Lab 8: C++ STL Lists

Points: 10

Description:

In this lab you will learn to use the Standard Template Library (STL) List Class. What is a list? A list is a linear series of nodes and each node can store some type of data. Your list will hold characters and you will use the list to store the reverse character sequence of text entered by the user without any spaces. If the user enters the text `"navy beans"` then the value stored in the list would be `"snaebyvan"` and each individual character would be a data element in the list. You will also need to display the contents of the list several times. Lastly you are going to remove some nodes/elements from the list and display the data elements(s) removed. The rules for what elements to remove are below:

- If the number of list elements is odd, remove the middle element
  `"hello" -> "helo"`
- If the number of list elements is even, remove the middle two elements
  `"helped" -> "heed"`

I placed comments with numbers to show you the area that you will have to modify. I also commented out for-loops and if-else structures so that the code would compile and run. You will need to uncomment and complete these areas to get everything working. Also, there will be at least one warning during compilation and you need to make sure there are no warnings when you submit the lab.

There are four example executions of the program below and you need to emulate the output as much as possible with what you turn in.

I have provided the stl_lab8.cxx code and a Makefile that works with the code file. You do not need to rename the code file or modify the Makefile when turning it in.

Strings:

I have been told that you might not know some things about the C++ String Object and there is at least one important aspect that you need to know for this lab. A String can be iterated over like an array using a for-loop. For instance:

```
string name = "Spongebob";

for(unsigned int i = 0; i < name.size(); ++i)
{
      cout << name[i] << endl;
}
```

Also, to get the size of the string you can use the `size()` function as used above in the for-loop. I used an unsigned integer in my for-loop, which is just a way to say that my variable is never going to hold a negative value.

Lists:

The STL List allows the user to specify the type of data contained in the node with some new syntax. Below is an example of creating a new list that stores integers.

```
list<int> numbers_list;
```

Nothing else can be put into this list except integer values. Replacing what is in the < > brackets with another data type will create a list that can store that specific type of data. To use lists in your program you need to include the correct header information:

```
#include <list>
```

With arrays, we could use bracket notation to access individual elements inside the array. For example:

```
int array = {1, 2, 3};          // int array containing three elements
cout << array[0] << endl;       // output the first value in the array
```

This will not work with a STL list and instead you will have to use something called an iterator. The syntax to create an iterator is ugly and as an example, if I were making an iterator for an integer list it would be:

```
list<int>::iterator it;
```

First we need to get the iterator to point to a valid location in the list. To get the iterator to point to the first list element for the *numbers* list the syntax would be:

```
it = numbers.begin();
```

Once the iterator is pointing to a valid list location it is possible to move the iterator forward and backward by one element in the list by using ++ and -- operators. Be careful not to move the iterator outside of the list bounds.

```
++it;
--it;
```

To access a specific element in the list that the iterator is pointing to you have to use the * operator. Here is an example:

```
*it = 23;
cout << *it << endl;
```

To figure out how to do the rest of the lab you will need to use the reference below:

http://www.cplusplus.com/reference/stl/list/

Goals:
Learning about C++ STL lists.

Deliverables:
Make sure that the **Makefile** and **stl_lab8.cxx** file for the program are in a directory named Lab8 and then zip the directory. Rename the zip file to your first name underscore last name (e.g. jeffrey_lamarche.zip) and submit the zip file on the Desire 2 Learn website in the Lab 8 dropbox.

Due Date and Time:
Friday October 26th 2012 at 11:59pm. Labs turned in after this time will lose one point per day late.

Hint:
The only list functions needed are: `begin()`, `end()`, `push_front()`, `erase()`, and `size()`

Examples of Output:

```
Please enter a string of text: ab

User text entered: ab

Reverse of text:   ba

List size too small for remove!
```

```
Please enter a string of text: abc

User text entered: abc

Reverse of text:   cba

List size before remove = 3
Removing elements(s) - b
List size after remove = 2

List contents after remove: ca
List contents reversed: ac
```

```
Please enter a string of text: abcd

User text entered: abcd

Reverse of text:   dcba

List size before remove = 4
Removing elements(s) - b c
List size after remove = 2

List contents after remove: da
List contents reversed: ad
```

```
Please enter a string of text: Isn't the C++ STL List great?

User text entered: Isn't the C++ STL List great?

Reverse of text:   ?taergtsiLLTS++Cehtt'nsI

List size before remove = 24
Removing elements(s) - S T
List size after remove = 22

List contents after remove: ?taergtsiLL++Cehtt'nsI
List contents reversed: Isn'ttheC++LListgreat?
```