Data Structures 1
First Semester, AY 2024-2025

Machine Problem #1M – Array and Array Operations

GAME Title: Pokémon Battle Royale: Power, Strategy, and Fatigue

Objective: Create a Pokémon battle simulation using arrays, where players select Pokémon for battle, and their health is adjusted based on battle outcomes and fatigue from repeated use.

Duration: 2 weeks including checking

Requirements:

1. **Use of Arrays**:
   Arrays must be used to store Pokémon names, health, power, poison and potions. You will manipulate these arrays throughout the exercise. Arrays must be used for storing, selecting, and deleting Pokémon.

2. **Player Pokémon Selection**:
   The end-user will select three or four Pokémon from the list to assign to Player 1 and Player 2. Each player's selections should be stored in a separate array.

3. **Remove Chosen Pokémon from the List**:
   After each player makes their selections, the chosen Pokémon should be deleted from the original list, ensuring that no Pokémon is used twice.

4. **Health Adjustments Based on Battle Results**:
   - **Health Decreases ("Bumababa")**: A Pokémon's health will decrease by 10 points if it loses a battle.
   - **Health Increases ("Tumataas")**: A Pokémon's health will increase by 5 points if it wins a battle.
   - **Fatigue Factor**: Every time a Pokémon is used in battle, regardless of whether it wins or loses, its health will decrease by 2 points due to fatigue.
   - **Health Adjustments**: Implement logic to adjust health based on wins, losses and fatigue.

5. **Battle**
   - **Hanap ng Kalaro**: Find a classmate to play as Player 2, while you will be Player 1.
   - **Multiple Battles**: You need to play **the game** against your classmate. After each battle, determine the winner based on the health of the Pokémon.
   - **Battle** continues until all selected Pokemon from both players have fought.
   - **Statistics**:
     At the end of the three rounds, output statistics:
     - How many battles Player 1 won?
     - How many battles Player 2 won?
     - Declare the overall winner based on the total number of wins.

6. **Submission**

- Program Checking during laboratory class
- Peer Evaluation coming from 2 – 3 classmates who played along with you.
- Upload a complete laboratory report.

Instructions:

1. **Create initial Pokémon list**.
   - Start by creating arrays for Pokémon names, health, power, poisons(damaging) and potions (healing).
   - Power (?), you can use the power variations that we have during prelims (same rulings applied on chance of winning)

2. **Player Pokémon Selection**
   - Player 1 will select 3-4 Pokémon from the list. Once a Pokémon is selected, it should be removed from the main list.
   - Player 2 will select 3-4 Pokémon from the remaining list.
   - Store each player's Pokémon in separate arrays (player1_pokemon[] and player2_pokemon[]).
   - Wanted to be fair: take turn to pick Pokémon
   - Added complexity: Who pick first? Bato-Bato pick?

3. **Simulate Battle:**
   - Randomly pair up each Pokémon from Player 1 and Player 2
   - Compare the power of each pair of Pokémon:
     - The Pokémon with higher power wins the battle
     - The winner's health increases by 5%
     - The loser's health decreases by 10%
     - Both Pokémon lose 2% points of health due to fatigue (napagod) in battle.
   - Add Complexity
     - you can use poisons and potions during the battle
     - You can add battle field where poisons and potions can be used (optional)

4. **Repeat for Three Rounds**
   - After each round, reset the game by selecting Pokémon again from the list (or reuse the previous selections).
   - Keep track of the number of wins for both Player 1 and Player 2 after each round.
   - Add complexity: Pokemon power decreases and increases based on poison and potion

5. **Output Statistics**
   - After three battles, display how many times Player 1 and player 2 won.
   - Declare the over all winner based on who won the most rounds
   -

Data Structures 1
First Semester, AY 2024-2025

Peer Evaluation Form for Pokémon Battle Program

Instructions:

Each student is required to evaluate their partner's performance based on the following criteria. Please provide honest and constructive feedback. This will help your classmate improve their programming skills and the overall experience of using the Pokémon battle program.

Name of Evaluator: _____
Name of Partner: _____
Date: _____

Evaluation Criteria

Criteria	Rating Scale (1-5)	Comments
Program Usability		
How easy and user-friendly was the Pokémon battle program?	1 - Difficult to use, 2 - Somewhat difficult, 3 - Average, 4 - Easy to use, 5 - Very user-friendly	
Array Operations and Accuracy		
Were Pokémon correctly selected, deleted, and stored in arrays as per the conditions?	1 - No array operations, 2 - Many errors, 3 - Some errors, 4 - Few errors, 5 - No errors	
Battle Logic Implementation		
Did the battle logic (power comparison, health adjustment, fatigue) work correctly?	1 - Incorrect battle logic, 2 - Many bugs, 3 - Some issues, 4 - Minor issues, 5 - Perfectly implemented	
Multiple Rounds Implementation		
Was the program able to simulate three rounds accurately?	1 - Did not work, 2 - Major issues, 3 - Some issues, 4 - Minor issues, 5 - Worked perfectly	
Statistics Output		
Were the correct statistics displayed after each round and at the end of the three rounds?	1 - Incorrect or missing statistics, 2 - Many issues, 3 - Some inaccuracies, 4 - Minor issues, 5 - Completely accurate	
Creativity/Originality		
Did the program demonstrate creative battle mechanics or unique features?	1 - No creativity, 2 - Minimal creativity, 3 - Some creativity, 4 - Quite creative, 5 - Very creative and unique	
Collaboration & Communication		
How well did your partner communicate and collaborate during the activity?	1 - No collaboration, 2 - Poor communication, 3 - Average collaboration, 4 - Good collaboration, 5 - Excellent teamwork	

Overall Rating: Out of 5: _____
General Feedback:
Please provide any additional feedback or suggestions for your partner:

Signature of Evaluator: _____

Rubric for Program Checking and Laboratory Report

1. Program Checking (Total: 50 points)

Criteria	Points	Description
Functionality	20	The program runs without errors, all features work as intended (0-20).
Code Structure and Organization	10	Code is well-organized, with proper indentation and logical flow (0-10).
Array Usage	10	Arrays are implemented correctly for storing and manipulating data (0-10).
Battle Logic Implementation	5	Correct application of battle mechanics and health adjustments (0-5).
User Interface and Experience	5	Program is user-friendly and easy to navigate (0-5).

Total for Program Checking: _____ / 50 points

---

2. Laboratory Report (Total: 50 points)

Criteria	Points	Description
Content and Completeness	20	All required sections are present (Algorithm,flowchart, output, peer evaluation.) (0-20).
Clarity and Organization	10	Report is logically structured and easy to read (0-10).
Analysis and Interpretation of Results	10	Includes detailed analysis of battle outcomes and statistics (0-10).
Formatting and Presentation	5	Report follows required formatting guidelines (fonts, spacing, etc.) (0-5).
Grammar and Spelling	5	Report is free of grammatical and spelling errors (0-5).

Total for Laboratory Report: _____ / 50 points

Overall Score: _____ / 100 points

Evaluation Criteria Descriptions:

- **Functionality:** Assess if the program operates without any bugs or errors and meets the specified requirements.
- **Code Structure and Organization:** Evaluate how well the code is structured, including proper use of functions and comments.
- **Array Usage:** Check if arrays are used correctly for storing and manipulating data.
- **Battle Logic Implementation:** Verify that the battle mechanics are accurately implemented, including health adjustments and fatigue effects.
- **User Interface and Experience:** Consider how user-friendly the program is, including prompts and instructions for the user.
- **Content and Completeness:** Ensure that all sections of the report are thoroughly covered and address the objectives of the exercise.
- **Clarity and Organization:** Look for logical flow and readability in the report's layout and writing style.
- **Analysis and Interpretation of Results:** Analyze how well the report interprets the outcomes of the battles and discusses statistics.
- **Formatting and Presentation:** Confirm adherence to specified formatting standards.
- **Grammar and Spelling:** Evaluate the overall writing quality for errors.