

1. **Scenario:** 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.
2. **Scenario:** 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.
3. **Scenario:** 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.
4. **Scenario:** 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.
5. **Scenario:** 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.
6. **Scenario:** 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.
7. **Scenario:** 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.
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.
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.

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.

## My answers (krithiksha):

### 1. Scenario:

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

- i. Get the user input
- ii. Check the user input is valid integer using `type()` function.
- iii. Put the condition in while loop, in order to check repeatedly
- iv. If condition pass, take the user input
- v. Otherwise, ask the user to give a valid integer input

### 2. Scenario

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.

- i. Get the list
- ii. Initialize the empty dictionary
- iii. Loop through the given list, and add the number as dictionary key and its frequent occurring count as its value
- iv. If the number repeats again the particular number key's value will be incremented by 1
- v. Now calculate the max value from the dictionary and return the number as the most frequently occurring number in a given list.

### 3. scenario:

1. 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.

- i. Get the 2 strings
- ii. Convert the 2 strings into lower case
- iii. Sort the 2 string
- iv. Compare the 2 strings, if it is equal then 'anagram'
- v. Otherwise , ' not a anagram'

### 4. scenario:

1. 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.

- i. Get the string
- ii. Initialize a vowel\_str variable to store vowels ["aeiou"]
- iii. Initialize a count =0 variable
- iv. Convert the string in lower case
- v. Loop through the string
- vi. Check if the character in string is present in vowel\_str
- vii. If yes, increment count by 1
- viii. After the loop execution return the count variable

### 5. scenario:

1. 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.

- i. Get the sentence
- ii. Separate the sentence using .split(' ') -> it becomes list of words in the sentence
- iii. Now reverse the list using .reverse()
- iv. merge the words in the list and make it as a string by using .join()
- v. return the sentence which is reversed by it's order of words

## 6. scenario:

1. 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.

- i. Get the list and n
- ii. Initialize 'missedNum = []'
- iii. Loop through the range of n
- iv. Check number in n with each item in list
- v. If condition fails, store the number (missed number in list) in the 'missedNum'
- vi. After the loop execution, return the missedNum list which contains missing number in a list containing n-1 numbers from 1 to n.

## 7. scenario:

1. 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.

- i. Get the withdrawal amount from user
- ii. Check if the withdrawal\_amount is lesser than the account\_balance of the user
- iii. If condition pass, withdraw the amount and send to user with current account balance details
- iv. Else: send message to user ' your balance is insufficient to withdraw the amount '

## 8. scenario:

1. A system needs to verify whether a given dataset contains duplicate entries.

Write logic to check whether a given list contains duplicate values.

- i. Get the list
- ii. Initialize list\_without\_duplicates=[]
- iii. Convert the list into set and convert as list and store in list\_without\_duplicates
- iv. Check list is equal to list\_without\_duplicates, if yes, list has no duplicate values
- v. Else: list has duplicate values

## 9. Scenario

1. 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.

- i. Get the integer
- ii. Use sum() function to sum the all digits of given integer
- iii. Return the sum

## 10. Scenario

1. 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.

- i. Get the sentence
- ii. Initialize the variable 'alpha = "abcdefghijklmnopqrstuvwxyz"
- iii. Initialize alphaCount variable
- iv. Convert the sentence into lower case, strip(), eliminate duplicates
- v. Loop through the each character of sentence
- vi. Check if the character is present in 'alpha'
- vii. If yes, increment alphaCount by 1
- viii. If alphaCount == 26 then sentence is pangram else not a pangram