CSCE 221 Assignment 5 Cover Page

First Name   Last Name   UIN

User Name   E-mail address

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				
Web pages (provide URL)				
Printed material				
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   Date

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 that is given in the Fig. 1.
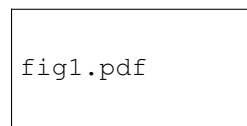
Figure 1: Students quiz score (input file)

- 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. Part of the roster is given by Fig. 2.

- 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.

- A hash table is defined as a vector of linked lists. The size of the hash table m is equal to the number of students in the roster. In case of a collision use the chaining method and place colliding elements in the same linked list selected by the hash function

$$h(x) = x \bmod m$$

  The chaining method on average is efficient if the number of elements from collisions in each linked lists is constant, $O(1)$, compared to all elements stored in the hash table.
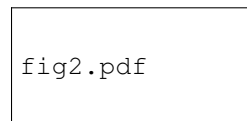
Figure 2: The roster file

- The implementation steps:

  1. (10 points) Read input.csv containing grades, see Fig. 1.
  2. (20 points) Use the Regex class to parse each row in input.csv.

2

- The regular expression are part of the C++ STL. To find more information about regular expression please read the chapter Dr. Bjarne Stroustrup's "*Programming. Principles and Practice Using C++*" the latest edition, Chapter 23, sections 23-6 through 23-9, "*Text Processing*" http://www.stroustrup.com/Programming/lecture-slides.html
- include the header file (`#include <regex>`)

3. (20 points) Create a hash table using student's UIN as a key and student's score as a value (key-value pair). Resolve collisions using the chaining method.

- Read the lecture notes and text book about Hashing, chap. 9.
- Display the statistics about the hash table: minimum, maximum, and average length of the linked lists in the hash table.

4. Read the roster.csv containing students grades (= class grading book).

5. Use the Regex to parse each row in roster.csv.

6. Look up the hash table by using the parsed UIN and recover the corresponding quiz score.

7. (30 points) Create a new file output.csv with appended scores, see Fig 3, see more details below:

   (a) read a line from roster.csv
   (b) extract UIN field and create the corresponding key
   (c) search the hash table using the key
      i. if the search is successful, update the line by appending the corresponding score
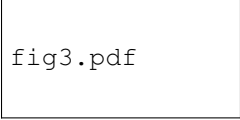   ii. if the search is unsuccessful, copy the roster line without any changes

Figure 3: The updated by quiz score student file (output file)

- Comment: the CSV file format is very important. You should learn what a plain-text CSV file is. Then, you can write the regex pattern string.

What to submit to CSNet?

- Your C++ source code with the header block including: your name, user name, section number and e-mail address

- (20 points) A report which should consists of the following parts:

   - The cover page.
   - Assignment number and its description.
   - Description of data structures and algorithms used by your program.
   - Description of input and output data. List all restrictions and assumptions that you have imposed on your input data and program.
   - How have you tested your program for corrections?
   - Which C++ features or standard library classes have you used in your program?
   - 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.
   - Write your conclusion.