

Polygonal Line Search

Multiple polygonal lines are given on the xy-plane. Given a list of polygonal lines and a template, you must find out polygonal lines which have the same shape as the template.

A polygonal line consists of several line segments parallel to x-axis or y-axis. It is defined by a list of xy-coordinates of vertices from the start-point to the end-point in order, and always turns 90 degrees at each vertex. A single polygonal line does not pass the same point twice. Two polygonal lines have the same shape when they fully overlap each other only with rotation and translation within xy-plane (i.e. without magnification or a flip). The vertices given in reverse order from the start-point to the end-point is the same as that given in order.

Figure 1 shows examples of polygonal lines. In this figure, polygonal lines A and B have the same shape.

Write a program that answers polygonal lines which have the same shape as the template.

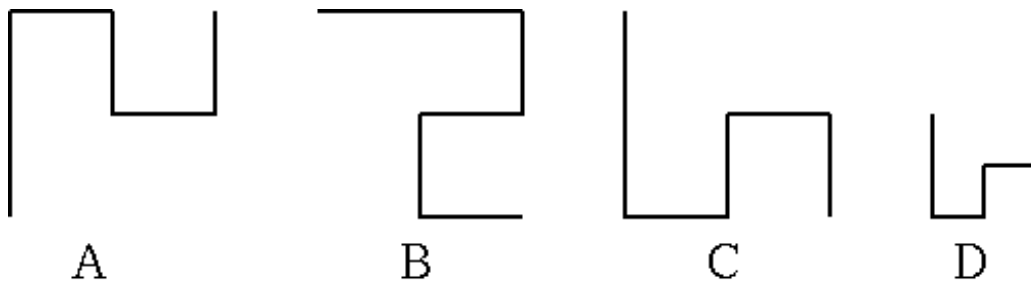


Figure 1: Polygonal lines

Input

The input consists of multiple datasets. The end of the input is indicated by a line which contains a zero.

A dataset is given as follows.

```
n
Polygonal line0
Polygonal line1
Polygonal line2
...
Polygonal linen
```

n is the number of polygonal lines for the object of search on xy-plane. n is an integer, and $1 \leq n \leq 50$. Polygonal line0 indicates the template.

A polygonal line is given as follows.

```
m
x1 y1
x2 y2
...
xm ym
```

m is the number of the vertices of a polygonal line ($3 \leq m \leq 10$). x_i and y_i , separated by a space, are the x- and y-coordinates of a vertex, respectively ($-10000 < x_i < 10000$, $-10000 < y_i < 10000$).

Output

For each dataset in the input, your program should report numbers assigned to the polygonal lines that have the same shape as the template, in ascending order. Each number must be written in a separate line without any other characters such as leading or trailing spaces.

Five continuous "+"s must be placed in a line at the end of each dataset.

Example

Input:

```
5
5
0 0
2 0
2 1
4 1
4 0
5
0 0
0 2
-1 2
-1 4
0 4
5
0 0
0 1
-2 1
-2 2
0 2
5
0 0
0 -1
2 -1
2 0
4 0
5
0 0
2 0
2 -1
4 -1
4 0
5
0 0
2 0
2 1
4 1
4 0
4
4
-60 -75
-60 -78
-42 -78
```

-42 -6
4
10 3
10 7
-4 7
-4 40
4
-74 66
-74 63
-92 63
-92 135
4
-12 22
-12 25
-30 25
-30 -47
4
12 -22
12 -25
30 -25
30 47
3
5
-8 5
-8 2
0 2
0 4
8 4
5
-3 -1
0 -1
0 7
-2 7
-2 16
5
-1 6
-1 3
7 3
7 5
16 5
5
0 1
0 -2
8 -2
8 0
17 0
0

Output:

1
3
5
+++++
3
4
+++++
+++++