Scroll to automagically
open the letter

Farid Nakhle

I. Mission CodeQuest: The Lost Utilities

Welcome, brave coder!

You have been chosen as the Guardian of the TUJ Digital Realm, and your quest is to restore the lost powers of its ListUtilities: a legendary artifact that maintains order in codename CIS2168; the chaotic lands of Data Structures.

A bit more than a year ago, upon my arrival to TUJ, the evil Bug Lord has scattered the sacred list functions across many of the campus buildings: Main building, Students Studio, Sancha Building, and most recently, the Owl Center! To complicate things further, I heard some of them are being sent for hiding in the new Kyoto Campus…

Only YOU can bring them back... To complete your mission, you must implement the ancient ListUtilities methods, test their magic in battle (your main method), and prove your worth to the Grand Council of Coders.

Do you have what it takes to become a Master of Lists?

📜 Your Quest Instructions

1. Create the sacred artifact: a Java class named ListUtilities.
2. Reconstruct the lost spells: implement the legendary methods, listed below.
3. Test your magic: summon all functions from your main method and use any data input you see fit.
4. Forge your tools wisely: use ArrayLists to store and manipulate data.
5. Prove your mastery: **REFRAIN FROM PLAGIARISM** and ensure all functions work flawlessly before facing the Council.

II. A Memo of the Lost Spells of ListUtilities

1. The Scroll of Uniqueness

- Legend says: A true list is one where no two elements are the same.
- Your task: Create a spell (a function) that checks if all items in a list are distinct.
- Input: A list of numbers or words.
- Output: true if all elements are unique, false otherwise.
- Example:
    - Input: [1, 2, 3, 4, 5, 5]
    - Output: false

2. The Multiples Mirror

- Legend says: Only numbers divisible by a chosen value will appear in this enchanted mirror.
- Your task: Create a method that returns only elements that are multiples of a given number.
- Input: A list of numbers and an integer n.
- Output: A new list containing only elements that are multiples of n.
- Example:
    - Input: [1, 25, 2, 5, 30, 19, 57, 2, 25], 5
    - Output: [25, 5, 30, 25]

3. The Rune of Word Length

- Legend says: Only words of a specific length can be deciphered by the Rune of Word Length.
- Your task: Create a method that filters a list of words by length.
- Input: A list of words and an integer representing the desired length.
- Output: A new list containing only words of the specified length.
- Example:
    - Input: ["I", "like", "to", "eat", "eat", "apples", "and", "bananas"], 3
    - Output: ["eat", "eat", "and"]

4. **The Spell of Permutations**

- Legend says: Two lists are bonded by an unseen force if they contain the same elements, regardless of order.
- Your task: Create a spell that checks if two lists are permutations of each other.
- Input: Two lists of the same type (numbers or words).
- Output: true if the two lists contain the same elements, false otherwise.
- Example:
  - Input: [1, 2, 4] and [2, 1, 4]
  - Output: true

5. **The Whispering Words**

- Legend says: A single sentence can be broken into individual words by the Whispering Words incantation.
- Your task: Create a method that transforms a sentence into a list of words.
- Input: A sentence as a string.
- Output: A list of words, split by spaces.
- Example:
  - Input: "Hello, world!"
  - Output: ["Hello,", "world!"]
- Extra Challenge! If you can remove punctuation to make it ["Hello", "world"], you will earn bonus experience points!

6. **The Vanishing Curse**

- Legend says: With a single spell, all unwanted elements in a list can be erased forever.
- Your task: Create a method that removes all occurrences of a specific item from a list.
- Input: A list and a specific item.
- Output: The list is modified to remove all occurrences of the item.
- Example:
  - Input: [1, 4, 5, 6, 5, 5, 2], item 5
  - Output: [1, 4, 6, 2]

Your Training Grounds (Testing Your Spells)

- In your main function summon each spell and display their effects.
- The Grand Council will only accept magic that has been properly tested.

Victory Conditions 🎖️ (Grading)

- Each spell correctly implemented and tested (15 points each).
- Code is neat and properly structured (10 points).
- Bonus: Remove punctuation in stringToListOfWords() (5 extra points).

The Final Trial (Submission)

- Upload your **.Java** scrolls (ListUtilities.java and your main) to the Grand Archives (also boringly known as Canvas).