Jimmy the Geek
Input File: JimmyIn.txt

Jimmy the geek has discovered a fool proof algorithm for predicting the winner of a playoff basketball game. It uses the average number of regular season points per game each team's offence ***s***cores, S, and each team's defense has ***a***llows, A. For example, if St. Joe's offence scores an average of 80 points per game, and Duke's defense allows an average of 41 points per game, then when they play each other St. Joe will score 60 points = (80 + 41) / 2. To determine the winner of the game, Jimmy would perform a similar calculation to determine the number of points Duke will score, and the winner of the game would be the team with the highest score. There will never be a tie.

Write a program that uses Jimmy's algorithm to determine the winner of the college basketball playoffs, which is a single elimination tournament (one loss and you're out). In the first round, team 1 plays team 2, team 3 plays team 4, etc. In the second round, the winner of the team 1 vs. team 2 game plays the winner of the team 3 vs 4 game, etc.

Inputs

The first line of input will contain the number of college playoff seasons to consider. This will be followed by a group of input lines for each season. The first line in a group will contain the number of teams in the playoffs, N, which will always be a power of 2. This will be followed by one line of input per team that contains two integers, separated by a space. The first integer will be the team's S point value the second integer will be the team's A point value. The first of these lines will be team 1's statistics, the second team 2's statistics, etc.

Outputs

There should be two lines of output for each season. The first line will contain the team number of the team that won the playoffs. The second line will contain the margin of victory (winner's minus looser's points scored) for each game the champion team won, each separated by a space. These will be in round number order, with the last integer being the margin in the championship game.

Sample Inputs

```
2
4
100 76
74 52
88 68
85 93
8
100 98
130 121
88 79
88 74
88 82
91 81
```

Sample Inputs (continued)

```
82 80
89 78
```

Sample outputs

```
1
1 2
4
2 2 2
```