

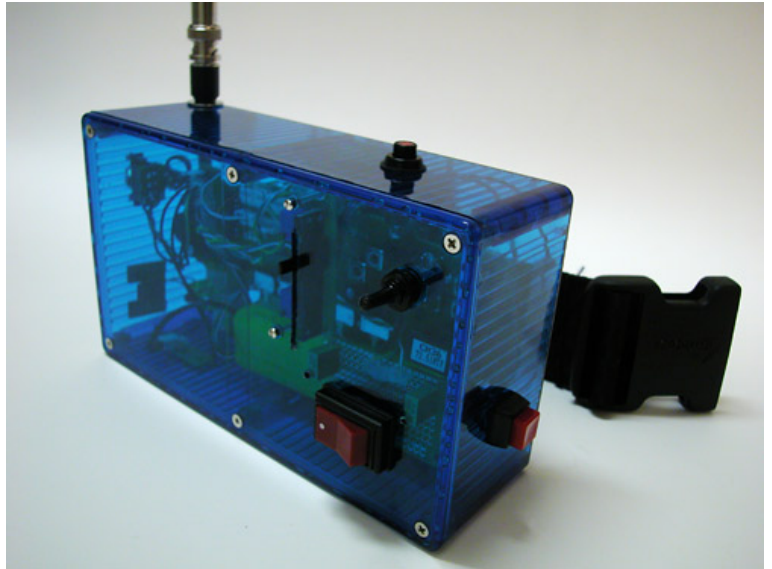
## 11746 Finding the Transmitter Again

SOS! Your boss has misplaced an important (and expensive) radio transmitter somewhere in a large, rectangular cornpatch. He refuses to explain why he took the transmitter out into the cornpatch, but he has tasked you with finding it. Thankfully, he has already set up three radio receivers in the cornpatch to help you narrow down the possible locations to search.

The receivers are designed in such a way that they can only measure the signal strength at their location, and not which direction it is coming from. Furthermore, they are too large to easily move.

You know that for this particular model of transmitter, the signal strength at a positive distance  $d$  is given by the expression  $P/d^2$ , where  $P$  is the power level of the transmitter. Unfortunately, you don't know what power level the transmitter is currently set to, only that it is positive.

Write a program that reads as input the locations and readings of the receivers, and outputs a list of possible locations of the transmitter.



### Input

The input begins with a number  $N$  on a single line, followed by  $N$  testcases.

Each testcase consists of three lines, each containing the location (as two integers  $x_i y_i$ ) and power reading (as an integer  $p_i$ ) from one of the receivers. You are assured that  $0 \leq x_i, y_i \leq 1,000$ ,  $0 < p_i \leq 100,000$ , and that the receivers are located at three distinct points.

### Output

For each testcase, output one line containing first the number of possible locations of the transmitter, followed by the possible locations themselves (in the form  $x_t y_t$ ). A location is possible if it is within the cornpatch ( $0 \leq x_t, y_t \leq 1,000$ ), and a transmitter there could produce the given receiver readings for some positive power level  $P$ .

The output coordinates should be rounded to exactly two decimal places, and multiple output locations should be sorted by increasing  $x$ -coordinate, breaking ties by increasing  $y$ -coordinate. You are guaranteed that the input data will determine a finite, non-zero number of possible locations. Furthermore, the input will be such that a small perturbation (on the order of  $10^{-6}$ ) in the receiver positions would not change the number of possible locations.

### Sample Input

```
2
100 0 50
```

```
100 100 50
0 100 50
100 0 50
100 100 50
0 100 25
```

**Sample Output**

```
1 50.00 50.00
2 67.71 50.00 332.29 50.00
```