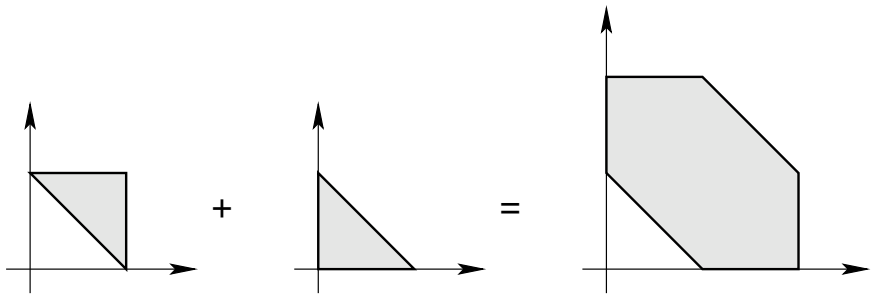
Polygon

PROBLEM

A polygon consists of all points on or enclosed by its border. A convex polygon has the property that for any two points *X* and *Y* of the polygon, the line segment connecting *X* and *Y* is inside the polygon. All polygons in this task are convex polygons with at least two vertices, and all vertices in a polygon are different and have integer coordinates. No three vertices of the polygon are collinear. The word "polygon" below always refers to such polygons.

Given two polygons *A* and *B*, the Minkowski sum of *A* and *B* consists of all the points of the form (x_1+x_2, y_1+y_2) where (x_1, y_1) is a point in *A* and (x_2, y_2) is a point in *B*. It turns out that the Minkowski sum of polygons is also a polygon. The figure below shows an example: two triangles and their Minkowski sum.



We study a reverse operation to the Minkowski sum. For a given polygon *P*, we are looking for two polygons *A* and *B* such that:

- *P* is the Minkowski sum of *A* and *B*,
- *A* has from 2 to 4 different vertices, i.e. it is a segment (2 vertices), a triangle (3 vertices) or a quadrilateral (4 vertices),
- *A* should have as many vertices, as possible, i.e.:
  - *A* should be a quadrilateral, if possible,
  - if *A* cannot be a quadrilateral, it should be a triangle, if possible,
  - otherwise it should be a segment.

Clearly, neither *A* nor *B* can be equal to *P* because then the other summand would have to be a point, which is not a valid polygon.

You are given a set of input files, each containing a description of a polygon *P*. For each input file you should find the polygons *A* and *B*, as required above, and create an output file containing descriptions of *A* and *B*. For the given input files such polygons *A* and *B* can always be found. If there are many correct results, you should find and output one of them. You should not submit any programs, just the output files.

INPUT

You are given 10 problem instances in the text files named `polygon1.in` to `polygon10.in`, where the number after `polygon` is the input number. Each input file is

organized as follows. The first line contains one integer N: the number of vertices of the polygon P. The following N lines describe the vertices in a counter-clockwise order, one vertex per line. Line I+1 (for I = 1, 2, ..., N) contains two integers X_I and Y_I, separated by a space: coordinates of the Ith vertex of the polygon. All input coordinates are non-negative integers.

OUTPUT

You are to submit 10 output files corresponding to the given input files which describe the required polygons A and B. The first line is to contain the text:
`#FILE polygon I`
where integer `I` ($1 \le$ `I` ≤ 10)is the number of the respective input file.

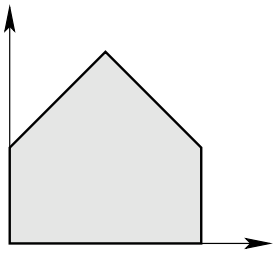
The output format is similar to the input format. The second line is to contain one integer N_A: the number of vertices in A ($2 \le N_A \le 4$). The following N_A lines describe the vertices of A in the counter-clockwise order, one vertex per line. Line I+2 (for I = 1, 2, ..., N_A) contains two integers X and Y, separated by a space: coordinates of the Ith vertex of the polygon A.

Line N_A+3 should contain one integer N_B: the number of vertices in B, ($2 \le N_B$). The following N_B lines describe the vertices of B in the counter-clockwise order, one vertex per line. Line N_A+J+3 (for J = 1, 2, ..., N_B) contains two integers X and Y, separated by a space: coordinates of the Jth vertex of the polygon B.

EXAMPLE INPUT AND OUTPUTS

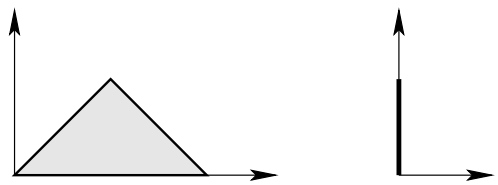
`polygon0.in`

```
5
0 1
0 0
2 0
2 1
1 2
```



For the above input, either of the below output files (see also the figures) is correct, since in both cases A is a triangle and it cannot be a quadrilateral.

```
#FILE polygon 0
3
0 0
2 0
1 1
2
0 1
0 0
```



```
#FILE polygon 0
3
0 0
1 0
1 1
3
0 1
0 0
1 0
```