**Hash Tables**

Last Modified: 23 Feb 2022

This assignment is due to be completed and submitted by noon Friday, 4 March.

1. Given the input 3406, 6672, 4669, 1509, 8446, 9991, 3723, 9381, 3441, in that order, and the hash function *h*(*x*) = 6 – (*x* mod 7), draw the resulting hash table. State and explain any assumptions you make.
2. Assume that conditions indicate a need to rehash the hash table from problem 1. Perform a rehash of that hash table, and show the hash table that results from the rehashing. Explain your process and results.
3. Every standard Unix system has a file /usr/share/dict/words which is a newline-delimited utf-8 file with many common words and names. The command $ wc -l /usr/share/dict/words will tell you how many lines are in the file (the wc command stands for wordcount; here, there is one word per line). Here is a hash function:
4. size\_t hashx(const string& key, size\_t table\_size)
5. {
6. size\_t hash\_value = 1;
7. for (auto character : key)
8. {
9. hash\_value = 43 \* hash\_value + static\_cast<size\_t>(character);
10. }
11. return hash\_value % table\_size;
12. }

Implement this algorithm using a table size that approximates a load factor of 1 when the function is used to hash the contents of /usr/share/dict/words as the keys to be hashed. Implement this function in a program that reads words from standard input. [Here](https://borax.truman.edu/310/304/template.cpp) is a program you should use as the template for your program to read one-word lines from standard input. Run your program with the command line

$ cat /usr/share/dict/words | ./program

Count the number of collisions the algorithm produces when the words are hashed. *Do not* actually implement a hash table! Rather, just count how many collisions there would be if you did implement a hash table. You should pay particular attention to efficiency in writing your program. An input file with 100,000 words should take less than a second to run and produce output.

1. State, explain, and justify the results you got from running the program in question 3. Be sure to explain what table size you used in your program, and why you chose that value, and what the load factor used by your program is.

To aid you in making a reasonable drawing of a hash table for problems 1 and 2, [here](https://borax.truman.edu/310/304/template.tex) is a LaTeX program template that shows how to use tikz to draw a picture of a hash table.

Create a LaTeX document that addresses questions 1, 2, and 4, and states the results you got in question 3. *Do not* copy a lot of code into your LaTeX document. You can put a little code in if you need to, or refer to line numbers of your program.

By the due date, use the [homework submission](https://borax.truman.edu/310/submit.php) page to submit the .cpp program for processing the words, the .tex source file, and the finished .pdf writeup.