Textual RPG

Loops

1. **Calculate Total Player Experience**

   - **Task:** Write a function that calculates the total experience of a player by summing up the experience values of all their completed quests.
   - **Inputs:** An array of quest objects.
   - **Example:**
     - Input: `[{ name: "Quest 1", experience: 100 }, { name: "Quest 2", experience: 150 }, { name: "Quest 3", experience: 75 }]`
     - Output: `325`
   - **Tip:** Use a loop to iterate over the quests and accumulate the experience values.

2. **Find Quest by Name**

   - **Task:** Write a function that finds a quest by its name in the given array and returns the quest object.
   - **Inputs:** An array of quest objects, a quest name.
   - **Example:**
     - Input: `[{ name: "Quest 1", experience: 100 }, { name: "Quest 2", experience: 150 }, { name: "Quest 3", experience: 75 }]` , questName: "Quest 2"
     - Output: `{ name: "Quest 2", experience: 150 }`
   - **Tip:** Use a loop to iterate over the quests, check if the quest name matches the specified name, and return the matching quest object.

3. **Sort Quests by Experience**

   - **Task:** Write a function that sorts the given array of quest objects in ascending order based on their experience values.
   - **Inputs:** An array of quest objects.
   - **Example:**

        - Input: `[{ name: "Quest 1", experience: 100 }, { name: "Quest 2", experience: 150 }, { name: "Quest 3", experience: 75 }]`
        - Output: `[{ name: "Quest 3", experience: 75 }, { name: "Quest 1", experience: 100 }, { name: "Quest 2", experience: 150 }]`
    - **Tip:** Use a loop to implement a sorting algorithm, such as bubble sort or insertion sort, to rearrange the quests based on their experience values.

4. **Find Enemies with High Health**

    - **Task:** Write a function that finds all enemy objects with health greater than a specified threshold in the given array and returns a new array with the matching enemies.
    - **Inputs:** An array of enemy objects, a health threshold.
    - **Example:**
        - Input: `[{ name: "Enemy 1", health: 80 }, { name: "Enemy 2", health: 120 }, { name: "Enemy 3", health: 65 }]`, threshold: 100
        - Output: `[{ name: "Enemy 2", health: 120 }]`
    - **Tip:** Use a loop to iterate over the enemies, check if the enemy's health is greater than the threshold, and add matching enemies to the new array.

5. **Calculate Matrix Sum**

    - **Task:** Write a function that calculates the sum of all numbers in a given matrix.
    - **Inputs:** A matrix (2D array) of numbers.
    - **Example:**
        - Input: `[[1, 2, 3], [4, 5, 6], [7, 8, 9]]`
        - Output: `45`
    - **Tip:** Use nested loops to iterate over the matrix and accumulate the sum.

6. **Find Maximum Value in Matrix**

    - **Task:** Write a function that finds the maximum value in a given matrix and returns it.
    - **Inputs:** A matrix (2D array) of numbers.
    - **Example:**
        - Input: `[[3, 5, 2], [9, 1, 7], [4, 8, 6]]`
        - Output: `9`

- **Tip:** Use nested loops to iterate over the matrix and update the maximum value as you find larger numbers.

7. **Count Even Numbers in Matrix**

   - **Task:** Write a function that counts the number of even numbers in a given matrix and returns the count.
   - **Inputs:** A matrix (2D array) of numbers.
   - **Example:**
     - Input: `[[1, 2, 3], [4, 5, 6], [7, 8, 9]]`
     - Output: `4`
   - **Tip:** Use nested loops to iterate over the matrix and increment a counter for each even number.

8. **Find Quests with High Rewards**

   - **Task:** Write a function that finds all quest objects with a reward greater than a specified threshold in the given array and returns a new array with the matching quests.
   - **Inputs:** An array of quest objects, a reward threshold.
   - **Example:**
     - Input: `[{ name: "Quest 1", reward: 50 }, { name: "Quest 2", reward: 100 }, { name: "Quest 3", reward: 75 }]`, threshold: 80
     - Output: `[{ name: "Quest 2", reward: 100 }]`
   - **Tip:** Use a loop to iterate over the quests, check if the quest's reward is greater than the threshold, and add matching quests to the new array.

9. **Calculate Total Enemy Power**

   - **Task:** Write a function that calculates the total power of all enemies in the given array by summing up their power values.
   - **Inputs:** An array of enemy objects.
   - **Example:**
     - Input: `[{ name: "Enemy 1", power: 50 }, { name: "Enemy 2", power: 80 }, { name: "Enemy 3", power: 65 }]`
     - Output: `195`

    - **Tip:** Use a loop to iterate over the enemies and accumulate their power values.