Safest Buildings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

PUBG is a multiplayer online battle royale video game. In the game, up to one hundred players parachute onto an island and scavenge for weapons and equipment to kill others while avoiding getting killed themselves. BaoBao is a big fan of the game, but this time he is having some trouble selecting the safest building.

There are n buildings scattering on the island in the game, and we consider these buildings as points on a two-dimensional plane. At the beginning of each round, a circular safe area whose center is located at $(0, 0)$ with radius R will be spawned on the island. After some time, the safe area will shrink down towards a random circle with radius r ($r \le R$). The whole new safe area is entirely contained in the original safe area (may be tangent to the original safe area), and the center of the new safe area is uniformly chosen within the original safe area.

The buildings covered by the new safe area is called the safe buildings. Given the radius of the safe areas and the positions of the buildings, BaoBao wants to find all the buildings with the largest probability to become safe buildings.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains three integers n ($1 \le n \le 100$), R and r ($1 \le r \le R \le 10^4$), indicating the number of buildings and the radius of two safe circles.

The following n lines each contains two integers x_i and y_i ($-10^4 \le x_i, y_i \le 10^4$), indicating the coordinate of the buildings. Here we assume that the center of the original safe circle is located at $(0, 0)$, and all the buildings are inside the original circle.

It's guaranteed that the sum of n over all test cases will not exceed 5000.

Output

For each test case output two lines.

The first line contains an integer m, indicating the number of buildings with the highest probability to become safe buildings.

The second line contains m integers separated by a space in ascending order, indicating the indices of safest buildings.

Please, DO NOT output extra spaces at the end of each line.

Example

standard input	standard output

2
3 10 5
3 4
3 5
3 6
3 10 4
-7 -6
4 5
5 4
	1
	1
	2
	2 3