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.

   Logic: Collect the input amount.

   if the amount entered is greater than zero, positive, if it is less than zero, negative else zero.

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.
   Logic
   Collect the passcode
   Convert the input to string.
   List the string value and convert them into integers.
   Summing up the individual values.

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.
   Logic:
   Collect the transaction ID.
   Convert it to strings.
   Use the strig reversal function.
   Convert the result into integer.

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.
   Logic:
   Collect the number.
   If the given number is greater than one, Use the for loop to find out the division reminder value. If it is zero, not the prime and exit. Else, prime number.

If the given number is less than or equal to one, not a prime number.

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.
   Logic:
   Define a function for the factorial[calculate_factorial].
   Collect the amount.
   If the value is zero or one, return 1.
   If it is negative, error.
   If it is greater than one, n *calculate_factorial(n-1).

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.
   Logic:
   Define a function for Armstrong number.
   Collect the input value and store it to a variable.
   Convert it to string and identify individual numbers and convert them to integer. Using loops, the total arrives and matches the input value. Based on the result, identify.

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.
   Logic:
   Collect the password.
   Take the first letter, last letter and middle words and store them with three different variables.
   Concatenate them with last letter, middle words and first letter.

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.

Logic:

Collect the number
Divide it with two.
Record the reminder.
Take the quotient for next division.
Continue the process till the quotient becomes zero.
Collect the reminders.
Read the reminder in the reverse order.

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.

   Logic:

   Collect the sentences.
   Select the words and count them one by one.
   Count the first word and store the word and the counted value. If the subsequent word count is greater replace the stored values.
   Like that we can identify 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.

    Logic:

    Define a function.

    Collect the words and remove the spaces, commas etc.

    Find the length. If both are not same, it is not anagram.

    If matches, sort the letters and find the match.

    If it is matching, anagram else no.