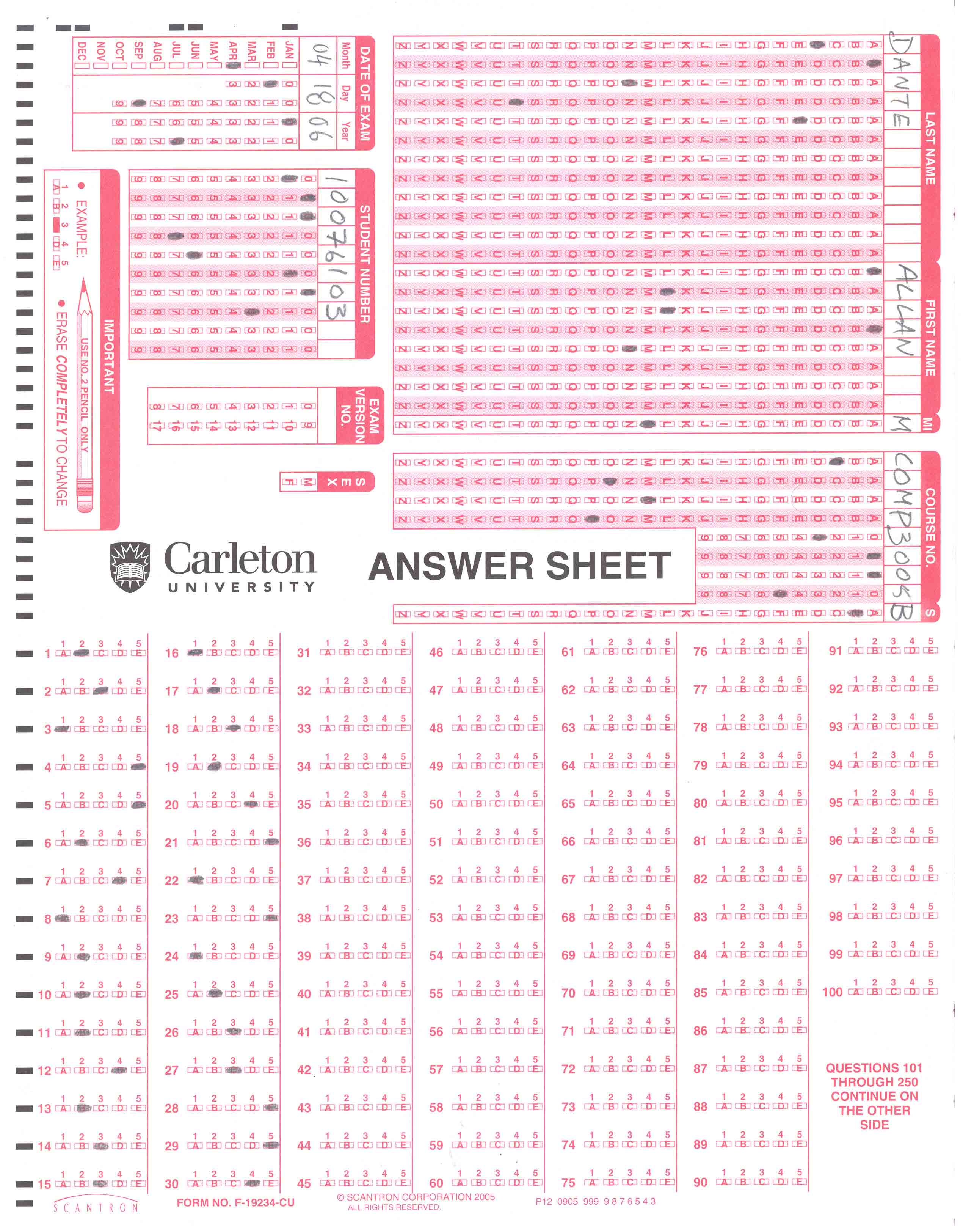
**Assignment #6** *(Mini Simulation Project)*





# *Problem Solving and Programming in C++*

# *Department of Computer Science*

# *Old Dominion University*

**Objectives:** The software testing process (*i.e.*, the **BBT**) consumes large amounts of data. Creating the test data is a crucial part of the testing process especially for new systems which are not in use yet. This assignment will give you an opportunity to practice how to create a good representative test data. After completing this assignment, students will be able to:

* Create random data from a given set of parameters
* Create a large number of test cases to test all possibilities
* Write a C++ program to **simulate** the functionality of a real machine and create a simulation data

This assignment will be submitted into two parts (*Part-A* and *Part-B*) – a separate Code::Blocks project is required for each part.

**Problem description**: You need to write a C++ program to help your professor marking the midterm for students in the CS-250 class at ODU. There are 5000 students writing the midterm for this class on scantron sheets. The exam includes 100 different questions. Questions number 5, 10, 15 and 25 are true/false, while all other questions are multiple-choice. The possible answers for the multiple choice questions are ‘A’, ‘B’, ‘C’, while the answers for the true/false questions are ‘A’, ‘B’.

Your task is to write a C++ program to simulate the machine which scans the students’ answer sheets and marks the exam. The machine starts by scanning the answer sheets and recording the answers into a file named “studentsAnswer.txt”. This file does not exist; you will need to create this file. The machine also records the information (*ID, name, ....)* and answers of each student in a separate line (*one-record*) into this file. To accomplish this you must implement the following functionalities of Part-A and Part-B:

***Part-A***: (*70 marks*)

Write a C++ program to generate the students data file “studentsAnswer.txt”. Each line in this file represents a student’s record which includes the:

className, studentID, lastName, firstName, studentAnswers

As shown below for one student:

CS-250, 10853, John, Adam, ACBCACCABBACABACBC AABBAB

The record above is just an example of one students, remember there are 100 questions and 5000 students. Please note that the student ID is a unique five digit number. You should use a *random number generator* to generate a unique ID for each student taking the midterm exam. Students do not have to answer all questions. For example, the student above did not provide an answer for question number **19** (*given that the total number of the exam questions is* 25). You must randomly generate the first-name, last-name and answers of each student. Please make sure that some students might leave some question(s) blank without answer(s). This should occur about 1% of the time.

**Important notes** about the studentsAnswer.txt file – which your C++ program will create:

1. All data records in the studentsAnswer.txt file must be randomly generated for all students (starting from student #2 to student #5000)
2. The data of the first student in the studentsAnswer.txt file must be created differently, as follow:
   1. The answers of the first student is copied from the firstStudentAnswersEntry.txt file
   2. The rest of the first student’s information is created randomly – *like all other students*

**Submission notes for Part-A:**

* Zip the entire Code::Blocks project containing all the .cpp, .h, .cbp files and name the zipped file “Assg6A\_cslogin.zip”, where the cslogin is your login ID for the computers at the Department of Computer Science at ODU
* You do not have to submit the txt file (studentsAnswer.txt) with your program for Part-A. This file will be deleted when the graders test your program.
* Submit the zipped file using the appropriate Blackboard link.

***Part-B***: (*30 marks*)

In this portion of the assignment, you will need to extend you program from part-A to grade the exam. Start a new Code::Blocks project for Part-B. Your program should read the data from the file “studentsAnswer.txt” – which you created in Part-A. Please make sure to read the data into the appropriate data structure (struct). Next start grading the exam according to the following rules:

A student receives **70** points plus:

* **2** points for a correct answer
* **0** for a blank answer
* A penalty of **-1** point for a wrong answer
* Students who submit more than 5 blank answers will receive an additional penalty of **-2** points
* The final score is scaled to the 100 point scale by:

totalStudentPoints = (totalStudentPoints / maximumPossiblePoints) \* 100;

You must initialize totalStudentPointsand maximumPossiblePointsproperly

The initial 70 points is to compensate for fact that since the answers are generated randomly many student will fail. The 70 points are added to introduce more variation to pass and fail.

When the professor submits the midterm-exam for automatic grading, he/she submits an answer key into a file named “answerKey.txt”. This file includes two lines:

* The first line in the “answerKey.txt” file is the answer key
* The second line is the name of the selected marking schema (***easy***, ***medium***, or ***hard***)

The two files (“answerKey.txt” and “markingScehma.txt”) are provided under the supporting files for this assignment on Blackboard. The letter grade for students is calculated based on the selected marking schema. Your program should save the results of grading into a text file named “results.txt”. This file should include records (*one record per student*). Each record in this file should include the:

class-name, studentsID, lastName, firstName, answers, letterGrade, numberOfCorrectAnswers, numberOfFalseAnswers, and numberOfBlanks, totalPoints ======> all separated by a comma

At the end of your results file, you should provide some **statistics** to help your professor analyze the exam results. Your statistics should indicate the follow information:

Number students attended the exam: ???

Number students received A: ???

Number students received B: ???

Number students received C: ???

Number students received D: ???

Number students received F: ???

**Submission notes for Part-B:**

* Zip the entire Code::Blocks project containing all the .cpp, .h, .cbp files along with the txt file studentsAnswer.txt and name the zipped file “Assg6B\_cslogin.zip”, where the cslogin is your login ID for the computers at the Department of Computer Science at ODU
* You must submit the txt file (studentsAnswer.txt) with your program for Part-B. This file is required for grading your assignment.
* Submit the zipped file using the appropriate Blackboard link.

**Sample Output:**

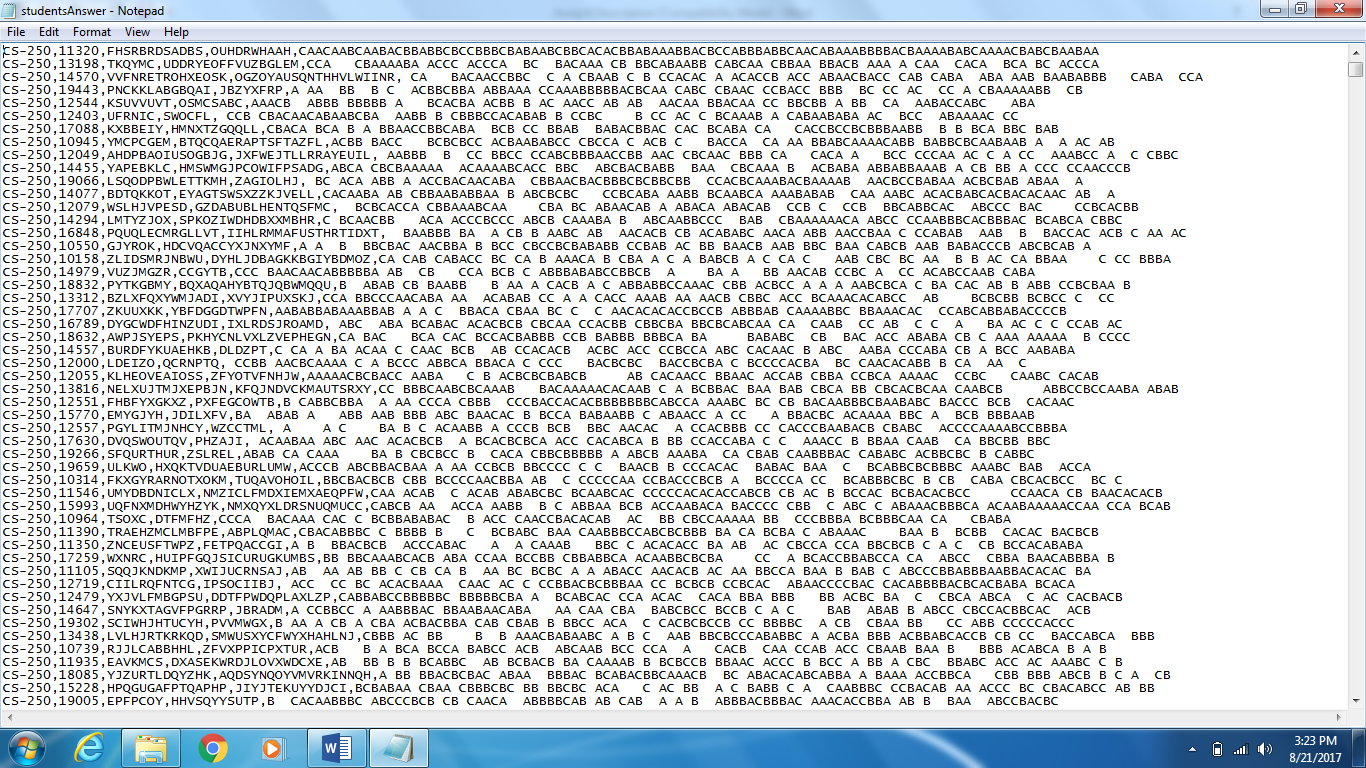


Figure 1: studentsAnswer.txt

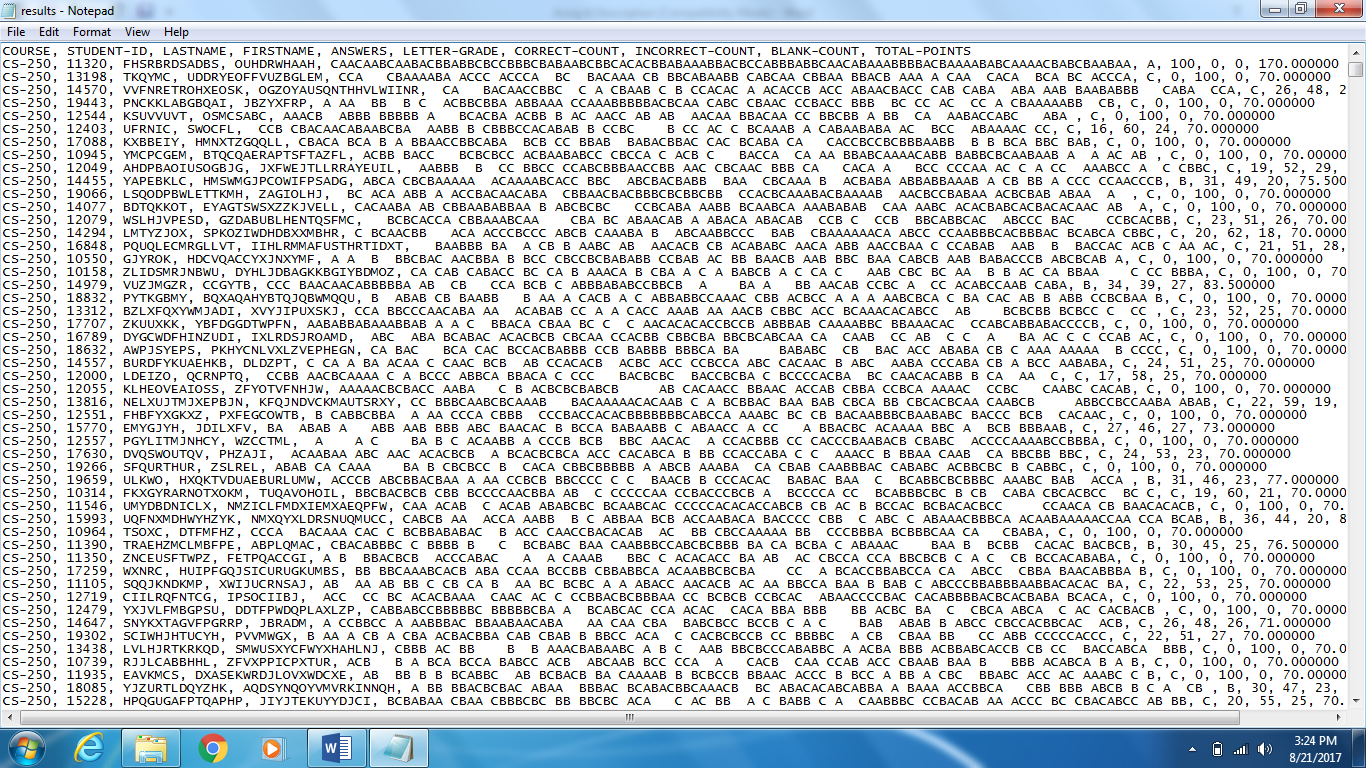


Figure 2: results.txt

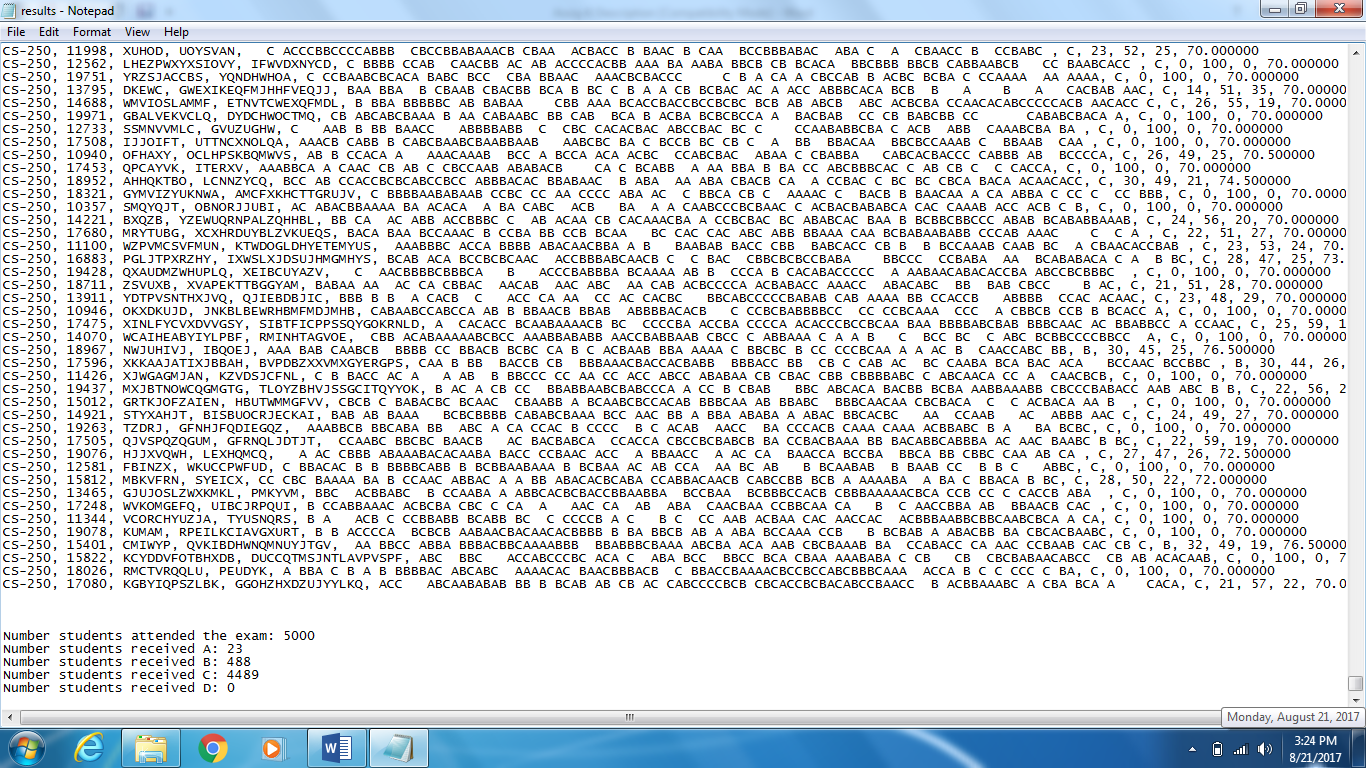


Figure 3: results.txt continue