SGU 110. Dungeon

time limit per test: 1 sec.
memory limit per test: 4096 KB

The mission of space explorers found on planet M the vast dungeon. One of the dungeon halls is fill with the bright spheres. The explorers find out that the light rays reflect from the surface of the spheres according the ordinary law (the incidence angle is equal to the reflectance angle, the incidence ray, the reflected ray and the perpendicular to the sphere surface lay in the one plane). The ancient legend says that if the light ray will reflect from the spheres in the proper order, than the door to the room with very precious ancient knowledge will open. You are not to guess the right sequence; your task is much simpler. You are given the positions and the radii of the spheres, the place where the laser shot was made and the direction of light propagation. And you must find out the sequence in which the light will be reflected from the spheres.

Input

The first line of input contains the single integer n ($1 \le n \le 50$) - the amount of the spheres. The next n lines contain the coordinates and the radii of the spheres xi, yi, zi, ri (the integer numbers less or equal to 10000 by absolute value). The last line contains 6 real numbers - the coordinates of two points. The first one gives the coordinates of the place of laser shot, and the second gives the direction in which it was made (the second point is the point on the ray). The starting point of the ray lies strictly outside of any sphere.

Output

Your program must output the sequence of sphere numbers (spheres are numbers from 1 as they was given in input), from which the light ray was reflected. If the ray will reflect more the 10 times, than you must output first 10, then a space and the word 'etc.' (without quotes). Notice: if the light ray goes at a tangent to the sphere you must assume that the ray was reflected by the sphere.

Sample Input 1
```
1
0 0 2 1
0 0 0 0 0 1
```
Sample Output 1
```
1
```

Sample Input 2
```
2
0 0 2 1
0 0 -2 1
0 0 0 0 0 100
```
Sample Output 2
```
1 2 1 2 1 2 1 2 1 2 etc.
```

Polygon (Estonia)

TASK

Write a program to find a convex polygon whose sides have the given lengths.

In this task, we consider a polygon to be convex if all its inner angles are strictly greater than 0 degrees and strictly less than 180 degrees.

INPUT

The input file name is `POLY.IN`. The first line of the file contains an integer N, the number of vertices of the polygon ($3 \le N \le 1000$). Each of the following N lines contains an integer a_i, the length of one side of the polygon ($1 \le a_i \le 10{,}000$).

OUTPUT

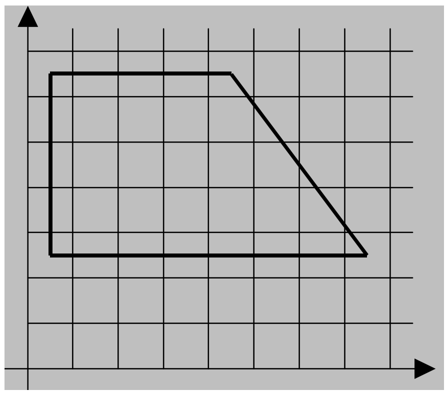
If the desired polygon can be constructed, the output file `POLY.OUT` should contain exactly N lines. Each line should contain two real numbers x_i and y_i ($|x_i| \le 10{,}000{,}000$, $|y_i| \le 10{,}000{,}000$) such that by connecting the points (x_i, y_i) and (x_{i+1}, y_{i+1}) for all $1 \le i < N$ and additionally the points (x_N, y_N) and (x_1, y_1) with line segments, we obtain a convex polygon. The lengths of the line segments must be equal to the numbers given in the input file, but not necessarily in the same order.

The vertices of the constructed polygon can be listed either clockwise or counterclockwise.

If the polygon cannot be constructed, print `NO SOLUTION` on the single line of the output file.

EXAMPLE

```
INPUT                  OUTPUT
4                      0.5 2.5
7                      7.5 2.5
4                      4.5 6.5
5                      0.5 6.5
4
```



GRADING

The grading program considers two lengths equal if they differ by less than 0.001.
Any standard floating point format is acceptable.

SGU 198. Get Out!

time limit per test: 2 sec.
memory limit per test: 65536 KB
input: standard
output: standard

Captain Faraway on his famous circular ship *Kolobok* is lost among the islands of the archipelago that he has just discovered. Now he wonders whether he can get out of there. Help him!

All islands in the archipelago can be composed of pieces that have circular form. You are given the map of archipelago and the position of captain. Find out whether captain can get out of there, i.e. can get as far from the point he is in the beginning as he likes.

Input

The first line contains N — the number of circular island parts ($1 \le N \le 300$). N lines follow, each containing x_i, y_i, r_i — coordinates of center and radius of the i-th circle. All coordinates and radii are real. Objects may overlap with each other in arbitrary way. All objects are considered solid.

The last line of the input file contains three real numbers — coordinates of the center of *Kolobok* and its radius.

You may consider *Kolobok* to be the perfect circle and that it is in the free area in the beginning. *Kolobok* can move along any trajectory and is so strong that he can even touch islands, but no nonzero part of island must intersect *Kolobok* during his motion. You may assume that making calculations with the precision of 10^{-6} is satisfactory.

Output

Output YES if *Kolobok* can leave archipelago and NO if it cannot.

Sample test(s)

Input

```
Test #1

7
2 2 1.1
-2 2 1.1
2 -2 1.0
-2 -2 1.0
```

```
2 -5 1.0
0 -8 1.0
-2 -6 1.0
0 0 1
```

```
Test #2
```

```
5
2 2 1.1
-2 2 1.1
2 -2 1.0
-2 -2 1.0
0 -3 0.01
0 0 1
```

Output

```
Test #1
```

```
YES
```

```
Test #2
```

```
NO
```

POI 2005 Stage 3: Dextrogyrate camel

Byteotia consists of *N* oasis in the desert, no three of which are collinear. Byteasar lives in one of these oasis and moreover he has an acquaintance in every other. Byteasar wants to pay a visit to as many of them as possible. He plans to travel on the back of his camel. The camel is as obstinate as a mule and thus moves in its own peculiar way:

- After departure from an oasis it moves along a straight line, until it gets to another oasis.
- The camel turns only at oasis, but it turns only right (clockwise) and by an angle from the interval [0°;180°] (the camel makes only one turn at an oasis, i.e. it will not turn by e.g. 200° as a result of two subsequent turns by 100°)
- The camel doesn't want to follow its own footprints.

Help Byteasar in planning such a route that he will be able to visit as many friends as possible. It should both begin and end in the oasis where Byteasar lives. It has to consist of segments connecting subsequently visited oasis. The route may not pass through any point two times, except the Byteasar's oasis, where the camel turns up twice: at the beginning and the end of the journey.

Byteasar's camel is initially facing a certain oasis and it has to start moving toward it. The direction the camel faces after returning from the journey is of no importance.

Task

Write a programme that:

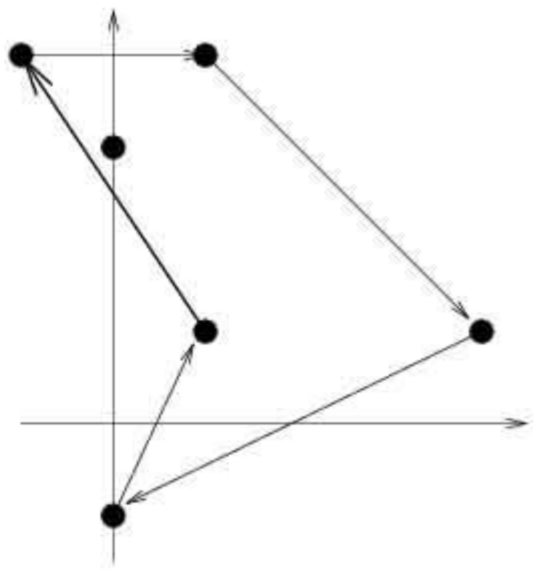
- reads from the standard input the camel's coordinates and the direction it faces as well as the coordinates of the Byteotian oasis,
- determines the maximum number of friends Byteasar can pay a visit to while sticking to the presented rules,
- writes the result to the standard output.

Input

In the first line of the standard input there is one integer *N* ($3 <= N <= 1\,000$) -- the number of oasis in Byteotia. The oasis are numbered from 1 to *N*. Byteasar lives in the oasis no. 1 and his camel is facing the oasis no. 2. In the following *N* lines the coordinates of the oasis are given. In the (i + 1)th line there are two integers x_i, y_i -- the horizontal and vertical coordinate of the *i*th oasis -- separated by a single space. All coordinates are from the interval from -16 000 to 16 000.

Output

In the first and only line of the standard output your programme should write one integer -- the maximum number of friends Byteasar can visit.



Example

For the following input data:

```
6
1 1
-1 4
0 -1
4 1
0 3
1 4
```

the correct answer is:

```
4
```

POI 2005 Stage 3: Special Forces Maneuvers

Secretly Organized Tactical Infantry Exercises (SORTIE) take place in the Błędowska Desert. The main part of the SORTIE shall be the disarmament of a bomb, hidden in an unknown location in the desert.

The first part of the manoeuvres is an airborne operation. The commandos jump individually, in a predetermined order, from a helicopter hovering above the desert. Upon landing, each commando digs in and moves no further. Only then does the next commando proceed.

There is a survival radius defined for each commando. If the distance between the commando and the bomb is equal to (or smaller then) the survival radius, the commando will perish should the bomb go off. The command wants to minimize the number of soldiers taking part in the operation, but it wants to be sure that at least one of them survives the possible explosion.

We shall assume for our purposes that the Błędowska Desert is a plain and we shall identify the positions of the dug in commandos with points on this plane. We are given a sequence of commandos to jump. None of them may miss his turn, i.e. if the *i*-th commando jumps from the helicopter, than all those before him have already jumped. For each of the commandos we know his survival radius and the coordinates of the point he is going to land in, should he be required to jump.

Task

Write a programme which:

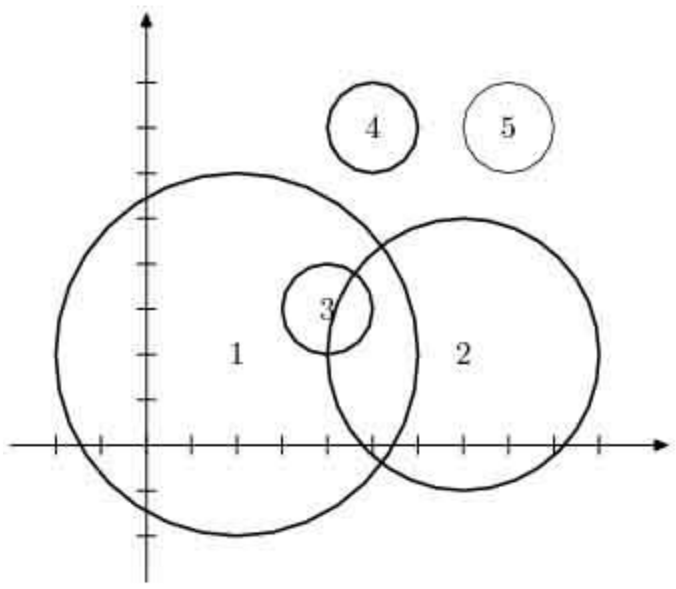
- reads from the standard input a description of each commando,
- calculates the minimal number of commandos to jump,
- writes the outcome to the standard output.

Input

In the first line of the standard input there is a single integer n ($2 <= n <= 2\,000$) -- the number of commandos. The following n lines contain descriptions of commandos -- one per line. The description of each commando is comprised of three integers: x, y and r ($-1000 <= x, y <= 1000$, $1 <= r <= 5000$). The point (x, y) denotes the landing place of the commando, and r denotes his survival radius. If the commando finds himself within the survival radius r from the bomb, he will perish should the bomb go off.

Output

In the first and only line of the standard output your programme should write a single integer - the minimal number of commandos required to jump in order to secure the survival of at least one of them, or a single word *NIE* (*NO* in Polish) if it is not possible to have an unconditional certainty that one of the commandos survives.



Example

For the input data:
```
5
2 2 4
7 2 3
4 3 1
5 7 1
8 7 1
```
the correct outcome is:
```
4
```

Attention!

This task may be solved using floating point types:

- `double` or `extended` in Pascal,
- `double` or `long double` in C and C++.

The use of `float` or `real` types may result in incorrect results due to the floating point arithmetic errors.

SGU 303. Great Berland Wall

Time limit per test: 4 second(s)
Memory limit: 65536 kilobytes
input: standard
output: standard

Berland is in civil war. This time peaceful solution is hardly possible. Collapse to the East and West Berland is inevitable! Commander-in-chief of grey, general Kruglyakovski, has made a decision to hasten the process and to surround his headquarters with a wall (which later be called Great Berland Wall). The wall should be built in such a way that the enemy's headquarters would stay at the other side of the wall.

Looking to the Berland map general Kruglyakovski noticed that the country consists of a number of provinces. Each province has the form of a simple (but possibly not convex) polygon. The country of Berland has no enclaves and does not contain any enclaves of other countries inside, i.e. it is possible to get from one point of Berland to another without leaving the country. The "passability" is known for each segment of the border of each province. "Passability" is the maximum speed the soldier can move with along the corresponding segment.

General Kruglyakovski decided that the wall will pass only along the borders of the provinces and will have the form of a simple polygon. He sent the group of soldiers to build the wall during the night. And you got a task to find such a form and position of the wall that the time of building would be minimal. The time of building of the section of the wall is equal to the "passability" of the corresponding border segment. The time of building of the wall is equal to the sum of building of its segments.

Input
The first line of the input file contains the number of segments of province borders — integer number N ($5 \le N \le 300$). The segments themselves follow further. Each segment is defined by five numbers x_1, y_1, x_2, y_2, v, where (x_1, y_1) and (x_2, y_2) are the coordinates of the segment end points and the integer number v ($1 \le v \le 1000$) is the "passability". Any pair of segments has not more than one common point and this point can only be their common end-point. Input data finishes with one more quadruple of numbers X_1, Y_1, X_2, Y_2, where (X_1, Y_1), (X_2, Y_2) are the coordinates of headquarters of the general Kruglyakovski and his opponent correspondingly. Both headquarters are situated strictly inside one of the provinces. The headquarters are situated in different provinces. All coordinates in the input are integer numbers less than 10^4 by the absolute value.

Output
Write to the first line of the output the total time of building of the wall. Write to the second line the number of border segments the wall will occupy. To the next line write the sequence of the numbers of the segments. The segments are numbered in the order they appear in the input. If there are several solutions choose any of them.

Example(s)

sample input	sample output
13	6
0 6 3 6 9 0 0 4 2 8	6
4 4 6 6 7 2 4 3 6 1	9 10 4 7 5 6
3 6 6 6 1 6 4 6 6 1	
4 2 6 4 1 0 0 0 6 6	
2 2 2 4 1 2 2 4 2 1	
0 6 2 4 5 2 4 4 4 4	
4 2 4 4 3	
3 3	
2 5	

349. Wolves and Sheep

Time limit per test: 3 second(s)
Memory limit: 65536 kilobytes
input: standard
output: standard

The herds of Berland are in danger! Wolves have attacked a pasture of sheep. The shepherd has decided to exterminate wolves in the neighborhood without causing any damage to the sheep. Thus he took a trophy gun, left to him by his grandfather and headed for the ambush. The gun is cast steel and fires with the armour-piercing shells, and the bullets go right through and can hurt a sheep if a wolf is being shot. The wolves and the sheep are represented by segments. The shepherd is in point (0, 0). The flying path of a bullet is a ray coming from point (0, 0). If the path and the segment, characterizing an animal, intersect — the animal dies. Please find the minimum of shots, that is necessary, to kill all the wolves. We rely upon your prudence, for every sheep should remain safe and sound.

Input
The first line describes two integers N and M ($0 \le N \le 10^5$; $0 \le M \le 10^5$) — is the amount of the wolves and the sheep accordingly. It is followed by $N + M$ lines. Every line contains four integer numbers X_1, Y_1, X_2, Y_2 ($-1000 \le X_1, X_2 \le 1000$; $1 \le Y_1, Y_2 \le 1000$), describing the segments. The first N lines describe the disposition of the wolves, the following M lines reveal the situation with the sheep. Segments can degenerate to points.

Output
Print the minimum amount of shots required to kill all the wolves. If you find this request quite impossible to fulfill without killing a single sheep, enter "No solution" (no quotation marks).

Example(s)

sample input	sample output

1 1
5 5 6 7
3 5 8 5
	No solution

sample input	sample output

2 1
1 1 2 3
-5 4 2 2
999 1000 1000 999
	1

POI 2005 Stage 3: Mirror trap

A mirror trap is a cuboid made of mirrors, the reflecting sides of which are facing the interior of the cuboid. Precisely in the geometric centre of the cuboid there is a miniature laser (whose dimensions we shall neglect). The task is to aim the laser in such a way that the beam travels the longest total distance possible and returns to the laser itself. By total distance we shall denote the sum of distances traveled by the laser beam in each of the three directions parallel to the edges of the mirrors (i.e. we are using the so called Manhattan (city) metric).

The dimensions of the trap are even integers. The edges and vertices of the trap, where distinct sides meet, do not reflect the laser beam. Inside the cuboid we shall introduce a cartesian coordinate system. Its axes are parallel to the edges of the trap and the laser shall be placed in the origin. The laser may be aimed at any integer point (a point whose all coordinates are integers) within the trap, the points on the surface of the mirrors included (with the single exception of the laser itself, i.e. the point (0, 0, 0)).

Task

Write a programme which:

- reads from the standard input the dimensions of the mirror trap,
- calculates such a point, that a laser beam fired from the laser it the direction of this point:
  - shall be reflected from the mirrors (but not necessarly from all of them),
  - shall neither intersect an edge nor a vertex of the mirror trap,
  - shall return to the laser, possibly from a different direction,
  - shall travel the longest total distance possible (in the sense of the definition provided).
- writes the outcome to the standard output.

Input

A single test consists of many mirror traps to be analysed. The first line of the standard input contains a single integer 1 <= K <= 1000, denoting the number of traps to be analysed. In the lines 2 ... K + 1 there are descriptions of the traps, a single per line. The description of the trap consists of three numbers 5 <= x, y, z <= 1000, separated by single spaces. The mirror trap has the dimensions of 2x2y2z.

Output

Your programme should write exactly K lines to the standard output. The i-th line should contain a solution for the i-th trap: three integers kx, ky, kz, separated by single spaces, - x <= kx <= x, - y <= ky <= y, - z <= kz <= z, (kx, ky, kz) <> (0, 0, 0). Those numbers signify that in the i-th trap the laser should be aimed at the point, whose coordinates are (kx, ky, kz).

Should there be a greater number of correct solutions, your programme ought to write out any one of them.