

## 2270 Cheap Fence

A farmer has decided to surround his garden in order to protect his fruits from the appetite of his neighbors. This man does not have a lot of money, but he is a very innovative person. He has found the possibility of reducing costs by using trees from his property as fence posts. He found that in order to maximize the area to be enclosed by the fence, the trees to be used as fence posts must be selected carefully avoiding the creation of inward angles in the fence. Also, the selection of tree posts must ensure that the fence will enclose all the trees that are in the property.

### PROBLEM

Given a set of trees (each tree is identified by a unique tree identifier) and the position of each tree in the property, write a program that obtains the following information:

- The minimum number of trees required to enclose all the trees in the property
- The sequence of trees that must be used as tree posts.

### Input

The input file contains several tests.

The first line of the test contains an integer value  $n$  ( $1 < n < 100$ ) representing the number of trees available in the garden.

The following  $n$  lines are triplets representing the tree id (integer number) and the position of the tree ( $x, y$  coordinates) in the garden (the coordinates are real numbers). The input ends when  $n = 0$ .

### Output

For each test, the first line displays the number of trees that must be used as fence posts. The next line displays the sequence of tree identifiers that will be used as fence posts.

### EXAMPLE

The next matrix represents the position of 11 trees in a property. A cell with a number indicates that the Tree whose identifier is the number in the cell is located in the  $x, y$  position of the garden. For example, tree 11 is in position (0,3). In this example, The number of fence posts required will be 6 and the sequence of tree identifiers that will be part of the fence posts is 11,6,4,10,9 and 8.

	0	1	2	3	4
0				11	
1	8		7		
2			1		
3	9	2			6
4				5	
5			3	4	
6	10				

**NOTES/HINT:** The first tree of the output sequence is always the last tree for which a position was provided in the input file. Your sequence in the output file must be printed in clockwise order.

### Sample Input

```
11
1 2 2
2 1 3
3 2 5
4 3 5
```

```
5 3 4
6 4 3
7 2 1
8 0 1
9 0 3
10 0 6
11 3 0
9
1 7.1 1.1
2 2.1 2.1
3 9.1 -1.1
4 9.1 2.1
5 10.1 -1.1
6 7.1 0.0
7 5.1 2.1
8 4.1 -2.1
9 4.1 3.1
```

### Sample Output

```
6
11 6 4 10 9 8
5
9 2 8 5 4
```