De Anza College
Summer 2016
CIS 22C: Data Abstraction and Structures
Programming Project 3 (Hash Tables)

For this programming project, you will be implementing a spelling checker. The spelling checker will accept words from a dictionary and a document and then output the words in the document that are not in the dictionary. You should store the words in the dictionary in a hash table and your program should output statistics about the hash table's performance. The instructions are as follows:

1. Implement the separate chaining **`HashTable`** template class as shown in Figures 5.6–5.10 (not including Figure 5.8) in the book *Data Structures and Algorithm Analysis in C++*. You should use your **`HashTable`** class to store the words (i.e. strings) in the dictionary. Use the hash function for strings shown in Figure 5.4.

2. Add a new method to the **`HashTable`** classed called **`printStats()`** that prints the minimum, maximum, and mean chain length for all the chains in the hash table.

3. Generate your own dictionary and document files

4. You may assume that the dictionary file always have the filename **`dictionary.txt`** and the document file always have the filename **`document.txt.`**

5. Save the dictionary and document files in the same directory as your executable code.

6. The dictionary file contains one word per line, and the first line will be the number of words in the file. The words are all in lowercase.

7. The document file contains lowercase words separated by spaces or new lines.

8. Your program should work with any dictionary and document files that comply with the above constraints.

9. Provide a main program. Your main program should do the following:

   (a) Create a hash table whose size is equal to the number of words in the dictionary.

   (b) Read the words from the dictionary and store each one in the hash table.

   (c) Output the statistics (i.e. minimum, maximum, and mean chain length) of the hash table after all words in the dictionary has been stored in the hash table.

   (d) Read the words from the document and output each word that is NOT in the dictionary.

The following are the additional instructions for this programming project:

- This programming project is to be done individually.

- Create a **`readme.txt`** file that describes exactly how to compile and execute your program.

- Collect your source codes, readme file, and other files needed to compile and execute your program into one ZIP file called YourFirstName_YourLastName_prog3.zip. **Please DO NOT include any executable files in your ZIP file.**

- Make sure you follow good object-oriented programming approach, good coding style, and proper documentation.

The grading for this programming project will be based not only on the correctness of the program, but also on the program's overall design, coding style, and documentation.

Submitting Assignments:

- Submit the ZIP file on Catalyst.