CS20 – Section 093 – Fall 2019
Programming Assignment 1 – Due Friday September 6 at 11:59PM

Linked list warm-up exercise (30 points)

Code functions that implement processing for linked lists. The linked lists in assignment 1 are used to hold data items of type string. The functions are:

void push_front (llh, string)
adds a node to the front of a list, designated by the header llh, containing the data string "string"

void push_back (llh, string)
adds a node to the back of the list

int list_length (llh);
returns the number of nodes in the list

string retrieve_front (llh)
retrieves the contents of the node at the front. Does not remove the node.
If the list is empty, throws an exception.

string retrieve_back (llh)
retrieves the contents of the node at the back. Does not remove the node.
If the list is empty, throws an exception.

void display_nodes (llh)
displays the data items in each node, producing a report that resembles the following:

```
node 0 data -> aaaaa bbbbb
node 1 data -> this is the string from node 1
node 2 data -> ccccc 33333
```

If the list is empty, display a message – do not throw an exception.

A user (aka main) defines and initializes linked list processing by doing two things:

(1) creates a struct representing a node of a linked list, with the following format:

```
struct LLnode
{
	LLnode * fwdPtr;
	string theData;
};
```

(2) defines a linked list header for each linked list, for example:

```
	LLnode * theLLHeader = nullptr;
```

In this case, the name of the header is theLLHeader, but it could be anything.

Use the main provided for your testing.