Instructions

- Attempt as many problems as you comfortably can. Focus more on quality and less on quantity.
- Problems are not listed in any particular order, start with the problem that you find the easiest.
- We are looking for the right output, but more importantly, we want to see how you think about a problem, design a solution and write clean, readable code.
- Seek help if you have trouble understanding the problem. You can send an email to (ajmal.ismail@arbisoft.com and jazib.babar@arbisoft.com). One of them should be able to answer any questions you may have.
- Even if you are unable to solve a problem fully, we'd still like to take a look at your code.
- Once done, create a zipped file on your desktop. Add your Full Name and date to the filename.

System related instructions

The following list of softwares is installed on your system (Ubuntu 16.04.02), you can solve the problems using any of them:

- Python2 / Python3
- C++ Compiler
- Oracle Java JDK/JRE 8
- Text Editor (ATOM)
- Browsers (Google Chrome, Firefox)
- Terminal

To create your code files

1. Open Atom text editor. (Click on icon ⚛ on the launcher bar to launch or press Windows key and search for Atom in Applications)
2. Create your project folder. File > Add Project Folder. (Do Remember the path where you create the project folder)
3. Create a new file inside the project folder for your code. File > New File. (Do remember to add extension of file after your file name e.g .cpp, .java or .py

To navigate to the directory where your code is using terminal

```
cd /path to your project folder.
```

For example if you created a project folder on Desktop, use this command: `cd ~/Desktop/Your folder name`

You can always check your current working directory by: `pwd`

***** ***Remember Linux is Case Sensitive***

To run a Python program there are two options.

Option 1: Call the interpreter yourself from terminal

For Python 2: `python <filename>.py`
For Python 3: `python3 <filename>.py`

Option 2: Let the code call the interpreter

1. Make sure the first line of your file has python interpreter path.
   `#! /usr/bin/python` or `#!/usr/bin/python3`
2. Make it executable: `chmod +x <filename>.py`
3. And run it using this command: `./<filename>.py`

To compile and run JAVA programs.

1. Write program and save the file as <filename>.java
2. Now to compile use this command from the terminal: `javac <filename>.java`
3. If everything works well you will notice a <filename>.class file is also created.
4. To run your program you compiled use the below code in terminal: `java <filename>`

To compile and run C++ programs.

1. Write program and save the file as <filename>.cpp
2. Now to compile use this command from the terminal: `g++ <filename>.cpp`
3. If everything works well you will notice a ‘a.out’ file is also created.
4. To run your program you compiled use the below code in terminal: `./a.out`

Someone from the HR department will be present throughout the test so if you have any trouble running any of the programs, please contact them immediately.

OOP Design - call center

Imagine you have a call center with three levels of employees: junior (J), technical lead (TL), product manager (PM). There can be multiple employees, but only one TL or PM. An incoming telephone call must be allocated to a Junior (J) who is free. If a junior can't handle the call, he or she must escalate the call to technical lead. If the TL is not free or not able to handle it, then the call should be escalated to PM. Design the classes and data structures for this problem. Implement a method getCallHandler().

Input

This is an open ended question. Feel free to make assumptions (write them down as comments in your code) where you see fit.

Output

The code file with the classes.
Some test code to mimic different scenarios defined above.

Counting Votes

Next turn of elections is near and we are already seeing that all political parties have started their campaigns at different levels. Meanwhile, a programmer is busy in creating a computerized system for this election. She has to create various different modules and deadline is very tight, she needs your help to create the counting module.

Input

Input will be read from the file. There will be multiple test cases in the input. First line of each test case will be an integer **N** (**N** <= 20), which will represent number of candidates. **N** lines follow, each containing a pair of strings separated by comma. First string will represent the name of the candidate and second string will represent the name of the party of that candidate. Note that a candidate can have no party, in that case second string will be `‘Independent’`. Next line will contain an integer **M** (M <= 10000) followed by **M** line, each with the name of the candidate for which a ballot is cast. Any name not in the list defined above should be considered as a rejected ballot.

Output

For each test case, output the name of the party of the winning candidate or word `‘independent’` if winning candidate has no party or word `‘tie’` if there is no winning candidate.

Sample Input

```
4
Batman, DC
Wonder Woman, Justice League
Superman, Marvel
Thor, Independent
7
Batman
Wonder Women
Wonder Women
Batman
Superman
Iron Man
Batman
```

Sample Output

```
DC
```

Meeting Rooms

We have several meeting rooms here at Arbisoft. Each team member is required to book the meeting room for a particular time before using it. Sometimes, our team members forget to see if the meeting room is already occupied in that time slot and they have to move to the other available room. While we are busy in creating new meeting rooms why don't you help us in creating a program that will check if a meeting room is already occupied.

Input

Input will be read from a file. Input will consist of multiple test cases. Each test case will begin with an integer B (`1 <= B <= 100`), donating number of bookings. B lines follow, consisting of start and end time of the booking (`hh:mm-hh:mm 24-hour format`). Stop processing the test cases when B is less than 1 and that test case should not be processed.

Output

For each test case, output "`conflict`" if bookings have conflicting times, and "`no conflict`" otherwise. Note that two bookings are not conflicting if a booking starts at the same time another booking ends.

Sample Input

```
3
03:00-03:50
18:00-19:00
03:50-04:30
2
11:00-12:00
10:00-13:00
0
```

Sample Output

```
no conflict
conflict
```

Independence Day

People all over the Pakistan celebrate Independence day on 14th of August every year. They dress up in green and white which are colors of Pakistan’s flag. We also celebrate Independence day at Arbisoft every year and beside wearing green and white we also decorate our office with these colors. This year we have decided to decorate our office with different patterns of green and white balloons. We will repeat these patterns to build almost infinite sequences of balloons. For example, If the pattern, **P**, is `WWGGG`, the infinite sequence would be `WWGGGWWGGGWW…`

We already have a huge stock of green balloons but no white balloons, so we want to know how many white balloons we will need between **I**th and **J**th balloon, inclusive (balloons are numbered with integers starting from 1). Can you help us?

Input

Input will be read from the file. The first line of input will indicate number of patterns that we have, **N** ($1 \le \mathbf{N} \le 100$). **N** lines follow. Each line will contain a string **P** ($1 \le$ length of $\mathbf{P} \le 100$) and two number **I** and **J** ($1 \le \mathbf{I} \le \mathbf{J} \le 10^{18}$), as defined above.

Output

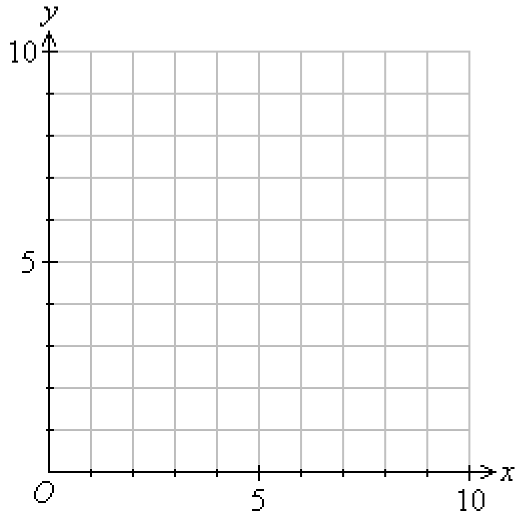
For each Pattern, output a number **X** on each line, where **X** is number of balloons between **I**th and **J**th balloon.

Sample Input/Output

`3`	
`WWGGG 3 7`	`2`
`WWGW 10 12`	`2`
`GW 1 10000000`	`5000000`

Glide Me!

You will be writing code to navigate a glider to pass through a hoop.
The glider will be flying in a 2-d cartesian coordinates system **limited to Quadrant 1 only** that looks something like this:



Feel free to look up 2-d cartesian coordinates system if you don't know already.

The instruction will look like this: "UUDSSDDDUSDUUU"

What this means:
"U" = Move Glider Up by 1 unit
"D" = Move Glider Down by 1 unit
"S" = Stationary (Keep going straight)

Please write control code for the glider that will consume the above code and move the glider to the appropriate position. For every instruction, the glider moves one position forward as well. Imagine a 2-d platformer game(flappy bird, for example) where the screen moves automatically. For example, glider starting at (0,0) geting instruction "USSUD" will take the following path: (1,1) (2,1) (3,1) (4,2) (5,1)
The goal of this simulation is to determine if your glider has passed through the vertical hoop identified by given "Top" and "Bottom" coordinates (assume *edges* of this hoop to be solid poles onto which glider can crash, moreover, assume that top coordinate of hoop will always be above bottom coordinate). You will also be given starting coordinates for the glider! Your input file will have 4 lines. Line 1: Starting coordinate, Line 2: Hoop Top, Line 3: Hoop Bottom, Line 4: instruction string.

Please print out the result in the end “Success” means glider passed through hoop. “Failure” means glider crashed or didn’t pass through hoop. Crash can also occur at bottom (y < 0)

Sample Input

```
(0,5)
(9,9)
(9,4)
UUDSSDUSDUUU
```

Sample Output

```
U: (1,6)
U: (2,7)
D: (3,6)
S: (4,6)
S: (5,6)
D: (6,5)
U: (7,6)
S: (8,6)
D: (9,5)
U: (10,6)
U: (11,7)
U: (12,8)
Success
```

Sample Input

```
(3,2)
(11,5)
(11,1)
DDDDUUSSUS
```

Sample Output

```
D: (4,1)
D: (5,0)
D: (5, -1)
Failure
```