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(Spring, 2022)

Deadline: 2022-03-02 23:59:59

Late submission will open for 24 hours after the deadline, with -50% point deduction.

Problem 1. Find Bugs!

Your friend has just started learning programming, and has spent hours on a textbook practice problem. He/She asks you for help, and sends you this piece of code:

```
int main()
{
    int a; b; c; // Store the number of students in classes A, B, and C.

    printf("How many students are in classes A, B, and C, respectively?/n");
    scanf("d% d% d%", a, b, c);

    if(a > b & a > c)
        int max = a;
        printf("Class A has the most students, ");
    if(b > a & b > c)
        int max = b;
        printf("Class B has the most students, ");
    if(c > a & c > b)
        int max = c;
        printf("Class C has the most students, ");

    int total = a + b + c;

    // A class can accept at most half of all students.
    if(max > total / 2)
        printf("and is overly enrolled!/n");
    if(max = total / 2)
        printf("and is full!/n");
```

```
25      if(max < total / 2)
26          printf("and still has open seats!/n");
27
28      return 0;
29  }
```

"The problem says, if more than one class share the same most students, your program should output **the one whose name appears first in alphabetical order**, but I don't understand what that means." said your friend pathetically.

Please help your friend debug by identifying and correcting all his/her silly mistakes. You may first fix compile errors so that the code runs, and then figure out why it (possibly) does not produce correct results.

Oh, your friend has just sent you sample inputs and outputs on OJ!

Sample inputs (black) and outputs (red):

(1):

How many students are in classes A, B, and C, respectively?

20 20 17

Class A has the most students, and still has open seats!

(2):

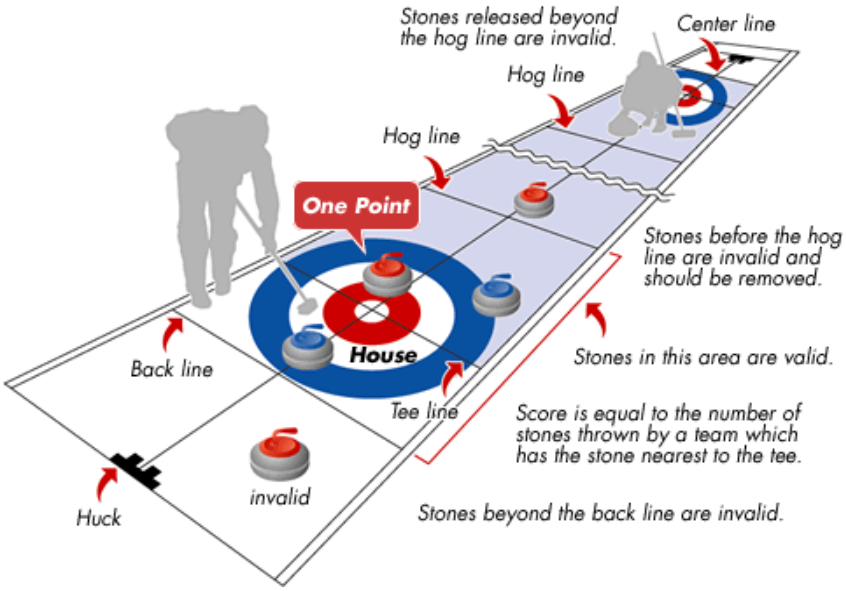
How many students are in classes A, B, and C, respectively?

10 24 14

Class B has the most students, and is full!

Problem 2. Curling

Curling is one of the most exciting and interesting events in Beijing 2022 Winter Olympics. Curling is a sport in which players slide stones on a sheet of ice toward a target area. Two teams take turns sliding stones (rocks) across the ice curling sheet toward the *house* (a circular target marked on the ice). The goal is to accumulate the highest score for a game. Points are scored for the stones resting closest to the center of the house after both teams have thrown all of the stones. You can have a look at the following figure for basic rules. More details are on https://en.wikipedia.org/wiki/Curling



The players can use brooms or brushes to sweep the ice in front of the stone to influence the path. "Sweeping a stone" can decrease the friction, which makes the stone travel a straighter path (with less curl) and a longer distance. A great deal of strategy and teamwork go into choosing the ideal path and placement of a stone for each situation, and the skills of the curlers determine the degree to which the stone will achieve the desired result.

Part A. How far does the stone slide?

In this part, you are going to help curling players to calculate the **sliding distance** of the stone. After the thrower has thrown the stone, the sweepers may use brooms to sweep the ice in front of the stone to make it faster. When you watch a curling game, you can always hear their yelling. The yelling is a effective way to exchange information between thrower and sweeper. Some common yellings are as follows:

- "HURRY": Sweep as fast as possible
- "WHOA": Stop sweeping

Given the time of "HURRY" and "WHOA", you are able to calculate the sliding distance. To simply the situation, you can assume the stone slides **in a straight line** with no rotations. Only the fraction will slow down the stone. The initial velocity of the stone is $3m/s$. **The acceleration of the stone is**

$-0.2m/s^2$ **without sweeping and** $-0.1m/s^2$ **with sweeping.** The sweepers are not sweeping in the beginning, and will immediately start/stop sweeping after hearing a yelling.

Input description:

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

In each of the following lines, you will receive a **floating point number** and a **character**, separated by a hyphen(-). The floating point number indicates the seconds after throw. The character is the content of yelling where "H" stands for "HURRY" and "W" stands for "WHOA". It is guaranteed that all yellings are given in chronological order.

Output description:

You need to output the sliding distance as follows:

"**The stone has slided for ***m**" with a new line(\n).

Note that the distance should be displayed with 3 decimal point digit.

Sample input:

2

1.00 - H

2.00 - W

Sample output:

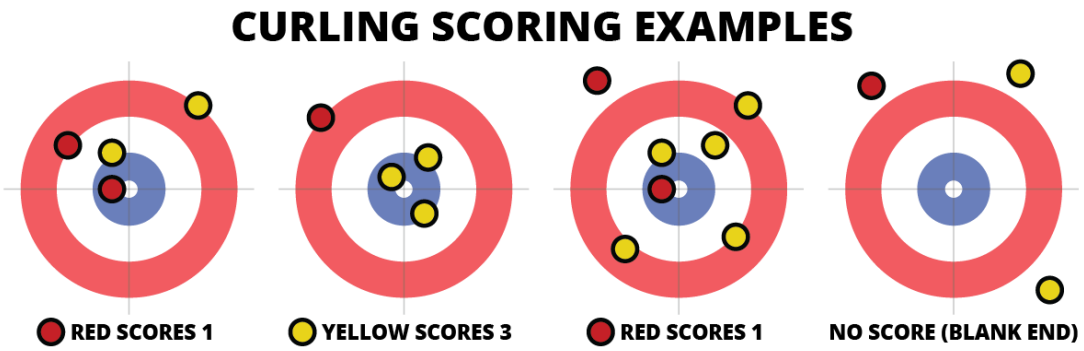
The stone has slided for 23.875m

Part B. Who wins?

A complete curling game consists of ten ends (individual matches). Points are scored for each end, according to the following rules:

- In one end, only the winning team scores points. The losing team would not get points.
- Any stone that is not touching any part of the house could not earn points.
- The team with the **stone closest to the button** will win an end, and receive **one point** for that stone.
- The winning team will earn **additional points**, **one point** for every other stone closer to the button than their opponent's **closest stone**.
- Neither of the teams will score if there are **no stones in the house**.

Some examples are shown in the following figure.



In this part, you can regard each stone as a mass point. The location of stones are represented in Cartesian coordinates. The center of the house (button) is set to be the origin. The location of each stone is given by coordinate (x, y).

Assume the radius of the house is **1.8m**. Both teams have 3 stones. Can you figure out the winner and the points they earn?

Input description:

You will receive 6 lines of input. The first 3 lines are the location of stones for **red** team and the next 3 lines are for yellow team.

In each line, you will receive the coordinate of a stone in the form of (x, y) where x, y is the coordinate.

It is guaranteed that:

- Each team has only 3 stones.
- None of the stones will overlap with each other.
- None of the stones will have the same distance to the button.
- None of the stones will be located at the boundary of the house.

Output description:

You need to output the winner and score as follows:

"**RED SCORES ***" or "**YELLOW SCORES ***" with a new line (\n)

If there are no winner, you need to output:

"**BLANK END**" with a new line (\n)

Sample input:

(-0.2,-0.2)

(-0.1,0.1)

(0.5,0.5)

(0.1,0.15)

(0.15,-0.5)

(0.5,-0.1)

Sample output:

RED SCORES 1