CSCE 221 Assignment 5 Cover Page

First Name Leilani Last Name Horlander-Cruz UIN 523008771

User Name leilanihc112 E-mail address leilanihc112@tamu.edu

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

Type of sources				
People	Ian Dickerson			
Web pages (provide URL)	http://cplusplus.com	http://piazza.com		
Printed material	Powerpoint slides			
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	Leilani		Horlander-Cruz	Date	4-13-17

\begin_inset Separator latexpar\end_inset

CSCE 221 — Programming Assignment 5 (100 points)

Due: April 13th, 2017 at 11:59 pm

Assignment Description (100 points)

The purpose of this project is to write a program to help record every students' grades in a spreadsheet roster. When TAs collect results from an online quiz platform, the platform can only provide the grades of students who have submitted their quizzes. If a student skips them, the online platform will not keep track of their grades. Therefore, when a TA downloads data from the platform, he/she will only have the submitted results which is actually a spreadsheet.

- However, we can have a thousand of students in a class but we have much less submissions. Then, how can we fill these collected scores into a spreadsheet which has

thousands of entries in the most efficient way? A straightforward solution is to scan the whole list for each collected score but it is too slow when the number of students is very large.

- Therefore, we consider using a hash table to record all collected scores since we will have the UINs and we can use them as unique keys in the hash table. Thus, for the roster, we read it row by row and we can get UIN. We look up the UIN in the hash table and we will find the corresponding score. Then, we can record collected scores in the roster.

Description of data structures and algorithms used by your program

Vectors

Vectors are much like arrays. They allocate data in contiguous memory. Unlike static arrays, vectors can be grown. The vectors in this program were of string type and were called to_search, to_search2, and UINs. These vectors held data that would be used to sort the data that would be printed the output file and would be compared to the input file for matches and matches2.

Structs

Structs are composite data type declarations that define physically grouped lists of variables to be placed under one name in a block of memory, allowing the variables to be accessed via a single pointer, or the struct declared name. The struct created in this program was called record and held strings called UIN and grade, which were then used to compare the hash table values to the matches in input.csv and roster.csv.

HashTable / Maps

HashTables are used for fast lookup when recording the data into an output file. They are vectors of linked lists. The size of the hash table is equal to number of the students in the roster. In C++, map uses hashing to store its objects. The key values are used to sort and uniquely identify the elements, while the mapped values store the content associated to this key. In this program, a map called records used the keys of type string, which were the UIN, and the struct record, which were the grades, as the type of mapped value.

Regex Patterns

Regular expressions are sequences of characters that define a search pattern. Usually this pattern is then used by searching algorithms for “find” operations on strings. These were used to find the UINs of the students who had grades from input.csv on roster.csv in order to be able to print their grades out on output.csv.

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

roster.csv and input.csv were the input files used and the output file was output.csv. The restrictions imposed by this data are the constant file names, meaning that these are the only files that can be used as input and output in this program. If the user were to want to use different input files, they would have to modify the code accordingly, as well as the output file name. Any output file with the same name will be replaced if the user were to run the code multiple times, including if they changed the input data.

How have you tested your program for corrections?

roster.csv and output.csv were compared to see if the grades were accurately appended.

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

#include <iostream>
#include <fstream>
#include <cstdlib>

#include <string>
#include <regex>
#include <unsorted_map>

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 computational results worked as expected. The unsorted_map feature in C++ was used, which is similar to a hash table, and uses the separate chaining method with buckets to resolve collisions, so the corresponding operations follow closely to the expectations from the hash algorithms. On average, the runtime will be a constant time of O(1) and O(n) in the worst case. bucket_count, which is the number of elements in the hash table, runs in the constant time and bucket_size, which is the size of the collision chain, runs in time linear to the size of the bucket. The O(1) run time is apparent in the statistics of the hash table that were obtained.

Minimum: 0
Maximum: 3
Average: .73913

Conclusion

This assignment allowed for freedom in determining which way the programmer wanted to solve the task at hand. Using information presented in class and some creativity, the code was written and tested to prove successful. Even though we were pretty much left to our own devices, I still see this as a strong learning experience and can say I like this type of assignment more than the other ones. I like being given a task and being allowed to solve as I please. It did, however, allow me to learn about hash tables and collision handling.