

Problem

Create a program to score results from a multiple-choice test, allowing partial credit for selected answers. The program, (mandatory name: `grader`), will read the student ID numbers and their answers from the input file, ("`anskey.dat`"), followed by the test key. The program will then process the answers given by each student and determine two scores, one without partial credit and one with partial credit. The program will then produce a report file that includes the following information:

- ID number and both scores for each student.
- Averages of both sets of scores.
- A histogram showing how many students scored in each decile.

This information must be clearly labeled and formatted to be both readable and attractive.

Input

The first line of the input file, ("`anskey.dat`"), contains two integers, separated by whitespace. The first integer specifies the number of students for whom data is given and the second specifies the number of questions given. Following the first line, the next `<number of students>` lines of the input file has the following format:

- A student ID number (9 contiguous digits).
- Two columns that should be disregarded.
- The responses provided by that student to the questions. Responses are expressed as single digits (0-9) with no intervening spaces. Note that a student may fail to respond to a question, indicated by a blank (space). It is not possible for a student to provide any response other than a single decimal digit or a blank.

Lines may have trailing whitespace (blanks and/or tabs) preceding the newline character. Each input line (except the first) will contain at least `M` characters before the newline, where $M = (9 + 2 + \text{number of questions})$. See the accompanying sample input file.

Following the last line of student data, there is an answer number header line that can be ignored. This line gives column headings 0-9 for the corresponding answer responses. The next `<number of questions>` key lines have the following format:

- Question number
- Character indicating the value of each possible response (10 in all). These characters will be separated by spaces and/or tabs. Possible characters and the corresponding value are as follows:

Character	Value
a	100%
b	70%
c	40%
d	20%
x	0%

There are some restrictions on the size of the input:

- There will be no more than 50 students.
- There will be no more than 50 questions.
- Each question will have at most 10 possible responses (0-9).

Each student response should be scored according to the table given above. Final scores, whether partial credit or not, should be reported as percentages, (out of 100), regardless of the number of questions asked. Scores and averages should be reported to two places after the decimal.

Coding requirements:

- Absolutely no file-scoped variables are allowed! File-scoped constants and type declarations are acceptable.
- Use constants instead of variables where appropriate.
- You must pass parameters by reference only when the called function needs to modify the value of the parameter. Note this applies to array parameters as well (pass by constant reference).
- You must implement at least four functions.
- Not counting declarations and comments, the body of `main()` must contain no more than 20 lines.

Grading

Due Dates: Submit: Monday Oct. 27 Hardcopy: Tues. Oct. 28

Two grades will be given for this assignment. The first will be given by the automatic grading system which will assign a grade based on the output. (See the course Web site: <http://ei.cs.vt.edu/~cs1344/> for more details.)

The second grade will assigned by the GTA. It will be based upon design, documentation and coding style. Turn in hard copies of the source code, output file and a top-down hierarchical design tree of the program. In addition, submit a diskette with files containing: all project workspace files: (project file, source code, etc.), executable image, (.exe), I/O files, and an ASCII, (text), *readme* file with execution instructions. The source code file must be named grader.cpp and the executable image named grader.exe. The disk should contain no subdirectories. All materials to be graded must be placed in a sealed 9" x 12" folder, neatly labeled with your name, course number and date.

In addition, the GTA may require you to demonstrate your program. To receive partial credit for programs that are non-working, or are not fully functional, a brief one or two paragraph description of the problem(s) must be included in the assignment folder. The location, routine minimum, must also be specified along with possible corrections that need to be made.

Sample Input											Sample Output		
15	20										ID Number	Without Partial	With Partial
548932001	21255124001011000111										=====		
229081368	21550166001011000111										548932001	85.00	88.50
593072704	212544170020 1202111										229081368	95.00	95.00
227131902	21250113001011000112										593072704	55.00	65.50
230044180	21550163001011000111										227131902	85.00	88.50
224235019	21251113002021202										230044180	100.00	100.00
244577135	21550063001021000111										224235019	50.00	62.50
368781746	21250013001011000111										244577135	90.00	90.00
231135020	21210013001021202121										368781746	85.00	88.50
219821682	21550163001011000111										231135020	60.00	67.00
227253655	21250163001011000111										219821682	100.00	100.00
426232843	21551164001011000141										227253655	95.00	95.00
027158513	55550413001011000111										426232843	90.00	92.00
006049924	21550163001011010111										027158513	80.00	83.50
231454006	5510156001011101111										006049924	95.00	95.00
0	1	2	3	4	5	6	7	8	9		231454006	65.00	65.00
1	x	x	a	x	x	x	x	x	x		=====		
2	c	a	c	x	x	x	x	x	x		Average	82.00	85.07
3	x	x	x	x	x	a	x	x	x				
4	x	x	x	x	x	a	x	x	x				
5	a	c	x	x	x	x	x	x	x				
6	x	a	x	x	x	x	x	x	x				
7	x	b	b	x	x	x	a	x	x				
8	x	x	c	a	a	c	x	x	x				
9	a	x	c	c	x	x	x	x	x				
10	a	x	x	a	x	x	x	x	a				
11	x	a	b	c	d	x	x	x	x				
12	a	x	x	x	x	x	x	x	x				
13	x	a	x	x	x	x	x	x	x				
14	x	a	x	x	x	x	x	x	x				
15	a	x	b	x	x	x	x	x	x				
16	a	x	x	x	x	x	x	x	x				
17	a	x	x	x	x	x	x	x	x				
18	x	a	x	a	x	x	x	x	x				
19	x	a	x	x	x	x	x	x	x				
20	x	a	x	x	x	x	x	x	x				