

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:

- a. Get the Number from User
 - b. Validate the Number is Positive/Negative/Zero
 - c. Do the 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.

Logic:

- a. Allow the user to input numerical passcode. Check the input and validate. If non-numeric then 'Invalid Passcode'
 - b. Sum the digit of a passcode
 - c. If sum of passcode is correct then open the digital locker
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:

- a. Get the user mobile app. for the reversing Transaction ID
- b. Validate thru OTP. If OTP validated then return the reverse to it's account#

- c. Ensure the return is reversed to account#
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:

- Allow the user to enter User ID
 - Validate the user ID has Prime-numbered or not
 - Show the message to user as, "User is Prime Numbered User ID and enable the features.
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:

- Get the user Input as Number
 - Validate the number is factorial is >1 or not using
 - Set constant value $x=1, n= \text{<user Input>}$, using for logic
For i in range(1, $n+1$):
 $X=x*i$
 - Print factorial of a 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.

Logic:

- a. Get user input as number
- b. Find the length of the number
- c. Using for loop, find each number is power of that length
- d. Multiple each power of a number and follow the same for digits
- e. Sum up the values of all each powered digit were it's multiplied
- f. If sum of the value must be equivalent to that user input no. called Armstrong number else not

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:

- a. Find the length of the password
- b. Using for loop, find the first and last character of the password
- c. Find first character as array of 0 and last character length -1. Swap the position
- d. Assign first character to last character and vice versa.

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:

- a. Get the decimal numbers from the user as input
- b. Convert decimal numbers to binary numbers
- c. For e.x Decimal Number is 25. Converted binary is 0b

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:

- a. Get the sentence from the user
- b. Find each words and count the word length in that sentence using for loop
- c. Compare the word and word length and hold the object in array list.
- d. Whichever has higher length then return it in that sentence

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:

- a. Get the user input as two Strings
- b. Remove the space, convert to lowercase and store long words
- c. Sort the char,Read each char in a word and compare with the variable. If words are equal print valid anagram else not