1. **Scenario:** A program needs to determine the largest prime factor of a given number for mathematical computations.
   Write logic to find the largest prime factor of a given number.

Ans:

- Get the numbers.

- Initialize the variable as largest to handle the factors.

-find the smallest prime factor as2 and keep dividing.

- Create a loop through each odd number, divisible by that number until it has no longer divisible.

- And store that number as largest. After loop ends, print the largest factor.

2. **Scenario:** A system needs to convert a Roman numeral string into its integer equivalent for numerical processing.
   Write logic to convert a given Roman numeral string into an integer.

Ans:

- Get the string.
- Read the string and map the roman symbol to values.
- Start from the left side of the given string.
- If the current value is >or = next value, add it.
- If the current value is < next value, subtract it.
- Continue until the end, and the total value needs to be print.

3. **Scenario:** A text-processing application analyzes a list of words to find the longest common prefix.
   Write logic to find the longest common prefix among a list of strings.

Ans:

- Get the list of string.
- Take the first string as a initial prefix.
- Compare that with the next string.
- if the string doesn't start with prefix, shrink prefix.

- otherwise continue till the string match will become empty.

4. **Scenario:** A calculator program takes two numbers and an operator to perform a mathematical operation.
   Write logic to take two numbers and an operator (+, -, *, /) and perform the corresponding operation.

Ans:

- Get the two numbers.
- Ask user to enter the operator.
- If the operator is +, perform addition. Or -, perform subtract or *,perform multiplication or / perform division.
- Once if.. elif condition ends, print the result.

5. **Scenario:** A lottery system assigns special numbers where Armstrong numbers have a special significance.
   Write logic to check whether a given number is an Armstrong number.

Ans:

- Get the number from user.
- Count the total digits from the given number.
- Then we have split the numbers into digits.
- then calculate each digit power of total digits. And the sum of those digits will be remain same as given number.
- if yes, print given number is armstrong number.
- otherwise Not an armstrong number.

6. **Scenario:** A task management application allows users to add, remove, and view tasks in a to-do list.
   Write logic to implement a to-do list where users can add, remove, and view tasks.

Ans:

- Create the list to store the task.
- Use operator .append() to add the task to the list.
- Use .remove() to remove the task from the list.

- To view the list use print statement for display

7. **Scenario:** A weather application requires a function to convert temperature between Celsius and Fahrenheit.
   Write logic to convert a given temperature from Celsius to Fahrenheit and vice versa.

Ans:

- Get the conversion which user want.
- If celsius to fahrenheit, use formula as multiple the number by 9/5 then add 32.
- If fahrenheit to celsius, use formula as subtract the 32 from the multiple of the given number by 5/9.
- then print the result.

8. **Scenario:** A security system needs to verify whether a given string has all unique characters.
   Write logic to check whether all characters in a given string are unique.

Ans:

- Get a string.
- Initialise unique character
- Split the string into character.
- Compare the each character with unique character.
- If the its not there, its unique and add it.
- if the character is present, it's not a unique.
- if the unique as character in it, print the statement as All the character in the given string is uniqu

9. **Scenario:** A text analyzer tool identifies the longest word in a given sentence for readability analysis.
   Write logic to find the longest word in a given sentence.

Ans:

- Get the sentence.
- Split as words from sentence.
- Check the len(word) and make the first len(word) as initial one.

- Compare other len(words) with intial one, if the other len(words) is greater then the initial one.update the new len(word)
- Continue this process, finalize the last new len(word) as the longest word in the given sentence.

10. **Scenario:** A password manager generates secure passwords based on user-defined length and complexity.

Write logic to generate a random password of a given length, containing uppercase, lowercase, digits, and special characters.

Ans:

- Get the password length from user.
- Use the string module to get the uppercase, lowercase, digits, and special characters.
- Combine all character in one group.
- Use random() to pick the characters randomly.
- And then print the final password.