421 Polygonal Puzzle

To help model the three dimensional volumes that describe each supercomputer choice over time, Valentine McKee and Jake Briggs decided to construct some cardboard mock-ups. Valentine began by cutting out polygons of each of the supercomputer criteria for each year. As she was bringing the cut-outs to Jake's lab, where they intended to turn the flat polygons into the actual volumes, Valentine bumped into Ren McCormack. Literally. And she dropped all of her polygon cut-outs onto the ground.

"Ren! Why don't you look where you're going," Valentine sighed. "I dropped all of my polygons. Now I'm going to have to start all over cutting these out." Valentine had labeled the polygons prior to cutting them out, but unfortunately, the labeling was not part of the actual polygon and was lost when they were cut out. "I don't know which is which now."

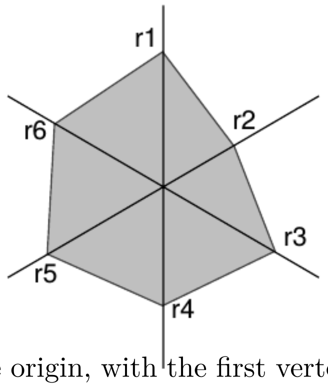
"Sorry." Ren looked downcast, then brightened. "Hey, do you want to go to the dance with me this weekend?"

"I don't think so, Ren," replied Valentine, "I think I'll be cutting out and sorting polygons all weekend." Valentine stooped and began picking up the cut-outs. Ren was a great dancer but he also tended to show it off. The one time that Valentine had gone to a dance with him, Ren had completely embarrassed her with his antics.

"Can't you use a computer or something to help you sort those things out?" Ren picked up one of the polygons and squinted at it. (Ren was a theater major.)

Valentine thought for a moment. "Nope," she said. "I'm just going to have to start over. Have fun at the dance this weekend, though." To herself, she thought, "Yes, I can use a computer. The only other thing I need is that camera that Jake already has in the lab to help me see the Big Picture."

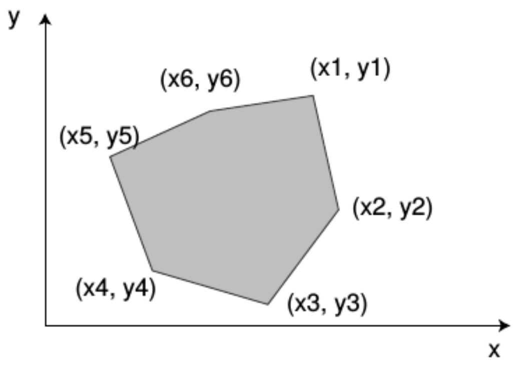
Input



The first line of input will consist of two integers separated by a space. The first integer specifies the number of polygons to be sorted. The second integer specifies the number of vertices for each polygon.

The rest of the input will consist of two lists of coordinates. The first is a list of coordinates describing the shape of each polygon in terms of the values on each of its axes (i.e., the values for `r1` through `r6` in the figure on the right).

The vertices of each polygon are distributed at equal angles around the origin, with the first vertex aligned at 12 o'clock. The vertices are given in clockwise order. The polygons in this list are unique with respect to rotation and dilation. That is, no polygon will be exactly like any other polygon (having the same vertices in the same locations), nor can any polygon be rotated about its center and/or uniformly scaled such that it will be exactly like another polygon in the list.



The second list of polygon coordinates corresponds to the polygons in the first list, but as measured by a camera. The vertices of each polygon in this case will be given as (x, y) pairs with respect to the camera's viewpoint origin and are given in clockwise order. The first vertex given is arbitrary, however.

Note that the camera's view of each polygon may include magnification or de-magnification (which may be different for each viewed polygon). Moreover, the camera can only resolve the polygon vertices to within plus or minus one pixel in its field of view, so there may be some jitter associated with the viewed vertex locations. For each polygon, the maximum amount of jitter at each vertex is guaranteed to be less than one percent of the distance from the origin to the vertex farthest away from the origin.

All of the polygons viewed by the camera will be right-side up.

Output

The output is a list of instructions indicating how to rearrange the scrambled polygons. That is, for each polygon in the second list, the output should indicate the polygon to which it corresponds in the first list, the amount by which it must be rotated in a clockwise direction to align the vertices. Thus, one line of output consists of an integer index i (indexing starts from 1), a floating point angle θ (with $0 \leq \theta < 360$) given to the nearest tenth of a degree.

Sample Input

```
2 4
1 1 2 1
1 1 1 1
1 3 2 3 2 2 1 2
2.5 3.5 2 2 1 2 1 3
```

Sample Output

```
2 45.0
1 135.0
```