CPSC 1375 Programming I HW6

(Scoring students on a multiple-choice exam)

This exercise is designed to introduce you to arrays and strings (i.e., arrays of characters) in C++.

Suppose a class of students takes a 20-question multiple-choice exam; each question has five choices (A,B,C,D,E), only one of which is correct. For simplicity, we assume that every student attempts to answer every question, i.e., there are no "blank" responses. The students' responses are put in a data file "tests.dat", each record of which consists of a student id (a string eleven characters long), followed by a blank, followed by that student's 20 responses (a string twenty characters long). Thus a typical record might look like:

120-49-5322 CDEDBAABECCDAAEBADDE

There is a second file called "answers.dat" which consists of a single record, namely the correct answers to the exam (a string twenty characters long). Thus, this record might look like:

CBEDBEBAECDDAEEBDEDE

in which case the above student has answered question 1 correctly, but blown question 2, etc.

Your job, as teaching assistant for the course, is to write a C++ program to read in the correct answers from the "answers.dat" file and then loop through the "tests.dat" file and produce the following GRADE SUMMARY REPORT. (The example below illustrates a class of size 23; of course, you should count as you read the student response file.)

(see next page)

student-id	number correct
430-52-6192	13
112-81-5225	19
120-49-5322	12
456-65-3211	2
990-45-0978	4
324-98-5445	6
112-34-3443	12
435-56-8790	18
452-23-5675	6
776-45-5454	7
567-71-9909	6
345-54-7834	9
555-12-2341	18
568-09-0096	20
664-61-0987	10
612-45-7687	5
567-23-1125	3
561-43-6781	6
498-12-4321	3
479-01-4867	4
489-90-0999	17
456-09-1111	19
561-66-6657	3

<------ same order as in the "tests.dat" file, i.e., no rearranging necessary

number of students taking exam = 23

question	A	B	C	D	E
1	5	1	13*	3	1
2	4	7*	1	7	4
3	2	3	2	2	14*
4	2	1	3	13*	4
5	2	8*	1	4	8
6	5	1	3	3	11*
7	5	7*	0	4	7
8	11*	4	0	3	5
9	3	2	1	2	15*
10	0	1	12*	3	7
11	6	2	4	10*	1
12	2	2	4	9*	6
13	9*	4	0	6	4
14	4	2	2	2	13*
15	3	3	1	1	15*
16	1	8*	4	3	7
17	6	3	1	7*	6
18	4	1	2	5	11*
19	0	1	1	10*	11
20	2	0	1	1	19*

<------ total number of students who chose each choice, by question; print asterisks next to the correct answers!!

(continued on next page)

Comments and programming hints:

(1). You will, of course, have to "hook up" and open two data files in this assignment!
(2). Please remember that arrays in C++ begin at position zero, not one.
(3). You'll need a 20 x 5 two-dimensional array of counters to build the "distribution chart" part of the output. Also, please recall that C++ does allow you to do arithmetic with characters, so it is easy to convert an 'A' to 0, 'B' to 1, 'C' to 2, etc.
(4). Try to "neatly arrange" the grade summary report, so that it reasonably resembles the illustration shown.
(5). Big hint on asterisk-printing: It is easy to decide whether or not you need to print an asterisk after an entry in the distribution chart; just check to see if this particular choice is the correct one! But asterisk-printing will throw off your column alignments; to ensure that it doesn't, you will need to print an asterisk ("*") if the choice is correct and a blank space (" ") if it isn't.