Assignment 2

Posted on: 1/22/14

Due on: 2/3/14 at 11:55 pm (SAKAI)

In this assignment, you will implement several methods in the **chain class (as mentioned in the class lecture)**.
The files we will provide to you:

- **chainNode.h**
- **chain.h**
- **chain.cpp**
- **linearList.h**
- **main.cpp**
- **Myexception.h**
- **Myexception.cpp**
- **LinkedList_Elements.txt**
- **Makefile**

You are asked to implement **FOUR NEW** methods in the **chain.cpp**. The detail of these methods are given below:

Part I:

Read **25** numbers from the **.txt** file (**LinkedList_Elements.txt)** provided to you and store them in the chain. (**Note that the first element in the list is the first number of the file, the second element is the second number and so on**). You should write your code inside the function **readAndStoreFromFile** in **chain.cpp**. The function prototype is given below:

```
void chain :: readAndStoreFromFile(char* fileName)
```

Part II:

From **the linked list created above**, find and **erase** the entries which contain numbers that are **multiple of** a given integer. You should write your code inside the function **eraseModuloValue** in **chain.cpp**. The function prototype is given below:

```
void chain :: eraseModuloValue(int theInt)
```

You will use this method to remove entries which are **MULTIPLE OF 5**, i.e., you need to call the above method with **theInt = 5.**

Part III:

On **the linked list generated after the operation of Part II,** rearrange the entries such that **all the entries with odd numbers precede all the entries with even numbers.** You should maintain the original ordering within even/ odd numbers. You should write your code inside the function **oddAndEvenOrdering** in **chain.cpp**. The function prototype is given below:

```
void chain :: oddAndEvenOrdering()
```

Part IV:

Reverse the linked list generated after the operation of Part III. Write your code in the body of the method **reverse** in **chain.cpp.** The function prototype is given below:

```
void chain :: reverse()
```

Creating and Extracting a tar file:

Linux/ Macintosh

- To create a tar file: **tar cvf (tar file name) (file 1) (file 2) (file 3)...**
- To extract the contents of a tar file: **tar xvf (tar file name)**

Windows

- You can use **ALZip** to create .tar file.

Things to Remember before Submission:

1. Check for the submission deadline (**both date and time**) and make sure you submit your **.tar file** before the deadline.
2. **LATE SUBMISSIONS ARE NOT ALLOWED.**
3. You should test your code on **thunder machine. TA will run your code on thunder machine and if it fails to compile there, you will be penalized.**
4. TA will only do **make** and then run the executable file, **e.g. ./main**.
5. You should submit **ONLY** a **.tar** file through **SAKAI** consists of **all the .h, .cpp and makefile.** The name of the **.tar** file should contain your name and UFID.
6. The output of your submission should be **exactly like (not "almost like")** the snapshot in Figure 1.
7. You can write any helper/ auxiliary method needed for the implementation of the four methods you are responsible for.

Figure 1: Snapshot of required input/ output