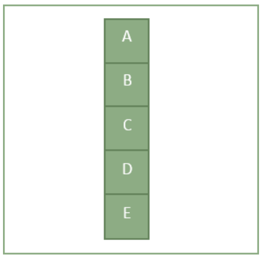
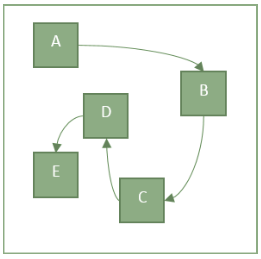


CSCI 501 / 701 - SP&P

Fall 2020 - Homework 2

ArrayList and LinkedList for C

Selim Temizer



Due date : Sunday, November 1st, 2020 (Submission through Moodle by 23:55 Astana Time)

During our lectures, we have seen how an ***Abstract Data Type*** (***ADT***) can be built with ***multiple implementations*** in C programming language (which does not have object-oriented features like function overloading and polymorphism). More specifically, we looked at how the ***Stack ADT*** can be implemented with ***arrays*** and ***linked lists***, and how developers can use both implementations simultaneously in their applications.

One of the most frequently used collections (container data structures) in Java programming language is the ***List ADT***, and the Java class library provides one array-based, and one (doubly) linked-list-based implementations of the List ADT, which are called ***ArrayList*** and ***LinkedList***, respectively. Java lists are very flexible and very practical in the sense that the user can choose the underlying implementation based on the needs of the application, and both implementations provide dynamically resizing storage for generic list elements. Unfortunately, the standard library of the C programming language does not contain dynamically resizing collections like the Java's ***List***, therefore C programmers need to implement their own dynamic data structures or use 3rd party libraries if they would like to work with such flexible containers.

In this homework assignment, we will build ***ArrayList*** and ***LinkedList*** implementations of the List ADT for the C programming language. The C-language version of our List ADT will serve ***exactly the same functionalities*** as the Java List ADT (with the exception of ***iterator*** support). In this respect, our C-language ***ArrayList*** and ***LinkedList*** implementations will be ***professional equivalents*** to their Java counterparts.

The homework bundle contains a directory named ***Code***. The C-language version of the List ADT is fully specified by the following 3 files in this directory: **List.h**, **ArrayList.h**, and **LinkedList.h**. There are just 2 files in this directory that need to be implemented by you, and those files are **ArrayList.c** and **LinkedList.c** (both files contain some useful code fragments in them as suggestions, but you are free to use or delete those code fragments). All other files are already fully implemented for you (including a sample user application). Note that you are free to use a ***singly***-linked-list or a ***doubly***-linked-list implementation for the ***LinkedList*** container.

There is another directory in the homework bundle that is named ***Tester***. This directory contains the **Unit Testing Framework** and a few **Test Suites** that I built for automatically grading your solutions. You can also test your implementations yourselves with the provided test framework, but this is an optional task that is not a required part of this homework assignment.

Bonus – Preparing Unit-Test Suites (5 points per test suite, up to 10 points)

As you can see in the ***Tester*** directory, test suites for ***creating*** lists (for both ***ArrayList*** and ***LinkedList*** implementations) and ***adding*** elements to lists (again for both ***ArrayList*** and ***LinkedList*** implementations) are already prepared. However, I need test suites for all other List

ADT functions for automatically grading your solutions. You may help me prepare the remaining test suites, which will improve your unit-test building skills and you may also earn some bonus points at the same time. (Note: In order to prepare high-quality tests, you need to carefully study all the files in the ***Tester*** directory.) I will keep a list of students who would like to build test suites as shown in **Figure 1**. If you want to earn bonus points, you can pick ***one*** or ***two*** functions from the list below and post your choice(s) on Piazza. Once I approve your requests (also on Piazza), you will be responsible for building test suites for that/those function(s) (for both ***ArrayList*** and ***LinkedList*** implementations). Requests will be granted on a ***first-come first-served*** basis on Piazza only.

Function	Tests		Bonus
	ArrayList	LinkedList	Student Name
create	X	X	
createFrom			
add	X	X	
addAll			
clear			
contains			
containsAll			
equals			
get			
indexOf			
isEmpty			
lastIndexOf			
remove			
removeElement			
removeAll			
replaceAll			
retainAll			
set			
size			
sort			
toArray			
print			
destroy			

Figure 1. List of functions that require test suites.

What to submit? (Use ***only ASCII characters*** when naming all of your files and folders)

I am only asking for the following files:

- The first file is the implemented **ArrayList.c** file.
- The second file is the implemented **LinkedList.c** file.
- If you would like to earn bonus by preparing test suites, you will need to submit 2 more files per each test suite. The details of this will be agreed upon and arranged through Piazza.

Submit all files through Moodle (there is no need to zip them, Moodle allows submission of multiple files).

Late submissions will **NOT** be accepted, therefore, start working on this homework assignment as early as possible, and try to have at least a working baseline system by the deadline. Good luck.