Problem A. Arithmetic Derivative

Input file:	`arithmetic.in`
Output file:	`arithmetic.out`
Time limit:	2 seconds
Memory limit:	1024 mebibytes

For natural numbers the *arithmetic derivative* is defined as follows:

- $p' = 1$ for any prime p,
- $(ab)' = a'b + ab'$ for all $a, b \in \mathbb{N}$ (Leibniz rule).

Input

The first line contains one integer, the number T of test cases ($1 \le T \le 40\,000$). Each of the following T lines contains one integer x ($1 \le x \le 10^9$).

Output

For each value of x output in a single line its arithmetic derivative x'. If there are multiple possible values of x' choose any of them.

Examples

arithmetic.in	arithmetic.out
3 8 13 21	12 1 10

Problem B. Banner Rotation

Input file:	`banner.in`
Output file:	`banner.out`
Time limit:	3 seconds
Memory limit:	1024 mebibytes

Goondex LLC Inc. is the largest company in Byteland specializing in Internet-related services and products. Their web search engine is one of the best in terms of ranking algorithm quality and the amount of pages in index, so it is rather popular and widely used. Search service is provided for free for all users, but (let's face the truth) the company needs to make money, so it follows the most common monetization scheme: advertising content is inserted into the web pages of Goondex portal, then banner shows and clicks are sold.

The search engine shows the user exactly one banner on each page with search results. No, there are no tons of ads at Goondex. In general, a single banner is chosen at random from the collection of size N with equal probability. When the user opens the next page located on Goondex host, a new banner is tossed randomly from the N available, and so on.

The user can click on banners if he founds them relevant. If the banner was clicked, it would not be shown more in the current session. So, the next banner would be chosen from the rest $N-1$ banners. The reason of such behaviour is simple: Goondex managers agree that this strategy makes advertisement less annoying for end users. All the data about daily clicks is reset every day: logs with user sessions are pushed to MapReduce cluster in the dead of night, and the next day begins from scratch. So the banner clicked yesterday can appear on the web page today again. If the user clicks on all the N banners during one day, we run into situation when there are no non-clicked banners left. In this case, luckily, no banners will be shown to the user on subsequent pages until the day ends.

Vasya is an inveterate Internet surfer and often uses the Goondex search engine. It depends but we may assume that the number of Goondex pages Vasya visits every day is distributed uniformly on the range of integers $[A, B]$. Sometimes banners attract Vasya's attention. The boy clicks on a banner with probability P. After being clicked the banner disappears and has a chance to be shown and clicked the next day only.

Hence the banner rotation mechanism at Goondex is non-deterministic, there is a problem that some banners may not be shown for a long period of time. How many days on average are required to show Vasya all the N existing banners?

Input

The only line of input contains three integers and one real number: N, A, B and P. Integer N ($1 \le N \le 50$) is the total number of banners available in the Goondex advertisement system. Integers A and B ($1 \le A \le B \le 50$) define the minimum and the maximum number of pages Vasya visits a day, inclusive. Real number P ($0 \le P \le 1$) means the probability of a click on a banner given with no more than three digits after decimal point (it may be also given as an integer 0 or 1).

Output

Output one real value: the expected number of days required to view all the banners. It is guaranteed that the answer exists and is finite. The absolute or the relative error should not be greater than 10^{-6}.

Examples

banner.in	banner.out
`2 1 1 0`	`3.00000000`
`5 6 7 1`	`1.00000000`
`2 2 2 0.500`	`1.28571429`
`3 1 4 0.42`	`2.17604701`

Note

Consider the first example. At the beginning one banner (1 or 2) is shown with a $\frac{1}{2}$ probability. It is not possible to see two banners during one day. In order to finish in two days Vasya needs to see a new banner on the second day (different from the shown on the first day), which is done with a $\frac{1}{2}$ probability. But, again, there is a $\frac{1}{2}$ chance that banners shown on the first and the second days will be equal, so the third day will be required. The probability to finish in three days is $\frac{1}{2} \cdot \frac{1}{2}$. We can continue ad infinitum. As a result, the expected day count can be calculated as follows:

$$E = \frac{1}{2} \times 2 + \frac{1}{2^2} \times 3 + \frac{1}{2^3} \times 4 + \ldots.$$

Let us evaluate the infinite series:

$$\begin{aligned} E = \sum_{i=1}^{\infty} \frac{i+1}{2^i} = \\ = \sum_{i=1}^{\infty} \frac{i}{2^i} + \sum_{i=1}^{\infty} \frac{1}{2^i} = \\ = \sum_{i=1}^{\infty} \frac{i}{2^i} + 1 = \\ = \frac{1}{2} \sum_{i=1}^{\infty} \frac{i}{2^{i-1}} + 1 = \\ = \frac{1}{2} \sum_{i=0}^{\infty} \frac{i+1}{2^i} + 1 = \\ = \frac{1}{2}(1+E) + 1, \end{aligned}$$

which means that

$$E = 3.$$

Now consider the second example. Note that Vasya clicks on each banner, so on the first day all five banners are going to be shown in random order, and the rest one or two pages will be definitely shown without advertisement.

Problem C. Code of the Tree

Input file:	`code.in`
Output file:	`code.out`
Time limit:	1 second
Memory limit:	1024 mebibytes

Romka thinks he has invented quite a good algorithm of encoding an undirected tree. For each vertex i he calculates one integer d_i: maximal distance from this vertex to other vertices of the tree. Then he will use the sequence $d = (d_1, d_2, \ldots, d_N)$ as a code of this tree.

Of course, Romka implemented this algorithm in a few minutes. Can you do this in a few hours?

Input

The first line of input contains one integer N ($2 \le N \le 100\,000$), number of vertices in a tree. Vertices are numbered from 1 to N. Each of the next $N-1$ lines denotes two vertices x_i, y_i ($1 \le x_i, y_i \le N$, $x_i \ne y_i$) that are connected by i-th edge.

Output

For each vertex output maximum distance d_i to another vertex of the tree.

Examples

code.in	code.out
5 1 2 1 3 1 4 1 5	1 2 2 2 2
4 1 2 2 3 3 4	3 2 2 3

Problem D. Decoding Tree Code

Input file:	`decode.in`
Output file:	`decode.out`
Time limit:	1 second
Memory limit:	1024 mebibytes

Some days ago Romka invented quite a good (as he thinks) algorithm of encoding an undirected tree. For each vertex i he calculates one integer d_i: maximal distance from this vertex to other vertices of the tree. Then he will use the sequence $d = (d_1, d_2, \ldots, d_N)$ as a code of this tree.

Now he tries to solve an inverse problem: given a tree code, reconstruct an initial tree. Can you do this faster than Romka?

Input

The first line of input contains one integer N ($2 \le N \le 100\,000$), number of vertices in a tree. The second line contains N integers d_i ($1 \le d_i \le N - 1$) separated by spaces: maximal distance from i-th vertex to other tree vertices.

Output

If there is no tree with the given code, print `Epic fail`. Otherwise, print the line `I got it` followed by $N - 1$ lines with tree edges. In case of several solutions print any of them.

Examples

decode.in	decode.out
5 1 2 2 2 2	I got it 1 2 1 3 1 4 1 5
5 4 3 2 3 4	I got it 1 2 2 3 3 4 4 5
3 1 1 1	Epic fail

Problem E. Election

Input file:	`election.in`
Output file:	`election.out`
Time limit:	1 second
Memory limit:	1024 mebibytes

> Voter turnout of 146.47 % was registered in Rostov region.
>
> "Russia 24" TV channel

It is a common thing that the sum of all the percentages in election results does not make up exactly 100 %. The reason is not always some kind of deception. Often the sum mismatch is simply caused by roundoff errors. For example, consider the following poll results chart:

- candidate Alice: 26 %,
- candidate Bob: 17 %,
- candidate Charlie: 58 %.

The sum of percentage points is $26\,\% + 17\,\% + 58\,\% = 101\,\%$!

The numbers look suspiciously, but if they are obtained, for example, from 25.51 %, 16.71 % and 57.78 % respectively by means of rounding to integers for better human-readable presentation, all seems to be fair: numbers before rounding add to 100 % precisely.

Assume that we use the following rule: we round to the nearest integer, and when two integers lie on the same distance, we choose the greater one. For example, numbers 0.5, 0.7, 1, 1.499 round to 1 and number 1.5 rounds to 2.

You are given the election results as a list of N integer percentage values after rounding. Please determine whether these results are consistent, i. e. we can build a list of N real values, each from the range $[0, 100]$, which give exactly 100 after summation.

Input

The first line of input contains an integer N ($1 \le N \le 1000$). The second line contains N space-separated integers belonging to the closed interval from 0 to 1000.

Output

Print `Yes` if the data are consistent and can represent real election results after rounding, and `No` otherwise.

Examples

election.in	election.out
3 26 17 58	Yes
2 49 49	No
1 146	No

Problem F. Factorial Number System

Input file:	`factoradic.in`
Output file:	`factoradic.out`
Time limit:	1 second
Memory limit:	1024 mebibytes

The *factorial number system*, also called *factoradic*, is a mixed radix numeral system: the i-th digit from the right has base $i+1$, which means that the digit must be less than or equal to i, and that (taking into account the bases of the less significant digits) its value to be multiplied by $i!$. If the number has the form $a_k a_{k-1} \ldots a_2 a_1$ in the factorial number system, then $0 \le a_i \le i$, and the number is equal to

$$\sum_{i=1}^{k} a_i \cdot i!.$$

It is known that every positive integer has a unique representation in factoradic form.

The string consists of numbers written in factoradic form and symbols "`+`" and "`-`" denoting the operations of addition and subtraction, respectively. Find the value of the expression and output it in factoradic form.

Input

The only line of input contains an expression built on the above guidelines. Each number in the expression is positive and is not greater than $2 \cdot 10^9$. Expression contains no more than 1000 numbers, and it is guaranteed that the calculation result will be positive and will not be greater than $2 \cdot 10^9$.

Letters from `A` to `Z` are used to denote digits $10, \ldots, 35$. Leading zeros are not allowed.

Output

Output the result of evaluating the expression in factoradic form. Leading zeros are not allowed.

Examples

factoradic.in	factoradic.out
321+321	1320
321-320	1

Problem G. Graphs. How Many of Them?

Input file: `graphs.in`
Output file: `graphs.out`
Time limit: 2 seconds
Memory limit: 1024 mebibytes

Consider an arbitrary connected labeled graph $G = (V, E)$ without loops and multiple edges. Formally, vertices of the graph G are labeled with the consecutive integers $1, \ldots, N$, where N is the number of vertices in the graph G, and for any pair of vertices v and u there exists a path in the graph G between v and u.

A *bridge* or *cut-edge* is an edge of a graph such that deletion of this edge increases the number of the connected components of the graph.

How many connected labeled graphs with N vertices have no more than B bridges?

Input

The only input line contains two integers N and B ($2 \le N \le 50$, $0 \le B \le N(N-1)/2$).

Output

Output the number of the connected labeled graphs with N vertices and no more than B bridges modulo 1 000 000 007.

Examples

graphs.in	graphs.out
2 1	1
4 0	10
4 1	22

Problem H. Halloween

Input file:	`halloween.in`
Output file:	`halloween.out`
Time limit:	2 seconds
Memory limit:	1024 mebibytes

In one city of Byteland every inhabitant has a *universal electronic card*. Those cards set an ID and a *special holiday number* for every citizen. On holidays residents present each other with gifts depending on various traditions.

On Halloween all the guys give the girls cookies with predictions. A guy presents cookies to a girl if his holiday number is divisible by hers, and he gives her the number of cookies equal to the result of the division of his number by hers.

The mayor wants to know how many cookies will be given this year. Knowing the holiday numbers of all the guys and girls in the city, calculate how many cookies will be given in total.

Input

The first line of input contains integers N and M ($1 \le N, M \le 10^6$), number of guys and girls, respectively. The second line contains N integers A_i ($1 \le A_i \le 10^6$): *holiday number* of the i-th guy. The third line contains M integers B_j ($1 \le B_j \le 10^6$): *holiday number* of the j-th girl.

Output

Output the total number of cookies that will be given by guys to girls.

Examples

halloween.in	halloween.out
3 2 5 10 15 5 3	11
10 10 6 7 8 9 10 11 12 13 14 15 1 1 2 2 3 4 5 6 7 8	291

Note

In the first sample the first girl will get six cookies (one from the first guy, two from the second one and three from the third one). The second girl will get five cookies from the third guy.

Problem I. Intelligent System

Input file:	`intelligent.in`
Output file:	`intelligent.out`
Time limit:	1 second
Memory limit:	1024 mebibytes

K conveyor belts for car production work at the factory of the company $A.A.A.$ (Alliance Automatic Automobiles). Each conveyor belt consists of N producing robots. You know the operation time for every producing robot. This time may be different for each robot. So, all the robots with number j located on K conveyors perform the same operation, but, maybe, with different speed. In order to accelerate the process of production you can transfer a partially assembled car from the current conveyor belt to the other, and the time of transferring can be neglected.

The production of a car goes through N stages. At the beginning the intelligent system is looking for the fastest way to construct the car taking into account the current workloads. Then the car runs all the production path from the first producer to the N-th according to the plan. Now the factory is standing idle, all the robots are free, so the system just takes the most productive robot for each stage.

One of the company's clients has ordered the production of a prototype of the innovative car "SuperMegaCar 3000". The marketing department thinks that if the prototype is released quickly, then the car will not be popular. But the slowdown in the production is very expensive. The managers have decided to help the customer. It is allowed to double the operating time of one of the producing robots. So, the intelligent system will calculate the fastest path based on the new values of the operating time. Find the maximum possible time of the car production after slowing down one robot twice.

Input

The first line of input contains integers K and N ($1 \le K, N \le 10^3$). Each of the next K lines represents one conveyor belt and contains N integers $A_{i,j}$ ($1 \le A_{i,j} \le 10^4$): each $A_{i,j}$ is the operating time of the j-th producing robot of the i-th conveyor belt.

Output

Output the maximum possible time of a single car production.

Examples

intelligent.in	intelligent.out
3 4 5 10 5 10 6 20 6 20 7 30 7 30	40
1 1 20	40

Problem J. Joining Transformation

Input file:	`joining.in`
Output file:	`joining.out`
Time limit:	3 seconds
Memory limit:	1024 mebibytes

Consider a string s which consists of the small latin letters.

Define the transformation f that maps a string to the lexicographically greatest one of all the strings obtained by removing zero or more characters from the initial string.

Your task is to find the result of applying f to several substrings of the sequence of strings. The sequence is built as follows. You are given the initial string s. Each of the next strings is generated by replacing some character with a new one.

As the total length of the transformation results can be quite big, you have to output only the symbol at the specified position for each transformation. If the resulting string is short enough and the desired position does not get inside it, output "`-`" (hyphen-minus, ASCII code 45).

Input

The first line of input contains the initial string s ($1 \le |s| \le 100\,000$) which consists of the small latin letters. The second line contains an integer Q ($1 \le Q \le 100\,000$), the number of queries. Next Q lines describe the queries.

A query to find a character at the specified position in the result of applying f to the substring of the current string s is described by four integers: 1, the query type, l and r, the query bounds, and k, the position of the character you are asked for ($1 \le l \le r \le |s|$, $1 \le k \le |s|$).

A query to replace a character is described by three items: 2, the query type, p, an index of the character in the string, c, a new character (a small latin letter) to put to the p-th position in the string replacing an old character ($1 \le p \le |s|$).

Output

For each query of the first type output the sought character, one per line.

Examples

joining.in	joining.out
abcabc 12 1 1 6 1 1 1 6 2 1 2 5 2 1 2 5 3 1 1 4 1 1 1 4 2 2 3 a 1 1 4 2 1 1 4 3 2 5 d 1 1 5 1 1 1 5 2	c c b - c a a a d -

Problem K. King of Byteland

Input file:	`king.in`
Output file:	`king.out`
Time limit:	6 seconds
Memory limit:	1024 mebibytes

After a successful military campaign Byteland expanded significantly and the current capital is on the outskirts of the kingdom. King Byte XVII decided to move the capital to a new city and build a new castle there.

There are N cities in Byteland that are connected by exactly $N - 1$ roads. One can reach from any city to any other using only these roads. For convenience, all the cities are numbered with integers from 1 to N.

To select the best city, the king demanded to collect some statistics. First of all he is interested in the answers to queries like "what cities are at a given distance from a possible new capital". Distance between two cities is measured as the number of roads passed during the trip and is the minimum possible such number for these cities. Byte XVII does not want to read huge reports, so for each question he added one restriction: from all the cities at the distance d from the city c find only the k-th in the sorted list (the cities are ordered by their numbers).

To help to prepare the report you are asked to find answers to the queries.

Input

The first line of input contains an integer N ($2 \le N \le 50\,000$), the number of cities in Byteland. Each of the next $N - 1$ lines contains two integers u and v: the numbers of cities connected by a road.

The next line contains an integer M ($1 \le M \le 100\,000$), the number of queries that must be answered. Next, M lines contain descriptions of queries, one per line. Each query is described by three numbers c, d and k ($1 \le c \le N$, $1 \le d \le 30$, $1 \le k$): the city of a possible new capital, the distance from the capital to cities, the 1-based index in the sorted list of cities at the distance d from c. It is guaranteed that there are at least k cities at the distance d from the city c.

Output

Output M answers for king's queries, one per line.

Examples

king.in	king.out
3 1 2 2 3 6 1 1 1 1 2 1 2 1 1 2 1 2 3 1 1 3 2 1	2 3 1 3 2 1

Problem L. Linear Programmer

Input file:	`linear.in`
Output file:	`linear.out`
Time limit:	1 second
Memory limit:	1024 mebibytes

As you probably know, linear programming (LP) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships. Linear programming is a special case of mathematical programming (mathematical optimization).

Vasya is said to be the best *linear* programmer of the students of his group. Vasya doesn't enjoy writing computer programs at all. "C++ or Java—it's a frightful bore!" Vasya thinks. It goes without saying that traditional ACM programming contests do not take any notice of him. But linear programming problems captivate our hero very much. Vasya spends hours optimizing functions using Dantzig's simplex algorithm, Khachiyan's ellipsoid method, Karmarkar's projective approach and other hard and sophisticated techniques…

Consider the following problem Vasya is puzzled over at present. Given n variables

$$x_1, x_2, \ldots, x_N$$

over the set of real numbers, you need to maximize the difference between the first and the last one, i. e.

$$x_N - x_1 \to \max$$

subject to M linear inequality constraints

$$\begin{cases} x_{v_1} - x_{u_1} \le \ell_1, \\ x_{v_2} - x_{u_2} \le \ell_2, \\ \ldots \\ x_{v_M} - x_{u_M} \le \ell_M. \end{cases}$$

Can you solve it?

Input

The first line of input contains two integers N and M ($2 \le N \le 100\,000$, $1 \le M \le 100\,000$), the number of variables and constraints respectively. Then M lines follow, each containing three integers. The $(i+1)$-th line of input defines the values of u_i, v_i ($1 \le u_i, v_i \le N$ and $u_i \ne v_i$) and ℓ_i ($1 \le \ell_i \le 100\,000$). All the numbers are separated with single spaces.

Output

If the set of feasible solutions is empty, i. e. the constraints are inconsistent and there exists no suitable real-valued vector $(x_1, x_2, \ldots, x_N)$, output `No solution`. If the objective function has no upper bound, print `Infinity`. Otherwise you should output the required maximum as a single number with absolute or relative error less than 10^{-6}.

Examples

linear.in	linear.out
2 1 1 2 3	3
2 1 2 1 3	Infinity

Problem M. Masha and Dasha

Input file:	`mashadasha.in`
Output file:	`mashadasha.out`
Time limit:	1 second
Memory limit:	1024 mebibytes

On twin sisters' Masha and Dasha birthday parents baked a big tasty cake in the form of a triangle. Sprinkles topped along the border have turned out especially delicious. . . The sisters are grown-up enough, both have their own circle of friends and want to treat them to the cake.

The parents try not to offend any of the sisters, so they decided to divide the cake into two parts of the same area. Moreover, the lengths of the strip of tasty topping left on the pieces' border should be equal.

The cutting line is supposed to be straight: it's not appropriate to offer the guests lots of crumbs. Alas, the parents cannot calculate the way a section should be made.

In order to solve this problem the parents took a large sheet of paper, drew a coordinate grid and placed the cake on top of the sheet in such a way that one of the corners is located in the origin, another one—in the point with coordinates $(c, 0)$, the last one—in the point (a, b).

Help the parents and determine how the cut should be made.

Input

The only line of input contains three integral values a, b, c ($-1000 \le a \le 1000$, $1 \le b, c \le 1000$).

Output

Output the coordinates of the start and end points of the cut in two separate lines. The points should lie on the triangle sides (more precisely, the distance from each point to the nearest side is expected to be less than 10^{-4}). The parts obtained after cutting the cake should have almost equal areas and almost equal sums of lengths of the original cake's edges (the absolute difference must be less than 10^{-4} in both cases).

If the cake could not be divided in the desired way, output a single string `Impossible`.

If the problem has several soultions, output any of them.

Examples

mashadasha.in	mashadasha.out
7 3 8	7.48445074 1.54664777 0.24128451 0.00000000