

Round 3 2013

[A. Cheaters](#)

**B. Rural Planning**

[C. Are We Lost Yet?](#)

[D. Observation Wheel](#)

[Contest Analysis](#)

[Questions asked](#)

Submissions

Cheaters

7pt	Not attempted 250/365 users correct (68%)
10pt	Not attempted 201/223 users correct (90%)

Rural Planning

9pt	Not attempted 97/137 users correct (71%)
13pt	Not attempted 9/28 users correct (32%)

Are We Lost Yet?

12pt	Not attempted 85/127 users correct (67%)
18pt	Not attempted 23/53 users correct (43%)

Observation Wheel

8pt	Not attempted 242/255 users correct (95%)
23pt	Not attempted 3/6 users correct (50%)

Top Scores

Gennady.Korotkevich	64
vepifanov	64
SnapDragon	64
tomconerly	64
mystic	64
mikhailOK	64
winger	64
dzhulgakov	64
qwerty787788	64
PavelKunyavskiy	64

Problem B. Rural Planning

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input  
9 points

Solve B-small

Large input  
13 points

Solve B-large

Problem

You have recently purchased a nice big farmyard, and you would like to build a fence around it. There are already **N** fence posts in your farmyard.

You will add lengths of fence in straight lines connecting the fence posts. Unfortunately, for reasons you don't fully understand, your lawyers insist you actually have to use *all* the fence posts, or things will go bad.

In this problem, the posts will be represented as points in a 2-dimensional plane. You want to build the fence by ordering the posts in some order, and then connecting the first with the second, second with third, and finally the last one with the first. The fence segments you create should be a polygon without self-intersections. That is, at each fence-post there are only two fence segments, and at every other point there is at most one fence segment.

Now that's easy, but you also actually want to preserve the fact your farmyard is big! It's not really fun to wall off most of your farmyard with the fences. So you would like to create the fence in such a way that the enclosed area is *more* than half of the maximum area you could enclose if you were allowed not to use all the posts.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. The first line of each test case contains the number **N** of posts. The posts are numbered from 0 to **N** - 1. Each of the next **N** lines contains two integers **X<sub>i</sub>** and **Y<sub>i</sub>** separated by a single space: the coordinates of the **i**-th post.

Output

For each test case, output one line containing "Case #x: ", where x is the case number (starting from 1), followed by **N** distinct integers from 0 to **N** - 1, separated by spaces. They are the numbers of the posts, in either clockwise or counter-clockwise direction, that you will use to build the fence. Note that the first and last posts are connected.

If there are multiple solutions, print any of them.

Limits

The posts will be at **N** unique points, and will not all lie on the same line.

Small dataset

1 ≤ **T** ≤ 100  
3 ≤ **N** ≤ 10  
-100 ≤ **X<sub>i</sub>**, **Y<sub>i</sub>** ≤ 100

Large dataset

1 ≤ **T** ≤ 30  
3 ≤ **N** ≤ 1000  
-50000 ≤ **X<sub>i</sub>**, **Y<sub>i</sub>** ≤ 50000

Sample

Input	Output
3	Case #1: 0 1 2 3
4	Case #2: 0 1 4 2 3
1 2	Case #3: 0 2 1
2 0	
0 0	
1 1	
5	
0 0	
1 1	
2 2	
0 2	
2 0	

```
3
0 0
1 0
0 1
```

In the first test case, there are three polygons we can construct, and two of them have a large enough area — the ones described by sequences 0 1 2 3 and 0 2 1 3. The polygon described by 0 1 3 2 would be too small. In the second test case, we have to make sure the polygon does not intersect itself, so, for instance, 0 1 2 3 4 or 0 1 3 4 2 would be bad. In the third case, any order describes the same triangle and is fine.

---

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform