**CS120L – C Programming Language Labs**
Fall 2014
**Week 8: Loops, Conditionals & Simple Problem Solving**

In this lab, you are only allowed to use functions from **stdio.h**, and **fabs** function from **math.h**. You are **not allowed** to use any other libraries.

You are also **not allowed** to use <u>arrays</u> in this lab.

For each task, 20 marks will be deducted if you violate any of the rules above.

# Task 1

In this task, you need to read in a list of students and their grades, and then print out the number of passed and failed students, together with the average grade for each group. In this task, the passing grade is 5, which means that any grades that are less than 5 are failing grades.

**Input**: the student ID numbers and the corresponding grades from user's inputs, where each line consists of a non-negative integer student ID and his/her grade, ranging from 0 to 10. The list of students ends when the user enters 0  0 (0  0 is not counted into that student list).

**Output**: The first line contains the number of passed students and their average grade. The second line contains the number of failed students and their average. The grades should have 2 decimal places.

For example, if the input is:

```
027358 7.5
326017 6
834114 2
235124 4.0
195830 3.5
983582 8
198045 6.5
0 0
```

Then the output will be:

```
4 7.00↵
3 3.17↵
```

 (↵ indicates the end-of-line character)

To check the equality of two floating point numbers, we cannot use the comparison operator "==" due to the inherent rounding errors. Instead, we should check the absolute difference of the two numbers (e.g. fabs(a-b)). If the absolute difference is less than some threshold, we deem that these two numbers are the same; otherwise, they are different. In this lab, let's make the threshold 0.00001.

# Task 2

In lab 5, you have found the $n^{th}$ digit of an integer number. A similar but slightly harder problem is to find the $n^{th}$ digit after the decimal point of a real number. For instance, the 4th digit after the decimal point of 12.34567 is 6 (the 1st digit is the leftmost one after the decimal point).

**Input**: a real number and the positive index n of the desired digit.
**Output**: the $n^{th}$ digit after the decimal point of that number.

Here are some sample test cases:
```
-12.34567 4
6↵

31.98 5
0↵

62.00005 2
0↵
```

## Command line for compiling

```
/usr/bin/gcc -Wall -Wextra -ansi -pedantic q1(2).c -o q1(2).exe
```

## Grading scheme

- 100 marks in total (50 marks each question).

- 10 marks for successful build.

- 10 marks for successful execution and correct output (2 marks * 5 hidden test cases).

  Note that your program should only print out the required output, ending immediately with an end-of-line character \n, and not any redundant character.

- 20 marks for correct implementation.

- 10 marks for good programming style. For the complete guide, please see the Assignment Guideline on CS120 page.

  In general, you should have clear variable names, reasonable comments to explain your code, and consistent indentation. Please use **2-blank spaces** instead of tab spaces.

## Output formatting

You should use the given sample input and output files to check your output format before submission. After compiling your program into **a.exe**, you can generate your output using:

```
./a.exe < q1.in > result.out
```

Then compare it with the sample output by typing:

```
diff result.out q2.out
```

You need to make sure that `diff` does not print out anything. Otherwise, your output is considered wrong and you will lose all the 10 marks for output correctness.


## Submission

Please name your source code for question 1 as **q1.c**, question 2 as **q2.c**, put them in a folder named **cs120&lt;session&gt;_&lt;your Digipen login id&gt;_&lt;labnumber&gt;**, in which &lt;**labnumber**&gt; is 8 this week and &lt;session&gt; is either a or b, and zip them in **cs120&lt;session&gt;_&lt;your Digipen login id&gt;_&lt;labnumber&gt;.zip** for submission.

Wrong submission file/folder name will cause 10 marks deducted.

Note that the file and folder names must be **lowercase**.

The deadline of submission is 24th October 23:59 and late submission will receive zero mark.