In-class objectives

Learn more about data hiding by implementing a simple container adapter and solving a programming problem with it.

Part 1

Start by discussing a container adapter (based either on the linked list from earlier in the week or a STL container from Fundamentals) for a circular, doubly linked list with these member functions/features:

- void insertFromFront (T &); // inserts an element of type T in the front
- void insertFromBack (T &); // inserts an element of type T at the back
- void deleteFront();
- void deleteBack();
- bool isEmpty();
- ~cdl_list();

Part II: Moving around this data structure

To be doubly, you should be able to move forward and backward. To be circular, the first element should consider the last element its preceding neighbor and the last element should point to the first element as its “next” neighbor.

Discuss adding the ability to move around your templated cdl_list by adding “iterator”-like functionality. What are the special cases you need to consider?

- void reset(); // returns to the front of the list
- void next(); // moves to the next element of list
- T currentValue(); // returns current value at top of the list
- void delete(); // removes an element from the list
- operator++(); // runs next

HINT: This exercise is much easier if you use a container that supports random access. Discuss why and/or how? Since we are encouraging data hiding in this exercise, any viable approach will receive full credit.
Part III: Putting it all together

In ancient times, a fair princess Buttercup had many suitors. Thinking her true love dead, she decides to use the following procedure to choose the knight she will marry. First, all suitors are lined up and given a number from 1 \ldots n. Then, because of a strange fascination with the number three, she would count to three and remove the third suitor considered in that round from contention. This process would repeat, returning to the front of the list once all suitors were considered.

For example here is the results of the process with n = 6:

123456 (3, you’re not the one for me)
12456 (6 … sorry)
1245 (4, no more)
125 (2, no I don’t choose you)
15 (5, at least you survive)
1

Buttercup’s true love Wesley was not killed but rather became a protégé of the Dread Coder SJE. As part of the Dread Coder’s crew, you have been asked to help Wesley determine which position in line he should stand.

Specifically:

1. How could you use the circular doubly linked list, and specifically Part II, to perform the simulation and display what spot Wesley should stand in for any arbitrary \( n \) number of suitors. Discuss pseudocode if you have time

2. Write brief text below on the back on how you used data hiding for this exercise. Are there any special advantages?