JOINING POINTS

"Joining points" is a single-player game. To play it, choose two integers greater than two and call them *g* and *r.* Then draw four points at the vertices of a square making the top two points green and the bottom two points red. Draw green points and red points inside the square taking care that no three points, including the four initial ones, are in the same line. Continue until the total number of green points equals *g* and the total number of red points equals *r*.

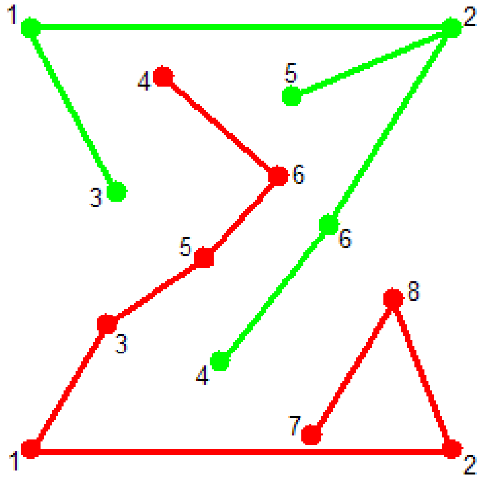
After the board is drawn, start joining points. Any two points can be joined by a line segment as long as:

- The two points to be joined are of the same color, and
- The line segment joining the points does not intersect any other previously drawn line segment (other than at the endpoints).

Two points *u* and *v* are said to be in the same *component* if it is possible to traverse from point *u* to point *v* using the line segments already drawn.

You win the game if you get all the green points in one component using exactly *g*-1 line segments, and all the red points in another component using exactly *r*-1 line segments. **It can be proven that if the points are drawn as described above, then there is always a way to win the game.**

You will be given a square board of size *s* with *g* green and *r* red points whose coordinates are represented by integer pairs (x_i, y_i). Green points are numbered from 1 to *g* with the top-left point at (0,*s*) being 1, the top-right at (*s*,*s*) being 2, and interior points ranging from 3 to *g* in arbitrary order. Red points are numbered from 1 to *r* with the bottom-left point at (0,0) being 1, the bottom-right at (*s*,0) being 2, and interior points ranging from 3 to *r* in arbitrary order.



The figure shows a sample game all the green points are joined into one component and all the red points are joined into another component.

You can see that no three points are in the same line, and that no two line segments intersect each other except at their endpoints.

TASK

Write a program that, given the coordinates of the *g* green points and the coordinates of the *r* red points, decides how to draw *g* - 1 green line segments and *r* - 1 red line segments so that all of the green points are in the same component, all of the red points are in another component, and no two line segments intersect each other.

CONSTRAINTS

3 ≤ *g* ≤ 50 000	Number of green points.
3 ≤ *r* ≤ 50 000	Number of red points.
0 < *s* ≤ 200 000 000	

INPUT

Your program must read the following data from the file `points.in`

`points.in`	DESCRIPTION

`6`
`0 1000`
`1000 1000`
`203 601`
`449 212`
`620 837`
`708 537`
`8`
`0 0`
`1000 0`
`185 300`
`314 888`
`416 458`
`614 622`
`683 95`
`838 400`
	LINE 1: Contains the integer *g.*
	NEXT *g* LINES: Each line contains two space-separated integers that represent the coordinates x_i and y_i of each of the *g* green points, starting from 1 to *g.*
	LINE *g+2:* Contains integer *r*.
	NEXT *r* LINES: Each line contains two space-separated integers that represent the coordinates x_i and y_i of each of the *r* red points, starting from 1 to *r.*

OUTPUT

Your program must write the following data to the file `points.out`

`points.out`	DESCRIPTION

`1 3 g`
`3 1 r`
`3 5 r`
`4 6 r`
`6 5 r`
`4 6 g`
`1 2 g`
`1 2 r`
`5 2 g`
`2 6 g`
`7 8 r`
`8 2 r`
	Your output file must contain (*g* – 1) + (*r* – 1) lines, one for every line segment drawn to join points.
	Each line must contain three space-separated entities: two integers and a character.The two integers represent the numbers of two points joined by that line segment. The character must be a ***g*** if the points are green and an ***r*** if the joined points are red.
	The order in which you list the line segments does not matter; neither does the order of the endpoints of each line segment.

GRADING

For a set of test cases worth a total of 35 points, every test run will meet the following requirements:
$3 \leq g \leq 20$
$3 \leq r \leq 20$