1. Given two integer numbers return their product only if the product is equal to or lower than 1000, else return their sum.

2. Write a program to iterate the first 10 numbers and in each iteration, print the sum of the current and previous number.

3. Write a program to accept a string from the user and display characters that are present at an even index number. For example, str = "pynative" so you should display ‘p’, ‘n’, ‘t’, ‘v’.

4. Write a program to remove characters from a string starting from zero up to n and return a new string.

   For example

   ```
       remove_chars("pynative", 4) so output must be tive. Here we need to
   remove first four characters from a string.

       remove_chars("pynative", 2) so output must be native. Here we need to
   remove first two characters from a string.
   ```

   Note: n must be less than the length of the string.

5. Write a function to return True if the first and last number of a given list is same. If numbers are different then return False.

6. Iterate the given list of numbers and print only those numbers which are divisible by 5

7. Print the following pattern, Print Pattern using for loop.

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

8. Write a program to check if the given number is a palindrome number. A palindrome number is a number that is same after reverse. For example 545, is the palindrome numbers.

9. Given a two list of numbers, write a program to create a new list such that the new list should contain odd numbers from the first list and even numbers from the second list.

10. Write a Program to extract each digit from an integer in the reverse order. For example, If the given int is 7536, the output shall be “6 3 5 7“, with a space separating the digits.

11.
    ```
    Hello, World!: Write a program that prints "Hello, World!" to the console.
    ```

12. Even or Odd: Write a program that takes an integer as input and prints whether it is even or odd.

13. Sum of Numbers: Write a program that asks the user for two numbers and prints their sum.

14. Factorial: Write a program that calculates the factorial of a given number. The factorial of a non-negative integer n is the product of all positive integers less than or equal to n.

15. Fibonacci Sequence: Write a program that prints the Fibonacci sequence up to a given number of terms. The Fibonacci sequence starts with 0 and 1, and each subsequent term is the sum of the two previous terms.

16. Prime Numbers: Write a program that checks whether a given number is prime or not. A prime number is a number greater than 1 that has no divisors other than 1 and itself.

17. Palindrome Checker: Write a program that checks whether a given string is a palindrome. A palindrome is a word, phrase, number, or other sequence of characters that reads the same forward and backward.

18. Rock, Paper, Scissors: Write a program that allows the user to play Rock, Paper, Scissors against the computer. The program should randomly generate the computer's choice and determine the winner based on the rules of the game.

19. Number Guessing Game: Write a program that generates a random number between 1 and 100 and asks the user to guess the number. The program should provide feedback on whether the guess is too high or too low until the correct number is guessed.

20. Reverse a String: Write a program that takes a string as input and prints the reverse of the string.

21. Word Count: Write a program that reads a text file and counts the number of occurrences of each word in the file. The program should then print the word and its count in descending order of frequency.

22. Anagram Checker: Write a program that takes two strings as input and checks whether they are anagrams of each other. Anagrams are words or phrases formed by rearranging the letters of another word or phrase.

23. Matrix Multiplication: Write a program that multiplies two matrices of arbitrary size. The program should take two matrices as input and output the resulting matrix.

24. Password Generator: Write a program that generates a random password for the user. The password should be a combination of uppercase and lowercase letters, numbers, and special characters. The user should be able to specify the length of the password.

25. Hangman Game: Write a program that allows the user to play the game of Hangman. The program should randomly select a word from a predefined list and prompt the user to guess letters until they either solve the word or run out of guesses.

26. File Encryption/Decryption: Write a program that encrypts or decrypts a file using a simple encryption algorithm. The program should take a file and a secret key as input and output the encrypted or decrypted file.

27. Web Scraper: Write a program that extracts data from a website of your choice. The program should scrape the web page, extract relevant information, and save it to a file or display it to the user.

28. Sudoku Solver: Write a program that solves a Sudoku puzzle. The program should take a partially filled Sudoku grid as input and output the solution.

29. Image Processing: Write a program that performs basic image processing operations, such as resizing, cropping, and applying filters, on an image file.

30. Tic-Tac-Toe AI: Write a program that allows the user to play Tic-Tac-Toe against an AI opponent. The AI should use a strategy to make intelligent moves and try to win the game.

31. 
    ```
    Word Frequency Counter: Write a program that takes a sentence as input and
    counts the frequency of each word in the sentence. Store the word
    frequencies in a dictionary and print the result.
    ```

32. Unique Elements: Write a program that takes an array of integers as input and returns a new array containing only the unique elements from the original array. The order of the elements in the new array should be the same as in the original array.

33. Dictionary Manipulation: Write a program that creates a dictionary representing a person's contact information, including their name, email address, and phone number. Allow the user to add, update, or delete entries in the dictionary.

34. Array Sorting: Write a program that takes an array of numbers as input and sorts the array in ascending order. You can use any sorting algorithm of your choice (e.g., bubble sort, insertion sort, or selection sort).

35. Word Anagrams: Write a program that takes a list of words as input and groups together words that are anagrams of each other. The program should return a dictionary where the keys are sorted strings of the characters in each word, and the values are lists of words that are anagrams.