Problem A. Fraction

Input file:	`frac.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Given are two integers: a and b. Find the value of a/b, written in base-k numeral system.

Input

The input file contains several tests, each on its own line. Each test consists of three integers $1 \le a \le 10\,000$, $1 \le b \le 10\,000$, $2 \le k \le 36$. In the last line of the input file there are two zeros. There are no more than 1000 tests in the input file.

Output

For each set of the input data output a string which contains the representation of a/b in base-k numeral system. Separate integral and fractional parts with a dot. If a/b is integer, do not output the dot. There should not be extra zeros. If a/b is a periodic fraction, put the period in parentheses.

Example

frac.in	standard output
10 5 2 1 2 8 8780 1 29 2794 6083 23 157 7 19 1 12 10 0 0	10 0.4 ACM 0.(ACM) 13.(82DAG5) 0.08(3)

Problem B. No Smoking!

Input file:	`smoking.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

John is a good guy, but he has a bad habit — he is a smoker. Jeffrey is John's friend, and he tries to make John to give up smoking. But John doesn't want to give up, so all Jeffrey's attempts were effectless.

Recently, Jeffrey invented a new way to make his friend to give up. John is somewhat messy, so his cigarettes do not lay in a single pack, but rather lay on a big table without any order. Jeffrey wants to take away some cigarettes such that John cannot notice that. John will not notice that some cigarettes are missing if the number of cigarettes that disappear in a single day does not exceed one. What is more, Jeffrey can take away only cigarettes which intersect some other cigarettes. Help Jeffrey to determine if he can materialize his plan.

Input

A cigarette is represented as a line segment. In the first line of the input file there is a number N ($1 \le N \le 125\,000$) — the number of cigarettes on John's table. The next N lines contain the descriptions of cigarettes: $(i+1)$-th line contains the coordinates of the endpoints of the i-th cigarette — the integer numbers x_1, y_1, x_2, y_2 ($-10\,000 \le x_1, y_1, x_2, y_2 \le 10\,000$).

Output

If Jeffrey can start materializing his plan, output a word "`YES`" in the first line of the output file. The second line should contain two integers i and j — the number of cigarette to take and the number of cigarette it intesects.

If Jeffrey cannot take away a single cigarette, output "`NO`".

Example

smoking.in	standard output
2 0 0 2 2 0 2 2 0	YES 1 2
2 0 0 0 5 5 0 5 10	NO

Problem C. Stars

Input file:	`stars.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

John likes watching stars. But it is difficult to watch the whole sky, so he watches only a certain part of the space, which is bounded by a cube of size $N \times N \times N$. This cube is split in small cubes of size $1 \times 1 \times 1$. There are three types of events:

- in a certain small cube, some new stars appear;
- in a certain small cube, some stars disappear;
- John's friend Jeffrey can visit him and ask to count stars in a certain part of the sky.

Input

The first line of the input file contains an integer N ($1 \le N \le 128$). The coordinates of small cubes are triples of integers ranging from 0 to $N - 1$.

The records of events follow, one record per line. Each line starts with an integer M.

- If M is equal to 1, four numbers follow: x ($0 \le x < N$), y ($0 \le y < N$), z ($0 \le z < N$), K ($-20\,001 < K < 20\,001$) — the coordinates of a small cube and the change in the number of stars.
- If M is equal to 2, six numbers follow: — x_1, y_1, z_1, x_2, y_2, z_2 ($0 \le x_1 \le x_2 < N$, $0 \le y_1 \le y_2 < N$, $0 \le z_1 \le z_2 < N$), which mean that Jeffrey asked to count the number of stars in small cubes (x, y, z) from the area $x_1 \le x \le x_2$, $y_1 \le y \le y_2$, $z_1 \le z \le z_2$.
- If M is equal to 3, then John decided to take a break. Such a record appears only once in the input file and is the last one.

The number of records does not exceed 100 002.

Output

For each Jeffrey's question output a single number on a separate line — the number of stars in a needed region.

Example

stars.in	standard output
2 2 1 1 1 1 1 1 1 0 0 0 1 1 0 1 0 3 2 0 0 0 0 0 0 2 0 0 0 0 1 0 1 0 1 0 -2 2 0 0 0 1 1 1 3	0 1 4 2

Problem D. Tree

Input file:	`tree.in`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Given a weighed tree. Find the distance between two given vertices.

Input

The first line of the input file contains an integer $N \leq 150\,000$ — the number of vertices in the tree. The vertices are numbered from 0 to $N - 1$.

In the following $N - 1$ lines there are triples of integers. Each triple u, v, w correspond to an edge of length w which connects the vertices u and v. Lengths of edges do not exceed 1000.

The next line contains an integer $M \leq 75\,000$ — the number of queries. Next M lines contain two integers each — u, v — the numbers of vertices to measure the distance between them.

Output

For each query output the distance in a separate line.

Example

tree.in	standard output
3 1 0 1 2 0 1 3 0 1 0 2 1 2	1 1 2

Problem E. Martian Army

Input file:	`army.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Many ages ago, Martians started using huge humanoid robots in wars. During the current campaign of conquering the Moon, all Martian army is located in headquarters on Mars, where each soldier controls its own robot using the Internet. There is a strict hierarchy in Martian army — each soldier, except for the general (there is only one general in the army), has a single immediate commander. The Martian Statute allows a soldier to conversate only to his subordinate. All conversations are done through the local network of the headquarters, where each soldier has its own computer. All computers are numbered from 1 to N, where N is the number of soldiers in the army. Robots are numbered in exactly the same way, the i-th robot is controlled from the i-th computer. The computer number of a subordinate is strictly greater than the computer number of his commander. The soldiers, apart from the computer number, are characterized by their *reliability* — a real number, the owner of the i-th computer has the reliability of A_i. Everyone on Mars know that the reliability of the general, which is equal to 1, and that the reliability of privates (privates are soldiers who don't have subordinates) is equal to 0.

One could naively think that the traffic in the headquarters network is free to the army. Unfortunately, it is not true: for each megabyte of traffic between the i-th computer and the computer of the commander of the i-th soldier the main Martian internet provider requires a payment of C_i Martian dollars. The situation is complicated by the fact that the volume of trafiic between any two computers is a state secret, so it's not known even to the provider. The provider does the following: once in a month it sends a bill, and the army accountants write down the total amount of money.

Let the commander and his subordinate have the computer numbers of i and k correspondingly. Due to regulation documents, the traffic between the computers i and k should not be less than $A_i - A_k$. In the beginning of a month, provider knows the hierarchy of the soldiers in the army and prices per megabyte of traffic, but they don't know the A_i values (except for the general and the privates) and surely they don't know the amounts of traffic that will be used by the army accountants to sign the bill. The provider wants to know what is the guaranteed sum of money it will get.

Input

In the first line of the input file the integer N, the size of the army, is given $2 \le N \le 100\,000$. Next $N - 1$ lines contain pairs of numbers K_i, C_i — the computer number of the commander of the owner of the i-th computer and the cost of one megabyte of traffic between the computers i and K_i. $1 \le K_i < i \le N$, $0 \le C_i \le 1000$.

Output

Output the real number — the minimum sum of money for the traffic that is guaranteed. The number should have exactly two digits after decimal point.

Example

army.in	standard output
7 1 10 2 5 2 3 3 1 3 2 3 3	8.00

Problem F. Farm

Input file:	`farm.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There is a farm somewhere on Earth. An old farmer runs the farm, who breeds eagles, beagles and violet piggies. When a new animal is born, the farmer needs to determine who it is. He can distinguish a violet piggie from all others, but to tell apart eagles and beagles he needs help from experts. An expert commission measures two parameters of the newborn — the wingspan and the ear length. Using these data, the commission decides whether it is an eagle or a beagle.

Experts work the following way. The i-th expert chooses two integer numbers a_i and b_i which do not exceed $2\,000\,000\,000$ by the absolute value. When an animal with the parameters (A, B) is born, the i-th expert counts the value $a_iA + b_iB$. If it is positive, then expert decides that an eagle is born, a negative value corresponds to a beagle. If it is zero, then the expert refrains from any guesses.

A decision for a certain animal is made by voting. If strictly more than a half of experts think that the animal is an eagle, the commission tells the farmer that the animal is an eagle. When more than a half of experts vote for a beagle, then a new beagle is recorded in the book. In all other cases the commission cannot say anything concrete, so the farmer thinks that the new animal is a violet piggie.

One fine day the farmer decided that it costs too much money to pay for the service of so many experts. Well, if the commission consists of four experts, and the first expert always speak the same as the third one, while the second expert is solidary to the fourth one, then it is possible to fire the third and the fourth expert without losing the quality.

There are N eagles and beagles at the farm (it is known for any of them if it is eagle or beagle). The farmer wants to find the minimal K such that a commission of K experts can determine all eagles to be eagles and all beagles to be beagles by appropriately choosing a_i and b_i.

Input

In the first line of the input file there is a number N — the number of eagles and beagles at the farm ($2 \le N \le 10\,000$). N lines follow, in each of them three integers – the parameters of the j-th animal — are given: A_j is the wingspan, B_j is the ear length, C_j is 1 if the animal is an eagle and 2 if it is a beagle. $0 \le A_j, B_j \le 10\,000$.

Output

If there is no commission that can satisfy the requirements of the farmer, output a single number -1. Otherwise, output the number K in the first line. In next K lines, output the numbers of a_i and b_i, separated by space — the numbers which the experts should use. You can output any numbers, satisfying the constraints, such that the commission makes the correct decision for each of N animals.

Example

farm.in	standard output
2 10 0 1 0 10 2	1 1 -1

Problem G. Of the Frogs

Input file:	`frogs.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

One fine day, N white and N black frogs decided to play. They found $2N + 1$ hubbles, numbered them by integers from 0 to $2N$ and occupied them in such a way that all white frogs sit on hubbles with numbers $0 \dots N-1$, all black frogs sit on hubbles with numbers $N+1 \dots 2N$, and the hubble N is empty.

The aim of the game is to swap white and black frogs, such that at the end of the game the first N hubbles are occupied by black frogs and the last N hubbles are occupied by white frogs. In one move, a black frog may jump either from the i-th hubble, $i > 0$, to the $(i-1)$-th hubble, if it is empty, or from the i-th hubble, $i > 1$, to the $(i-2)$-th hubble, if it is empty and the $(i-1)$-th hubble is occupied by a white frog. A white frog may jump either from the i-th hubble, $i < 2N$, to the $(i+1)$-th hubble if it is empty, or from the i-th hubble, $i < 2N-1$, to the $(i+2)$-th hubble, if it is empty and $(i+1)$-th hubble is occupied by a black frog.

Usually white and black make moves in turn, but this game is cooperative, so black frogs and white frogs can move in any order. If after 1 000 000 moves the frogs still didn't swapped, they got tired and jump away from the hubbles to the water.

Given N, determine if the frogs can reach the aim of their game.

Input

In the input file, there is a single integer N ($1 \le N \le 499$).

Output

If the frogs cannot swap, output the number -1. Otherwise, in the first line output K — the number of moves to reach the aim of the game — and in the second line output the numbers C_i ($1 \le i \le K$), such that i-th move is made by a frog sitting at the C_i-th hubble. If there are many solutions, output any of them.

Example

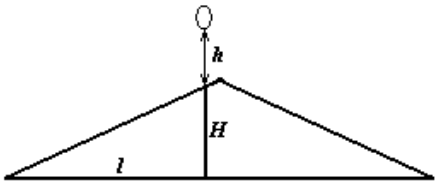
frogs.in	standard output
1	3 2 0 1

Problem H. Ryaba the Hen

Input file:	`hen.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There once was an old man and an old woman and they had a hen called Ryaba. Once the hen laid an egg, not an ordinary one, but a golden one... Well, stop, this is a tale of the past. Listen to the tale of the future!

There once was an old man and an old woman and they had a hen called Ryaba. Once the hen laid an egg, not an ordinary one, but made of titan. Then Ryaba left to kill Sarah Connor, and the oldies left deciding what to do with the egg. The old man hit it and hit it, but could not break it. The old woman hit it and hit it, but could not break it. They called the mouse to help. The mouse hit it and hit it, but could not break it. It finally got angry and rolled the egg up to the roof to throw it away from there — maybe it will break! The roof has the profile shown in the picture below:



The lower left vertex of the isosceles triangle in the picture has the coordinates of $(0, 0)$. The mouse reached a point of (l, H), raised the egg at the height of h above that point and let it fall. The egg fell down, jumped away from the roof, flew a little bit, hit the roof again, and again, until it fell down to the ground. The egg didn't break, but left a number of dents on the roof surface. How many dents there are?

The mouse didn't study in a college, but it knows that the acceleration of free fall is 10, that the egg jumps elastically (specifically, the angle of incidence equals to the angle of reflection) from any roof point (including the lowest one), and one can ignore the size of the egg during calculations. The point (l, H) is not the topmost point of the roof.

Input

In the first line of the input file there is a number N ($1 \le N \le 1000$) — the number of test cases. In each of the next N lines the test cases are located, each containing three integer numbers in a line: H, l, and h, such that $1 \le H, l \le 10^{100}$, and $1 \le h \le 10$ (don't forget the mouse is not tall).

Output

For each test case output the number of dents on the roof.

Example

hen.in	standard output
2 1 2 1 1 2 10	1 1

Problem I. Lunocode

Input file:	`lunocode.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

During the defencive war against Martians, the programmers from the Moon invented a new way of coding the information. Information is represented as a stripe A of size 10×1000, consisting of zeros and ones. The top left cell of such a stripe has the coordinates $(1, 1)$, and the lower right is $(10, 1000)$. To transmit the information without corruption, a funny principle was invented: in any stripe A, for all i from 1 to 999 the following condition is true: the set $\{j | A[j][i] = 0 \text{ and } A[j][i + 1] = 1\}$ has not more than K elements. When the information is received, this condition is checked, and if it is not satisfied for some i, one cannot trust the received information. This principle has become widely used and received a name: "the lunar control condition".

However, Martian programmers, who participated some time ago in Martian programming contests, managed to break the cipher. Now, lunar programmers use matrices of size $M \times N$ instead of stripes. Surely, for each matrix, for all i from 1 to $N - 1$, the lunar control condition is held. So far, the Martians learned only how to count the total number of matrices of the given size. Can you do the same?

Input

The input file contains three integer numbers M, N, K ($1 \le M, N \le 40$, $0 \le K \le M$).

Output

Output the answer — the number of matrices of the given size which satisfy the lunar control condition.

Example

lunocode.in	standard output
2 1 0	4
2 2 1	15

Problem J. Planes

Input file:	`planes.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Have you ever been to an airport where planes depart with an interval of one minutes, or even less? Did you think how many planes there are in the air? And what about the whole Earth?

Let's think that the Earth is an ideal sphere with the center at $(0, 0, 0)$ and the radius of 6370 kilometers. Most passenger planes fly at the height of no more than 15 kilometers. If you could look at the Earth from a very distant point, the planes would be points on its surface.

Consider a certain moment of time. There are N planes in the air. A plane with the number of i is at the point of intersection of the Earth surface with a ray originating at $(0, 0, 0)$ to the direction (X_i, Y_i, Z_i). Your task is to find the maximum number of planes you can see at a very big distance from the Earth. More precisely, you can observe a certain open hemisphere of the Earth surface.

Input

The first line of the input file contains an integer N which does not exceed 150. The next N lines contain the triples X_i, Y_i, Z_i, which describe the planes. All the numbers do not exceed 600 by their absolute value. At least one number from each triple is not a zero.

Output

Output the maximum number of planes that you can simultaneously observe.

Example

planes.in	standard output
6 0 0 1 0 0 -1 0 1 0 0 -1 0 1 0 0 -1 0 0	3

Problem K. Spy Satellites

Input file:	`satellites.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Martian spy satellites made a photo of some territory located at the dark side of the moon. All one can see at the photo is a set of light points. The general of the Martian army supposed that these points are the secret objects of the lunar army located at their bases. Before conquering the Moon, it is useful to know how many bases the lunar army has. As there is no other data, the experts decided to find the number of bases seen on the photo.

Martians think that the bases are the sets of points which satisfy the following condition: for each base, the distance between any two objects at that base is strictly less than the distance between any object at that base and any object at any other base. The distance between two objects with coordinates (A, B) and (C, D) is considered to be $\sqrt{(A-C)^2 + (B-D)^2}$.

Input

The input file consists of several blocks of test data. In the first line of a block there is an integer N — the number of objects at the photo. Each of the following N lines contain the coordinates of an object. The coordinates are integers which do not exceed 10^4 by absolute value. All objects have different coordinates. The blocks are separated by empty lines. The last block consists of a single integer 0 and should not be processed.

In each input file the sum of N does not exceed 5 000, the sum of N^2 does not exceed 400 000, the sum of N^3 does not exceed 250 000 000.

Output

You should determine the possible numbers of bases seen on the photos. For each block, output a string of zeros and ones of length N. For example, a string 110 denotes that on the photo there can be seen one or two bases, a string 011 means that two or three bases can be seen.

Example

satellites.in	standard output
4 -1 -1 1 1 1 -1 -1 1 4 1 0 2 4 1 1 0 1 0	1001 1101

Problem L. Control on Schedule

Input file:	`timetable.in`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

The Major of the city finally noticed that there are too many minibuses, and citizens nearly stopped using trams. In the city there are N minibus stops, some of them are connected by roads. If two stops are connected by a road, then a minibus can move from the first stop to the second stop, as well as from the second stop to the first stop, without intermediate stops. No pair of stops is connected by two or more roads, and no road connects a stop with itself. A minibus stops at all the stops at its route.

When an order to decrease the number of minibus routes was issued, the minibus organization decided to retain only circular routes, which do not contain self-intersections (i.e. each stop appears only once at the route). Each pair of routes should be different by at least one road. As the organization does not want to shrink its market share, it decided to create as many routes as possible to satisfy the conditions above. The routes were numbered from 1 to K. Each route is served by exactly one minibus.

There is a second problem, however — they need to build a timetable for controllers. A timetable is a table where columns correspond to routes and rows correspond to the moments of time where control is performed. If in a cell $[T, I]$ there is a number X, it means that a minibus serving the route I stops for the control at the stop X in the moment of time T. A cell may be empty. During the day, each minibus should stop at each stop for the control exactly one. This means that in each column there are as many nonempty cells as there are stops in the route. In a single moment of time, no two minibuses can be checked at the same stop, and, surely, a minibus cannot be at different stops in the same time.

Find the minimum possible number of rows in the timetable.

Input

In the first line of the input file there are integers N ($3 \le N \le 15$) and M — the number of stops and roads, correspondingly. The next M lines contain pairs of stops which are connected by a road. Stops are numbered from 1 to N.

Output

Output the minimum number of rows in the timetable.

Example

timetable.in	standard output
4 4 1 2 2 3 1 3 1 4	3