CSCI/CMPE 3333 Assignment Five
Instructor: Zhixiang Chen

In this homework assignment, I would like you to have some fun playing hash functions.

Design a "Super Hash Function": This is a group project. Each group can have at most two team members. If you would like, you can work alone. The goal is to design a "super hash function" to hash lowercase English letter strings of length 20 into integers (four byte int type) with a collision rate as low as possible. You have to carry out numerous "design-challenge" steps: first design and implement a hash function, and then try all you can to detect collisions for random strings with 20 lowercase English letters. Repeat these until you have strong confidence that your hash function is indeed *"super",* that is, with practically zero collision if possible.

Special Requirement for Your Program: For ease of grading, I highly recommend all of you to use the following format to declare your hash function:

int superHash (const string & str)

In addition, save your hash function in separate header file named as *"supperHashHeader.h".* The grader will either copy your function, or include the header file, into his driver program.

Here is the fun part: The base score for this assignment is 250. The grader will generate a random list of 1,000,000 lowercase English letter strings of length 20 and save these strings into a file. He'll use these strings to challenge your "super hash function" and count how many collisions your "super hash function" will have on these strings. For each collision detected, the grader will take 1 point off from your base score. Say, the total number of collisions is T, then the final score for you is *max{0, 250 – T}.*

Note: The list of these 1,000,000 random strings used by the grader is confidential. Moreover, the same list will be used to challenge all groups' "super hash functions."

Note: Please don't try to fool the grader with some strategies as follows:

- Assign the first string to 1, the second to 2, the third to 3, and so on.
- Design a hash function to hash strings. Once a collision occurs, rehash the string or assign some integer that has been used hashed to for the string.

If the grader finds you try to fool around, he will give you a zero grade.

A simple note about the expected number of collisions:

In the context of this assignment, essentially you need to hash strings of length 20 to integers from 1 to $m = 2^{32}$ =4,294,967,296. When hashing n = 1,000,000 random strings, the expected number of collisions is

$$n - m \times \left(1 - \frac{1}{m}\right)^{n}$$

$$= 1{,}000{,}000 - 2^{32} \times \left(1 - \frac{1}{2^{32}}\right)^{1{,}000{,}000}$$

$$\approx 117$$

Hence, expectedly you shall receive 250-117 = 133 points.

However, some simple, straightforward calculation shall give you a more promising result

$$
\begin{aligned}
&m = 4294967296 \\
&m - n = 4293967296 \\
&m / 2 = 4293967296 \\
&m / n = 4294.97 \\
&m \approx 4295 \times n
\end{aligned}
$$

The above indicates that for each random string, you have about 4295 many possible integers to hash to. If somehow your hash function can emulate uniform distribution well, then there is a good chance that your hash has no collision. Therefore, there is a chance that you can receive 250 points.

Below is a very nice reference to hash collision estimation by Professor Rosa Orellana in the Math Department at Dartmouth College:
https://math.dartmouth.edu/archive/m19w03/public_html/Section6-5.pdf

You got to be a little bit careful to use Theorem 6.15 in Professor Rosa Orellana's note, where the number of hash values is assumed to be larger than the number of possible slots (or positions or integers).

For your convenience, the document referred above is attached in this assignment.

Acknowledgement:
Samuel Perales told me (4/27/2016) during our Watson Group Meeting that the original requirement for Home Work 5 may be too challenging and that everyone may expectedly receive 0 points, according to the grading formula ***grade = max{0, 200 – 5T}.*** The reason behind his concern is that he learned from some source over the Internet that the expected number of collision for the given assignment is about 117. I then verified that that number is right. However, if one can use a good pseudo-random number generator, the original requirement is achievable with a higher grade. Nevertheless, I decided to relax the requirement to the current state. I thank Samuel for his input.

Due Date:
The due date will be given via Blackboard.

Warning:
Any submission one week after the due date will not be accepted.

How to submit your work?
Please upload your source program files and your test results to Blackboard.