PYTHON DEVELOPER

TASK - 6

39. **Sudoku Validator**
    Objective: Validate whether a given Sudoku board configuration is valid.
    Input: A 9x9 2D list representing a Sudoku board.
    Output: `True` if valid, otherwise `False`.
    Hint: Check rows, columns, and 3×33 \times 33×3 grids for duplicates.

---

40. **Word Frequency in Text**
    Objective: Count the frequency of each word in a given text.
    Input: A string of text.
    Output: A dictionary where keys are words and values are their counts.
    Hint: Use `split()` to separate words and a dictionary to store counts.

---

41. **Knapsack Problem (0/1)**
    Objective: Solve the 0/1 knapsack problem using dynamic programming.
    Input: A list of weights, a list of values, and a maximum capacity.
    Output: The maximum value that can be carried within the capacity.
    Hint: Use a dynamic programming table to keep track of the maximum values at each capacity.

---

42. **Merge Intervals**
    Objective: Merge overlapping intervals in a list of intervals.
    Input: A list of intervals where each interval is represented as a pair of integers [start,end][start, end][start,end].
    Output: A list of merged intervals.
    Hint: Sort the intervals by start time and merge if the start of the current interval is less than or equal to the end of the previous one.

---

43. **Find the Median of Two Sorted Arrays**
    Objective: Find the median of two sorted arrays.
    Input: Two sorted lists.

Main Flow Services and Technologies Pvt. Ltd.
Contact Us. +91 9389641586, +91 97736 99074
Email-Add. contact.mainflow@gmail.com
www.mainflow.in

Output: The median value of the two lists.
Hint: Use binary search or merge the two arrays and find the median.

---

44. **Maximal Rectangle in Binary Matrix**
    Objective: Find the area of the largest rectangle in a binary matrix (matrix containing only 0's and 1's).
    Input: A 2D binary matrix.
    Output: The area of the largest rectangle formed by 1's.
    Hint: Use dynamic programming by treating each row as the base of a histogram and applying the largest rectangle in histogram technique.

---

45. **Largest Sum Contiguous Subarray (Kadane's Algorithm)**
    Objective: Find the largest sum of a contiguous subarray in an array of integers.
    Input: A list of integers.
    Output: The maximum sum of the subarray.
    Hint: Use Kadane's Algorithm which runs in linear time.

---

46. **Word Ladder Problem**
    Objective: Given two words, find the shortest transformation sequence from the start word to the end word, changing only one letter at a time.
    Input: Two words and a dictionary of words.
    Output: The length of the shortest transformation sequence.
    Hint: Use breadth-first search (BFS) and treat each word as a node in a graph.

---

6. Command-Line RPG Game

- **Description**: Design a role-playing game where players explore a text-based world, fight enemies, and collect items to progress.
- **Challenges**:
    - Create a dynamic map with different locations and events.
    - Implement a combat system with health, attack, and defense stats.
    - Save and load game progress.
- **Skills**: Object-oriented programming, file handling, and game mechanics.

6. Command-Line RPG Game

- **Restriction**: **Text-based interface only** (no graphical user interface).
- **Reason**: By limiting the project to a **command-line interface**, students are forced to focus on **game mechanics** like combat, item management, and world-building. This helps build **logical thinking** and **complex program structures** without the distraction of graphics. The project teaches how to design games that depend entirely on **logic and text-based feedback**.
- **Learning Outcome**: Students will learn to develop **interactive games**, utilize **object-oriented programming (OOP)** for characters and items, and implement game mechanics like **turn-based combat** and **inventory systems**.

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.