

Scenario:1

A user is required to enter a valid number in a form, but users sometimes input invalid data. Write logic to repeatedly prompt the user until they enter a valid integer.

Logic-1:

1. Fix the valid number.
2. Get the user Entered number.
3. If user “entered number” is not match to “valid number” means, it will give prompt until user has to enter valid number.

Scenario: 2

A data analysis tool processes a list of numbers and needs to identify the most frequently occurring value. Write logic to find the most frequently occurring number in a given list.

Logic-2:

1. Import the `Counter` class from the `collections` module.
2. Get a list of numbers as input.
3. Check whether the list contains any values.
4. If the list is empty, return 'None'.
5. Use the `Counter` class to count how many times each number appears.
6. Return the most frequently occurring number along with its count.

Scenario: 3

A text-processing application needs to compare words and check if they are anagrams (contain the same letters in a different order). Write logic to determine whether two given strings are anagrams.

Logic-3:

1. Get the two input strings.
2. Check whether all letters of one string occur in the other string (order of letters does not matter).
3. Split the strings into individual characters and compare each letter one by one.
4. If all letters occur in both strings with the same frequency, they are anagrams..

Scenario: 4

A speech analysis program needs to count the number of vowel sounds in a given input. Write logic to count the number of vowels in a given string.

Logic-4:

1. get input string from user.
2. check whether the given string has vowels or not .
3. if yes, count as 1 like wise, untill the end

Scenario: 5

A text-editing software includes a feature to reverse the order of words in a sentence for stylistic effects. Write logic to reverse the order of words in a sentence while keeping the words themselves intact.

Logic-5:

1. Get the input sentence.
2. Split the sentence into words using spaces.
3. Reverse the list of words.
4. Join the reversed list back into a string with spaces.
5. Return or print the result.

Scenario: 6

A missing number is detected in a sequence of values stored in a database. Write logic to find the missing number in a list containing n-1 numbers from 1 to n.

Logic-6:

1. Get the list of numbers (some missing)
2. Define the full set from 1 to n.
3. Convert your input list into a set.
4. Subtract the input set from the full set → this gives the missing numbers.

Scenario: 7

An ATM machine processes withdrawal requests and needs to ensure that users cannot withdraw more than their account balance. Write logic to allow a withdrawal only if the balance is sufficient.

Logic-7

1. Get the withdrawel amount from the user.
2. fix the balance_amount.

3. if `withdrawal_amount <= balance_amount`:

 subtract the withdrawal amount from balance amount and return updated balance amount

4. otherwise; return insufficient balance. your balance amount is 'return updated balance amount'.

8.Scenario:

A system needs to verify whether a given dataset contains duplicate entries. Write logic to check whether a given list contains duplicate values.

Logic-8:

1. Get the input list.
2. Convert the list to a set (since sets don't allow duplicates).
3. Compare the length of the list and the set:
4. If lengths are equal → No duplicates.
5. If lengths are not equal → Duplicates exist.

9.Scenario:

A digital calculator includes a feature to sum the digits of a number for verification purposes. Write logic to calculate the sum of all digits in a given integer.

Logic-9:

1. Get the value of "n" from user.
2. Using for loop fix the ranges and get each number and add with total (initially total is zero).
3. Return total (sum of digits)

10. Scenario:

A language-learning app wants to verify whether a given sentence is a pangram (contains every letter of the alphabet at least once).

Write logic to check if a given sentence is a pangram.

Logic-10:

1. get the all_letter in lowercase from string class.
2. get the sentence from user, convert into lower case, and filter the unique value.
3. check whether given sentence is subset of all_letters or not.
 - if yes, means return 'its pangram'
 - otherwise, 'it's not a pangram'