PYTHON DEVELOPER
TASK - 4

25. **Find Missing Number**
    Objective: Find the missing number in an array of size nnn, containing numbers 111 to n+1n+1n+1.
    Input: A list of integers.
    Output: The missing integer.
    Hint: Use the sum formula Sum(n)=n(n+1)2\text{Sum}(n) = \frac{n(n+1)}{2}Sum(n)=2n(n+1) and subtract the sum of given numbers.

---

26. **Check Balanced Parentheses**
    Objective: Check if a string of parentheses (( ), { }, [ ]) is balanced.
    Input: A string of parentheses.
    Output: `True` if balanced, otherwise `False`.
    Hint: Use a stack to match opening and closing parentheses.

---

27. **Longest Word in a Sentence**
    Objective: Find the longest word in a given sentence.
    Input: A string (sentence).
    Output: The longest word.
    Hint: Split the string into words and compare lengths.

---

28. **Count Words in a Sentence**
    Objective: Count the number of words in a sentence.
    Input: A string (sentence).
    Output: Integer representing the word count.
    Hint: Use the `split()` method to divide the string by spaces and count elements.

---

29. **Check Pythagorean Triplet**
    Objective: Determine if three numbers form a Pythagorean triplet (e.g., a2+b2=c2a^2 + b^2 = c^2a2+b2=c2).
    Input: Three integers aaa, bbb, ccc.

Output: `True` if they form a triplet, otherwise `False`.

Hint: Ensure ccc is the largest, then verify the equation.

---

30. **Bubble Sort**
    Objective: Implement the bubble sort algorithm to sort a list of integers.
    Input: A list of integers.
    Output: Sorted list in ascending order.
    Hint: Repeatedly swap adjacent elements if they are in the wrong order.

---

31. **Binary Search**
    Objective: Implement the binary search algorithm to find a target in a sorted array.
    Input: A sorted list of integers and a target integer.
    Output: The index of the target or `-1` if not found.
    Hint: Use a divide-and-conquer approach by checking the middle element.

---

32. **Find Subarray with Given Sum**
    Objective: Find a contiguous subarray whose sum equals a given value SSS.
    Input: A list of integers and an integer SSS.
    Output: The indices of the subarray or `-1` if no such subarray exists.
    Hint: Use the sliding window technique or prefix sums.

---

4. Log Analysis System

- **Description**: Write a program to parse large log files (e.g., web server logs) and extract insights like the most frequent IP addresses, response codes, or URLs accessed.
- **Challenges**:
  - Efficiently process large text files.
  - Handle various formats and error cases in logs.
  - Summarize and display data meaningfully.
- **Skills**: File I/O, data parsing, and data aggregation.

Empowering Tomorrow, Today with Main Flow

4. Log Analysis System

- **Restriction:** File size must be limited to 100 MB for log files.
- **Reason:** Log analysis is important for many real-world applications, but dealing with large datasets can become cumbersome. Limiting file size helps students focus on data parsing and insight extraction, allowing them to develop efficient file handling techniques without being overwhelmed. It also simulates real-world constraints like processing time and memory limits.
- **Learning Outcome:** Students will learn how to efficiently process large datasets, extract meaningful insights, and handle file-based data using basic algorithms and data structures like dictionaries and lists.

Deadline Compliance

- **Restriction**: **Submit the project within 7 days** from the start date.
- **Reason**: Meeting deadlines is crucial in the real-world software development environment. This restriction helps students practice **time management** and **task prioritization**. In professional settings, tight deadlines are often the norm, and learning to meet them without compromising quality is an essential skill.
- **Learning Outcome**: Students will learn to manage their time effectively, complete projects under pressure, and **deliver results on time**, which are all important skills in the workplace.