Problem A. Elementary

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

The famous black pearl of the Borgias has been stolen and Sherlock is on the case. Now, the first step that needs to be taken is determining some parameters of the valuable jewel, in particular its volume. Luckily Scotland Yard is in possession of a 3D model of the missing gem, which, surprisingly enough for a pearl, is a tetrahedron. The task is obviously too boring for Sherlock, so he asked you to perform the calculation for him.

Input

The first line of input contains the number of test cases t ($t \leq 100$). The input for a single test case consists of four lines.The i-th of them contains three integers x_i, y_i, z_i ($-1000 \leq x_i, y_i, z_i \leq 1000$), which denote the coordinates of the i-th vertex of the tetrahedron.

Output

For each test case in the only line of output print a single floating point number representing the volume of the pearl. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Examples

standard input	standard output
2 0 0 0 10 0 0 0 10 0 0 0 10 1 2 3 6 5 4 12 8 9 11 12 13	166.666667 16.666667

Problem B. Cross Spider

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

The Bytean cross spider (*Araneida baitoida*) is known to have an amazing ability. Namely, it can instantly build an arbitrarily large spiderweb as long as it is contained in a single plane. This ability gives the spider an opportunity to use a fancy hunting strategy. It does not need to wait until a fly is caught in an already built spiderweb; if only the spider knows the current position of a fly, it can instantly build a spiderweb to catch the fly.

A cross spider has just spotted n flies in Byteasar's garden. Each fly is flying still in some point of a 3D space. The spider is wondering if it can catch all the flies with a single spiderweb. Write a program that answers the spider's question.

Input

The first line of the input contains an integer n ($1 \leq n \leq 100\,000$). The following n lines contain a description of the flies in a 3D space: the i-th line contains three integers x_i, y_i, z_i ($-1\,000\,000 \leq x_i, y_i, z_i \leq 1\,000\,000$) giving the coordinates of the i-th fly (a point in a 3-dimensional Euclidean space). No two flies are located in the same point.

Output

Your program should output a single word `TAK` (i.e., *yes* in Polish) if the spider can catch all the flies with a single spiderweb. Otherwise your program should output the word `NIE` (*no* in Polish).

Examples

standard input	standard output
4 0 0 0 -1 0 -100 100 0 231 5 0 15	TAK
4 0 1 0 -1 0 -100 100 0 231 5 0 15	NIE

Problem C. Line Teleportation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

In a 3d space (which is indeed a space in regular sense) there are n lines each of which is a teleporter. It means that a spaceship that reaches any point of a line may instantly teleport to any other point of the same line.

Spaceship moves with a unit speed and is now located at the point (x_1, y_1, z_1). What is the minimum time needed for it to reach the point (x_2, y_2, z_2)?

Input

The first line of input contains a single integer n ($0 \le n \le 100$).

In the following two lines there are six integers x_1, y_1, z_1 and x_2, y_2, z_2 — the coordinates of the start point and the end point.

Each of the following n lines contains six integers $x_{i,1}, y_{i,1}, z_{i,1}, x_{i,2}, y_{i,2}, z_{i,2}$, the coordinates of two distinct points defining the i-th line. Some of the lines may coincide.

All the coordinates in the input are integers not exceeding 10^4 by the absolute value.

Output

Output the minimum time needed for spaceship to reach the finish point from the start point. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Example

standard input	standard output
2 0 0 0 1 1 1 0 0 0 1 0 0 1 1 0 1 1 1	1.000000

Problem D. Segment Distance in 3D

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You are given two segments in a 3d space. You have to find the distance between these two segments. Distance between segments is defined as a minimum distance between any point of the first segment and any point of the second segment.

Input

The first line contains six space-separated integers, the endpoints of the first segment.

The second line contains the description of the second segment in a similar manner.

It is guaranteed that each of the segments has distinct endpoints.

All the coordinates do not exceed 1000 by the absolute value.

Output

Output the distance between two segments. Your answer will be considered correct if its relative or absolute error does not exceed 10^{-6}.

Examples

standard input	standard output
0 0 0 1 0 0 0 2 0 0 2 1	2.0000000000
2 1 0 0 1 0 0 0 0 3 3 0	0.0000000000

Problem E. 3D Printing

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

3D printing is a technique for manufacturing items from a digital template. The printer lays down layers of a polymer material, building an entire 3D object as a series of at plates of varying shapes, stacked upon one another. The polymer is initially sticky enough so that the plates printed on top of one another will adhere. After the object dries or cures, the resulting objects can be quite durable.

Consider a 3D printer in which objects to be printed are described as a template, consisting of a combination of several convex polyhedra (i.e., at-surfaced objects such that a line from one interior point to another interior point never passes outside the volume of the object). Write a program to determine the total volume of polymer required to sculpt an object from a given template.

Input

There will be several test cases in the input. Each test case will begin with a line with a single integer n ($1 \leq n \leq 100$) representing the number of polyhedra in that template.

The subsequent lines describe the n polyhedra. Each polyhedron begins with a line containing an integer f ($3 < f < 30$), which is the number of faces on the polyhedron. Following that line is a series of lines describing polygons that comprise the faces. Each such line begins with an integer v ($3 \leq v \leq 24$), which is the number of vertices. Following v, on the same line will be $3v$ real numbers, representing the v vertices as (x, y, z) coordinates. For example, if $v = 3$, then the line would be $v\ x_1\ y_1\ z_1\ x_2\ y_2\ z_2\ x_3\ y_3\ z_3$.

All the coordinates will be between -100 and 100, inclusive. Vertices are presented in sequential order; there will be an edge of the polygon from (x_1, y_1, z_1) to (x_2, y_2, z_2), from (x_2, y_2, z_2) to (x_3, y_3, z_3) and so on.

The polygons are closed, so there is an implied edge from the last vertex in a polygon back to the first. All of the vertices of a face will be coplanar. Edges will not cross, and each vertex will lie on exactly two edges. No three (or more) vertices in a polygon will be collinear. None of the polyhedra in any given test case will overlap. The input will end with a line with a single 0, which shouldn't be proceeded.

Output

For each template, print a real number on its own line, indicating the volume of polymer required in cubic centimeters. The volume should be printed with a precision of 0.01.

Example

standard input	standard output
2 6 4 10 10 0 10 15 0 15 15 0 15 10 0 4 10 10 0 10 15 0 10 15 20 10 10 20 4 10 15 0 15 15 0 15 15 20 10 15 20 4 15 15 0 15 10 0 15 10 20 15 15 20 4 10 10 0 15 10 0 15 10 20 10 10 20 4 10 10 20 10 15 20 15 15 20 15 10 20 6 4 0 0 0 0 25 0 25 25 0 25 0 0 4 0 0 0 0 25 0 0 25 0.5 0 0 0.5 4 0 25 0 25 25 0 25 25 0.5 0 25 0.5 4 25 25 0 25 0 0 25 0 0.5 25 25 0.5 4 25 0 0 0 0 0 0 0 0.5 25 0 0.5 4 0 0 0.5 0 25 0.5 25 25 0.5 25 0 0.5 0	812.500

Problem F. 3D Model

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given N points in 3-dimensional space. Consider the convex polygon of minimum space, containing all those points. Your task is to print number a_k of faces with k vertices for each integer k such as $a_k \neq 0$.

Input

First line of the input contains one integer T — number of test cases ($1 \leq T \leq 100$). Each test case starts with line consisting one integer N — number of pointf ($4 \leq N \leq 30$). Each of next N lines contain three integers X, Y and Z ($-1000 \leq X, Y, Z \leq 1000$) — coordinates of the points. It is guaranteed that any convex polygon containing those points have nonzero volume.

Output

For each test case print the line "`Case #A:`", where A is number of test-case (1-based). Then print M lines: number of vertices of face k_i, colon ('`:`'), space and number of faces with k_i vertices a_{k_i}, Lines must be printed in the increasing order of k_i.

Examples

standard input	standard output
2 6 0 0 0 2 0 0 0 2 0 2 2 0 3 1 0 1 1 1 5 0 0 0 1 1 0 0 2 0 2 1 0 1 1 1	Case #1: 3: 5 5: 1 Case #2: 3: 4

Problem G. Asteroids

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Association of Collision Management (ACM) is planning to perform the controlled collision of two asteroids. The asteroids will be slowly brought together and collided at negligible speed. ACM expects asteroids to get attached to each other and form a stable object.

Each asteroid has the form of a convex polyhedron. To increase the chances of success of the experiment ACM wants to bring asteroids together in such manner that their centers of mass are as close as possible.

To achieve this, ACM operators can rotate the asteroids and move them independently before bringing them together. Help ACM to find out what minimal distance between centers of mass can be achieved. For the purpose of calculating center of mass both asteroids are considered to have constant density.

Input

Input file contains two descriptions of convex polyhedra.

The first line of each description contains integer number n — the number of vertices of the polyhedron ($4 \le n \le 60$). The following n lines contain three integer numbers x_i, y_i, z_i each — the coordinates of the polyhedron vertices ($-10^4 \le x_i, y_i, z_i \le 10^4$).

It is guaranteed that the given points are vertices of a convex polyhedron, in particular no point belongs to the convex hull of other points. Each polyhedron is non-degenerate.

The two given polyhedra have no common points.

Output

Output one floating point number — the minimal distance between centers of mass of the asteroids that can be achieved. Your answer must be accurate up to 10^{-5}

Example

standard input	standard output
8 0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1 5 0 0 5 1 0 6 -1 0 6 0 1 6 0 -1 6	0.750000

Problem H. Connected Rings

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

There are two circles with radius 1 in 3D space. Please check two circles are connected as chained rings.

Input

Input consists of two ring descriptions. Each description consists of seven integers: center coordinates c_x, c_y, c_z, coordinates of the vector normal to the plane containing the ring n_x, n_y, n_z and positive radius r. All numbers in the input do not exceed 100 by the absolute value.

It is guaranteed that the rings do not have common points.

Output

Output `YES` if the rings are connected and `NO` otherwise.

Examples

standard input	standard output
0 0 0 0 0 1 1 0 1 0 1 0 0 1	YES
0 0 0 0 0 1 1 0 0 0 1 0 0 2	NO

Problem I. Black hole

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

In 3141 the black hole problem become very big for the citizens of San Francisco. San Francisco is located not on the Earth, but in space. It is a network of orbit stations which are connected between each other with tunnels. The tunnels could not be broken or become disconnected. The tunnels can freely pass through each other.

In this problem you can assume that San Francisco is a closed polyline in *3D* space without self-intersection and touching.

The black hole can trap the orbit stations and tunnels. To protect San Francisco from that it was decided to build the special devices in the open space. This devices generate the lines. Nothing could intersect these lines, neither orbit station, nor tunnel. The devices were placed in the space. But nobody sure that the devices were located correctly.

The devices are located correctly if there is no black hole that could trap San Francisco — the lines will not let tunnels pass through them. A black hole could appear in any point of the space except the orbit station or a tunnel. Also a black hole cannot appear on any line. The tunnels cannot be disconnected. If the devices are located incorrectly then San Francisco can be moved to a black hole completely and the lines can not help.

Input

On the first line of the input file you are given a single integer n — the number of vertices of the polyline ($3 \le n \le 1000$). On the next n lines you are given the description of the vertices of the polyline — integers x, y, z ($-10^4 \le x, y, z \le 10^4$). On the next line you are given one single integer m ($1 \le m \le 1000$) — number of lines. All lines are parallel to the OZ axe. On the next m lines you are given lines description — integer coordinates x, y ($-10^4 \le x, y \le 10^4$). It is guaranteed that all the given lines do not have common points with polygon.

Output

Output one single word "`Yes`", if the polyline could be adsorbed to some point not located on the devices without breaking its edges, otherwise output "`No`".

Examples

standard input	standard output
4 0 0 0 2 0 0 2 2 0 0 2 0 1 1 1	No
4 0 0 0 2 0 0 2 2 0 0 2 0 1 3 3	Yes

Problem J. The Worm in the Apple

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Willy the Worm was living happily in an apple — until some vile human picked the apple, and started to eat it! Now, Willy must escape!

Given a description of the apple (defined as a convex shape in 3D space), and a list of possible positions in the apple for Willy (defined as 3D points), determine the minimum distance Willy must travel to get to the surface of the apple from each point.

Input

Input file will begin with a line with a single integer n ($4 \leq n \leq 1,000$), which tells the number of points describing the apple. On the next n lines will be three integers x, y and z ($-10,000 \leq x, y, z \leq 10,000$), where each point (x, y, z) is either on the surface of, or within, the apple. The apple is the convex hull of these points. No four points will be coplanar. Following the description of the apple, there will be a line with a single integer q ($1 \leq q \leq 10^5$), which is the number of queries — that is, the number of points where Willy might be inside the apple. Each of the following q lines will contain three integers x, y and z ($-10,000 \leq x, y, z \leq 10,000$), representing a point (x, y, z) where Willy might be. All of Willy's points are guaranteed to be inside the apple.

Output

For each query, output a single floating point number, indicating the minimum distance Willy must travel to exit the apple. Output this number with 4 decimal places of accuracy. Output each number on its own line, with no spaces, and do not print any blank lines between answers.

Example

standard input	standard output
6 0 0 0 100 0 0 0 100 0 0 0 100 20 20 20 30 20 10 4 1 1 1 30 30 35 7 8 9 90 2 2	1.0 2.8868 7.0 2.00

Problem K. Spheres

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Plane is called tangent to the sphere if its intersection with a sphere consists of exactly one point.

You are given three spheres in a 3d space. Find all of their common tangent planes (i.e. planes that are tangent to each of the spheres).

Input

The first line of input contains an integer T, the number of test cases.

Then there follow T test cases. Each test case consists of three lines, each of which contains three integers x_i, y_i, z_i, r_i ($-500 \le x_i, y_i, z_i \le 500$, $1 \le r_i \le 500$), the center coordinates and the radius of the i-th sphere.

It is guaranteed that the sphere centers are three distinct points not belonging to the same line.

Output

In the first line of output print the integer k, the number of common tangent planes.

In the following k lines output four numbers a_j, b_j, c_j, d_j, coefficients of the plane equation $a_j x + b_j y + c_j z = d_j$, it should also be true that $a_j^2 + b_j^2 + c_j^2 = 1$.

It is recommended to output coefficients with at least eight digits after the decimal point.

Example

standard input	standard output
1 0 0 0 1 0 2 0 1 2 0 0 1	4 0.0000000000000000 1.0000000000000000 0.0000000000000000 1.0000000000000 1.0000000000000000 0.0000000000000000 0.0000000000000000 1.0000000000000 0.0000000000000000 0.0000000000000000 1.0000000000000000 -1.0000000000000 0.0000000000000000 0.0000000000000000 1.0000000000000000 1.0000000000000

Problem L. Cubes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

In a 3d space with Cartesian coordinate system there are two cubes. The following information is known:

- side length of each cube is 2;
- center of each cube is located in the coordinate system origin;
- coordinates of the first cube vertices $A_1A_2A_3A_4A_5A_6A_7A_8$ are the following: $A_1(1,1,1)$, $A_2(1,-1,1)$, $A_3(-1,-1,1)$, $A_4(-1,1,1)$, $A_5(1,1,-1)$, $A_6(1,-1,-1)$, $A_7(-1,-1,-1)$, $A_8(-1,1,-1)$;
- vertices of the second cube $B_1B_2B_3B_4B_5B_6B_7B_8$ are enumerated such that it is possible to translate corresponding vertices of first and second cubes by rotating the first cube (A_1 moves to B_1, A_2 to B_2 and so on)

You are given the coordinates of the second cube. Find the volume of the intersection of these two cubes.

Input

Input consists of no more than 8 lines, each of which contains three real numbers with no more than 20 digits after the decimal points, the coordinates of a vertex of $B_1B_2B_3B_4B_5B_6B_7B_8$.

Output

Output the intersection volume. Your answer will be considered correct if its relative or absolute error does not exceed 10^{-6}.

Examples

standard input
1.0000000000 -1.0000000000 1.0000000000 1.0000000000 -1.0000000000 -1.0000000000 -1.0000000000 -1.0000000000 -1.0000000000 -1.0000000000 -1.0000000000 1.0000000000 1.0000000000 1.0000000000 1.0000000000 1.0000000000 1.0000000000 -1.0000000000 -1.0000000000 1.0000000000 -1.0000000000 -1.0000000000 1.0000000000 1.0000000000
standard output
8.0000000000000000000

standard input
1.4142135623730950488016887242097 0 1 0 -1.4142135623730950488016887242097 1 -1.4142135623730950488016887242097 0 1 0 1.4142135623730950488016887242097 1 1.4142135623730950488016887242097 0 -1 0 -1.4142135623730950488016887242097 -1 -1.4142135623730950488016887242097 0 -1 0 1.4142135623730950488016887242097 -1
standard output
6.62741699796952078000

Problem M. Another Short Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Consider an initial set A consisting of N points in three-dimensional space. Each point is associated with a real value p_i that is equal to the probability that i-th point is included in the final set B. Your task is to calculate the expected number of *border points* for convex hull of B.

A point $p \in B$ is called a *border point* for the convex hull of set B if there is no way to choose points $a, b, c, d \in B$ such that p lies strictly inside the tetrahedron $abcd$.

Input

The first line contains the number of points N ($1 \le N \le 50$). The next N lines describe points of A in the form $p_i\ x_i\ y_i\ z_i$, which are the probability of the i-th point to be included in the final set and the coordinates of the i-th point ($0.00 \le p_i \le 1.00$, $-1000 \le x_i, y_i, z_i \le 1000$). Probabilities are given with exactly two digits after the decimal point.

It is guaranteed that no two points of A coincide, no three points of A are collinear and no four points of A are coplanar.

Output

Output the expected number of *border points*. Your answer will be considered correct if its absolute error is less then 10^{-6}.

Examples

standard input	standard output
5 0.20 0 0 0 0.40 0 0 4 0.60 0 4 0 0.80 4 0 0 0.50 1 1 1	2.4808000000