

CSE 331 - Project 2

Linked Lists

TA: James Daly

dalyjame@msu.edu

Due Date: 11:59 pm Sep 26, 2014

1 Project Description

In this project, you will complete an implementation of a linked list. You are given skeleton code (i.e., `Main.cpp`) that performs reading inputs and printing outputs as well as a class declaration for a linked list (`LinkedList.h`) and a complete list node implementation (`Node.h`). Your job is to complete the implementation of the following methods in the `LinkedList` class:

1. **PushFront**: Inserts an item at the head of the list.
2. **PushBack**: Inserts an item at the tail of the list.
3. **PopFront**: Deletes the item at the head of the list.
4. **PopBack**: Deletes the item at the tail of the list.
5. **Filter**: Deletes all items that **do not** have the desired property keeping only ones with the property.
6. **ForEach**: Invokes a function on each element of the list.

Some other methods have been provided for you but may depend on these methods to function. You may add additional methods to the `LinkedList` class (such as for the insertion and deletion of arbitrary items) as desired. Note that the `LinkedList` class is templated; any other methods you add should also be templated. You may not modify the `Node` class in any way.

Filter and **ForEach** are examples of higher-order functions, taking a function as input and invoking it on the contents of the list. They add a lot of versatility without adding lots of extra methods and are common in functional programming. For example, it is not necessary to have both an **Evens()** and an **Odds()** method since **Filter** can be used to find either result. The main function and some of the provided methods use these to great extent. You can invoke the passed function in the same way as any other function: **isIn(x)** will return true if x has the desired property.

Each of the **Push** and **Pop** methods should run in $O(1)$ time. **Filter** and **ForEach** should run in $O(n)$ time.

Make no other changes to the files other than completing the assigned methods. You must provide your own code; you may not copy another's code. Remember to read the project guidelines before you start coding. Remember to update the size member variable when you insert or delete an item.

2 Input Test Cases

The program takes two inputs. The first input is a file containing integers line-by-line. The second is some integer k . Your program will read the integers from the file and store them inside of the linked list.

Your program will then output the list and the size of the list. It will then filter the list, keeping only the numbers that are divisible by k , and output the new list and size. Finally, it will filter a copy of the original list, keeping only the numbers that are not divisible by k , again outputting the results. The **main** method should already do this; you simply need to implement the relevant functionality in the **LinkedList** class.

Your task is to execute the program on each dataset and report how many items are divisible by 2, 3, and 5 according to the following format:

```
input_file div_2 div_3 div_5
```

Example:

```
n10 5 3 2
n20 10 6 4
n30 15 10 6
..
```

Where `div_2` is the number of elements divisible by 2, `div_3` is the number of elements divisible by 3, and `div_5` is the number of elements divisible by 5. Note: use the CSE black server as a running machine. Other test cases will be executed after the due date for grading.

3 Project Deliverables

The following files must be submitted via Handin no later than 11:59 pm Sep 26, 2014.

1. **LinkedList.h** - contains your implementation of the **LinkedList** class.
2. **project2.pdf** - file containing your answers to written questions

4 Written Questions

1. Report your execution results for each of the given inputs as detailed above.

2. Let A and B be linked lists as defined in this project. Provide pseudo-code for copying B to the end of A .
3. What is the running time of your answer for question 2?
4. Repeat question 2 except now you are allowed to destroy B in the process.
5. What is the running time of your answer for question 4?