

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.

1. Read input from user.
2. Check if the input is $>$ than 0 then it is positive.
3. Else if input is less than 0 then it is negative.
4. Else input is 0 then 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.

1. Read input from user.
2. Convert the number into individual digit.
3. Initialize sum = 0.
4. Using for loop we can add digit. Repeat until it add all digit.

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.

1. Read the input number.
2. Keep the number in a list.
3. Reverse the number in a list.
4. Print the reversed 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.

1. Read the input number.
2. If the number is less than 2, print not prime.
3. Loop from 2 to the square root of the number.
4. If the number is divisible by any of these number, print not prime.
5. If no divisors are found, print prime.

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.

1. Read the input.
2. Check if $n=0$ or 1 , return 1 .
3. Else find factorial using formula $(n (n-1))$.
4. Print the output.

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.

1. Read the input.
2. Count the digit of given input.
3. Findout the sum of power of each digit.
4. If sum of power is equal to given input, then the num is Armstrong.
5. Else not an Armstrong num.

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.

1. Read the input.
2. Check the length of string greater than 1.
3. Put the string into list.
4. Now swap the index of first and last character.
5. Print the result.

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.

1. Read the input, Check the number is ≥ 2 .
2. Divide the input by 2. note the remainder.
3. After that divide the Quotient by 2. note the remainder.
4. Continue until Quotient becomes 0.
5. Read the remainder from bottom to top.

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.

1. Read the input.
2. Split the sentence into words.
3. Find the maximum length word, store in temp variable.
4. Using loop, Check the current word is longer than stored variable.
5. If, yes Print the word otherwise check next 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.

- 1. Read two input string & convert it into lowercase or uppercase.*
- 2. Sort the 2 input strings (asc or dec).*
- 3. Now compare the two sorted input string.*
- 4. If both are same then print anagrams.*
- 5. Else not a anagrams.*

H O P E L A R N I N G