

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.

- a. Get the input data
- b. If the input is not a number then throw an error to the user.
- c. Example – type conversion method -> integer to float is possible. Trying float(input value) if true or no error then we take that the input is a number.

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.

- a. Identify the list
- b. Identify the max count of the number. `Max(numberslist, numberslist.count)`

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.

- a. Input the strings [2 inputs]
- b. Using the sorted function, compare the 2 strings
- c. If String 1 and 2 are equal, then return true else return false.

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.

- a. Input the string through input function
- b. Have the vowels defined in a variable
- c. Using for loop, loop the text input
- d. Compare the text input and the vowels [for i in vowels]
- e. Count += 1
- f. Print the count value.

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.

- a. Input the string
- b. Reverse the string using negative indexing
- c. `r = 'abc'`
- d. `r[::-1]`

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.

- a. `MissingNumber = 0`
- b. `List = [1,2,3,4,5,7,8,9,10]`
- c. For i in list

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.

- a. Check the account balance
- b. Declare the withdrawal amount
- c. If Withdrawal amount > account balance -> Error message ' Cannot be withdrawn higher than balance
- d. Else Withdraw the amount and show balance
 - i. `New_balance = account balance – withdrawal 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.

- a. Identify the list
- b. Convert the list into a set
- c. `New_unique_list = set(list)`

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.

- a. Input the number
- b. Assign the initial total to zero
- c. Using type conversion, convert the number into string and once again change it into integer
 - i. `sum(Int(digit) For digit in number)`
- d. Assign this value to the total

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.

- a. Input the string
- b. Declare variable with all the alphabets – 'a....z'
- c. Iterate the input string using for loop
- d. If the I value is not present in the input sentence, then it is not a pangram, else pangram
- e. For I in alphabets:
 - i. If I not in input Sentence
 - ii. Then not a pangram, else pangram