American University of Armenia, CSE
CS121 Data Structures A, C
Fall 2021

Homework Assignment 3

Due Date: Tuesday, October 12 by 23:59 electronically on Moodle

Please solve the programming tasks either in Java or C++, following good coding practices (details are on Moodle).

1. **(15 points)** Implement a variant of the `DoublyLinkedList` class that doesn't use header and trailer sentinels. Use it in a sample program to test the functionality and make sure it is correct. Discuss the comparative pros and cons of the implementations with the sentinels and without.

2. **(10 points)** Extend the `SinglyLinkedList` class with a `removeLast` method/function that removes and returns the last (tail) element of the sequence. Use it in a sample program. What is the running time of your method/function? Why is such a method/-function typically not included in the public interface of the SLL?

3. **(15 points)** Implement the Queue ADT using an array as the underlying data structure such that the front of the queue always corresponds to index 0. Test it in a sample program. Specify and discuss the execution times of each of the Queue ADT methods/functions in this implementation.

4. **(25 points)** The task is to simulate Q&A during office hours. A number of students arrive one after another for OH, each having one or more questions. Then, they ask their questions in a circular fashion: the first-arriving student asks the first question, after which this student moves to the end of the line. After asking a question, the second-arriving student also moves to the end of the line to wait for their turn to ask another question, and so on. If a student has no more questions, they leave the line instead of moving to the end. The OH stops when there are no more questions left.

   Using an appropriate **efficient** data structure to represent the line of the students, implement a program that simulates this process. The program input consists of a few lines. Each line includes the name and surname of one student, followed by the number of questions this student has when they arrive for OH. The output should specify the order of the questions in a format illustrated in the example below.

`sample input`	`sample output`

`Dennis Ritchie 3`
`Bjarne Stroustrup 1`
`James Gosling 2`
	`Dennis Ritchie asks question 1`
	`Bjarne Stroustrup asks question 1`
	`James Gosling asks question 1`
	`Dennis Ritchie asks question 2`
	`James Gosling asks question 2`
	`Dennis Ritchie asks question 3`

   Note that your program should include a `Student` class with a proper OOP structure and minimal required structure and functionality.

5. **(25 points)** Implement the Deque ADT using a *circular dynamic array*. Note that all operations should have $O(1)$ running time. Your `DynamicArrayDeque` class should implement the `Deque` interface given in the textbook. Write a program to test all the methods of your class.

6. **(10 points)** Write a generic method/function for reversing the contents of a stack without using any additional data structures. Specify the time and memory complexity of your method/function. Test it in a program.