1. **Scenario:** You are developing a banking application that categorizes transactions based on the amount entered.
   Write logic to determine whether the amount is positive, negative, or zero.

   1. Get the user input for the amount
   2. Create the elif condition to check the amount is positive, negative, zero
   3. If amount>0 its positive
   4. Elif amount=0 its zero, else the amount is negative

2. **Scenario:** A digital locker requires users to enter a numerical passcode. As part of a security feature, the system checks the sum of the digits of the passcode.
   Write logic to compute the sum of the digits of a given number.

   1. Create the input for the passcode
   2. Convert the int to string
   3. Initialize the variable to zero
   4. For each digit to add the variable
   5. Print the sum of digit

3. **Scenario:** A mobile payment app uses a simple checksum validation where reversing a transaction ID helps detect fraud.
   Write logic to take a number and return its reverse.

   1. .Create the function for reverse_number and get the input
   2. Convert number in to string and reverse it.
   3. int(str(n)[::-1])
   4. Convert it back to number
   5. Print the reverse_number

4. **Scenario:** In a secure login system, certain features are enabled only for users with prime-numbered user IDs.
   Write logic to check if a given number is prime.

   1. Get the user ID from input.
   2. If the user ID is less than or equal to 1, it is not prime.

    3. Start checking divisibility from 2 up to the square root of the user ID.
    4. If the user ID is divisible by any of these numbers, it is not prime.
    5. If no divisors are found, the user ID is prime.
    6. Enable the secure features only if the user ID is prime.

5. **Scenario:** A scientist is working on permutations and needs to calculate the factorial of numbers frequently.
   Write logic to find the factorial of a given number using recursion.
    1. Get the number input from the user.
    2. If the number is 0 or 1, factorial = 1.
    3. Otherwise, multiply the number by the factorial of (number - 1).
    4. Repeat calculation until reaching 1.
    5. Return the final multiplied result as the factorial of the given number.

6. **Scenario:** A unique lottery system assigns ticket numbers where only Armstrong numbers win the jackpot.
   Write logic to check whether a given number is an Armstrong number.
    1. Create the input field for the ticket number
    2. Convert number to string to count digits
    3. armstrong_sum = sum(int(digit) ** num_digits for digit in num_str)
    4. Create the if condition for check the number is armstrong number or else not armstrong number

7. **Scenario:** A password manager needs to strengthen weak passwords by swapping the first and last characters of user-generated passwords.
   Write logic to perform this operation on a given string.
    1. Create the input field for get the user input password

2. If the password length is greater than 1 swap first and last letter

3. strengthened = password[-1] + password[1:-1] + password[0]

8. **Scenario:** A low-level networking application requires decimal numbers to be converted into binary format before transmission.
   Write logic to convert a given decimal number into its binary equivalent.

   1. Create the input field for get the decimal number
   2. Convert the decimal number into binary and slice from index 2
   3. binary = bin(num)[2:]
   4. print the binary number

9. **Scenario:** A text-processing tool helps summarize articles by identifying the most significant words.
   Write logic to find the longest word in a sentence.

   1. Create the input field for get the sentence
   2. Use split() for break the sentence to word separated by spaces
   3. Find the longest_word using the len()
   4. Print the longest_word

10. **Scenario:** A plagiarism detection tool compares words from different documents and checks if they are anagrams (same characters but different order).
    Write logic to check whether two given strings are anagrams.

    1. Get the user input for two string and store it in two variables

    2. call anagram check function and sort()

    3. Now compare the two words, if both are equal it is anagrams

    4. else it is not anagrams