Computer Science II — CSci 1200
Lab 8
Generic Functions; Maps, Part 2

Introduction

Download the following two files, which will be used for Checkpoints 2 and 3.

```
http://www.cs.rpi.edu/academics/courses/spring07/cs2/lab08/macbeth.txt
http://www.cs.rpi.edu/academics/courses/spring07/cs2/lab08/checkpoint2-3-start.cpp
```

Once you have downloaded these files, you need not access the network any further. Turn off all network connections.

Checkpoints

1. The standard library provides a number of generic functions, each of which can be used with a variety of containers (data structures). These functions interact with the containers through iterators. You already have some experience with one of these functions, `std::sort`, which requires a begin iterator, an end iterator and, optionally, a comparison function. As we have already seen with `std::sort`, some containers have their own specialized version of these functions — in particular, `std::list` has its own sort function.

   One of the more useful generic functions is `std::find`, together with its close relative, `std::find_if`. Both are declared in the `algorithm` header file. `std::find` takes 3 arguments — a begin iterator, an end iterator, and a value — and returns an iterator that refers to the first location in the sequence, starting from the begin iterator, that contains the value (assuming `operator==` is defined on the type stored in the container). If the value is not in the sequence, the end iterator is returned. Therefore, testing the returned iterator to see if it is the end iterator determines if the value is in the sequence. The second form, `std::find_if`, also takes 3 arguments — a begin iterator, an end iterator, and a boolean function whose single argument has the same type as the values stored in the container — and returns an iterator referring to the first location in the sequence for which the function returns true. If the function does not return true for any value from the begin iterator up to (but not including) the end iterator, `std::find_if` returns the end iterator. `std::find` and `std::find_if`

work on any container for which iterators have `operator++` defined. They are usually applied to vectors, lists and arrays but not `std::map` and `std::set` because these two have their own version of `find` which exploits the internal structure of the container to much work faster than `std::find`.

Now onto the actual work of the checkpoint: Write a simple program that explores the use of both `std::find` and `std::find_if` In your main program create a vector of ints and a list of doubles and add values to these sequences. Next, write a simple bool function that returns true if the double value it is given is positive. Finally, write code that shows each of `std::find` and `std::find_if` both (a) actually finding a value and (b) not finding a value. Output an appropriate message in each case.

2. In the rest of this lab you will write a program that reads text from one file, breaks the text into words, and eliminates words that are shorter than 4 characters in length. Then the program will determine which words are 3 or fewer words away from other words (the distance between words is defined below, in Checkpoint 3). This provides an association between words, which may be used by a search engine.

   In Checkpoint 2, write a program that has two command-line arguments — the input file and the output file. The program should open and read the input file, find the words, eliminate those that have fewer than 4 letters, and it should store the other words into a vector in the order they are read. It should then output the vector to the specified output file. Words are defined as an uninterrupted sequence of alphabetic characters (letters). Convert all letters to lower case. You may start with the code in `checkpoint2-3-start.cpp`, which does a great deal of the work for you. Use the file `macbeth.txt` to test your program.