You are an IT and Statistics specialist in the National Bomb Defusing Squad. You have to use your programming and statistical skills to find out locations of probable suicide bomb attacks. Sometimes you also have to find the blast radius of bombs from the death toll. In this specific problem, you will have to find the expected number of deaths due to a suicide bomb attack in a crowded place. For simplicity, in this problem you can assume:

(1) Each person can be considered as a point in a 2-D Cartesian plane. So an (x, y) coordinate can be used to denote the location of a person. Here x and y are always non-negative integers.

(2) All the persons in a scenario are equally likely to carry a bomb. But exactly one person will carry the bomb.

(3) More than one person can be at the same coordinate.

(4) A suicide bomb has a blasting radius R. Everyone within the blasting radius dies. For example a person at location (p, q) is carrying a bomb and there is another person at (m, n). When the bomb explodes at (p, q) the person at (m, n) will die if his distance from (p, q) is not more than R. The problem for you to solve is — for a given scenario you will have to calculate the death toll for up to thousand values of R.

Input

The input file contains maximum 7 test cases.

First line of each test case contains two integers N $(0 < N \leq 3000)$ and Q $(0 < Q \leq 1000)$. Here N denotes the number of people in the scenario and Q denotes the number of queries to follow. Each of the next N lines contains two integers (x_i, y_i) that denotes the Cartesian coordinate of one person in the scenario. These integers are non negative and do not exceed 25000. Each of the next Q lines contains a single integer R_j $(0 < R_j < 40001)$. Input is terminated by a line containing two zeroes.

Output

For each set of input produce Q lines of outputs. Each of this Q lines contains output for one query — the expected number of deaths if the blast radius of the bomb is R_j. This value should be rounded to two digits after the decimal point. **Print a blank line after the output for each test case.**

Sample Input

```
4 3
1 1
1 2
12 3
40 40
1
10
100
0 0
```

Sample Output

```
1.50
1.50
4.00
```