Programming Assignment 4
Linked List
COSC 052

In this assignment you are requested to implement a parameterized (Template) Linked List. Your implementation should handle existing built in types and classes.

After implementing your Linked List class use it to store the names of all bitmap files in a provided directory. (Do not hardcode the directory in your source. The user will supply the directory as an argument on the command line when they run your program.)

Your implementation should support the following interfaces

- **LinkList**()
  - constructor
- **LinkList**(LinkList) // create a new link list from an existing link list
  - copy constructor – creates a new link list by making a copy of an existing link list
- void **push_front**(elem)
  - inserts an item at the front of the link list
- void **push_back**(elem)
  - inserts an item at the rear of the link list
- void **pop_front**()
  - deletes the item at the front of the link list. Item is not returned.
- void **pop_back**()
  - deletes the item at the back of the link list. Item is not returned.
- void **remove**(int i)
  - deletes the item at position ***i*** in the list. Item is not returned.
- void **clear**()
  - deletes all items in list and
- elem **get**(int i)
  - returns a copy of the item stored at position ***i*** in the link list. Does not remove item from list
- int find(elem)
  - returns the index of ***elem***, if found else returns -1.
- int **size**()
  - returns the number of elements stored in a list

Once you have developed your link list class, use it to hold the names of bitmap graphic files stored in a directory on your computer. Use _findfirst()/_findnext() for Windows or opendir()/readdir() for Unix based systems.

Once you have stored the names of the bitmap files in your Link list, create a link list of bitmap images using CImg's CImgList() object and display them on your screen.

MUST HAVE:

- Design Documentation
  - (UML diagram)
  - brief explanation of your design (methods, parameters types, utility functions, etc.)
- Test code exercising each of the above methods
  - show your link list works with integers
  - show it works with ***strings*** or ***char ****
- In destructor method
  - print out the value contained in each node as it is deleted
- Allow user to specify directory to search on command line

An example code using <int>:

```
. . .
LL<int> mylist;
       mylist.push_back(100);
       mylist.push_back(200);
       mylist.push_back(300);
       mylist.push_back(400);

       cout << "Contents of mylist..." << endl;
       for(int i = 1; i <= mylist.size(); i++)
              cout << "Content of mylist [" << i << "]: " << mylist.get(i) << endl;


       LL<int> mylist2(mylist);  // create a new list from an existing list

       cout << "\n\nUsing Copy Constructor mylist2(mylist)..." << endl;
       cout << "Contents of mylist2..." << endl;
       for(int i = 1; i <= mylist2.size(); i++)
              cout << "Content of mylist2 [" << i << "]: " << mylist2.get(i) << endl;

       cout << "Erasing item at index 2 in mylist.erase(2)" << endl;
       mylist.erase(2);

       cout << "\ncontents of myList after erasing item at index 2..." << endl;
              for(int i = 1; i <= mylist.size(); i++)
              cout << "Content of mylist [" << i << "]: " << mylist.get(i) << endl;
```

example output from code snippet:

```
Contents of mylist...
Content of mylist [1]: 100
Content of mylist [2]: 200
Content of mylist [3]: 300
Content of mylist [4]: 400


Using Copy Constructor mylist2(mylist)...
Contents of mylist2...
Content of mylist2 [1]: 100
Content of mylist2 [2]: 200
Content of mylist2 [3]: 300
Content of mylist2 [4]: 400
Erasing item at index 2 in mylist.erase(2)

contents of myList after erasing item at index 2...
Content of mylist [1]: 100
Content of mylist [2]: 300
Content of mylist [3]: 400


In Link List Destructor...
deleting Item 100 from link list
deleting Item 200 from link list
deleting Item 300 from link list
deleting Item 400 from link list

In Link List Destructor...
deleting Item 100 from link list
deleting Item 300 from link list
deleting Item 400 from link list
```