

**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.

- Read the input number.
- If the number is greater than 0 print Positive.
- If the number is less than 0 print Negative.
- Else 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.

Read the input number.

- Convert the number into individual digits.
- Initialize a variable as sum is 0.
- Each number added to the sum variable.
- Print the 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.

- Read the number.
- Convert the number into a string.
- Reverse the string.
- Convert the string into a number.
- 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.

- Read the given number.
- Check if the given number is divisible by 2, print "Not Prime".
- If the number not found any divisors, then print "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.

- Read the input number.
- Set the factorial number as 1, add +1 in every loop check.
- Iterate till the given value of fact =fact\*current value.
- After the iteration complete till given number, print the 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 given number.
- Sum the number of given digits and raised the power of each number of its own given digits.
- If the number equals to the given number, then print, "It is Armstrong Number".
- If the number not equals to the given number print, "It 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.

- Read the given string.
- If the string length is less than 2, print the string as it is.
- If the given string length is greater than 2, swap the first and last character and keep the middle string as it is.
- 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.

- Read the input decimal number.
- Represent an empty binary string.
- While the number is greater than 0.
- Divide the decimal number by 2 to get a remainder.
- Binary is the sum of remainder and binary string.
- Reverse the binary string order to get 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.

- Input the sentence from the user.
- Initialize the longest word as a variable.
- Split the words by using, .split().
- Iterate each word store current word check and update as a longest word.
- 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.

- Input two strings from the user.
- Makes each character to lower case and remove spaces.
- Sort the characters of the strings.
- If the characters are matches print “Anagram”.
- Else, print, “Not an anagram”.