Data Structures Programming Problems

Problem 1: Floating-Point Number Management System (Using Array, LinkedList, or Stack)

A data analytics student wants to create a simple program that can manage and analyze a collection of **floating-point numbers** (for example: 12.5, 3.14, or -7.8). The student wants to test how different data structures handle adding, removing, and viewing data.

You are tasked to create a **menu-driven Java program** that allows the user to choose which data structure to use (Array, LinkedList, or Stack) and then perform different operations on floating-point numbers.

The program should allow the following operations:

1. Add a new floating-point number to the data structure.
2. Remove the most recently added number.
3. View the most recently added number without removing it.
4. Display all numbers currently stored in the data structure.
5. Check if the data structure is empty.
6. End the program.

Program Flow Example:

```
Select Data Structure:
1. Array
2. LinkedList
3. Stack
Enter choice: 3

--- Floating-Point Number Management Menu ---
1. Add Number
2. Remove Number
3. View Top Number
4. Display All Numbers
5. Check if Empty
6. Exit
```

Goal: This exercise helps students understand how data is stored, accessed, and managed in different data structures when working with decimal numbers.

Problem 2: Product Inventory Management System (Using Array, LinkedList, or Stack)

A small retail shop wants to create a program that can keep track of its **product names** using data structures. Each time a new product arrives, it should be added to the list. When a product is sold, it should be removed. The store manager should also be able to view which product was added most recently and see whether the inventory is empty.

You are tasked to create a **menu-driven Java program** that allows the user to choose a data structure (Array, LinkedList, or Stack) to manage product names.

The program should allow the following operations:

1. Add a new product name to the inventory.
2. Remove the most recently added product.
3. View the most recently added product without removing it.
4. Display all products currently stored in the inventory.
5. Check if the inventory is empty.
6. End the program.

Program Flow Example:

```
Select Data Structure:
1. Array
2. LinkedList
3. Stack
Enter choice: 1

--- Product Inventory Management Menu ---
1. Add Product
2. Remove Product
3. View Top Product
4. Display All Products
5. Check if Empty
6. Exit
```

Goal: This problem helps students understand how data structures handle storing and retrieving string data, which is commonly used in inventory and list management applications.