2778 Radius of the Holedox

An ironclad disk, named *Holedox*, planned to attack the Mars. It would move in a plane, under the control of an automated program. The automated program is a sequence of (dx, dy) pairs, indicating differences of x-coordinates and y-coordinates between two contiguous positions of the disk center in the plane. For each movement, say from (x, y) to $(x + dx, y + dy)$, the *Holedox* would keep still at (x, y) for a period, move to $(x + dx, y + dy)$ rapidly, and then keep still again at $(x + dx, y + dy)$ to wait for the next movements.

However, the *Holedox* would be detected by the Mars Security Bureau (MSB), who set up m sensors given by their coordinates (x_i, y_i) in the plane where $1 \le i \le m$. Because the *Holedox* moved incredibly fast, it could be detected only when it was keeping still. At that time, every sensor would activate its state to 1 when it was inside *Holedox*, deactivate its state to 0 when it was outside the *Holedox*, and set its state to 0 or 1 randomly when it was exactly on the *Holedox*.

On Sept. 11th of 3001, the attack began. However, after t movements of the *Holdedox*, the MSB intercepted its automated program. Now, based on the program and the $t + 1$ previous sensor states, could you please write a program to help the MSB to calculate the minimum possible radius of the *Holedox*?

To understand the problem clearly, let us consider the following instance described in the Sample Input. Suppose there were three sensors which are $A(0, 0)$, $B(1, 0)$ and $C(1, 1)$.

As shown in Figure 1, accordingly to the program of "(1,0)(0,2)", a unit-radius disk moved from (0,0), to (1,0), and to (1,2) during the first three still periods. Therefore, the sensor states of A, B and C might become (1,1,0), (1,1,1) and (0,0,1) respectively, consonant with the observations given in the input. Besides that, it is easy to verify that the unit size is the minimum possible radius for the sample input.

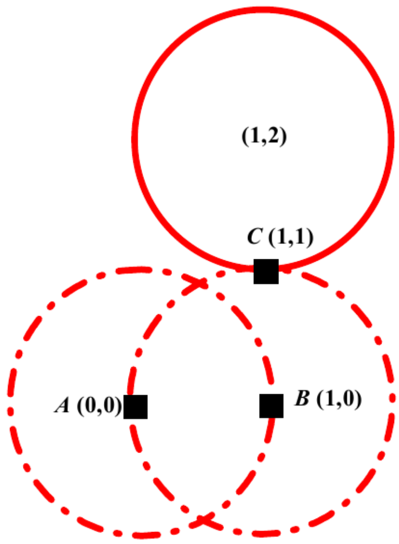

Fig 1. Sample Instance

Input

The input file consists of several test cases. Each case begins with an integer m to indicate the number of sensors, whose coordinates, (x_i, y_i), are pairs of real numbers, given in the next m lines, where $1 \le m \le 50$ and $1 \le i \le m$. Then, an integer t, where $0 \le t \le 500$, is given to indicate the number of previous movements, followed by a line of m binaries representing the sensor states when the *Holedox* started to move. Among the next $2t$ lines, the $(2j - 1)$-th line gives (dx_j, dy_j) for the j-th movement in the intercepted program, and the $(2j)$-th line gives m binaries which is the sensor states after the j-th movement of the *Holedox*, where $1 \le j \le t$. Finally, each case is terminated by an empty line, while the input file is terminated by $m = 0$.

Output

For each test case, output a line of a real number, rounded to the third place after the decimal point, to indicate the minimum possible radius of the *Holedox*. If no possible radius exits for a certain case, please output '`-1.000`' instead.

Sample Input

```
3
0 0
1 0
1 1
2
1 1 0
1 0
1 1 1
0 2
0 0 1

0
```

Sample Output

```
1.000
```