Due: Wednesday April 6, beginning of class

Description:

Professor Proton teaches an online *Science Wow!* class aimed at middle school students (grades 6 and 7 in the States). Her goal is to introduce the student to general science concepts and the scientific method in a way that is fun and engaging.

Professor Proton gives short online quizzes to see how well the students are doing. While she wants a big picture view (how many questions were answered correctly), she also wants to keep track of which questions students find the most difficult so she can adjust her presentations.

To that end she has asked *B*arking *D*ogs *S*oftware to create an application that will grade a multiple-choice quiz and provide information on how many students missed each question.

The number of questions on a quiz can change from week to week. The quiz results are sent to Prof. P in a text file named `quizdata.txt`. The first line of the file contains the number of questions on the quiz followed by a space and then an *n*-character string of the correct answers (the quiz answer key). Each line that follows the key contains a three-digit integer student ID followed by a space and then that student's answers. There is an example below.

Professor Proton asks that the program produce an output file, <last name>Report.txt (for example, I would generate `GrayReport.txt`), containing your name at the top, the answer key, each student's ID and each student's score as a percentage, and then information about how many students missed each quiz question. The output file should be nicely formatted. There is an example below. Dave, lead Minion on the project, has done some preliminary design and determined that the solution ***must*** have two functions.

> `getAnswers` reads the answers (key or student answers) from an open input file into an array. This function takes three parameters: a FILE* variable, the # of responses to read in, and an answers array.
>
> `gradeQuiz` grades a single instance of a quiz, returning the number of correct answers and updating which questions were missed. This function takes four parameters: a key, a set of answers to grade, the questions missed array, and the number of entries to process (e.g., the number of quiz questions).
>
> ***Additionally***, you ***must*** use dynamic memory allocation to create the memory for the arrays you use.

Here is a sample input file and the corresponding output.

Sample input file: quizdata.txt ← input *must* come from this file. Grr!

```
5 dbbac
111 dabac
102 dcbdc
251 dbbac
```

Sample output file: GrayReport.txt

```
Simon Gray

      Quiz Report
Question 1  2  3  4  5
Answer   d  b  b  a  c

ID     Score(%)
111      80
102      60
251     100

Question  1  2  3  4  5
Missed by 0  2  0  1  0
```

Submit:

Your file should be named as follows: Lastname_p3.c. For example, I would send **G**ray_p3.c. Email me your C source file to me. You do not have to submit a design. Obviously you should do one!