

Rural Planning

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_i and Y_i 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

Memory limit: 1GB.

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

Small dataset (Test set 1 - Visible)

Time limit: 30 seconds.

$$1 \leq T \leq 100$$

$$3 \leq N \leq 10$$

$$-100 \leq X_i, Y_i \leq 100$$

Large dataset (Test set 2 - Hidden)

Time limit: 60 seconds.
 $1 \leq T \leq 30$
 $3 \leq N \leq 1000$
 $-50000 \leq X_i, Y_i \leq 50000$

Sample

Sample Input

```
3
4
1 2
2 0
0 0
1 1
5
0 0
1 1
2 2
0 2
2 0
3
0 0
1 0
0 1
```

Sample Output

```
Case #1: 0 1 2 3
Case #2: 0 1 4 2 3
Case #3: 0 2 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 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.