

# GONDOR

[English](#)

[Vietnamese](#)

The legendary land of Gondor had a network of sparks to quickly alert the entire land of an emergency.

Every spark is manned by an archer with several arrows and instruction in which order to light the other sparks.

More precisely, when his own spark is lit, the archer next to it lights his arrows and shoots one at every other spark that has not yet been lit, in the order in which his instructions say. The archer does so until he is out of arrows (or sparks to shoot at). The archers are very precise so every arrow hits its target. The time or an arrow to travel some distance is equal to that distance, while the time or an archer to shoot all his arrows is negligible. Sauron's army is approaching Gondor so the spark at Minas Tirith has been lit. Write a program that, given the layout o sparks in the coordinate plane, the number of arrows and instructions or every archer, calculates the time indices at which each of the sparks will be lit.

## Input

The first line contains an integer  $N$  ( $1 \leq N \leq 100$ ), the number o sparks. The sparks are numbered from 1 to  $N$ . The spark in Minas Tirith, which has been lit at time 0, is spark number 1.

Each of the following  $N$  lines describes one spark. The description of one spark is composed of :

The integers  $X$  and  $Y$  ( $1 \leq X, Y \leq 1000$ ), the coordinates of the spark;

An integer  $S$  ( $1 \leq S \leq 100$ ), the number of arrows;

$N-1$  distinct integers between 1 and  $N$ , the instructions or the archer. The instructions are the order in which, once his spark is lit, the archer will consider shooting arrows at other sparks.

No number will appear more than once in the list, nor will an archer be instructed to shoot an arrow at his own spark.

The input will be such that no two sparks will be lit at the same time.

## Output

Output  $N$  decimal numbers, each on a single line, the times at which the sparks light up, in order from spark 1 to  $N$ . Your output must be accurate to  $\pm 0.01$ .

## Example

**Input:**

```
4
1 1 1 2 3 4
1 2 1 4 1 3
2 1 1 2 1 4
```

2 2 1 3 2 1

**Output:**

0.000000

1.000000

3.000000

2.000000

**Input:**

5

4 3 2 5 2 4 3

4 5 1 4 1 5 3

4 4 1 1 4 5 2

2 4 1 5 2 3 1

3 4 2 2 4 3 1

**Output:**

0.000000

2.000000

4.414214

2.414214

1.414214