A very brief note about lists

We started with arrays. A drawback to using an array is that it can't grow or shrink. One solution to that problem is vectors. A vector can grow and shrink as you need it to. However, even a vector has a drawback. You can insert and remove entries from only one end. A list is like a vector, but can do more. You can insert and remove entries from either end.

Here are details for a list.

1. You need a #include:

```
#include <list>
```

2. When you create a list, you need to tell what type of data the list will contain. This is very similar to what you do for a vector. For example, if the list will contain integers, you would declare it

```
list <int> name_of_list (initial_size);
```

Here name_of_list is the name that you will use to access the list and initial_size is the initial size of the list. Like a vector, initial_size can be 0.

3. Now, you use the push_back (value) and pop_back() functions to insert and remove entries like a vector. But, you can also use push_front (value) and pop_front() to insert and remove entries to and from the front of the list.

4. There is a front() function to see what is at the front of the list and a back() function to see what is at the back of the list. Neither of these functions changes the list in any way; they just examine the data in the list. A goo idea would be to use the combination

```
front_value_of_list = list_name.front();

list_name.popfront();
```

to get the item at the front of the list and then remove it.

5. Finally, there is also a size() function to determine the size of the list, and a clear() function to erase all the elements in the list in case you want to start over.