

Vector, String and Iterator

You'll find a source file, sortWords.cpp and an input file, input.txt on the course homepage. You can also download the exercise files with the following curl commands:

```
curl -O https://www.csc2.ncsu.edu/courses/csc230/exercise/exercise25/sortWords.cpp
curl -O https://www.csc2.ncsu.edu/courses/csc230/exercise/exercise25/input.txt
curl -O https://www.csc2.ncsu.edu/courses/csc230/exercise/exercise25/expected.txt
```

The job of this program is to read a list of words from standard input, sort them using the sort() template algorithm. Then, you'll print the list twice, once using the index operator and then using an iterator to traverse the list. When you're done, you should be able to run the program like this:

```
$ ./sortWords < input.txt
```

```
-- Backward --
```

```
words
to
this
this
test
sort
see
really
read
program
print
of
of
list
it
is
if
can
and
and
a
a
-- Forward --
a
a
and
and
can
if
is
it
list
of
of
print
program
read
really
see
sort
test
```

this
this
to
words

I've left several places for you to add code:

- First, fill in the function parameter for `readWords()` to pass in the word list by reference. The `printBackward()` and `printForward()` functions don't need to change what's on the word list, so you can pass it by const reference for them.
- Fill in the body of `readWords()` to read a list of words from standard input and store them in the vector of words. Remember, you can use the `>>` operator to read into strings from standard input. It will read space-delimited tokens (i.e., words).
- In main, add code to call the `sort()` template algorithm to sort the word list in lexicographic order. Remember, by default `sort()` will compare elements using their less-than operator, and less than on strings compares them in lexicographic order, so ...
- Fill in the body of `printBackward()` to use the index operator (e.g., `words[i]`) to print the word list in reverse sorted order, one word per output line (starting with the last word on the list and ending with the first word).
- Fill in the body of `printForward()` to use iterators to print the word list in sorted order, one word per output line. Use the `auto` keyword when you declare the iterator variable. Iterators for traversing a const container have a slightly different type, `const_iterator` rather than `iterator`. With `auto`, you won't have to worry about this. The compiler will figure out the type the iterator needs to have based on the type of the value you initialize it with.
For this part, **don't** use the simplified for loop syntax, **for (type var : some_container)**. I want you to try using iterators directly.

When you're done, submit a copy of your completed `sortWords.cpp` file using the `exercise_25` assignment on Moodle.