

- **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 details from the user
  - Check the entered amount is positive or negative or zero
  - Print the result
- **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 a user enter numerical passcode as a list
  - Iterate the list one by one as integer
  - Add the digits while iterating
  - Then print sum of total

**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. Read the input number from the user
  2. Convert into string
  3. Reverse the string
  4. Convert back to number
  5. Print the reverse number
- **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. Read the number
    2. If the number less than 2. Print not a prime number
    3. If the number is divisible by 2 and square of any other number. Print not a prime number
    4. If the number divisible by same number and not other number. Print prime number

- **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. Read the input number
  2. If the number is 0 or 1. Print one
  3. If the number is greater than 1. Multiply all numbers upto  $n-1$ .
  4. Print the result
- **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. Read the input from user
  2. Check the count of given number.
  3. Split the given number and square the number with count
  4. Add those number and check whether given number and added number is equal or not
  5. If it equal. Print given number is Armstrong
  6. Else it is not Armstrong
- **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. Read the input
  2. If the string less than 2. Print as it is
  3. Swap the first and last character of the string. Without modifying middle character
  4. Print the modified string.
- **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. Read the input
2. Assign empty string for binary representation
3. Check the number  $> 0$
4. Divide the input number by 2 and save the remainder in binary string
  - a. Update the input number and repeat the process
5. Reverse the binary string and print the result

- **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. Read the input sentences
2. Assign variable to store longest word
3. Split the sentences as individual words
4. Loop the each words.
5. If the current word is longer the stored word then update longest word
6. Print longest word

- **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. Read two input string
2. Remove space and convert the both string to lowercase
3. Sort the character of both string and compare
4. If two string are identical. Print Anagrams
5. Else not a Anagrams