Problem A. Ability Draft

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Dota 2 is a competitive multiplayer computer game. In this game, two teams of n players are fighting each other in order to destroy the enemy main building, Ancient. Each player controls one hero, and each hero has exactly s standard abilities and exactly one ultimate ability.

Ability Draft is a fun mode of Dota 2. In this mode, at the start of the game each player gets random hero without abilities at all. Then players pick abilities in certain order from the pool of abilities provided by the game. At his turn, player may either pick any of the remaining standard abilities, or, if he didn't pick his ultimate ability, pick any of the remaining ultimate abilities. The pool contains enough abilities of both kinds so that each player can always pick his s standard abilities and one ultimate ability.

The strength of the team is the sum of strengths of all abilities of all heroes in this team. All players are picking abilities in such a way that the difference between the strength of their own team and the strength of the enemy team at the end of draft procedure is maximum possible.

Knowing the strengths of abilities in the pool and the order of ability picks, find the difference of strengths of the teams if all players perform their picks optimally.

Input

The first line of input contains two integers n and s ($1 \le n \le 5$, $1 \le s \le 3$) — the number of players in the team and the number of standard abilities of the heroes.

The second line contains $2n \cdot (s + 1)$ indices of players — the order of ability picks. Each index from 1 to $2n$ appears exactly $s + 1$ times in this array. Players with indices between 1 and n belong to the first team, and players with indices between $n + 1$ and $2n$ belong to the second team.

The third line contains an integer p_s ($2n \cdot s \le p_s \le 36$) — the number of standard abilities in the pool.

The fourth line contains p_s integers from 1 to 10^6 — strengths of the standard abilities.

The fifth line contains an integer p_u ($2n \le p_u \le 12$) — the number of ultimate abilities in the pool.

The sixth line contains p_u integers from 1 to 10^6 — strengths of the ultimate abilities.

Output

Output a single integer — the difference of strengths of the teams after all ability picks.

Examples

standard input	standard output
1 1 1 2 2 1 2 5 3 2 7 2	3
1 2 2 1 1 2 2 1 4 4 8 8 9 2 6 7	2
2 1 1 3 4 2 2 4 3 1 6 1 4 4 8 9 11 5 14 11 10 8 5	-1

Problem B. Short Random Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	512 megabytes

There are lots of things to do on this contest besides this problem, so let's make it quick.

You are given a tree consisting of n vertices. Length of each edge is a real number chosen independently and uniformly at random between 0 and 1. Find the expected value of the diameter of such tree.

Input

The first line of input contains the only integer n ($2 \le n \le 100$), the number of vertices in the tree.

Each of the next $n - 1$ lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$) describing endpoints of i-th edge.

Output

Output the answer as a value of a rational number modulo $10^9 + 7$.

Formally, it is guaranteed that under given constraints the expected value of diameter of such random tree is always a rational number $\frac{p}{q}$ (p and q are integer and coprime, q is positive), such that q is not divisible by $10^9 + 7$, which is a prime number (in case somebody missed it).

Output such integer a between 0 and $10^9 + 6$ that $p - aq$ is divisible by $10^9 + 7$.

Examples

standard input	standard output
5 1 2 2 3 3 4 4 5	2
5 4 2 2 3 3 1 3 5	283333337

Note

In the first sample case answer is 2 since each edge always belongs to the diameter adding 0.5 to diameter expected length.

In the second sample case expected length of diameter is $\frac{101}{60}$ that corresponds to value of $a = 283333337$ (since $101 - 60 \cdot 283333337 = -17000000119$ is divisible by $10^9 + 7$).

Problem C. Block, Stock and Two Smoking Galaxy Notes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

I decided to start a new fancy project related to cryptocurrency, deep learning, self-driving cars and maybe mobile voice assistance (will decide that later). I already have a team consisting of n promising software engineers and last thing that has to be done is choosing a techlead among them.

All engineers except the techlead should be divided into teams consisting of one or two engineers (recently I read first ten pages of the book "Agile Software Development: Programming in Pairs" and found the described technique very useful!). For each pair of engineers I know if they can interact with each other *effectively*.

The choice of techlead and distribution into teams is *effective* if any two-member team consists of two engineers that can interact effectively, and in any team there is at least one engineer who can interact effectively with the techlead.

I want you to find an appropriate company structure as fast as possible, so that our startup can make an IPO or ICO (not quite sure yet what that means, no time for that now), or determine that it is impossible and the world of success and glory is not for me (at least for today).

Input

The first line of input contains two integers n and m ($2 \le n \le 1000$, $0 \le m \le 10\,000$), the number of software engineers and the number of successfully interacting pairs.

Each of the next m lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), the indices of engineers forming an effectively interacting pair.

I guarantee that all unordered pairs of engineers are different.

Output

Print a single word "`No`" if it is impossible to create a company structure satisfying my requirements.

Otherwise, print a word "`Yes`" in the first line.

In the second line print two integers l, k ($1 \le l \le n$, $\left\lceil \frac{n-1}{2} \right\rceil \le k \le n-1$), the index of the techlead and the number of teams.

Each of the next k lines should contain two integers t_1 and t_2 defining a team. If the team consists of two members, t_1 and t_2 should be indices of the engineers forming it, otherwise t_1 should be the index of the only engineer in the team and t_2 should be -1.

If there are multiple correct answers, print any of them.

Examples

standard input	standard output
5 4 1 2 2 3 3 4 4 5	Yes 3 2 2 1 4 5
4 4 1 2 2 3 3 4 4 1	Yes 1 2 2 3 4 -1
4 3 1 2 2 3 3 1	No

Problem D. Lunch Queue

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	512 megabytes

It is lunch time! That's why n employees are coming to the canteen to have lunch. It is known that i-th employee works in the team c_i and has impudence a_i.

Suppose there are currently l people in the queue. Positions in the queue are numbered from 1 to l from the beginning of the queue. When a person comes to the canteen, he tries to get as close to the beginning of the queue as possible by staying next to one of his teammates. Formally, if a newcomer gets into the queue at the position k ($1 \le k \le l+1$), he shifts the k-th person and everybody behind him one position back and gets the position k. In particular, $k = 1$ corresponds to the beginning of the queue and $k = l+1$ corresponds to the end of the queue.

An employee i can stay at the position k only if two conditions are held:

- After getting to the position k, at least one of the neighbors of newcomer is from the same team c_i;
- A newcomer is not shifting too much people for his level of impudence, namely: $k \ge l + 1 - a_i$.

Among all k satisfying the conditions above the minimum possible is chosen. If there are no suitable values of k, a newcomer simply takes place after the last person in the queue.

For example, if there are five people in the queue working in teams 1, 3, 4, 1, 3 respectively (starting from the beginning of the queue) and the newcomer works in team 1 and has impudence 3, he stays at the position 4 of the queue. After that, the sequence of the employee teams becomes 1, 3, 4, **1**, 1, 3. Note that impudence value of 3 also allowed him to get to the position 3, but there would be no teammates next to him in that case, so the first condition is violated.

Knowing the order in which employees arrive and their values of c_i and a_i, find out how the queue will look like in the end.

Input

The first line contains an integer n ($1 \le n \le 400\,000$), the number of employees.

Each of the next n lines contains two integers c_i, a_i ($1 \le c_i \le n$, $0 \le a_i \le 400\,000$), the team and the impudence of the i-th employee.

Output

Output n distinct integers from 1 to n which are the indices of people in the queue from beginning of the queue to the end after all employees get to the canteen. Employees are indexed from 1 to n in order of their appearance in the input data.

Example

standard input	standard output
8 1 0 2 1 2 2 1 1 3 2 1 3 2 3 2 5	1 3 8 2 7 6 4 5

Problem E. Oneness

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Define *oneness* of the number x to be a number of integers $d > 1$ dividing x, whose decimal representation consists only of digit 1. For example, $oneness(121) = 1$ and $oneness(1221) = 2$.

Number n is obtained using the following algorithm involving a pseudo-random number generator. You are given two integers l and s_0, which are the length of the decimal representation of n and the generator seed.

The digits $d_0 d_1 \ldots d_{l-1}$ of the number n are generated by the following recursions:

$$d_i = \lfloor s_i / 1024 \rfloor \mod 10$$
$$s_{i+1} = (747796405 s_i - 1403630843) \mod 2^{32}$$

It is guaranteed that d_0 is non-zero.

Calculate the total oneness over all integers between 1 and n.

Input

The first line contains two integers l, s ($1 \le l \le 250\,000$, $0 \le s_0 < 2^{32}$), the number of digits in the decimal representation of n and the seed of pseudo-random number generator that is used to create the decimal representation of n.

Output

Output the sum of oneness over all integers between 1 and n.

Examples

standard input	standard output
1 1024	0
2 1048	1
4 11312	123
4 31415926	942

Note

In sample tests n equals 1, 11, 1221 and 9359 respectively.

Problem F. Shuffle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given a string of even length $s = s_1 \dots s_n$, we define shuffle operation which transforms a string into a new string according to the following rule:

$$\text{shuffle}(s) = s_1 s_3 \dots s_{n-1} s_2 s_4 \dots s_n$$

For example, shuffle(`abcdef`) = `acebdf`.

You are given two strings of equal even length, s and t. How many times do you have to apply shuffle operation to s in order to get t as a result?

In the other words, find minimum k such that $\underbrace{\text{shuffle}(\text{shuffle}(\dots \text{shuffle}}_{k \text{ times}}(s)\dots)) = t$ or report that it is not possible to reach t in any number of operations.

Input

The first line of input contains a string s, the second contains a string t ($|s| = |t|$, $2 \le |s| \le 10^6$, $|s|$ is even). Both strings consist of lowercase English characters.

Output

Print minimum non-negative k such that it is possible to obtain t from s by applying shuffle operation k times (or maybe not applying at all if $k = 0$), or print `-1` if it is impossible.

Examples

standard input	standard output
abcdef aedcbf	2
petrozavodsk poztsvoedark	3
qwerty ytrewq	-1

Problem G. Piecewise Linearity

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	512 megabytes

Alice is obsessed with linear functions and especially their plots that are always so mysteriously straight. Recently she found out a plot of function $f(x) = |x - 1|$ that impressed her a lot: it was twice as mysterious and beautiful since it consisted not only of one straight-line segment, but of two of them!

Alice immediately thought of a function that is $n \geq 2$ times as mysterious as a linear function. Formally, she came up with a piecewise linear function $f(x)$, whose plot consists of n straight-line segments. Function $f(x)$ is defined by $n+1$ points $P_0, P_1, \ldots, P_{n-1}, P_n$ belonging to its plot and allowing to reconstruct it in a following manner. Plot of function $f(x)$ is a polyline consisting of two rays P_1P_0, $P_{n-1}P_n$ and $n-2$ line segments P_1P_2, ..., $P_{n-2}P_{n-1}$. Each point P_i is defined by its Cartesian coordinates (x_i, y_i), which are both integers. It is guaranteed that $x_i > x_{i-1}$ for all i between 1 and n, i.e. given polyline is a plot of some function $f(x)$. Please, refer to the Note section for more details.

Now Alice asks you if it is possible to express her function $f(x)$ as a linear combination of terms of form $|x-a_i|$. Formally, your task is to find out if there exist two finite sequences of **real** numbers $\lambda_1, \lambda_2, \ldots, \lambda_m$ and $a_1, a_2, \ldots, a_m$ such that the following equation holds:

$$f(x) = \sum_{i=1}^{m} \lambda_i |x - a_i|$$

Input

First line of input contains an integer n ($2 \leq n \leq 100\,000$), the number of segments in a polyline that is a plot of Alice function.

In the i-th of next $n+1$ lines (indexed from zero) there are two integers x_i, y_i ($-10^6 \leq x_i, y_i \leq 10^6$), coordinates of point P_i.

It is guaranteed that $x_0 < x_1 < \ldots < x_n$.

Output

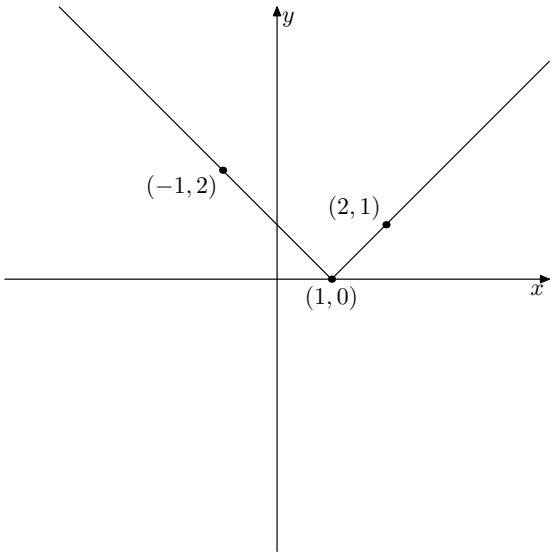
If it is possible to express $f(x)$ as a linear combination of terms of form $|x - a_i|$, print the only word "Yes" (without quotes). Otherwise print the only word "No" (without quotes).

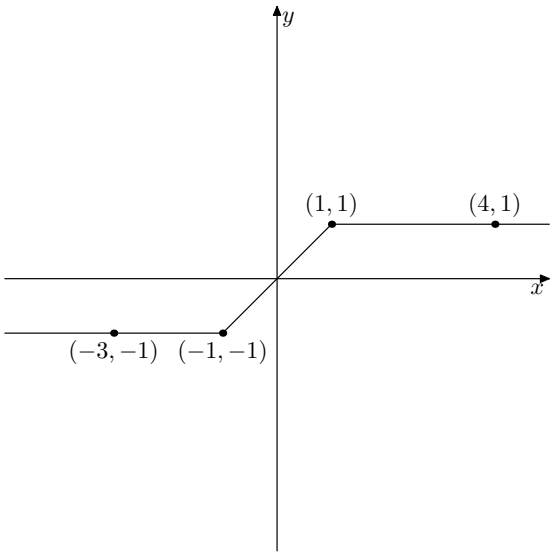
Examples

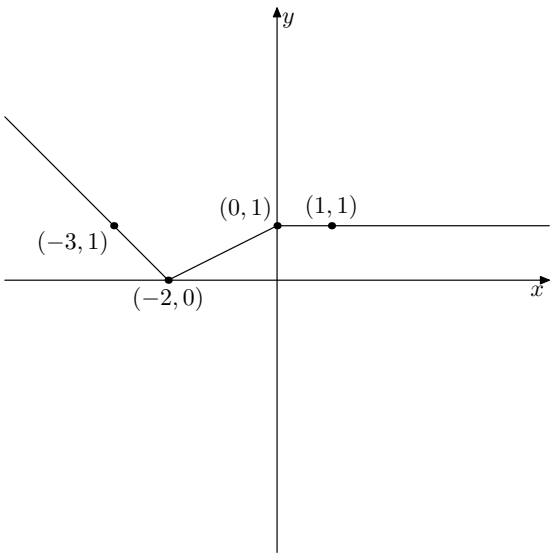
standard input	standard output
2 -1 2 1 0 2 1	Yes
3 -3 -1 -1 -1 1 1 4 1	Yes
3 -3 1 -2 0 0 1 1 1	No

Note

Pictures for the sample cases are given below:







Problem H. Sketch

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Given a sequence a consisting of integers $a_1, \ldots, a_n$, consider all its non-decreasing subsequences of length k. Among all of these take the one with smallest last element. We denote the value of this element with s_k.

The sequence $s(a) = s_1, \ldots, s_l$ is a *sketch* for sequence a, where l is the length of the longest non-decreasing subsequence of a.

Building a sketch of the sequence is a standard task while finding the length of the longest non-decreasing subsequence. Here we consider an opposite problem: given a sketch with some missing entries, find any sequence producing this sketch.

Formally, you are given a sequence $t_1, \ldots, t_k$ where each element is either a positive integer or -1. You have to find a sequence $a_1, \ldots, a_n$ with each element being an integer between 1 and m, inclusive, satisfying the following properties: length of the sketch of a should be equal to k, and for each index i between 1 and k, if $t_i \neq -1$, then $s_i = t_i$ should hold.

Input

First line contains three integers k, n, m ($1 \le k \le 300\,000$, $1 \le n \le 300\,000$, $1 \le m \le 10^9$), the length of the sketch, the length of the desired sequence and the upper bound on element values respectively.

Next line contains k numbers $t_1, \ldots, t_k$ ($1 \le t_i \le m$ or $t_i = -1$), the sketch itself.

Output

If a sequence with such sketch exists, output the elements of a desired sequence. In case of multiple answers you may output any of them.

If no valid sequence exists, output a single integer -1.

Examples

standard input	standard output
3 4 10 3 7 7	3 7 8 7
3 5 10 3 -1 7	3 6 5 4 7
4 2 10 1 2 3 4	-1

Note

Sketch of the answer sequence in example 2: 3 4 7.

Problem I. $\leq$ or $\geq$

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

This is an interactive problem.

There are n stacks, each containing k positive integers not exceeding 10^9. You play a game with the jury program. Initially you know only the topmost value of each stack. Game proceeds as follows:

- At the beginning of each turn you choose some integer x.
- After that, jury program selects one of the options: "$\leq$" or "$\geq$". Denote the chosen relation as R.
- For every non-empty stack, if the topmost element t satisfies the inequality "$t\ R\ x$", t is being removed from the stack. This procedure is performed only once with each stack.
- Finally, you are told what was the chosen relation R and current state of each stack, that is either information that the stack is empty or the stack topmost number.

In all tests except sample $n = 10\,000$, $k = 10$. Your task is to clear all stacks in no more than 50 moves.

Note that the jury program is **adaptive**, i.e. stack contents are not fixed and may change "on the run" depending on the output of your program.

Interaction Protocol

The first line of input contains two integers n, k, the number of stacks and the size of each stack. For the sample case $n = 4$, $k = 2$. In all other tests $n = 10\,000$, $k = 10$.

The second line of input contains n integers that are the topmost elements of each stack.

At the beginning of each turn you should output a single integer x ($1 \leq x \leq 10^9$).

Next line of input will contain the string, that will contain either the word "`End`", or one of the strings "`<=`" or "`>=`".

In case of "`End`", your program should immediately terminate with zero exit code. It may happen if your program successfully cleared all the stacks, if it made an incorrect query or if after 50 queries there are still non-empty stacks.

Otherwise, the string denotes the chosen relation R, and the next line of input will contain n integers, each of which will be either 0 if the corresponding stack is empty, or its topmost element otherwise.

All values in the stacks are positive integers not exceeding 10^9.

Make sure your output does not get buffered, for instance, by calling `fflush(stdout)` in C/C++, `System.out.flush()` in Java or `sys.stdout.flush()` in Python after printing each number.

Example

standard input	standard output
4 2 1 2 3 4 <= 5 6 3 4 >= 0 0 3 8 <= 0 0 7 8 <= 0 0 0 8 End	 2 4 3 7 8

Note

In the sample test there are four fixed stacks of size 2:

```
1  2  3  4
5  6  7  8
```

Next we describe the interaction example as shown in "Example" section. Note that, although the stacks are fixed, the interactor in the system may behave not exactly in this way even if you ask the same queries.

In the first query $x = 2$ and $R =$ "$\leq$". After the query numbers 1 and 2 are removed and stacks look like this:

```
      3  4
5  6  7  8
```

In the second query $x = 4$ and $R =$ "$\geq$". Numbers 5, 6 and 4 are removed, and the stacks' state is:

```
      3
      7  8
```

Problem J. Stairways

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

Elevators are often overcrowded. Sometimes you have to wait for several minutes until one of them arrives, and even after bear all the people who take a ride from the first floor to the second, to the third... Two eternities might pass until you finally reach your workplace.

Having this in mind, several Yandex programmers from the fifth floor decided to take stairs instead. Not only it helps to avoid queues, they thought, but also helps them to be in shape.

Avoid queues, they thought. So nave people. Stairways are so narrow that you cannot overtake a slow person which walks before you in slower pace. Even more, only two identical stairways lead to the fifth floor. That's why fast people quickly become anxious.

There are n people working on the fifth floor. Each morning they walk into the building in natural order (from 1-st to n-th). The problem is that they have different speed. Unless interrupted by anyone, i-th person would reach the fifth floor by the time t_i. However, if a slower colleague walks before him along the same stairway, i-th person has to slow down and he arrives exactly as soon as the slow colleague does.

If someone expected to reach his workplace at the time x but in fact arrived only at the time y, he says $y - x$ swear words. Programmers are smart enough to notice that if each of them chooses the stairway wisely, then less swear words will be pronounced in total. However, they could not find the optimal assignment. Help them!

Input

In the first line there is a single integer n ($1 \le n \le 100\,000$), which is the number of programmers.

In the second line there are n integers $t_1, \ldots, t_n$ ($1 \le t_i \le 10^9$), which are their expected arrival times listed in the same order in which programmers enter the building.

Output

Output the minimum possible total number of swear words said by the moment everyone reaches the workplace.

Examples

standard input	standard output
5 100 4 3 2 1	6
6 5 1 6 2 7 3	0

Note

In the first sample case the optimal way would be to let the slowest first programmer take the first stairway, and let everybody else take the second stairway. In such case the third programmer will pronounce 1 swear word, the fourth programmer will pronounce 2 swear words, and the fifth programmer will pronounce 3 swear words.

In the second sample case the optimal way would be to let first, third and fifth take the first stairway, and to let the remaining programmers take the second stairway.

Problem K. Hiding a Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

XOR-scanner is a device which scans a sequence of integers and accepts it if and only if the bitwise XOR of all numbers in the sequence is equal to zero.

You have a tree with n vertices, labeled with integers from 1 to n. You want to write down this tree in a standard format for the programming contest problem:

n
u_1 v_1
...
u_{n-1} v_{n-1}

Here n is the number of vertices and u_i, v_i are vertices connected by the i-th edge.

You want the XOR-scanner to accept your output. It might be not the case initially, so you can change the labels of some vertices of the tree. After this operation all vertices must have distinct integer labels from 1 to 10^9, inclusive.

For each vertex it is known if it is possible to change its label. Find a way to relabel some allowed vertices (possibly, keeping some labels or all of them as is) in such a manner that the XOR-scanner accepts the tree representation or report that it is impossible.

Input

In the first line there is an integer n ($2 \le n \le 100\,000$), the number of vertices in a tree.

Next line contains n integers, i-th of them is 0 if the label of i-th vertex is fixed and 1 if it can be changed.

Each of the next $n - 1$ lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n$), which denote the endpoints of the edges.

Output

If the desired relabeling exists, print the relabeled tree in the same format as it is given in the statement, keeping the order of edges and the order of endpoints of the each edge. The bitwise XOR of all printed numbers must be zero.

If it is impossible, print a single number `-1`.

Example

standard input	standard output
5 0 1 0 1 0 1 3 1 4 2 5 3 5	5 1 3 1 7 2 5 3 5