Due Date: Tuesday, April 5th @ 11:59pm

Points: 100

This is an individual assignment.

Objective:

For this assignment you will be testing two claims made in the textbook.

Claim 1:

On page 413 the textbook claims the 33, 37, 39 and 41 are good choices for the ***a*** in the polynomial hash code.

Claim 2:

On page 416 it is claimed that selecting a prime number larger than N for the ***p*** in the MAD method reduces the number of collisions.

You are to test these claims in the following fashion.

Claim 1:

Write a **polynomialHashCode** method that implements the polynomial hash code described on page 413 of the textbook. This method should have two parameters. The first parameter is a String that is the key to be converted into a hash code. The second parameter is an int that is the value for ***a*** in the polynomial had code formula.

Test your polynomialHashCode method on the words in words.txt file included with this assignment. For values of ***a*** between 30 and 45 inclusive, determine the total number of collisions and the maximum for of collisions for each value of ***a***.

Claim 2:

Write a **madCompression** method that implements the Multiply-Add-and-Divide that compresses a hashCode using the MAD method in the textbook. This method should have five parameters. The first parameter is an in that is the hashCode to be compressed. The second parameter is an int that is the value for ***N***. The third parameter is an int that is the value for ***p***. The forth is an int that is the value for ***a*** and the final parameter is an int that is a value for ***b***.

Your method should check to see that the values for ***a*** and ***b*** satisfy the conditions stated in the textbook and if these conditions are not satisfied you method should throw an illegal parameter exception.

Test your madCompression method for various values of ***p*** where ***p*** is the next prime number after N +/- 5. For example, if N is 8 the next prime number would be 11 so you would test your method for 6 <= ***p*** <= 16. In this case N would be determined by the number of entries in the word.txt file. Be sure to allow for an appropriate load factor when determining your value for N. You can assume that you will be using the open addressing approach to collision handling.

Create a NetBeans project whose name follows the convention Lab109-LastFM

Create a Client that meets the following requirements:

* Tests the polynomialHashCode method as described earlier
* Prints an ASCII table of the results similar to table 10.1
* Tests the madCompression method as described earlier
* Prints an ASCII table of the results similar to table 10.1

When your print your ASCII tables be sure that values over 999 use the comma thousands separator.

Things to turn in:

* Open a Microsoft Word document named using the Lab109-LastFM.docx convention
* Copy and Paste the source code of each of your classes
  + Include all of the classes you created/transcribed for this assignment.
  + Do not include classes that you copied from previous assignments.
* Run your client program and then copy the contents of the Output Windows to the clipboard and paste it into your Word document.
* Next, export your NetBeans project to a zip archive.
* Finally, on blackboard, submit both your Word document file and your zipped project file in one submission.