CSCE 221 Cover Page

Homework Assignment # PA-4

First Name: Justin Last Name: Lee UIN: 727000824

Username: 727000824 E-mail address: jlee232435@gmail.com

Please list all sour es in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you a get a lower number of points or even zero, read more in the Aggie Honor System O- e http://aggiehonor.tamu.edu/

Type of sources	Textbook	Lecture Slides	Website	
People				
Web pages (provide URL)		Dr. Leyk's Hash Slides	https://www.geeksforgeeks.org/csv-file-management-using-c/	
Printed material	Data Structures and Algorithm Analysis in C++ by Mark A. Weiss			
Other sources				

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work. *On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.*

Your Name: Justin Lee Date: 4/13/20

Assignment number and its description.

The PA4 assignment attempts to make a HashTable by storing student UINs from a .csv file. The HashTable is then used to search for the unique UINs of each student with hash keys and input their associated grades. The program creates a "output.csv" upon compilation with all the correct quiz grades inputted for the students who took the quiz as well as some statistical output of the HashTable's minimum, maximum, and average lengths.

Description of data structures and algorithms used by your program.

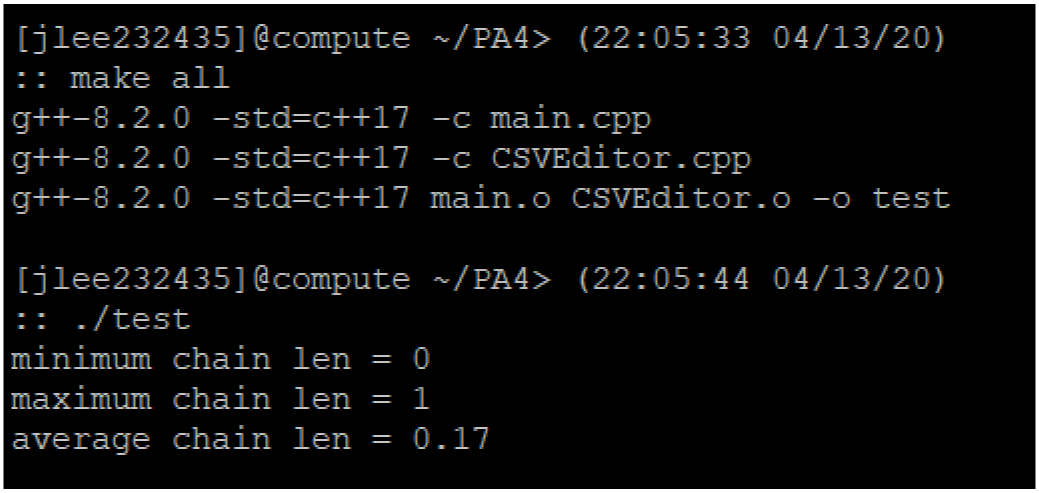
The main data structure used in this program was the Hash Table. The Hash table data structure is a vector of linkedlists that was used in the program to store key-value pairs. The UIN of the student would then be used as a hash key to easily access a student's grade. Another data structure that was used was the vector, which is simply a sequence container representing size mutable arrays. This was used to access particular values in the list and to perform some crucial functions such as the push_back() which helped insert data into the hash table itself.

Description of input and output data. List all restrictions and assumptions that you have imposed on your input data and program.

The input.csv file had a list of student grades, where it was than read through the I/O stream to be parsed with Regex and eventually get all the UINs and grades for each student. This data would then be stored into the Hash Table, where the roster.csv would then be parsed with Regex again to get the UIN as a key for the hash. This hash key would allow for the search of a student's information according to the UIN in the hash table. The roster is then updated in the output.csv with all the grades added in.

It was assumed that each UIN of 9 digits was unique to the student and that the quiz grades were smaller than 100. Restrictions that were imposed were that the input data's third and fourth column on the .csv spreadsheets only had digits. Another restriction was that the HashTable had a size of 100 entries.

How have you tested your program for corrections?



```
[jlee232435]@compute ~/PA4> (22:05:33 04/13/20)
:: make all
g++-8.2.0 -std=c++17 -c main.cpp
g++-8.2.0 -std=c++17 -c CSVEditor.cpp
g++-8.2.0 -std=c++17 main.o CSVEditor.o -o test

[jlee232435]@compute ~/PA4> (22:05:44 04/13/20)
:: ./test
minimum chain len = 0
maximum chain len = 1
average chain len = 0.17
```

I used the given main.cpp test file to print out my output .csv, which had all the correct data matching from the input.csv and the roster.csv. As for the Hash Table statistics, they matched properly with what was expected of the program. The program itself was run through a makefile where typing "make all" would compile the program and "./test" would output hash table data as well as create the output.csv.

Which C++ features or standard library classes have you used in your program?

The STL classes that I used in my program were iostream, fstream, sstream, regex, string, utility, and vector. These were all used to help read/write from the .csv files, parse the data, and update the hash table.

Provide the statistics about the hash table. Are the computational results about the hashing consistent with the expected running time for the hashing algorithm? Justify your answer.

The hash table had a minimum length of 0 for the lists with no values for quizzes, a maximum length of 1 for every UIN that contains a quiz grade, and an average length of 0.17 for the total number UIN's with quiz grades per total number of UINs. The min, max, and average functions all had a runtime of O(n) as all the elements needed to be traversed.

The hash table itself is just a vector of linked lists which have a node with a key-value pair of (UIN, grade). Thus, the computational results about the hashing are indeed consistent with the expected running time for the hashing algorithm as both the insert and search functions have a constant run time because only one element exists in each given list.

Write your conclusion.

In conclusion, this project has increased my comprehension of hash tables, Regex, and file handling with I/O streams. This program served the purpose of implementing I/O streams for data reading and writing, Regex for parsing the data, and the hash table for updating the students' quiz grades into a new roster.