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.

   - Enter the amount
   - if the number greater than 0 print(positive)
   - elif if the number less than 0 print (nagative)
   - else the number zero print(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.

   - To Enter the number
   - then use for sum =0
   - then use for statement
   - print sum of digits

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.

   - Enter the transation id number
   - the number convert to the string number
   - then using reverse id `[::-1]`
   - `print(reversed id)`

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.

   - Enter the number
   - then if is prime (num)
   - print the is prime number
   - else print is not the 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.

   - To Read the number
   - if the number 1 or 0
   - ruturn 1 else ruturn 1 multiple to factorial number of(n-1)
   - then print result

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.

   - Read the input number
   - then check if the number is an armstrong number or not
   - then initialize sum
   - find the sum of cube of each digit
   - to display the result if num==sum
   - print is an armstrong number
   - else pring is 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.

   - To read the input string
   - then use if statement
   - Swap the first and last characters using slicing
   - For strings of length 1 or less, no swap needed
   - else print(res)

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.

   - To read the decimal number
   - res empty string

- while greater than 0
- then divied 2 the reminnder stroe the binary digits
- then print (res)

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.

   - Read the input sentence
   - then use split the sentence word
   - Initialize an empty string to store the longest word
   - Iterate through each word
   - then use for loop
   - Update 'res' if the current word is longer than 'res'
   - then pring (res)

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.

    - Read the two input string
    - then using if statement
    - sorted(s1)—sorted(s2)
    - then print (yes)
    - else then print (no)