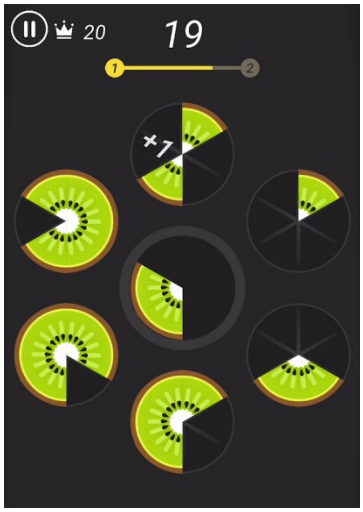
Slices

Filename: `slices`



Kelly and Jim are obsessed with a new mobile game called Slices! In Slices, the goal of the game is to fill up different circles with slices of fruit and pies while scoring points by placing pieces and completing full circles. After each level, the score required to move on increases.

Kelly and Jim may finish the levels at different times. Sometimes one player gets much further ahead than the other. When this happens, the other player becomes very angry and can't focus on anything in life other than catching up. (In extreme cases of differences between the two friends' progress, one player may forever hate the other.)

Specifically, if there ever is a time that Kelly is more than k levels ahead of Jim, Jim will forever hate Kelly. Similarly, if Jim is ever greater than k levels ahead of Kelly, Kelly will forever hate Jim.

The Problem:

Given the order which Kelly and Jim complete levels in Slices and the maximum number of levels one friend can be ahead of the other without anyone getting angry, determine whether Kelly hates Jim, Jim hates Kelly, they both hate each other, or if their friendship can last another day.

The Input:

The first line contains a single, positive integer, s, representing the number of scenarios to analyze. The first line of each scenario contains two positive integers, n and k ($1 \leq n \leq 1{,}000$; $1 \leq k \leq 1{,}000$), representing the number of events to follow, and the max number of levels one friend can be ahead of the other without putting a strain on their friendship, respectively. The next n lines each contain a single string (either "Kelly" or "Jim") representing who solved a level next and an integer, x ($1 \leq x \leq 1{,}000$), representing which level they just completed. It is guaranteed that the players solve the levels in order starting at 1.

The Output:

For each scenario, if neither friend hates the other, output a single line containing "`Everything is good`". If both players hate each other, output a single line containing "`Their friendship is doomed`". Otherwise, output either "`Kelly hates Jim`" or "`Jim hates Kelly`" depending on which friend hates the other.

Sample Input:

```
3
11 3
Kelly 1
Kelly 2
Jim 1
Kelly 3
Jim 2
Jim 3
Jim 4
Jim 5
Jim 6
Jim 7
Kelly 4
3 2
Jim 1
Kelly 1
Kelly 2
6 1
Kelly 1
Kelly 2
Jim 1
Jim 2
Jim 3
Jim 4
```

Sample Output:

```
Kelly hates Jim
Everything is good
Their friendship is doomed
```