TASK	GONDOR	MAJMUN	JEZERO
input	standard input		
output	standard output		
time limit (per test case)	1 second		
memory limit (heap+stack)	64 MB		
points	**30**	**50**	**70**
	150		

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 for an arrow to travel some distance is equal to that distance, while the time for 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 of sparks in the coordinate plane, the number of arrows and instructions for 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 of sparks. The sparks are numbered 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 for 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.

Note: 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 ±0.001.

EXAMPLE TEST DATA

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
```

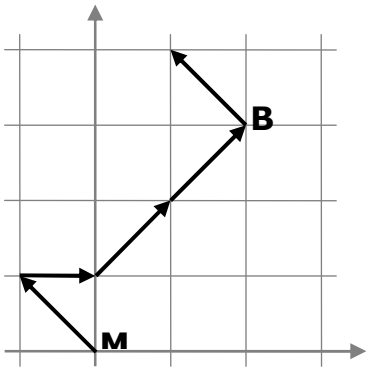
output
```
0.000000
2.000000
4.414214
2.414214
1.414214
```

Coming home after a hard day at school, Ivica is ready to relax playing the computer game "Monkey & banana". In the game, the monkey is located in a jungle, modeled as a coordinate plane. Every point with integer coordinates represents a tree.

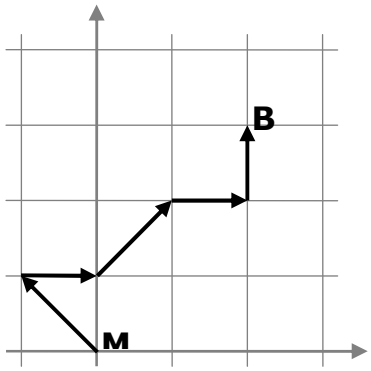
The monkey is initially located at tree (X_M, Y_M) facing up i.e. towards the tree (X_M, Y_M+1). The monkey is controlled with the keys 0 to 7. When the key K is pressed, the monkey turns 45 degrees left K times and then jumps to the **first tree he sees in his new direction**.

The game lasts until the monkey jumps exactly N times. After that, the score is calculated from the distance between the monkey and the banana tree, which is located at coordinates (X_B, Y_B). The lower the distance, the bigger the score.

Ivica played one game and is now interested if he could have done better **changing at most one key press**. Write a program that determines the smallest possible ending (Euclidean) distance between the monkey and the banana tree (it is possible that the current score cannot be improved).



The first example.



Solving the first example – the fourth key press should be 7.

INPUT

The first line of input contains four integers X_M, Y_M, X_B and Y_B ($0 \le X_M, Y_M, X_B, Y_B \le 1\,000\,000$), the initial coordinates of the monkey and the coordinates of the banana tree.

The next line contains the integer N ($1 \le N \le 100\,000$), the number of jumps (key presses) in the game played.

The last line contains a string of N digits between 0 and 7, the keys that Ivica pressed.

OUTPUT

Output a single decimal number, the smallest achievable distance. Your output must be accurate to ±0.001.

EXAMPLE TEST DATA

input
```
0 0 2 3
5
15102
```

output
```
0.000000
```

input
```
5 5 10 5
3
000
```

output
```
2.000000
```

input
```
0 0 10 10
9
700003000
```

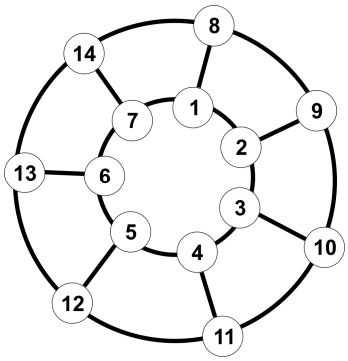
output
```
1.414214
```

Ivica has decided to take a walk in a nearby park. There is a small lake in the park, and an island in the middle of the lake, with N fountains arranged in a circle, connected by paths. The outer edge of the lake also contains N fountains arranged in a circle, connected by paths.

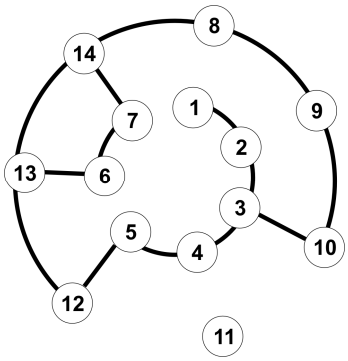
Each fountain on the island (inner fountain) is also connected by a bridge with exactly one outer fountain. More precisely, if the inner fountains are numbered 1 to N in clockwise order, and the outer fountains are numbered N+1 do 2·N, also clockwise, then there is a bridge connecting fountains F and N+F.

Therefore, the park contains exactly 2·N fountains and 3·N paths (including bridges), is in the left figure below.

Some of the paths are being cleaned by volunteers and are temporarily unusable. An example park with some of the paths unusable is shown in the right figure.



Example park for N=7. All paths are usable.



Same park with some of the paths unusable. The figure corresponds to the first example.

Ivica starts his walk near some fountain. He proceeds to use paths so that he never visits the same fountain or uses the same path twice. The walk ends when Ivica reaches the fountain where he started.

Write a program that calculates the **number of distinct walks** Ivica can make, and outputs the remainder when that number is **divided by 1 000 000 007**. Two walks are different if they don't contain the same paths (so the starting fountain and order of traversal don't matter). For the park in the right image above, there are three walks: 13-6-7-14-13, 8-9-10-3-4-5-12-13-14-8, and 8-9-10-3-4-5-12-13-6-7-14-8.

INPUT

The first line contains the integer N ($2 \leq N \leq 100\,000$), the number of inner or outer fountains.

Each of the following three lines contains a string of N characters '0' and '1' describing the availabilities of the paths. A zero in some position represents an unavailable path, while a one represents an available path. The three strings are:

1. The paths connecting the inner fountains in a cycle. The paths are given in clockwise order, starting with the path connecting fountains N and 1.
2. The paths connecting the outer fountains in a cycle. The paths are given in clockwise order, starting with the path connecting fountains 2·N and N+1.
3. The bridges. They are given in clockwise order, starting with the bridge connecting fountains 1 and N+1.

OUTPUT

Output the number of distinct walks modulo 1 000 000 007.

EXAMPLE TEST DATA

input
```
7
0111101
1110011
0010111
```

output
```
3
```

input
```
8
11111111
11111111
00001000
```

output
```
2
```

input
```
9
010111110
010111000
111101010
```

output
```
4
```