Statement

You are a large ball on an infinite horizontal plane. You start at time 0, touching the plane at the location $(0, 0)$. Your initial weight is w.

You are free to roll in any direction you like, at 1 unit of distance per second.

On the plane there are n objects, numbered from 0 to $n - 1$. For the purpose of this task we will consider them to be points. Object i is located at coordinates (x_i, y_i) and has weight z_i.

If you are at the same location as one of the objects and your weight is **strictly greater than twice** the weight of the object, you may spend 1 second to assimilate it. Doing so turns you into a bigger sphere – the weight of the assimilated object is added to your weight.

Your primary objective is to create a sphere that is as heavy as possible. Your secondary objective is to achieve the primary objective as quickly as possible.

Special rules

Resubmissions for this problem do not generate penalty minutes.

(Note that the 20-submission limit is still in place.)

Input format

The first line of the input file contains the number t of test cases. The specified number of test cases follows, one after another.

Each test case starts with a line containing the integers n and w. Then, n lines follow. The i-th of these lines (numbering from 0) contains the integers x_i, y_i, and z_i.

Output format

For each test case output two lines. The first line should contain the number k of items you want to collect, and the second line should contain a space-separated sequence of items in the order you want to collect them. (If k is zero, the second line should be present and empty.)

Subproblem K1 (100 points, public)

Input file: K1.in

There are $t = 5$ test cases, each worth 20 points. In each test case:

- $n \leq 200$
- all coordinates are between -10^7 and 10^7, inclusive
- the objects are at mutually distinct locations
- no object is located at $(0, 0)$
- all weights are between 1 and 10^7, inclusive, and their sum does not exceed 10^9

Each test is scored as follows:

A solution that doesn't describe a valid way of creating the biggest possible sphere in the format specified above scores 0. Otherwise, its score is determined using the time t it needs to produce the final sphere, using the following formula:

Let t_{opt} be the smallest time in which the same set of objects you actually collect in your solution can be collected by a ball with initial weight $w = 10^9$. Your score drops linearly from 20 points if your t equals t_{opt} to 0 points if $t = 4t_{opt}$ (or more).

(Note that usually there will be no valid solution that actually has $t = t_{opt}$. If that happens, it is impossible to score the full 20 points for that test case. This is intentional.)

The final score for the task is rounded to two decimal places.

Example

input	output
2 3 100 1 1 49 -1 -1 50 2 3 95 2 100 1 0 100 2 0 40	3 0 1 2 1 1

In the first test case the optimal path for a ball that already starts huge would be to collect the objects in the order 1, 0, 2. This would require $\sqrt{2} + 2\sqrt{2} + \sqrt{5} \approx 6.4787$ seconds of travel and 3 seconds of assimilating. Hence, $t_{opt} \approx 9.4787$.

We cannot take the path 1, 0, 2 because our ball is initially too small to assimilate object 1. Collecting the objects in the order 0, 1, 2 is the only order that works. E.g., by the time we go pick up object 1, our weight is already 149, so we are heavy enough to assimilate it. This solution requires $t = \sqrt{2} + 2\sqrt{2} + \sqrt{25} + 3 \approx 12.2426$ seconds.

For the above solution we would score 18.06 points.

In the second test case our ball will never be heavy enough to pick up object 0. The optimal solution is therefore to roll for 2 seconds and to pick up object 1. (Note that we rolled across object 0 without attempting to assimilate it.)

This solution would score the full 20 points.

Grading

For each test case the grader looks at the two lines of your output file that should contain its answer, and tries to judge it. For syntactically correct submissions the grader will report a rounded score for the test case. For other submissions it will report a short string describing the type of the error as follows:

- nok: failed to parse k
- noit: failed to parse the sequence in which you collect items
- badit: sequence must contain k distinct valid items
- toobig: item too heavy to be picked up when you wanted
- left: final ball is not as heavy as possible