

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:

- i. Read the given input amount
 - ii. If the amount is greater than 0, print "Positive (Deposit)"
 - iii. If the amount is less than 0, print "Negative (Withdrawal)"
 - iv. Else, print "Zero (No 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:

- i. Read the given input.
 - ii. Initially, consider a sum variable is 0.
 - iii. Using while loop and extracting the last digit of the number is obtained by using the modulus 10.
 - iv. The digit is added to another variable each time the loop is executed.
 - v. This loop terminates when the value of the number is 0.
 - vi. Print the total sum of the number
-

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:

- i. Read the given input
 - ii. Convert the number into string
 - iii. Reverse the string
 - iv. Convert the reverse string into number
 - v. Print the reverse 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.

Logic:

- i. Read the given number
- ii. Using for loop and if statement, if the given number is less than or equal to 1 and divisible by any number except 1.
- iii. Then print "It is not a Prime Number"
- iv. Else, print "It is a 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.

Logic:

- i. Read the given input number
 - ii. If factorial number is 0 or 1, return 1
 - iii. Else, Using for loop and the function calls with multiply the number and number-1.
 - iv. The function calls with decremented number until number is equal to 1 or 0.
 - v. Print the factorial 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.

Logic:

- i. Read the given input number.
 - ii. Find the number of digits in the given number.
 - iii. Initialize a variable sum to zero.
 - iv. Extract each digit from the given number and raise it to the power of the number of digits and add it to the sum.
 - v. If the sum is equal to the given number, then print "It is an Armstrong number"
 - vi. Else, print "It is 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.

Logic:

- i. Read the given input string
- ii. If the string has less than 2 characters then print original string itself.
- iii. Else, concatenated the last character, middle part, first character
- iv. 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.

Logic:

- i. Read the given decimal number
 - ii. Initialize the empty string for binary equivalent
 - iii. Using while loop, decimal number is divided by 2 successively and store the remainder
 - iv. Reverse the remainder order
 - v. Add the remainder reverse order to binary equivalent
 - vi. Print the binary equivalent.
-

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:

- i. Read the given sentence
 - ii. Split the sentence into words using string split
 - iii. Using for loop to iterate through the words and check the longest word by comparing their length.
 - iv. Return the word with most characters and store as longest word
 - v. Print the longest 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.

Logic:

- i. Read the given two string
- ii. Check the length of each string are equal
- iii. Counts the frequency of each character in both strings.
- iv. If the character counts match for all characters, then print "Anagrams"
- v. Else, print "Not an Anagram"