
7. Lockers

Program Name: Lockers.java

Input File: lockers.dat

James left Lloyd's homework in his locker and, being the obnoxious friend that he is, James refuses to tell Lloyd the locker number. Instead, James wants to play a game with him. James will have Lloyd stand in the hall where the locker is located and have Lloyd read a locker number to him. James will help Lloyd by telling him if he is getting closer or further away from his locker until Lloyd eventually finds the correct locker.

Since James is lazy, he wants you to write a program for Lloyd to use on his laptop so James does not have to go with Lloyd in this endeavor. He can also use the program for other unsuspecting classmates who make the mistake of giving him their homework.

Input

The first line of input will contain a single integer n that indicates the number of lockers to be found. Each of the next n lines will contain the following:

- The number of the first locker in the hall followed by a space.
- The number of the last locker in the hall followed by a space.
- The number of the target locker (the locker that contains the homework) followed by a space.
- The first locker number guessed followed by a space.
- The remaining locker numbers guessed in the order that they were guessed separated by a space.

Note: All lockers are located on one side of the hall. The locker numbers are in sequential order. All locker numbers in the input file will be in the range given.

Output

If the target locker is the first locker guessed, print the locker number, a space, and `FOUND ON FIRST TRY`. Otherwise, for each locker number guessed, except the first, print the locker number followed by a space and one of the following:

- `COLDER` if he has moved further away from the target locker,
- `WARMER` if he has moved closer to the target locker,
- `SAME` if he is the same distance from the target locker as the previous guess
- `FOUND` when he finds the target locker.

Print exactly one blank line before continuing to the next locker to be found.

Example Input File

4

```
500 1000 859 575 600 650 732 840 950 850 900 875 859
100 400 150 300 350 200 175 125 400 150
1000 2000 1575 1428 1300 1400 1450 1700 1600 1500 1575
500 700 650 650
```

Example Output to Screen

600 WARMER
650 WARMER
732 WARMER
840 WARMER
950 COLDER
850 WARMER
900 COLDER
875 WARMER
859 FOUND

350 COLDER
200 WARMER
175 WARMER
125 SAME
400 COLDER
150 FOUND

1300 COLDER
1400 WARMER
1450 WARMER
1700 SAME
1600 WARMER
1500 COLDER
1575 FOUND

650 FOUND ON FIRST TRY