

Modern graphic computer programs can, among other, even more stunning capabilities, fill a closed region. Though not all of them can protect the user from accidentally choosing to fill the background rather than the inner part. Besides being a channel hopper at home your boss' favourite hobby is colouring the pictures, you cannot protest long about adding this magnificent protection feature to his graphic program.

This means that your job is to write a program, which determines whether a point belong to a polygon, given the array of its vertices.

To make life a bit simpler you may assume that:

- all edges of the polygon are vertical or horizontal segments
- lengths of all the edges of the polygon are even integer numbers
- co-ordinates of at least one vertex are odd integer numbers
- both co-ordinates of any vortex cannot be divisible by 7 at the same time
- the investigated point P has both co-ordinates being even integer numbers
- the polygon has at most 1000 vertices
- co-ordinates of the vertices lay in the range: -10000..10000.

Input

Input data may consist of several data sets, each beginning with a number of polygon's vertices (n). Consecutive n lines contain co-ordinates of the vertices (x followed by y). Then go the co-ordinates of investigated point P. Input data end when you find 0 the number of polygon's vertices.

Output

For each polygon and each point P you should print one character (in separate lines): 'T' when P belongs to the polygon or 'F' otherwise.

Sample Input

```
4
1 1
1 3
3 3
3 1
2 2
12
1 1
1 9
3 9
3 5
5 5
5 9
7 9
7 1
5 1
5 3
3 3
3 1
4 2
0
```

Sample Output

```
T
F
```