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.

   Ans:

- Get the number from user.
- Check if the number is valid or invalid.
- If the number is valid, proceed with next procedure.
- If the number is invalid, throw error and ask for the valid input.
- This procedure repeats until the user provides valid one.

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.

   Ans:

- Create a list of number.
- Create a dictionary to store the frequency of the repeated number.
- Check if any number is repeating in the list and update the count.
- Identify the maximum repeated number from dictionary.
- Print the most frequently occurred 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.

   Ans:

- Get string 1 and string 2.
- Sort the string 1& 2
- Compare the string 1 and string 2.
- If both has same letter, print as anagram.
- Else print 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.

   Ans:

- Get the word from user.
- Create a set of vowels character with lowercase and uppercase.
- Initialize count=0
- Create a loop through each character in the word.
- If the character is equal to the vowels character, then increase count+=1.
- Once the loop ends, print the number of vowels in the word.

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.

   Ans:

   - Get the sentence.
   - Split the sentence into the words as list.
   - Use slicing method to reverse the list.
   - Then join the words back into sentence from list.
   - Then print it.

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.

   Ans:

   - Get the list of number.
   - Calculate the expected sum from 1 to n using the formula n*(n+1)//2.
   - And then the actual sum using the list of number.
   - To find missing number, subtract the actual sum from expected sum.
   - Then display the missing number.

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.

   Ans:

   - Read the account balance.
   - Get the amount withdrawal.
   - Check if the withdrawal amount is less than or equal to the account balance, allows user to proceed to withdraw and update the balance.
   - Otherwise print insufficient balance.

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.

Ans:

- Get the list of value.
- Initialize an empty set to store the values.
- Create a loop through the value, if the value is already in the set, duplicate exist.

- Otherwise, add the value to the set
- If the duplicate found is true, print duplicates are found. Otherwise, print no duplicates found.

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.

   Ans:

- Read the input number
- Convert them into the individual digits.
- Initialize sum =0
- Create a loop through the digits and add the digits with sum.
- Once loop ends, print the sum of the 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.

    Ans:

- Read the sentence.
- Make the sentence to lowercase.
- Initialize the empty set to store the letters.
- Create a loop through each character in the sentence.
- If the set contains between a to z, then add it to the set.
- After loop ends, If the set contains all 26 letters from a to z, print the sentence is pangram.
- Otherwise print the sentence is not pangram.