CSCE 221 Assignment 4 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 4 (100 points)

Due: April 13th, 2020 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.

Echo Wells	accumsan.neque@acmetus.com	800000079	94
Candace Bishop	adipiscing@bibendum.com	800000057	49
Conan Daugherty	aliquam.arcu.Aliquam@nequenon.net	800000053	96
Emmanuel Chan	arcu.Vestibulum@Etiamvestibulummassa.com	800000062	41
Beatrice Morgan	augue@tempus.edu	800000055	47
Ruth Woods	augue@vitae.org	800000001	99
Ursula Dickson	congue.In@velsapien.com	800000051	60
Donna Dennis	consectetuer@turpis.net	800000021	64
Sydnee Holloway	cubilia.Curae.Phasellus@nibhdolor.net	800000003	66
Tarik Sargent	cubilia.Curae@vehiculaet.edu	800000009	73
Kiona Santana	Cum.sociis.natoque@tristiquealiquetPhasellus.co.uk	800000069	68
Travis House	Donec.at@risusMorbi.net	800000046	40
Basil Moon	Donec.nibh@Donecnonjusto.org	800000081	54
Barbara Walter	Donec@placeratvelitQuisque.edu	800000005	84
Echo Haynes	dui.in.sodales@eget.net	800000074	64
Keely Graham	dui@anteipsumprimis.co.uk	800000043	47
Taylor Roberson	Duis.mi.enim@aceleifendvitae.edu	800000052	72
Bertha Rodriguez	Duis.volutpat.nunc@uteros.ca	800000032	44
Deacon Richards	eget.dictum@eget.com	800000066	90
Mikayla Grant	eget.dictum@tincidunttempusrisus.edu	800000044	37
Tucker Mullen	eget.metus.eu@lacus.net	800000090	31
Ruth Marquez	eget@Fusce.org	800000070	89
Vance Everett	elit.pellentesque@bibendum.edu	800000050	77

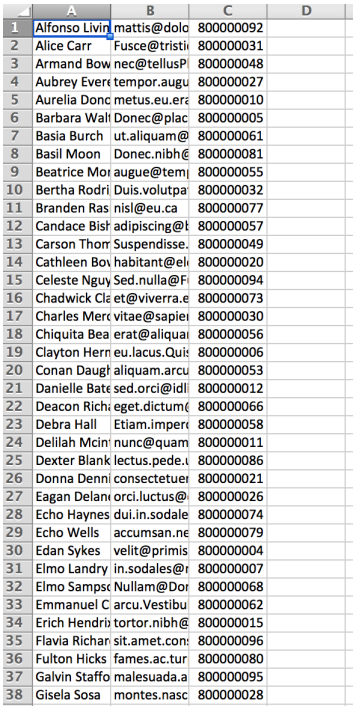
Figure 1: Students quiz score (input file)

- 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.

- The implementation steps:
  1. (10 points) Read input.csv containing grades, see Fig. 1.



	A	B	C	D
1	Alfonso Livin	mattis@dolo	800000092	
2	Alice Carr	Fusce@tristi	800000031	
3	Armand Bow	nec@tellusP	800000048	
4	Aubrey Evere	tempor.augu	800000027	
5	Aurelia Donc	metus.eu.era	800000010	
6	Barbara Walt	Donec@plac	800000005	
7	Basia Burch	ut.aliquam@	800000061	
8	Basil Moon	Donec.nibh@	800000081	
9	Beatrice Mor	augue@tem	800000055	
10	Bertha Rodri	Duis.volutpa	800000032	
11	Branden Ras	nisl@eu.ca	800000077	
12	Candace Bish	adipiscing@	800000057	
13	Carson Thorr	Suspendisse.	800000049	
14	Cathleen Bov	habitant@el	800000020	
15	Celeste Nguy	Sed.nulla@F	800000094	
16	Chadwick Cla	et@viverra.e	800000073	
17	Charles Merc	vitae@sapie	800000030	
18	Chiquita Bea	erat@aliqua	800000056	
19	Clayton Hern	eu.lacus.Qui	800000006	
20	Conan Daugh	aliquam.arcu	800000053	
21	Danielle Bate	sed.orci@idl	800000012	
22	Deacon Richa	eget.dictum@	800000066	
23	Debra Hall	Etiam.imper	800000058	
24	Delilah Mcin	nunc@quam	800000011	
25	Dexter Blank	lectus.pede.(	800000086	
26	Donna Denni	consectetuer	800000021	
27	Eagan Delan	orci.luctus@	800000026	
28	Echo Haynes	dui.in.sodale	800000074	
29	Echo Wells	accumsan.ne	800000079	
30	Edan Sykes	velit@primis	800000004	
31	Elmo Landry	in.sodales@r	800000007	
32	Elmo Sampso	Nullam@Dor	800000068	
33	Emmanuel C	arcu.Vestibu	800000062	
34	Erich Hendri	tortor.nibh@	800000015	
35	Flavia Richar	sit.amet.con	800000096	
36	Fulton Hicks	fames.ac.tur	800000080	
37	Galvin Staffo	malesuada.a	800000095	
38	Gisela Sosa	montes.nasc	800000028	

Figure 2: The roster file

2. (20 points) Use the Regex class to parse each row in input.csv.
   - 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. 5.
   - 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

- 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.

Alfonso Livin	mattis@dolo	800000092	
Alice Carr	Fusce@tristi	800000031	
Armand Bow	nec@tellusP	800000048	
Aubrey Evere	tempor.augu	800000027	
Aurelia Donc	metus.eu.era	800000010	
Barbara Wall	Donec@plac	800000005	84
Basia Burch	ut.aliquam@	800000061	
Basil Moon	Donec.nibh@	800000081	
Beatrice Mor	augue@tem	800000055	47
Bertha Rodri	Duis.volutpa	800000032	
Branden Ras	nisl@eu.ca	800000077	
Candace Bish	adipiscing@	800000057	49
Carson Thom	Suspendisse.	800000049	
Cathleen Bov	habitant@el	800000020	
Celeste Nguy	Sed.nulla@F	800000094	
Chadwick Cla	et@viverra.e	800000073	
Charles Merc	vitae@sapie	800000030	
Chiquita Bea	erat@aliqua	800000056	
Clayton Herr	eu.lacus.Quis	800000006	
Conan Daugh	aliquam.arcu	800000053	96
Danielle Bate	sed.orci@idl	800000012	
Deacon Richa	eget.dictum@	800000066	90
Debra Hall	Etiam.imper	800000058	
Delilah Mcin	nunc@quam	800000011	
Dexter Blank	lectus.pede.u	800000086	
Donna Denni	consectetuer	800000021	64
Eagan Delan	orci.luctus@	800000026	
Echo Haynes	dui.in.sodale	800000074	64
Echo Wells	accumsan.ne	800000079	94
Edan Sykes	velit@primis	800000004	
Elmo Landry	in.sodales@r	800000007	
Elmo Sampso	Nullam@Dor	800000068	
Emmanuel C	arcu.Vestibu	800000062	41
Erich Hendri	tortor.nibh@	800000015	
Flavia Richar	sit.amet.con	800000096	
Fulton Hicks	fames.ac.tur	800000080	
Galvin Staffo	malesuada.a	800000095	
Gisela Sosa	montes.nasc	800000028	

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

What to submit to eCampus?

- 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.