ASSIGNMENT 2C

Assignment 6 tests your knowledge of user-defined lists, stacks, queues, and priority queues (Chapter 24). You should start with the code from the textbook/presentation and adapt it to the assignment at hand and use the appropriate names and add the additional code and requirements bellow.

Part 1. Design the following 6 classes (with exact[1] names and functionality, replace **YourName** with your actual name):

Class	Description
YourNameList	Program the complete version of user-defined `MyList` interface from Chapter 24 (meaning you need to start with the code from the textbook/presentation and use the names and requirements listed here, and add the code for all methods that were not coded in the textbook/presentation and have a "`Left as an exercise`" comment). Add an additional **YourNameToString** method that returns a string with the elements from the list in the format: $Element_{Position0}$, $Element_{Position1}$, …, $Element_{PositionN-1}$.
YourNameArrayList	Program a complete version of the user-defined `MyArrayList` class from Chapter 24 (all fields and methods) that uses the `YourNameList` instead of `MyList` and add an additional **YourNameOutput** method that prints it in the format: ARRAYLIST=$Element_{Position0}$, $Element_{Position1}$, …, $Element_{PositionN-1}$.
YourNameLinkedList	Program a complete version of the user-defined `MyLinkedList` class from Chapter 24 (all fields and methods) that uses the `YourNameList` instead of `MyList` and add an additional **YourNameOutput** method that prints it in the format: LINKEDLIST=$Element_{Position0}$, $Element_{Position1}$, …, $Element_{PositionN-1}$.
YourNameStack	Program a complete version of the user-defined `GenericStack` class from Chapter 24 and 19 (all fields and methods) that uses the `YourNameArrayList` instead of the pre-defined `ArrayList` used in textbook/presentation and add an additional `YourNameOutput` method that prints it in the format: STACK=$Element_{Position0}$, $Element_{Position1}$, …, $Element_{PositionN-1}$.
YourNameQueue	Program a complete version of the user-defined `GenericQueue` class from Chapter 24 (all fields and methods) that uses a `YourNameLinkedList` instead of the pre-defined `LinkedList` used in textbook and add an additional `YourNameOutput` method that prints the list in the format: QUEUE=$Element_{Position0}$, $Element_{Position1}$, …, $Element_{PositionN-1}$.
YourName Assignment2C Driver class main	Create an array of 10 integers, read the values from the user and create instances of the `YourNameArrayList`, `YourNameLinkedList`, `YourNameStack`, and `YourNameQueue` user-defined lists and test/demonstrate their functionality by using all their methods (e.g. `add`, `remove`, `get`, `indexOf`, etc) and `YourNameToString` and `YourNameOutput` for all of them.

Part 2. Implement the 6 classes in NetBeans IDE and JAVA: Create a JAVA project called **YourNameAssignment2C** (same name as the driver class)., add the **YourNameList**, **YourNameArrayList**, **YourNameLinkedList**, **YourNameStack**, and **YourNameQueue** classes to the project (each class should have a separate JAVA file) and add the requested code to the 6 classes. Your program should be **user-friendly** (prompt the user for the input using a descriptive message) and be **well-documented/commented** (have comments for every line of code explaining what they do) and use only concepts learned in class so far.

Part 3. Create the screenshot document for your code and output: Create a Microsoft Word document from the **YourName-Assignment2C.docx** template and call it ***YourName-Assignment2C*.docx** (replace ***YourName*** with your actual name). Add to the document your **complete screenshots** of the NetBeans editor window showing the **complete JAVA source code** for all of the 6 JAVA classes in the appropriate sections and **complete screenshots** the **complete output**. If the entire class JAVA source code or the output does not fit in one screenshot or the screenshots cannot be easily read, create multiple screenshots and add them to the screenshot document. Please keep the screenshots in order (look at the line numbers). If your output is longer than a line and does not fit on one screen, Wrap Text in your output panel.

Part 4. Submit your work: Submit ***YourName-Assignment2C.docx*** on eCampus under the **Assignment 2C.** Do not archive the files (e.g. no RAR, ZIP, etc.) or submit other types/formats of files (e.g. no CLASS, JAVA, PDF, etc.

[1] Use the exact names (spelling, caps), parameters, returned values, functionality, and do not add or remove fields or methods. You will need to adapt the names to have this exact names and cases, to earn credit for the assignment. Do not use other ways to code the structures or corresponding pre-defined structures.