Computer Science II — Homework 1 — Word Search .

This assignment is due Thursday, January 25 at 11:59:59pm and is worth 60 points toward your homework grade. You must submit both a C++ source code file and `readme.txt` file. A template for the latter will be provided on the course web page. In solving this problem you may use any C++ construct or algorithm you wish from Lectures 1 and 2, but not beyond. Lecture 2 and the associated examples, posted on the web, are particularly important.

The program will be provided two input files — a keyword file and a search file — and it must produce one output file. The form of the command line will therefore be

```
keyword_search keywords.txt search-file.txt results.txt
```

See the example code from Lecture 2 for ways of reading command line arguments and attaching them to input and output streams.

Both input files will contain sequences of words, separated by whitespace. There will be no characters in the files other than whitespace and lower case alphabetic characters. You may assume the keyword file contains no repeated words.

Your program must determine the number of times each keyword exists as an exact match in the search file. Your program should output the keyword strings in lexicographic order (the order imposed by the `<` operator on strings), with one keyword and its number of occurrences on each line of output. Following this output, the program should output a blank line followed by the keyword that occurs most often. If there are several such keywords, all should be output.

Your program should do the search efficiently; more points will be given to more efficient techniques. Here are some hints in this regard:

- Binary search on an ordered vector of N strings can be assumed to work in $O(\log N)$ time, and the non-recursive version of binary search is faster in actual execution time than the recursive version.

- You may assume that there will be many more search words than keywords.

Your `readme.txt` file should contain your name, a summary description of your algorithm (just a few sentences will suffice), and a simple analysis of the efficiency of your program. If you find the latter difficult, do the best you can and do not worry about it too much for now.

Finally, here is an example of the input and output from our solution to this assignment. The keywords file is

```
 apple pear banana cherry orange  grapefruit
```

The search file is

```
  once upon an orange morning the grapefruit went to buy an apple but
only found a pear actually several pears and an orange

  the grapefruit
bought a pear and another pear but then the grapefruit wanted an apple
anyway so the grapefruit went to the banana and orange store
```

```
   there the grapefruit found
applesauce and once again an orange but again did not find an apple the
grapefruit went to orange lane where the rest of the orange family was
waiting
```

The output file contains

```
apple 3
banana apple 3
banana 1
cherry 0
grapefruit 6
orange 6
pear 3

Max keyword occurrences = 6
Number of max keywords = 2
Max keywords:   grapefruit orange
```