Project: Hashing Twitter

Due Date: 10/28/2015

Description

In this project you will be given a set of data containing Twitter posts in JSON format and store them in a hash table. Each post will contain three values: created_at, text, and screen_name.

Here is an example of a twitter record

```
{
   “create_at”:“2013-05-04T03:40:53Z”,
   “text”:“\"Sleepy sweet home Alabama, roll tide roll.\"”,
   “screen_name”:“CarolineAshby1”
}
```

Each twitter record will be separated by a comma; however, the text of a Twitter message may contain commas and the parts of a twitter message are separated by commas.

Your key will be the screen_name and the value will be a linked list of the user’s tweets in ascending order by create_at. You should insert each tweet as they are seen into the table. You should be using a data-structure to store the tweet data and that data-structure will be the value in the hash table.

Requirements

Your program will read in the Twitter posts from a file where the filename will be passed in through the command line.

Once you have read in the Twitter data you will prompt the user with a menu which must be in this order:

1. Enter new Tweet
2. Delete all tweets for a given screen_name
3. View Tweets
4. Quit

Option 1 as in this order: Tweet -> Username
Option 2: Prompt for a screen_name. Remove all tweets from the hash table for the given user. Report if the name is not found.
Option 3: Prompt for the screen_name and display all tweets from that user, again report if the name is not found.
Option 4: Quits the program.

You should use the djb2 hash function shown in class.

Style

Your program should show good style. You should extract specific functionality into functions and classes when it makes sense. You should try to keep your functions around 25 lines long. Your main method should be relatively small. You should not leave in code which has been commented out. Comments should be short and to the point.

You should have at minimum, but hopefully more, a HashTable/Dictioanary/HashMap/etc. class. This class should have the following public methods and functions:

Put – Takes in a key and a value and saves it in the table
Get – Given a key returns the value associated with it
Contains – Given a key determines if there is a value associated with it in the table
Count – Returns the number of items in the table
Remove – Takes in a key and removes the value associated with it

Additionally, it would be beneficial to have some sort of structure which could hold Tweets per user. Also you might be interested in a menu class which takes in a list of strings and how many to show per page.

What can and cannot be used

Here is a list of things you are allowed to do as well as things you cannot do. If you do anything on the "cannot do" list then you will receive a LARGE deduction, up to full points, to your grade.

Can

1. Use vector and arrays
2. Use string class
3. Use functions found in the C++ standard library and STL

Cannot

1. Use code which is not yours
2. Use C, i.e you must you I/O streams, classes , etc
3. Use std::map or other built in data-structures other than vector

Deductions

You will receive deductions based on the areas above: requirements, good style, and using forbidden items. Additional deductions will be applied if the submission requirements are not met as well. If there are any issues with compiling you will not receive any credit unless it is a trivial mistake, in which case you will receive significant deductions. If you do not submit a makefile you will receive significant deductions as well.

The breakup of points will be:

Requirements	60%
Good Style:	40%
Allowable Content	0-100%

Submission

You will submit your assignment to Blackboard. You must make sure you submit your project is zipped correctly. As before you must not include anything but your source code needed to compile and the makefile which will be used to compile it. The program which will be created must be named project4.