COSC2320: Data Structures and Algorithms
HW0: Finding the most repeated word in a document

1 Introduction

You will write a C++ program to find the most repeated word in a document. You can use any algorithm and data structures that you prefer, as long as the results are correct. It is preferred, but not necessary, that your algorithm is as efficient as possible, both in time to complete as well as memory management. **You are not allowed to use the STL Library.**

2 Input

The input is one text file, with at most 100 different words, separated by spaces (repeated spaces should be considered as one space. You can assume that words will be at most 100 characters long.

Example of input and result file:

```
input.txt
------------------------------------------------------------
The hungry, hungry caterpillar
 Eric Carle
------------------------------------------------------------
Result: hungry=2
```

Notice that words can be followed by punctuation signs and that some of them can be capitalized. This should not interfere with your results: a capitalized word is the same as a non-capitalized word for the purposes of this homework. Uppercase letters can be anywhere in the word, for ex.: ”Hungry”, ”hungry” and ”hUngry” should be counted as the same word. Punctuation can also be ignored.

3 Program and input specification

The main program should be called ”topWord”.

Syntax:

```
topWord filename="input.txt"
```

Note that the file name will not necessarily be the same every time, so your program will have to take that into account. You can use the Command Line Parser that is provided in the T.A.'s homepage.

The output should be sent to the screen in the form

```
Result: hungry=2
```

Please note that the output should always be in lowercase, regardless of how the words appear in the text.

4 Requirements

In order to successfully complete this assignment you must:

- Download the Borland C++ compiler. Links to the download page as well as instructions on its usage are in the T.A.'s homepages. It is recommended that you do a couple of short programs with the compiler in order to familiarize yourself with its usage.
- Once you have some experience with the compiler start your assignment. Remember that all the following homework sets will build upon each other, so it is important that you start with the right foot. Try to write your classes and procedures in a way that can be used in the future and **don't forget to comment your code**
- Do not use the STL library.
- Your program should write error messages to the screen. Your program should not crash, halt unexpectedly or produce unhandled exceptions. Consider empty input, zeroes and inconsistent information. Each exception will be -10.
- Test cases. Your program will be tested with 10 test cases, going from easy to difficult. You can assume 80% of test cases will be clean, valid input files. If your program fails an easy test case 10-20 points will be deducted. A medium difficulty test case is 10 points off. Difficult cases with specific input issues or complex algorithmic aspects are worth 5 points.
- A program not submitted by the deadline is zero points. A non-working program is worth 10 points. A program that does some computations correctly, but fails several test cases (especially the easy ones) is worth 50 points. Only programs that work correctly with most input files that produce correct results will get a score of 80 or higher. In general, correctness is more important than speed.
- Since this homework will only be "Pass/Fail" if your program scores more than 50 points, you will pass.