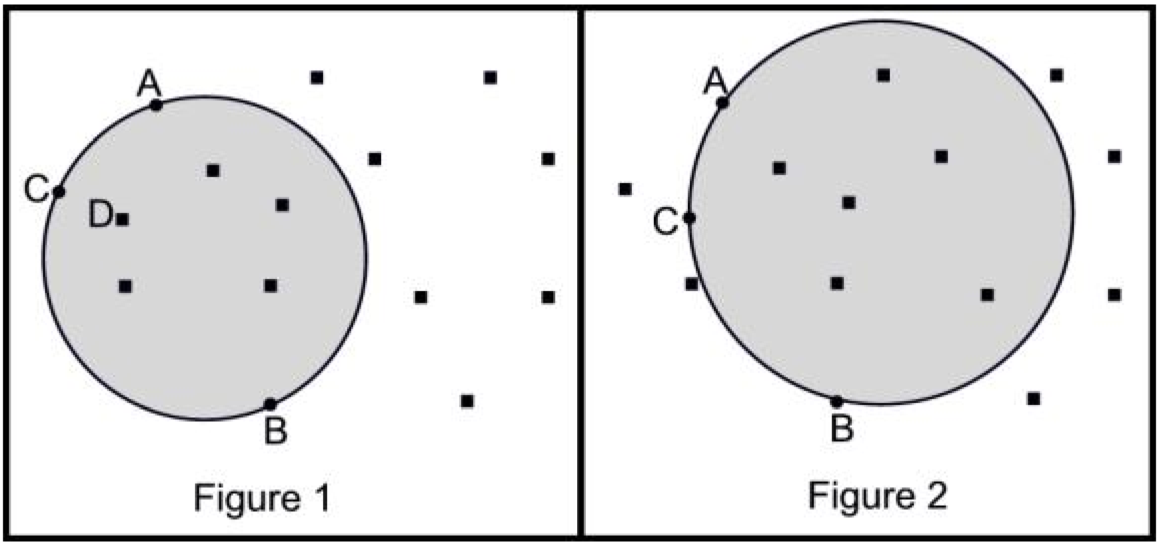
12064 Count the Points Inside

We know that if three different points are not collinear then there is a unique circle that passes through them. When three points are collinear we get a unique straight line passing through them and a straight line can be considered as part of a circle of infinite radius. Given several thousand points and three special points **A**, **B** and **C** in a two dimensional Cartesian coordinate system your job is to find out how many points are contained in the circle passing through the three special points. Please note that your solution must be very efficient.



In the figure above the three special points are shown as small filled circles and normal points are shown as small filled squares. In Figure 1 we can see a circle passing through three special points **A**, **B** and **C**. There is also a normal point **D**. There are total five points strictly within the circle and three points are on the boundary. So the circle contains eight points in total. When the point **D** is turned into special point **C** we get another circle, which is shown, in Figure 2. This larger circle contains total nine points.

Input

The input file contains at most ten sets of input. The description of each set is given below.

First line of each set contains six integers A_x, A_y, B_x, B_y, N and Q. Here (A_x, A_y) is the coordinate of **A** and (B_x, B_y) is the coordinate of **B**, N $(0 < N \leq 50000)$ is the total number of points (excluding **A** and **B**) and Q $(1 \leq Q \leq 1000)$ is the number of queries. Each of the next N lines contains two integers x_i, y_i, where (x_i, y_i) is the Cartesian coordinate of the i-th point $(1 \leq i \leq N, -10000 \leq x_i, y_i \leq 10000)$. Each of the next Q lines contains one integer S_C $(1 \leq S_C \leq N)$, which indicates that the S_C-th point is to be considered as point **C** for the current query. For example, in the first sample input the point with serial no **4** is **(5, 5)** as there are three points before it (**(-1, -1)**, **(-1, -1)** and **(3,3)**) in the input sequence. Please note that for a single set of input, point **A** and **B** are fixed. Only the position of point **C** is being altered for each query.

The input is terminated by a case where $N = Q = 0$. This case should not be processed. You can see from the first sample input that more than one point can have the same coordinate. Such points should be considered as different points.

Output

For each set of input produce $Q + 1$ lines of output. The description of the output for each set is given below:

The first line contains the serial number of the output as shown in the output for sample input. Each of the next Q lines contains an integer D which indicates how many points will be within the circle passing through **A**, **B** and **C**. The points on the boundary of the circle are also considered inside the circle. So you should also consider **A**, **B** and **C** within the circle. If the radius of the circle is greater than **100000** then simply print the line '`Impossible`' instead of the integer D.

Sample Input

```
0 10 10 0 6 2
-1 -1
-1 -1
3 3
5 5
-1 -1
12 16
4
1
0 10 10 0 8 2
-1 -1
2 2
4 4
6 6
8 8
12 12
14 14
16 16
3
5
0 10 10 0 0 0
```

Sample Output

```
Case 1:
Impossible
7
Case 2:
8
7
```