

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 input from the user.
- * Use the if-elif condition to find whether the amount is positive, negative or zero.
- * if amount is greater than 0, print it is positive
- * elif amount is less than 0, print it is negative
- * else amount is equal to 0, print it is 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.

- * Get the user input for passcode
- * Convert the number into a string
- * Each digit of the number add it to the sum variable
- * Print the sum.

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.

- * Get the user input
- * Convert the number into a string and reverse the string using `[::-1]`
- * Convert it to number and print the number.

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.

- * Get the input from the user
- * And check whether the number is prime or not.
- * If the number is only divisible by 1 and itself.
- * If yes, print it is prime. else print it is not prime.
- * If prime features are enabled, else not enable.

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.

- *Get the number to calculate factorial.
- * Call factorial function and pass the number as an argument
- * If the number is less than or equal to 0, it returns 1 to avoid infinite loop.
- *If not multiply with that number and recursive to factorial function with minus 1 of that 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.

- * Get the user input for a ticket number.
- * Find the number of digits in that number.
- * Take power to each digit of a ticket number with the number of digits in it and sum all the values.
- * If the sum is equal to the original number, print "Armstrong Number".
- * Else, print "Not an 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.

- * Get the user input for passwords.
- * Use swap function to swap first and last characters of that password and store it in a different variable.
- *Print the modified 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.

- *Get the user input for a decimal number.

- * Use bin() to convert it into binary or use loop to divide the number by 2 and then its quotient till we get 0

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.

- * Get the user input for a sentence

- * Use split () to split the sentence into an individual words

- * Create variable to store a value

- * Use len () to find the length of the word and compare it.

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.

- * Get the user input for two strings

- *Sort the characters of both string

- * Then compare two words, if both are equal, it is an anagram.

- * Else, it is not an anagram.