Important message on plagiarism

The single most important point for you to realize before the beginning of your studies at ShanghaiTech is the meaning of "plagiarism":

Plagiarism is the practice of taking someone else's work or ideas and passing them off as one's own. It is the misrepresentation of the work of another as your own. It is academic theft; a serious infraction of a University honor code, and the latter is your responsibility to uphold. Instances of plagiarism or any other cheating will be reported to the university leadership, and will have serious consequences. Avoiding any form of plagiarism is in your own interest. If you plagiarize and it is unveiled at a later stage only, it will not only reflect badly on the university, but also on your image/career opportunities.

Plagiarism is academic misconduct, and we take it very serious at ShanghaiTech. In the past we have had lots of problems related to plagiarism especially with newly arriving students, so it is important to get this right upfront:

You may…

• … discuss with your peers about course material.
• … discuss generally about the programming language, some features, or abstract lines of code. As long as it is not directly related to any homework, but formulated in a general, abstract way, such discussion is acceptable.
• … share test cases with each other.
• … help each other with setting up the development environment etc.

You may not …

• … read, possess, copy or submit the solution code of anyone else (including people outside this course or university)!
• … receive direct help from someone else (i.e. a direct communication of some lines of code, no matter if it is visual, verbal, or written)!
• … give direct help to someone else. Helping one of your peers by letting him read your code or communicating even just part of the solution in written or in verbal form will have equal consequences.
• … gain access to another one's account, no matter if with or without permission.
• … give your account access to another student. It is your responsibility to keep your account safe, always log out, and choose a safe password. Do not just share access to your computer with other students without prior lock--out and disabling of automatic login functionality. Do not just leave your computer on without a lock even if it is just for the sake of a 5--minute break.
• … work in teams. You may meet to discuss generally about the material, but any work on the homework is to be done individually and in privacy. Remember, you may not allow anyone to even just read your source code.

With the Internet, "paste", and "share" are easy operations. Don't think that it is easy to hide and that we will not find you, we have just as easy to use, fully automatic and intelligent tools that will identify any potential cases of plagiarism. And do not think that being the original author will make any difference. Sharing an original solution with others is just as unethical as using someone else's work.

CS100 Homework 1 (Fall, 2019)

In this homework, you are required to do some simple programs with C programming language, starting to get familiar with it while solving some simple problems. This homework tries to pave the way for your later homework and projects.

Percentage of this homework over the whole score: **7%**

Submission deadline:

2019-09-24 23:59

Problem 1: Searching maximal and minimal numbers (30%)

In this problem, you are required to write a C program to search for maximal and minimal values in a series of floating-point data. Here you would like to input n values. You can update some variables as you read in new values. Finally, you will output the minimal and maximal values of all your inputs. The following is a demo process for the program if you can do it successfully: (red indicates input)

```
Please type number of inputs: 5
3.2
1.6
7.8
8.3
5.1
Max value: 8.3
Min value: 1.6
```

Input description:

The first number is an integer **n**, which represents how many float point numbers the user wants to input. The following **n** inputs are **n floats with at most 1 decimal digit**, which are the **n numbers** you want to calculate the **min** and **max**.

Note that it is not certain how many inputs there are, so we recommend not using an array.

Output description:

First, your program should output a prompt **Please type number of inputs:** .

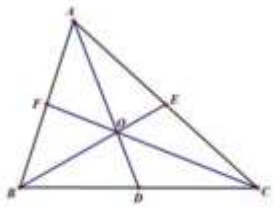
Then, after the user input **n** and **n numbers**, your program should output two lines, the first line is **Max value:** , and the maximum value of the inputs followed. The second line is **Min value:** , and the minimum value of the inputs followed.

*** All of your outputs should be printed with "printf" to only 1 decimal point digit floating point number, even if a number is an integer.**

*** The value of inputs are restricted to be between -10000.0 and 10000.0.**

Problem 2: Calculating the geometric center of a 2D triangle (30%)

In this problem, you are required to write a C program in order to calculate the geometric center of any given 2D triangle, see the following figure:



Note that we do not require you to draw such an image; only necessary parameters for the geometric center are required to be output. Thus, you will input three 2D vertex coordinates, and then output the geometric center of this triangle as well as the maximum and minimum distance from this geometric center to the 3 given vertices.

Note that there are multiple methods to compute the center coordinates. The simplest method is,

$$center_x = \frac{x1 + x2 + x3}{3}$$

$$center_y = \frac{y1 + y2 + y3}{3}$$

where x1, x2, x3 are x coordinates of three inputs; y1, y2, y3 are y coordinates of three inputs;

And the distance between two points can be calculated as

$$d(A\ to\ B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

The following is a demo process for the program if you can do it successfully: (red indicates input)

```
Input vertex 1: 0.0 1.0
Input vertex 2: -1.0 0.0
Input vertex 3: 1.0 0.0
The center of the triangle is: 0.0 0.3
The maximum distance between center and vertices: 1.05
The minimum distance between center and vertices: 0.67
```

Input description:

The input contains **3 lines**, for each line, it contains **two numbers**, indicating the **x**

coordinate and **y coordinate** for a vertex in the triangle.

*** Note that we assume all inputs are valid! (It means you do not need to consider the illegal cases)**

Output description:

1. output a prompt **Input vertex 1:** , then, receive two inputs.
2. output a prompt **Input vertex 2:** , then, receive two inputs.
3. output a prompt **Input vertex 3:** , then, receive two inputs.
4. output a prompt **The center of the triangle is:** , and followed by the values of **center_x** and **center_y**.
5. output a prompt **The maximum distance between center and vertices:** , and then the maximum distance between the center and the vertices.
6. output a prompt **The minimum distance between center and vertices:** , and then the minimum distance between the center and vertices.

*** Center coordinates should be printed with "printf" to display only 1 decimal point digit, and distances should be printed with "printf" to display only 2 decimal point digit.**
*** The values of coordinates are between -10000.0 and 10000.0.**

Problem 3: Whether you have a bad mood in a day? (40%)

As you have all experienced, before your first semester starts, there are many lectures (like school introduction, fire safety and such) for new students, and they are MANDATORY.

Of course, the contents of those lectures are very beneficial to new students, but there are always some who don't like the lectures, and here begins our story of Gezi Wang, a lazy freshman.

Gezi Wang's day begins when he wakes up at **8:00** with mood level **100**. 100 is also the maximum limit of his mood level.

During his day, several lectures will take place. Gezi Wang has to attend these lectures, but his mood level will drop.

> **For the first 60 minutes of a lecture, his mood level drops by 0.4/minute.**
> **After 60 minutes, his mood level drops by 0.8/minute.**

If his mood drops to or below 0, we will assume he has to be sent to the hospital.

When Gezi Wang does not have a lecture, he immediately returns to his dorm and sleeps.

> **His mood recovers by 0.5/minute in this situation.**

His mood level will not exceed the maximum limit 100.

His day ends at **22:00**. Your goal is to find out his mood level at the end of the day.

Input description:

You will receive **an integer** in the first line, indicating the number of lectures today.

In each of the following line, you will receive **two 24-hour times**, separated by a hyphen (-). (**Sample: 11:00-13:00**) This indicates the time period when a lecture starts and ends.

It's guaranteed that the inputs are as valid as you can imagine, specifically that:

1. There is no lecture before 8:00 and after 22:00.
2. The lectures will be given in time order. In other words, after a line of "14:00-14:30", you won't receive a time period such as "9:00-10:00".
3. The end time of a lecture is always later than the beginning time.
4. No lectures overlap, and there are always breaks between lectures. (And no matter how short a break is, he sleeps!)

Note that it is not certain how many lectures there are, so we recommend not using an array.

Output description:

Your output will be one line at the beginning and one line in the end.

At the beginning, you should write:

"**How many lectures today?**", followed by a new line(\n).

After all inputs, you must give your output in the following format:

If Gezi Wang is not sent to hospital during his day, you should write:

His mood level is * at the end of the day.**

*** The mood level should be printed with "printf" to display only 1 decimal point digit, even if it is an integer.**

Otherwise, you should write:

Gezi Wang has been sent to hospital.

The following is a demo process of the program: (red indicates input)

```
How many lectures today?
3
9:00-11:00
15:00-17:00
17:30-18:50
His mood level is 98.0 at the end of the day.
```

Hint: As you need to calculate time differences repeatedly, you might want to write a function, which will possibly have time as parameters and return the difference.