Challenge Statement

This challenge is about fixing and improving the given C++ project. The included project only contains one file, ‘buggy.cxx’ which contains both bugs and design deficiencies.

The idea is to:

- Fix ‘buggy.cxx’ so that it runs correctly
- Improve upon the design of ‘buggy.cxx’
- Modify the project so that it is up to par with *your* project and coding standards

Details

The application 'buggy.cxx' performs the following functions:

a) It reads a list of ASCII words from STDIN, a word per input line, terminated by the word 'end'.
b) It removes the duplicates, sorts the words alphabetically, and prints them with the number of how many times each word was encountered.
c) Then it repeatedly asks the user to enter a word and tries to look it up in the word list which was entered initially. It terminates when it encounters EOF.

The project has a number of bugs and design deficiencies. Please, fix all problems that you can find and make all the improvements you deem necessary so that the result satisfies your personal standards (what would you expect a well maintained C++ project to look like? File structure? Class structure / encapsulation? Testing tools? Build tools?).

The only restriction is that the multi-thread nature of the application and the overall structure MUST be preserved. (In other words, you shouldn't, for example, convert it to a single thread, even if you think it would work better in that way).

Some (but not all) the things you might want think about when improving upon the project:

- Modern C++ coding standards, paradigms and styles
- Proving the correctness of the program
- Project tooling (ex. reproducible builds)
- Code / project style and structure

What to return back to us

1. The project folder and all its contents.
2. A CHANGELOG file in the project folder outlining your changes to the code and project (alternatively you may use version control history to document your changes). Please explain why you made each change.
3. A README file in the project folder containing instructions on how to build and execute your program.
4. Please zip or tar everything in a directory named `yourfirst.lastname/` and return via email.
5. In your email response please let us know roughly how many hours you spent on this exercise.