Codeforces Beta Round #28 (Codeforces format)

A. Bender Problem

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Robot Bender decided to make Fray a birthday present. He drove n nails and numbered them from 1 to n in some order. Bender decided to make a picture using metal rods. The picture is a closed polyline, which vertices should be nails (in the given order). The segments of the polyline should be parallel to the coordinate axes. Polyline is allowed to have self-intersections. Bender can take a rod and fold it exactly once in any place to form an angle of 90 degrees. Then he can attach the place of the fold to some unoccupied nail and attach two ends of this rod to adjacent nails. A nail is considered unoccupied if there is no rod attached to it (neither by it's end nor the by the fold place). No rod could be used twice. It is not required to use all the rods.

Help Bender to solve this difficult task.

Input

The first line contains two positive integers n and m ($4 \le n \le 500, 2 \le m \le 500$, n is even) — the amount of nails and the amount of rods. i-th of the following n lines contains a pair of integers, denoting the coordinates of the i-th nail. Nails should be connected in the same order as they are given in the input. The last line contains m integers — the lenghts of the rods. All coordinates do not exceed 10^4 by absolute value. Lengths of the rods are between 1 and $200\,000$. No rod can be used twice. It is guaranteed that all segments of the given polyline are parallel to coordinate axes. No three consecutive nails lie on the same line.

Output

If it is impossible to solve Bender's problem, output `NO`. Otherwise, output `YES` in the first line, and in the second line output n numbers — i-th of them should be the number of rod, which fold place is attached to the i-th nail, or `-1`, if there is no such rod.

If there are multiple solutions, print any of them.

Sample test(s)

input

```
4 2
0 0
0 2
2 2
2 0
4 4
```

output

```
YES
1 -1 2 -1
```

input

```
6 3
0 0
1 0
1 1
2 1
2 2
0 2
3 2 3
```

output

```
YES
1 -1 2 -1 3 -1
```

input

```
6 3
0 0
1 0
1 1
2 1
2 2
0 2
2 2 3
```

output

```
NO
```

B. pSort

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day n cells of some array decided to play the following game. Initially each cell contains a number which is equal to it's ordinal number (starting from 1). Also each cell determined it's favourite number. On it's move i-th cell can exchange it's value with the value of some other j-th cell, if $|i - j| = d_i$, where d_i is a favourite number of i-th cell. Cells make moves in any order, the number of moves is unlimited.

The favourite number of each cell will be given to you. You will also be given a permutation of numbers from 1 to n. You are to determine whether the game could move to this state.

Input

The first line contains positive integer n ($1 \le n \le 100$) — the number of cells in the array. The second line contains n distinct integers from 1 to n — permutation. The last line contains n integers from 1 to n — favourite numbers of the cells.

Output

If the given state is reachable in the described game, output `YES`, otherwise `NO`.

Sample test(s)

input
5 5 4 3 2 1 1 1 1 1 1
output
YES

input
7 4 3 5 1 2 7 6 4 6 6 1 6 6 1
output
NO

input
7 4 2 5 1 3 7 6 4 6 6 1 6 6 1
output
YES

C. Bath Queue

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n students living in the campus. Every morning all students wake up at the same time and go to wash. There are m rooms with wash basins. The i-th of these rooms contains a_i wash basins. Every student independently select one the rooms with equal probability and goes to it. After all students selected their rooms, students in each room divide into queues by the number of wash basins so that the size of the largest queue is the least possible. Calculate the expected value of the size of the largest queue among all rooms.

Input

The first line contains two positive integers n and m $(1 \le n, m \le 50)$ — the amount of students and the amount of rooms. The second line contains m integers $a_1, a_2, \ldots, a_m$ $(1 \le a_i \le 50)$. a_i means the amount of wash basins in the i-th room.

Output

Output single number: the expected value of the size of the largest queue. Your answer must have an absolute or relative error less than 10^{-9}.

Sample test(s)

input
1 1 2
output
1.0000000000000000000

input
2 2 1 1
output
1.5000000000000000000

input
2 3 1 1 1
output
1.3333333333333350000

input
7 5 1 1 2 3 1
output
2.5021696000000070000

D. Don't fear, DravDe is kind

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A motorcade of n trucks, driving from city «Z» to city «3», has approached a tunnel, known as Tunnel of Horror. Among truck drivers there were rumours about monster DravDe, who hunts for drivers in that tunnel. Some drivers fear to go first, others - to be the last, but let's consider the general case. Each truck is described with four numbers:

- v — value of the truck, of its passangers and cargo
- c — amount of passanger on the truck, the driver included
- l — total amount of people that should go into the tunnel before this truck, so that the driver can overcome his fear («if the monster appears in front of the motorcade, he'll eat them first»)
- r — total amount of people that should follow this truck, so that the driver can overcome his fear («if the monster appears behind the motorcade, he'll eat them first»).

Since the road is narrow, it's impossible to escape DravDe, if he appears from one side. Moreover, the motorcade can't be rearranged. The order of the trucks can't be changed, but it's possible to take any truck out of the motorcade, and leave it near the tunnel for an indefinite period. You, as the head of the motorcade, should remove some of the trucks so, that the rest of the motorcade can move into the tunnel and the total amount of the left trucks' values is maximal.

Input

The first input line contains integer number n ($1 \le n \le 10^5$) — amount of trucks in the motorcade. The following n lines contain four integers each. Numbers in the i-th line: v_i, c_i, l_i, r_i ($1 \le v_i \le 10^4, 1 \le c_i \le 10^5, 0 \le l_i, r_i \le 10^5$) — describe the i-th truck. The trucks are numbered from 1, counting from the front of the motorcade.

Output

In the first line output number k — amount of trucks that will drive into the tunnel. In the second line output k numbers — indexes of these trucks in ascending order. Don't forget please that you are not allowed to change the order of trucks. If the answer is not unique, output any.

Sample test(s)

input
5 1 1 0 3 1 1 1 2 1 1 2 1 1 1 3 0 2 1 3 0
output
4 1 2 3 5

input
5 1 1 0 3 10 1 2 1 2 2 1 1 10 1 1 2 3 1 3 0
output
3 1 3 5

E. DravDe saves the world

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

How horrible! The empire of galactic chickens tries to conquer a beautiful city «Z», they have built a huge incubator that produces millions of chicken soldiers a day, and fenced it around. The huge incubator looks like a poligon on the the plane Oxy with n vertices. Naturally, DravDe can't keep still, he wants to destroy the chicken empire. For sure, he will start with the incubator.

DravDe is strictly outside the incubator's territory in point $A(x_a, y_a)$, and wants to get inside and kill all the chickens working there. But it takes a lot of doing! The problem is that recently DravDe went roller skating and has broken both his legs. He will get to the incubator's territory in his jet airplane LEVAP-41.

LEVAP-41 flies at speed $V(x_v, y_v, z_v)$. DravDe can get on the plane in point A, fly for some time, and then air drop himself. DravDe is very heavy, that's why he falls vertically at speed F_{down}, but in each point of his free fall DravDe can open his parachute, and from that moment he starts to fall at the wind speed $U(x_u, y_u, z_u)$ until he lands. Unfortunately, DravDe isn't good at mathematics. Would you help poor world's saviour find such an air dropping plan, that allows him to land on the incubator's territory? If the answer is not unique, DravDe wants to find the plan with the minimum time of his flight on the plane. If the answers are still multiple, he wants to find the one with the minimum time of his free fall before opening his parachute

Input

The first line contains number n ($3 \le n \le 10^4$) — amount of vertices of the fence. Then there follow n lines containing the coordinates of these vertices (two integer numbers x_i, y_i) in clockwise or counter-clockwise order. It's guaranteed, that the fence does not contain self-intersections.

The following four lines contain coordinates of point $A(x_a, y_a)$, speeds $V(x_v, y_v, z_v)$, F_{down} and speed $U(x_u, y_u, z_u)$. All the input numbers are integer. All the coordinates don't exceed 10^4 in absolute value. It's guaranteed, that $z_v > 0$ and $F_{down}, z_u < 0$, and point A is strictly outside the incubator's territory.

Output

In the first line output two numbers t_1, t_2 such, that if DravDe air drops at time t_1 (counting from the beginning of the flight), he lands on the incubator's territory (landing on the border is regarder as landing on the territory). If DravDe doesn't open his parachute, the second number should be equal to the duration of DravDe's falling down. If it's impossible for DravDe to get to the incubator's territory, output `-1 -1`. If the answer is not unique, output the answer with the minimum t_1. If the answers are still multiple, output the answer with the minimum t_2. Your answer must have an absolute or relative error less than 10^{-6}.

Sample test(s)

input

```
4
0 0
1 0
1 1
0 1
0 -1
1 0 1
-1
0 1 -1
```

output

```
1.00000000 0.00000000
```

input

```
4
0 0
0 1
1 1
1 0
0 -1
-1 -1 1
-1
0 1 -1
```

output

```
-1.00000000 -1.00000000
```

input

```
4
0 0
1 0
1 1
0 1
0 -1
1 1 1
-1
1 1 -1
```

output

```
0.50000000 0.00000000
```