, 8]`

Explanation: The Fibonacci series up to 10 is generated as [0, 1, 1, 2, 3, 5, 8].

### 6. Identifying Palindromes

**Difficulty:** Easy

**Topics:** Basic Programming, String Manipulation

**Description:** Write a program to check if a string or number is a palindrome.

**Example:**

Input: `string = "radar"`

Output: `Palindrome`

Explanation: "radar" reads the same backward as forward.

## 7. Crafting Star Patterns

**Difficulty:** Easy

**Topics:** Basic Programming, Patterns

**Description:** Write a program to create different star patterns (e.g., pyramid, diamond).

**Example:**

Input: `patternType = "pyramid", height = 5`

Output:

```
  *
 ***
*****
*****
*****
```

Explanation: A pyramid pattern with a height of 5 is generated.

## 8. Finding the Factorial of a Number

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to compute the factorial of a given number.

**Example:**

Input: `number = 5`

Output: `120`

Explanation:  $5!$  (factorial) is  $5 \times 4 \times 3 \times 2 \times 1 = 120$ .

## 9. Summing Digits of a Number

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to calculate the sum of digits of a number.

**Example:**

Input: `number = 1234`

Output: `10`

Explanation: The sum of the digits  $1 + 2 + 3 + 4 = 10$ .

## 10. Finding the Greatest Common Divisor (GCD)

**Difficulty:** Easy

**Topics:** Basic Programming, Number Theory

**Description:** Write a program to find the GCD of two numbers.

**Example:**

Input: `a = 48, b = 18`

Output: 6

Explanation: The GCD of 48 and 18 is 6.

#### 11. Finding the Least Common Multiple (LCM)

**Difficulty:** Easy

**Topics:** Basic Programming, Number Theory

**Description:** Write a program to find the LCM of two numbers.

**Example:**

Input: `a = 12, b = 15`

Output: 60

Explanation: The LCM of 12 and 15 is 60.

#### 12. Counting Vowels and Consonants in a String

**Difficulty:** Easy

**Topics:** Basic Programming, String Manipulation

**Description:** Write a program to count vowels and consonants in a given string.

**Example:**

Input: `string = "hello world"`

Output: `Vowels: 3, Consonants: 7`

Explanation: "hello world" contains 3 vowels (e, o, o) and 7 consonants (h, l, l, w, r, l, d).

#### 13. Reversing a String

**Difficulty:** Easy

**Topics:** Basic Programming, String Manipulation

**Description:** Write a program to reverse a given string.

**Example:**

Input: `string = "programming"`

Output: `"gnimmargorp"`

Explanation: The reversed string of "programming" is "gnimmargorp".

#### 14. Finding the Largest and Smallest Numbers in an Array

**Difficulty:** Easy

**Topics:** Basic Programming, Arrays

**Description:** Write a program to find the largest and smallest numbers in an array.

**Example:**

Input: `array = [4, 7, 1, 8, 5]`

Output: `Largest: 8, Smallest: 1`

Explanation: The largest number in the array is 8 and the smallest is 1.

#### 15. Sorting an Array

**Difficulty:** Easy

**Topics:** Basic Programming, Sorting Algorithms

**Description:** Write a program to sort an array of numbers in ascending order.

**Example:**

Input: `array = [3, 1, 4, 1, 5, 9]`

Output: `[1, 1, 3, 4, 5, 9]`

Explanation: The array sorted in ascending order is [1, 1, 3, 4, 5, 9].

**16. Finding the Sum of Elements in an Array****Difficulty:** Easy**Topics:** Basic Programming, Arrays**Description:** Write a program to find the sum of elements in an array.**Example:**Input: `array = [1, 2, 3, 4, 5]`Output: `15`

Explanation: The sum of the elements in the array is 15.

**17. Checking for Armstrong Numbers in a Range****Difficulty:** Easy**Topics:** Basic Programming, Number Theory**Description:** Write a program to find all Armstrong numbers within a given range.**Example:**Input: `range = [1, 500]`Output: `[1, 153, 370, 371, 407]`

Explanation: Armstrong numbers between 1 and 500 are 1, 153, 370, 371, and 407.

**18. Generating Multiplication Tables****Difficulty:** Easy**Topics:** Basic Programming, Mathematical Computations**Description:** Write a program to generate multiplication tables for a given number.**Example:**Input: `number = 4`

Output:

```
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
```

Explanation: The multiplication table for 4 up to 5 is generated.

**19. Finding Prime Numbers in a Range****Difficulty:** Easy**Topics:** Basic Programming, Number Theory**Description:** Write a program to find all prime numbers within a given range.**Example:**Input: `range = [10, 30]`Output: `[11, 13, 17, 19, 23, 29]`

Explanation: Prime numbers between 10 and 30 are 11, 13, 17, 19, 23, and 29.

**20. Checking for Perfect Numbers****Difficulty:** Easy**Topics:** Basic Programming, Number Theory**Description:** Write a program to determine if a number is a perfect number.

**Example:**

Input: `number = 28`

Output: `Perfect Number`

Explanation: 28 is a perfect number because its divisors (1, 2, 4, 7, 14) sum up to 28.

**21. Calculating the Sum of Even Numbers in a Range**

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to find the sum of all even numbers within a given range.

**Example:**

Input: `range = [1, 10]`

Output: `30`

Explanation: The sum of even numbers between 1 and 10 is  $2 + 4 + 6 + 8 + 10 = 30$ .

**22. Calculating the Sum of Odd Numbers in a Range**

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to find the sum of all odd numbers within a given range.

**Example:**

Input: `range = [1, 10]`

Output: `25`

Explanation: The sum of odd numbers between 1 and 10 is  $1 + 3 + 5 + 7 + 9 = 25$ .

**23. Finding the Fibonacci Number at a Specific Position**

**Difficulty:** Easy

**Topics:** Basic Programming, Sequences

**Description:** Write a program to find the Fibonacci number at a specific position.

**Example:**

Input: `position = 5`

Output: `5`

Explanation: The Fibonacci number at position 5 is 5 (sequence: 0, 1, 1, 2, 3, 5).

**24. Printing Prime Numbers Less Than a Given Number**

**Difficulty:** Easy

**Topics:** Basic Programming, Number Theory

**Description:** Write a program to print all prime numbers less than a given number.

**Example:**

Input: `number = 20`

Output: `2, 3, 5, 7, 11, 13, 17, 19`

Explanation: The prime numbers less than 20 are 2, 3, 5, 7, 11, 13, 17, and 19.

**25. Finding the Number of Digits in a Number**

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to count the number of digits in a given number.

**Example:**

Input: `number = 12345`

Output: 5

Explanation: The number 12345 has 5 digits.

## 26. Checking if a Number is a Narcissistic Number

**Difficulty:** Easy

**Topics:** Basic Programming, Number Theory

**Description:** Write a program to check if a number is a narcissistic number (where the sum of its digits raised to the power of the number of digits equals the number itself).

**Example:**

Input: `number = 153`

Output: `Narcissistic Number`

Explanation: 153 is a narcissistic number because  $1^3 + 5^3 + 3^3 = 153$ .

## 27. Generating a Pattern of Numbers

**Difficulty:** Easy

**Topics:** Basic Programming, Patterns

**Description:** Write a program to generate number patterns (e.g., sequential numbers in a matrix).

**Example:**

Input: `rows = 3`

Output:

```
1
2 3
4 5 6
```

Explanation: A number pattern with 3 rows is generated.

## 28. Finding the Sum of the Digits of the Factorial of a Number

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to find the sum of the digits of the factorial of a given number.

**Example:**

Input: `number = 4`

Output: 9

Explanation: The factorial of 4 is 24, and the sum of the digits (2 + 4) is 6.

## 29. Finding the Largest Palindrome in a String

**Difficulty:** Easy

**Topics:** Basic Programming, String Manipulation

**Description:** Write a program to find the largest palindrome in a given string.

**Example:**

Input: `string = "babad"`

Output: `"bab"` or `"aba"`

Explanation: Both "bab" and "aba" are valid palindromes in the string.

## 30. Finding Missing Numbers in a Sequence

**Difficulty:** Easy

**Topics:** Basic Programming, Arrays

**Description:** Write a program to find missing numbers in a sequence from 1 to n.

**Example:**

Input: `sequence = [1, 2, 4, 5]`

Output: `[3]`

Explanation: The missing number in the sequence from 1 to 5 is 3.

### 31. Generating a Pascal's Triangle

**Difficulty:** Medium

**Topics:** Arrays, Mathematical Computations

**Description:** Write a program to generate Pascal's Triangle up to a given number of rows.

**Example:**

Input: `rows = 4`

Output:

```
1
1 1
1 2 1
1 3 3 1
```

Explanation: Pascal's Triangle with 4 rows is generated.

### 32. Finding the Median of an Array

**Difficulty:** Medium

**Topics:** Arrays, Sorting

**Description:** Write a program to find the median of an array of numbers.

**Example:**

Input: `array = [3, 1, 2, 4, 5]`

Output: `3`

Explanation: The median of the sorted array [1, 2, 3, 4, 5] is 3.

### 33. Calculating the Power of a Number

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to calculate the power of a number.

**Example:**

Input: `base = 2, exponent = 3`

Output: `8`

Explanation: 2 raised to the power of 3 is 8.

### 34. Checking for an Anagram

**Difficulty:** Easy

**Topics:** String Manipulation

**Description:** Write a program to check if two strings are anagrams.

**Example:**

Input: `string1 = "listen", string2 = "silent"`

Output: **True**

Explanation: "listen" and "silent" are anagrams of each other.

### 35. Finding the Sum of Prime Numbers in a Range

**Difficulty:** Medium

**Topics:** Number Theory, Mathematical Computations

**Description:** Write a program to calculate the sum of all prime numbers within a given range.

**Example:**

Input: **range** = [1, 10]

Output: **17**

Explanation: The sum of prime numbers between 1 and 10 is  $2 + 3 + 5 + 7 = 17$ .

### 36. Finding the N-th Triangular Number

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to find the N-th triangular number.

**Example:**

Input: **N** = 4

Output: **10**

Explanation: The 4th triangular number is 10 (sum of the first 4 natural numbers).

### 37. Checking for Perfect Squares

**Difficulty:** Easy

**Topics:** Mathematical Computations

**Description:** Write a program to determine if a number is a perfect square.

**Example:**

Input: **number** = 16

Output: **True**

Explanation: 16 is a perfect square ( $4^2 = 16$ ).

### 38. Finding the Sum of Squares of Digits

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to find the sum of the squares of the digits of a number.

**Example:**

Input: **number** = 123

Output: **14**

Explanation: The sum of the squares of digits is  $1^2 + 2^2 + 3^2 = 14$ .

### 39. Generating a Square Matrix of a Given Size

**Difficulty:** Medium

**Topics:** Arrays, Matrix Operations

**Description:** Write a program to generate a square matrix of a given size and fill it with sequential numbers.

**Example:**

Input: **size** = 3

Output:



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

Explanation: A 3x3 matrix is generated with sequential numbers.

#### 40. Calculating the Sum of Digits of a Number Until Single Digit

**Difficulty:** Medium

**Topics:** Mathematical Computations

**Description:** Write a program to keep summing the digits of a number until a single digit is obtained.

**Example:**

Input: `number = 9875`

Output: `2`

Explanation: The sum of digits is  $9 + 8 + 7 + 5 = 29$ , and then  $2 + 9 = 11$ , and finally  $1 + 1 = 2$ .

#### 41. Finding the Count of Specific Digits in a Number

**Difficulty:** Easy

**Topics:** Basic Programming, String Manipulation

**Description:** Write a program to count the occurrences of a specific digit in a number.

**Example:**

Input: `number = 122333, digit = 3`

Output: `3`

Explanation: The digit 3 occurs 3 times in the number 122333.

#### 42. Generating a Fibonacci Sequence Using Recursion

**Difficulty:** Medium

**Topics:** Recursion, Sequences

**Description:** Write a recursive program to generate the Fibonacci sequence up to a given number.

**Example:**

Input: `number = 5`

Output: `0, 1, 1, 2, 3`

Explanation: The Fibonacci sequence up to 5 is generated.

#### 43. Finding All Divisors of a Number

**Difficulty:** Easy

**Topics:** Basic Programming, Mathematical Computations

**Description:** Write a program to find all divisors of a given number.

**Example:**

Input: `number = 12`

Output: `1, 2, 3, 4, 6, 12`

Explanation: The divisors of 12 are 1, 2, 3, 4, 6, and 12.

#### 44. Finding the Average of Numbers in an Array

**Difficulty:** Easy

**Topics:** Arrays, Mathematical Computations

**Description:** Write a program to calculate the average of numbers in an array.

**Example:**

Input: `array = [1, 2, 3, 4, 5]`

Output: `3`

Explanation: The average of the numbers is  $(1 + 2 + 3 + 4 + 5) / 5 = 3$ .

#### 45. Finding the Mode of Numbers in an Array

**Difficulty:** Medium

**Topics:** Arrays, Statistical Analysis

**Description:** Write a program to find the mode (most frequent number) in an array.

**Example:**

Input: `array = [1, 2, 2, 3, 4, 4, 4]`

Output: `4`

Explanation: The most frequent number in the array is 4.

#### 46. Determining the Length of a String Without Using Built-In Functions

**Difficulty:** Medium

**Topics:** String Manipulation

**Description:** Write a program to determine the length of a string without using built-in functions.

**Example:**

Input: `string = "hello"`

Output: `5`

Explanation: The length of the string "hello" is determined without using built-in functions.

#### 47. Generating a Number Pyramid

**Difficulty:** Medium

**Topics:** Patterns, Basic Programming

**Description:** Write a program to generate a pyramid of numbers (e.g., 1, 12, 123, etc.).

**Example:**

Input: `rows = 4`

Output:

```
1
12
123
1234
```

Explanation: A number pyramid with 4 rows is generated.

#### 48. Finding the Sum of Prime Factors of a Number

**Difficulty:** Medium

**Topics:** Number Theory, Mathematical Computations

**Description:** Write a program to find the sum of the prime factors of a given number.

**Example:**

Input: `number = 12`

Output: `5`

Explanation: The prime factors of 12 are 2 and 3, and their sum is  $2 + 3 = 5$ .

**49. Finding the Second Largest Number in an Array****Difficulty:** Medium**Topics:** Arrays, Sorting**Description:** Write a program to find the second largest number in an array.**Example:**Input: `array = [10, 20, 4, 45, 99]`Output: `45`

Explanation: The second largest number in the array is 45.

**50. Finding the Longest Substring Without Repeating Characters****Difficulty:** Medium**Topics:** String Manipulation, Sliding Window**Description:** Write a program to find the longest substring without repeating characters in a given string.**Example:**Input: `string = "abcabcbb"`Output: `"abc"`

Explanation: The longest substring without repeating characters is "abc".