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.

- Get the amount from the user
- If the amount > 0 then it's a positive trans
- Else if the amount < 0 then it's a negative trans
- Else return zero transaction

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.

- Get the passcode from the user as a string
- Init SUM=0
- Use a for loop to iterate through each number and add it to SUM
- Iterate until the all the numbers are summed up

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.

   a. Get the number from the user as a string value
   b. Use print(num[::-1]

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.

   a. Get the number from the user
   b. Using if condition, check if the number divisible by any other number other that 1 and its own
   c. If yes, return prime

d. Else return not-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.

   a. Get the number for which the factorial has to be applied
   b. If the number is 0, then return 1
   c. Else find the factorial by using the formula n*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.

   a. **Get the input from the user**
   b. **Count the digits of the number and save as p**
   c. For every digit of the number, use power and save in diff variables
   d. Sum up all the variable
   e. If sum=input number then its an amstrong number
   f. Else its not an amstromg 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.

   a. **Get the password from the use and save it in the variable str**
   b. **Assign last char to variable l**
   c. **Assign first char to variable f**
   d. x=l
   e. l=f
   f. f=x

g. now print the new string

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.

   a. **Get the number from the user**
   b. **Divide the number by 2 and save the reminder either 0 or 1**
   c. **Repeat the process until the quotient is 0**
   d. **Finally write the reminders we got 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.

   a. **Get the input sentence from the user**
   b. **Convert the sentence into a list of all those words**
   c. **Find the length of all the items in the list**
   d. Intialise a variable to store the length of 1st word of the list
   e. Iterate thru the loop and if the length of the subquent word is greater than the previous one then udpate the variable with the current word
   f. Print the final 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.

    - Input str1 from user
    - Input str2 from user
    - Convert the strings to lower case

- sort the string in the alphabetical order
- compare both strings.
- If equal print anagram
- Else print not an anagram