Name:

Description: We will be hashing objects into hash sets with 2 different sizes and using 3 different hash functions:

* Array sizes: 5749, 2857
* Hash functions: , ,

Note: Use separate chaining to handle collisions

We will be using the Stanford Graph Base that contains a list of the 5757 five-letter words in English. Here is a link: <https://www-cs-faculty.stanford.edu/~knuth/sgb-words.txt>

* Complete the following steps for each word in the file:
  1. Compute the [hash code](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) of the word.
  2. Compute the index, i, where the word would be placed in the array.
  3. Add 1 to x[i] to keep up with the number of times a word hashes to location i.
* When you have read the file completely, compute and print the following statistics for each unique hash set:
  1. The number of empty positions (buckets) in the array.
  2. The maximum number stored in a single array location (the longest bucket chain).
  3. The average of all non-zero array entries (the average bucket chain length).
* Given the above results, which hash function and array size would you recommend.

\*Given the results, we’d recommend using the first hash function . There are less collisions with this hash function, resulting in a more efficient algorithm. (See results below).

