Rectangles Packing

You might be aware of the Packing Problem or the Perfect Square Dissection problem known from the Scottish Book. Here, your task is similar: given a rectangle *R* and a given set *S* of rectangles: r_1, r_2, ... r_n try to pack axis-aligned rectangles from *S* into *R* in such a way that the unused space in *R* is as small as possible.

Input

First, you are given *t* ($t < 100$) - the number of test cases. Each of the test cases starts with two integers: *Rx*, *Ry* - describing the size of *R*. In the next line you are given *n* ($n < 100$) - the number of objects in the set. In the successive *n* lines, the descriptions of the *S* elements follow. The *i*-th line consists of two integers: rx_i, ry_i - the size of r_i.

Output

For each of the test cases output *k* ($0 <= k <= n$) - the number of elements from *S* you pack into *R* and in the successive *k* lines: *i* - the number of the rectangle to use, x_i, y_i - the coordinates of r_i in *R* and one character: either o (for original alignment) or r (for rotated alignment).

Coordinates are relative to *R*. Coordinates of vertices of *R* are: (0,0), (*Rx*, 0), (*Rx*, *Ry*), (0, *Ry*). Coordinates of vertices of a rectangle r_i: *i* x_i y_i o are: (rx_i, ry_i), (rx_i+x_i, ry_i), (rx_i+x_i, ry_i+y_i), (rx_i, ry_i+y_i), while coordinates of vertices of a rotated rectangle r_i: *i* x_i y_i r are: (rx_i, ry_i), (rx_i+y_i, ry_i), (rx_i+y_i, ry_i+x_i), (rx_i, ry_i+x_i).

Do not use the same r_i twice. No two different r_i, r_j may overlap.

Scoring

The score awarded to your program for a given test case is the area of the used rectangles.
The score awarded to your program for a given test set is the sum of points awarded for all cases in the set.
The overall score of the program is the sum of scores obtained for correctly solved test sets.

The number of points given in the ranking is scaled so that it is equal to 10 for the registered contestant whose solution has the highest score, and proportionally less for all solutions with lower scores.

Example

Input:
```
3

7 7
5
1 3
2 1
1 4
4 4
6 6
```

```
6 2
3
1 5
1 5
1 2

3 3
1
4 4
```

Output:

```
4
5 1 1 o
1 0 0 r
2 3 0 o
3 0 1 o

3
1 0 0 r
2 0 1 r
3 5 0 o

0
```

Scoring: The exemplary solution will score 57 points (45 + 12 + 0).

Input data sizes

s	t	n	Rx*Ry	l
1	10	<20	<40	2s
2	10	<20	<100	2s
3	10	<30	<120	2s
4	10	<30	<400	2s
5	10	<50	<200	2s
6	20	<40	<10	5s
7	20	<100	<10	5s
8	20	<40	<2500	5s
9	20	<80	<2500	5s
10	20	<100	<10000	5s

s	- test set number
t	- the number of test cases
n	- the number of rectangles
x*y	- the size of R
l	- time limit

Please note

- Till the last week of the series, all submitted codes will be visible to all users and tested on temporary data sets only.
- For the last week of the series, submissions will be visible to the submitting contestant, only, and tested on the full set of test cases. (All earlier solutions will be rejudged).