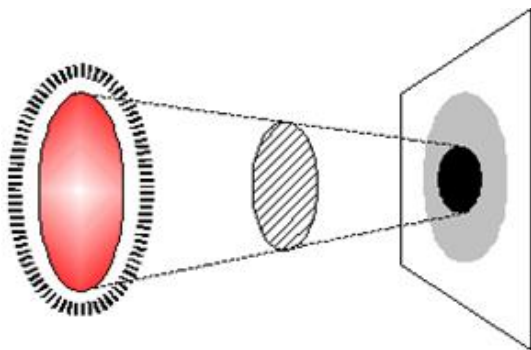


Little Dipto is fascinated by shadows. He spends hours holding up various objects in front of the wall, watching the shadow grow and shrink as he moves the object. But these days he has been developing a special interest in shadows of polygons; convex polygons to be precise. But he cannot seem to be able to predict the size of the shadow before he actually cuts out a paper polygon and forms a shadow of it on the wall. He has now turned to you for help.



Input

The input consists of at most 60 test cases. Each test case starts with an integer N , the number of vertices in the polygon given. The value of N satisfies $3 \leq N \leq 10000$. The next N lines contain a pair of integers each, x and y , specifying the coordinates of each vertex of the polygon in counterclockwise order (the z -coordinate of each vertex is always 0). These coordinate values will always be between -10000 and 10000 inclusive. The last line consists of two real numbers $0 \leq R \leq 100$ and $1 \leq D \leq 1000$. These two numbers specify the light source: it is a flat circular source of uniform intensity along the plane $z = D$, centred on $(0, 0, D)$ and radius R . The wall is a flat, smooth and infinitely large surface along the plane of $z = -D$. You may assume that there is no other light source in the vicinity. You may also assume that the input data will always be such that it specifies a polygon of non-zero area and that the same point will not be present in the data multiple times.

The last input is followed by a single '0' on the line, which should not be processed.

Output

There should be one single line of output for each test case. It should be formatted as 'Case c : u p ', where c is the case number with the first case designated as case 1, u is the area of the umbra (the part of the shadow that is completely dark, usually near the center) and p is the area of the rest of the shadow (the lighter part of the shadow that is only partially dark) called penumbra. The diagram above should help clarify the terms. Both areas should be within a relative error bound of $1e-6$ of the actual value. See the sample output below for further clarification.

Sample Input

```
5
0 0
10 0
10 10
1 10
0 9
5 10
0
```

Sample Output

```
Case 1: 100.000000 770.681952
```