1. **Finding the Sum of Digits of a Number Until Zero**
   Difficulty: Easy
   Topics: Basic Programming, Mathematical Computations
   Description: Write a program to repeatedly sum the digits of a number until the result is zero.
   Example:
   Input: `number = 123`
   Output: `6`
   Explanation: Sum of digits is 1 + 2 + 3 = 6; sum of digits of 6 is 6 (which is a single digit).

2. **Generating a Multiplication Table for a Range**
   Difficulty: Easy
   Topics: Arrays, Basic Programming
   Description: Write a program to generate multiplication tables for numbers within a specified range.
   Example:
   Input: `start = 2`, `end = 4`
   Output:

```
2 x 1 = 2   3 x 1 = 3   4 x 1 = 4
2 x 2 = 4   3 x 2 = 6   4 x 2 = 8
2 x 3 = 6   3 x 3 = 9   4 x 3 = 12
2 x 4 = 8   3 x 4 = 12  4 x 4 = 16
```

3. **Calculating the Sum of a Series (1 + 1/2 + 1/3 + ... + 1/n)**
   Difficulty: Medium
   Topics: Mathematical Computations
   Description: Write a program to calculate the sum of the series 1 + 1/2 + 1/3 + ... + 1/n up to the nth term.
   Example:
   Input: `n = 4`
   Output: `2.083333`
   Explanation: Sum of the series is 1 + 1/2 + 1/3 + 1/4 ≈ 2.083333.

4. **Finding All Pairs of Elements in an Array that Add Up to a Given Sum**
   Difficulty: Medium
   Topics: Arrays, Hashing
   Description: Write a program to find all pairs of elements in an array whose sum equals a specified target.
   Example:
   Input: `array = [1, 2, 3, 4, 5]`, `target = 5`
   Output: `[(1, 4), (2, 3)]`
   Explanation: Pairs that sum up to 5 are (1, 4) and (2, 3).

5. **Generating a Diamond Pattern of Stars**
   Difficulty: Medium
   Topics: Patterns, Basic Programming
   Description: Write a program to create a diamond pattern with stars of a given size.

Example:
Input: `size = 5`
Output:

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

6. **Counting the Number of Palindromic Substrings in a String**
   Difficulty: Medium
   Topics: String Manipulation
   Description: Write a program to count how many palindromic substrings exist in a given string.
   Example:
   Input: `string = "aaa"`
   Output: `6`
   Explanation: Palindromic substrings are "a", "a", "a", "aa", "aa", "aaa".

7. **Generating a Matrix with Multiplication Table**
   Difficulty: Easy
   Topics: Arrays, Matrix Operations
   Description: Write a program to generate a matrix where each element at position (i, j) is the product of i and j.
   Example:
   Input: `size = 3`
   Output:

```
1 2 3
2 4 6
3 6 9
```

8. **Finding the GCD of Multiple Numbers**
   Difficulty: Medium
   Topics: Mathematical Computations
   Description: Write a program to find the GCD (Greatest Common Divisor) of an array of numbers.
   Example:
   Input: `array = [12, 24, 36]`
   Output: `12`
   Explanation: The GCD of 12, 24, and 36 is 12.

9. **Finding the Sum of the First N Odd Numbers**
   Difficulty: Easy
   Topics: Mathematical Computations
   Description: Write a program to calculate the sum of the first N odd numbers.

Example:
Input: `N = 5`
Output: `25`
Explanation: Sum of the first 5 odd numbers (1 + 3 + 5 + 7 + 9) is 25.

10. **Finding the Number of Perfect Numbers Up to a Given Limit**
    Difficulty: Medium
    Topics: Number Theory
    Description: Write a program to find how many perfect numbers exist up to a given limit.
    Example:
    Input: `limit = 30`
    Output: `1`
    Explanation: There is only one perfect number (6) up to 30.

11. **Finding the Largest Prime Factor of a Number**
    Difficulty: Medium
    Topics: Number Theory
    Description: Write a program to find the largest prime factor of a given number.
    Example:
    Input: `number = 28`
    Output: `7`
    Explanation: The prime factors of 28 are 2 and 7, with the largest being 7.

12. **Generating a Matrix of Fibonacci Numbers**
    Difficulty: Medium
    Topics: Arrays, Matrix Operations
    Description: Write a program to generate a matrix where each element is a Fibonacci number.
    Example:
    Input: `size = 3`
    Output:

```
1 1 2
3 5 8
13 21 34
```

13. **Finding the Sum of the First N Prime Numbers**
    Difficulty: Medium
    Topics: Prime Numbers, Mathematical Computations
    Description: Write a program to calculate the sum of the first N prime numbers.
    Example:
    Input: `N = 4`
    Output: `17`
    Explanation: The sum of the first 4 prime numbers (2 + 3 + 5 + 7) is 17.

14. **Checking for a Balanced Bracket Sequence**
    Difficulty: Medium
    Topics: String Manipulation, Stack

Description: Write a program to check if a given string with brackets is balanced.
Example:
Input: `string = "{[()]}"`
Output: `True`
Explanation: The brackets are balanced.

15. **Finding the Sum of Numbers in a String**
    Difficulty: Easy
    Topics: String Manipulation
    Description: Write a program to extract and sum all numbers present in a given string.
    Example:
    Input: `string = "The numbers are 12 and 34"`
    Output: `46`
    Explanation: The sum of numbers 12 and 34 is 46.

16. **Finding the Longest Consecutive Sequence in an Array**
    Difficulty: Medium
    Topics: Arrays, Hashing
    Description: Write a program to find the longest sequence of consecutive numbers in an array.
    Example:
    Input: `array = [100, 4, 200, 1, 3, 2]`
    Output: `4`
    Explanation: The longest consecutive sequence is [1, 2, 3, 4].

17. **Generating a Matrix with a Spiral Pattern**
    Difficulty: Medium
    Topics: Arrays, Matrix Operations
    Description: Write a program to generate a matrix filled with numbers in a spiral pattern.
    Example:
    Input: `size = 3`
    Output:

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

18. **Finding All Subsets of a Set**
    Difficulty: Medium
    Topics: Combinatorics
    Description: Write a program to generate all possible subsets of a given set of numbers.
    Example:
    Input: `set = [1, 2]`
    Output: `[[], [1], [2], [1, 2]]`
    Explanation: The subsets of [1, 2] are the empty set, [1], [2], and [1, 2].

19. **Checking for Perfect Squares in a Range**
    Difficulty: Easy

Topics: Mathematical Comput

ations
Description: Write a program to check which numbers in a given range are perfect squares.
Example:
Input: `start = 1`, `end = 10`
Output: `[1, 4, 9]`
Explanation: Perfect squares between 1 and 10 are 1, 4, and 9.

20. **Finding the Sum of Diagonal Elements in a Matrix**
    Difficulty: Easy
    Topics: Matrix Operations
    Description: Write a program to calculate the sum of the diagonal elements in a square matrix.
    Example:
    Input: `matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`
    Output: `15`
    Explanation: The sum of the diagonal elements (1 + 5 + 9) is 15.

21. **Finding the Second Smallest Number in an Array**
    Difficulty: Easy
    Topics: Arrays
    Description: Write a program to find the second smallest number in an array.
    Example:
    Input: `array = [12, 13, 1, 10, 34, 1]`
    Output: `10`
    Explanation: The second smallest number in the array is 10.

22. **Generating Pascal's Triangle Up to N Rows**
    Difficulty: Medium
    Topics: Combinatorics
    Description: Write a program to generate Pascal's Triangle up to N rows.
    Example:
    Input: `N = 3`
    Output:

```
1
1 1
1 2 1
```

23. **Finding the Sum of Digits of the Product of Two Numbers**
    Difficulty: Easy
    Topics: Mathematical Computations
    Description: Write a program to find the sum of the digits of the product of two given numbers.
    Example:
    Input: `number1 = 12`, `number2 = 34`
    Output: `9`
    Explanation: The product of 12 and 34 is 408, and the sum of its digits is 4 + 0 + 8 = 12.

24. **Checking for Palindromic Numbers in a Range**
    Difficulty: Medium
    Topics: Mathematical Computations
    Description: Write a program to check for palindromic numbers within a given range.
    Example:
    Input: `start = 1`, `end = 100`
    Output: `[1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99]`
    Explanation: Palindromic numbers between 1 and 100 are the numbers listed.

25. **Generating a Matrix with Alternating 0s and 1s**
    Difficulty: Easy
    Topics: Arrays, Matrix Operations
    Description: Write a program to generate a matrix where the elements alternate between 0 and 1.
    Example:
    Input: `size = 3`
    Output:

```
0 1 0
1 0 1
0 1 0
```

26. **Finding the Count of a Specific Word in a String**
    Difficulty: Easy
    Topics: String Manipulation
    Description: Write a program to count how many times a specific word appears in a given string.
    Example:
    Input: `string = "hello world hello"`
    Output: `2`
    Explanation: The word "hello" appears 2 times in the string.

27. **Finding the Largest Sum of a Subarray**
    Difficulty: Medium
    Topics: Arrays, Dynamic Programming
    Description: Write a program to find the largest sum of any contiguous subarray.
    Example:
    Input: `array = [-2, 1, -3, 4, -1, 2, 1, -5, 4]`
    Output: `6`
    Explanation: The largest sum is 6, from the subarray `[4, -1, 2, 1]`.

28. **Generating a Right-Angle Triangle Pattern of Numbers**
    Difficulty: Easy
    Topics: Patterns
    Description: Write a program to create a right-angle triangle pattern with numbers.
    Example:
    Input: `height = 4`
    Output:

```
1
12
123
1234
```

29. **Finding All Divisors of the Product of Two Numbers**
    Difficulty: Medium
    Topics: Number Theory
    Description: Write a program to find all divisors of the product of two given numbers.
    Example:
    Input: `number1 = 6`, `number2 = 10`
    Output: `[1, 2, 3, 5, 6, 10, 15, 30]`
    Explanation: The product of 6 and 10 is 60, and its divisors are listed.

30. **Finding the Longest Sequence of Consecutive 1s in a Binary Array**
    Difficulty: Medium
    Topics: Arrays, Binary Operations
    Description: Write a program to find the longest sequence of consecutive 1s in a binary array.
    Example:
    Input: `array = [1, 1, 0, 1, 1, 1]`
    Output: `3`
    Explanation: The longest sequence of 1s is `[1, 1, 1]` with length 3.

31. **Calculating the Sum of the First N Fibonacci Numbers**
    Difficulty: Medium
    Topics: Fibonacci Sequence, Mathematical Computations
    Description: Write a program to calculate the sum of the first N Fibonacci numbers.
    Example:
    Input: `N = 5`
    Output: `12`
    Explanation: The first 5 Fibonacci numbers are 1, 1, 2, 3, 5, and their sum is 12.

32. **Checking for a Repeated Substring in a String**
    Difficulty: Medium
    Topics: String Manipulation
    Description: Write a program to check if a substring is repeated within a given string.
    Example:
    Input: `string = "abab"`
    Output: `True`
    Explanation: The substring "ab" is repeated.

33. **Finding the Median of a List of Numbers**
    Difficulty: Medium
    Topics: Sorting, Mathematical Computations
    Description: Write a program to find the median value of a list of numbers.
    Example:
    Input: `list = [3, 1, 4, 1, 5]`

Output: `3`
Explanation: After sorting the list to [1, 1, 3, 4, 5], the median is 3.

34. **Finding the Number of Words in a String**
Difficulty: Easy
Topics: String Manipulation
Description: Write a program to count the number of words in a given string.
Example:
Input: `string = "Hello world"`
Output: `2`
Explanation: There are 2 words in the string.

35. **Generating a Matrix with a Diagonal Pattern**
Difficulty: Medium
Topics: Matrix Operations
Description: Write a program to create a matrix where elements form diagonal lines of a given pattern.
Example:
Input: `size = 4`
Output:

```
1 0 0 0
1 1 0 0
1 1 1 0
1 1 1 1
```

36. **Finding the Sum of the First N Even Numbers**
Difficulty: Easy
Topics: Mathematical Computations
Description: Write a program to calculate the sum of the first N even numbers.
Example:
Input: `N = 4`
Output: `20`
Explanation: The first 4 even numbers are 2, 4, 6, 8, and their sum is 20.

37. **Finding the Count of Digits Greater Than a Specific Value**
Difficulty: Easy
Topics: Mathematical Computations
Description: Write a program to count how many digits in a number are greater than a specific value.
Example:
Input: `number = 54321`, `value = 3`
Output: `2`
Explanation: The digits

greater than 3 in 54321 are 5, 4, and 4, so the count is 2.

38. **Generating a Pattern of Prime Numbers**
Difficulty: Medium

Topics: Prime Numbers, Patterns
Description: Write a program to generate a pattern where each row contains the first few prime numbers.
Example:
Input: `rows = 3`
Output:

```
2
2 3
2 3 5
```

39. **Finding the Common Elements in Two Arrays**
Difficulty: Medium
Topics: Arrays
Description: Write a program to find common elements between two arrays.
Example:
Input: `array1 = [1, 2, 3, 4]`, `array2 = [3, 4, 5, 6]`
Output: `[3, 4]`
Explanation: The common elements between the two arrays are 3 and 4.

40. **Finding the Sum of the Squares of All Even Numbers Up to N**
Difficulty: Medium
Topics: Mathematical Computations
Description: Write a program to calculate the sum of squares of all even numbers up to a given N.
Example:
Input: `N = 4`
Output: `20`
Explanation: The even numbers up to 4 are 2 and 4, and their squares are 4 and 16. The sum is 20.

41. **Generating a Pattern of Increasing Numbers**
Difficulty: Easy
Topics: Patterns
Description: Write a program to create a pattern where numbers increase with each row.
Example:
Input: `rows = 3`
Output:

```
1
12
123
```

42. **Finding the Largest Element in Each Row of a Matrix**
Difficulty: Easy
Topics: Matrix Operations
Description: Write a program to find the largest element in each row of a matrix.

Example:
Input: `matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`
Output: `[3, 6, 9]`
Explanation: The largest elements in each row are 3, 6, and 9.

43. **Checking for Anagram Pairs in a List of Strings**
Difficulty: Medium
Topics: String Manipulation
Description: Write a program to find pairs of strings in a list that are anagrams of each other.
Example:
Input: `strings = ["listen", "silent", "hello", "world"]`
Output: `[("listen", "silent")]`
Explanation: "listen" and "silent" are anagrams.

44. **Finding the Frequency of Each Character in a String**
Difficulty: Easy
Topics: String Manipulation
Description: Write a program to count the frequency of each character in a given string.
Example:
Input: `string = "hello"`
Output: `{'h': 1, 'e': 1, 'l': 2, 'o': 1}`
Explanation: The frequency of each character in the string "hello" is shown.

45. **Generating a Matrix with Random Numbers**
Difficulty: Easy
Topics: Random Number Generation, Matrix Operations
Description: Write a program to generate a matrix filled with random numbers.
Example:
Input: `rows = 2`, `columns = 3`
Output:

```
3 8 1
7 4 6
```

46. **Finding the Length of the Longest Word in a String**
Difficulty: Easy
Topics: String Manipulation
Description: Write a program to find the length of the longest word in a given string.
Example:
Input: `string = "Find the longest word"`
Output: `8`
Explanation: The longest word is "longest" with length 8.

47. **Finding All Triplets in an Array That Sum to Zero**
Difficulty: Medium
Topics: Arrays, Sorting
Description: Write a program to find all unique triplets in an array that sum to zero.

Example:
Input: `array = [-1, 0, 1, 2, -1, -4]`
Output: `[[-1, -1, 2], [-1, 0, 1]]`
Explanation: The unique triplets that sum to zero are listed.

48. **Generating a Square Matrix with Random Values**
Difficulty: Easy
Topics: Random Number Generation, Matrix Operations
Description: Write a program to generate a square matrix where each element is a random value.
Example:
Input: `size = 3`
Output:

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

49. **Finding the Difference Between the Sum of Even and Odd Numbers in an Array**
Difficulty: Easy
Topics: Arrays, Mathematical Computations
Description: Write a program to calculate the difference between the sum of even and odd numbers in an array.
Example:
Input: `array = [1, 2, 3, 4, 5, 6]`
Output: `4`
Explanation: The sum of even numbers is 12, and the sum of odd numbers is 8. The difference is 4.

50. **Generating a Triangle Pattern of Stars with a Given Height**
Difficulty: Easy
Topics: Patterns
Description: Write a program to create a triangle pattern of stars with a specified height.
Example:
Input: `height = 4`
Output:

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