

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 number from the user
 - ii) Add the variable = `float(input("Enter the valid number:"))`
 - iii) Now the user can input the valid integer = `55.5`

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 inputs from the user
 - ii) Variable add the numbers
 - iii) Get the variable as Counts
 - iv) Check on the `most_frequently_occurring_number`
 - v) `print("Most frequently occurring number in a given list:"{most_frequently_occurring_number})`

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.
 - i) Input the string from the user
 - ii) Define variable and add the string
 - iii) Using if condition compare the values
 - iv) `if a==b`
 - v) `print(anagram) or not anagram`

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.
 - i) Define the string from the user
 - ii) add the sting into the variable
 - iii) input the vowels (I.E) a,e,i,o,u
 - iv) using print option ("Number of vowels:",variable)

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.

- i) Input the string from the user
- ii) Add the word
- iii) Add the variable as reversed
- iv) Input the string[::-1]
- v) print the variable

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.

- i) Get the inputs from the user
- ii) Define the variable and add the number in that
- iii) Missing = Define the missing_number(Variable)
- iv) Print(Variable,missing)

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.

- i) We can write the Control structures using if, elif and else.
- ii) Define the account balance
- iii) if the account balance withdrawing is more then the condition then it shouldn't allow

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

- i) Get the inputs from the user
- ii) List enter the numbers
- iii) Add the variable as Duplicates = [item in list, count in list if count>1]
- iv) print(duplicates)

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.

- i) Get the numbers from the user
- ii) Input the number in the variables
- iii) Add float(input("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.

i) **Alphabet=input()**