Problem A. Virus

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

The immune system of Sponge Bob is a rectangle $n \times m$ consisting of immune cells. Sponge Bob doesn't want to go to school tomorrow, so he needs to get sick.

In order to get sick he is ready to infect some of his immune cells with the ARVI virus. After initial infection, the virus propagation process takes place: every millisecond all healthy cells which have two or more infected neighbors become infected, and nothing happens with already infected cells. Here we consider two cells as neighbors if they share an edge. Sponge Bob will get sick only if all his immune cells will be infected. Since the infecting immune cells is quite painful, Sponge Bob wants to minimize the number of initially infected cells.

Find and output any example of the initial infection with the minimum possible number of infected cells, which leads to Sponge Bob's illness.

Input

The single line contains two integers n and m ($1 \leq n, m \leq 10^3$) — the size of Sponge Bob's immune system.

Output

Print a description of any initial infection with the minimum possible number of infected cells, which leads to Sponge Bob's illness. Output must contain n lines of m characters: 1 if the corresponding cell of the immune system is initially infected, and 0 otherwise. Do not separate the characters with spaces or other delimiters.

Examples

standard input	standard output
1 3	101
2 1	1 1

Problem B. Expected length of the minimum cycle

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

You are given a positive integer N, find the expected length of the minimum cycle in permutation of integers from 1 to N. All permutations are equiprobable.

Consider the answer is an irreducible fraction $\frac{A}{B}$. Output $A \cdot B^{-1} \mod P$, where P is a given prime number. It is guaranteed that $gcd(B, P) = 1$.

Input

The only line of input contains two integers N and P ($1 \le N \le 10^4$, $10^4 < P \le 10^9+33$). It's guaranteed that P is a prime number.

Output

Output the answer to the problem in a single line.

Examples

standard input	standard output
2 1000000007	500000005
3 1000000007	666666673

Problem C. Antipalindrome

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

Consider an alphabet consisting of lowercase and uppercase English letters. Let's enumerate the letters of the alphabet in such a way that lowercase letters have numbers from 1 to 26 (in alphabetical order) and uppercase letters have numbers from 27 to 52 (also in alphabetical order). For example, symbol b has number 2, symbol Y has number 51.

You are given a string s consists of the first k symbols of alphabet. You are also given a matrix C of size $k \times k$, consisting of non-negative integers. The element C_{ij} denotes the cost to change the symbol number i to the symbol number j at some position of the string s. It's guaranteed that $C_{ii} = 0$.

You should change some symbols of the string s in such a way that s will not contain any palindrome substring of length more than 1. Also, find the cheapest way to do that.

Note the changes of the symbol number i to the symbol number j at different positions are counted separately. Note also that you can change the symbol at each position not more than once.

Input

The first line contains one integer k $(1 \le k \le 52)$ — the size of the alphabet.

The second line contains a string s $(1 \le |s| \le 5 \cdot 10^6)$ constisting of only lowercase and uppercase English letters with numbers not greater than k.

The i-th of the next k lines contains k integers C_{ij} $(0 \le C_{ij} \le 10^9)$ — the cost to change the symbol number i to the symbol number j.

Note that the length of the string s may be quite large, so use fast methods to read it (for example function `scanf` in `C++` language and `BufferedReader` class in `Java` language).

Output

If it is possible to get a string without palindrome substrings of length more than 1 print the single integer c — the minimum cost to obtain such a string. Otherwise print the only integer "`-1`" (without quotes).

Examples

standard input	standard output
3 aaa 0 7 5 1 0 1 1 1 0	12
3 abc 0 1 1 1 0 1 1 1 0	0

Problem D. Long Nim

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

Two players are playing famous game Nim. They are given n heaps of stones, i-th of them contains a_i stones. Two players alternate taking any positive number of stones from any single heap. The goal is to be the last to take a stone. It's obvious that this game has predetermined result in case both players are acting optimally. In this problem you should answer how long the game will last if losing player is trying to play as long as possible, but the winning player is trying to finish it as soon as possible without making any moves that could lead him to lose the game. You should also output one of the first possible turns of the first player, that leads to the described result.

Input

The first line of input contains a positive integer n ($1 \le n \le 10^5$).

The second line contains n positive integers a_i ($1 \le a_i \le 10^{12}$) — the sizes of heaps.

Output

In the first line output a positive integer — how many turns the game will last if both players are playing optimally.

In the second line output the turn of the first player in format i k, where i is the index of the heap and k is the number of stones taken from the i-th heap. If there are multiple possible turns which lead to the optimal result, you may output any of them.

Examples

standard input	standard output
2 1 3	3 2 2
2 2 2	4 1 1

Problem E. Guess Table

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

Jury has created a $n \times m$ matrix A consisting of zeroes and ones. You should guess it.

In order to do that you can send no more than 200 queries. At the end you should print the desired matrix.

Each query is described by the matrix B of size $a \times b$ $(1 \le a \le n, 1 \le b \le m)$ consisting of zeroes, ones and questions. The answer to the query is 1 if the matrix A contains continuous submatrix that corresponds to the given pattern B. Otherwise the answer will be equal to 0.

Formally the answer will be equal to 1 if and only if exists some values (i, j) $(1 \le i \le n - a + 1, 1 \le j \le m - b + 1)$, so for each (x, y) $(1 \le x \le a, 1 \le y \le b)$ B_{xy} is a question or $B_{xy} = A_{i+x-1,j+y-1}$.

Also there is another restriction for the query matrix B: there should be no row and no column fully consisting of questions.

Note the problem is interactive, so you should flush your output after each query and at the end of the guess. To do that use function `fflush(stdout)` in `C++` language and `PrintWriter.flush()` in `Java` language.

Input

You are given two integers n and m $(1 \le n, m \le 13)$ — the size of the matrix to guess.

Output

At the end you should print a line with a string `"Answer:"` (without quotes) on the first line. After that print n lines with m symbols “`0`” and “`1`” — the desired matrix. Don’t forget to flush you output after that.

Interaction Protocol

On the first line of each your query print a string `"Query:"` (without quotes) and two integers a, b $(1 \le a \le n, 1 \le b \le m)$ — the size of the submatrix. After that print a lines with b symbols “`0`”, “`1`” and “`?`”. Don’t forget to flush you output after that.

After each query you will get an answer — a single line with number 1 if the desired matrix contains given submatrix and 0 otherwise.

Example

standard input	standard output
2 3	Query: 2 1 0 0
1	Query: 1 2 01
1	Query: 2 2 01 0?
1	Query: 2 3 101 ??0
0	Answer: 101 001

Problem F. Restrooms

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

MIPT university administration is planning to make repairs in the main corridor. Above all, they are going to repair all n restrooms located along the corridor and numbered from 1 to n. Initiative group of MIPT students and professors has made several requests of the following types:

- There should be at least one women's restroom in the segment between l_i^{th} to r_i^{th} restroom inclusive.
- There should be at least one men's restroom in the segment between l_i^{th} to r_i^{th} restroom inclusive.

You should answer if it is possible to satisfy all these requests, and, in case it is possible, output any possible arrangement.

Input

In the first line you are given three integers n, w, m ($1 \le n \le 10^6$, $0 \le w, m \le 10^6$) — number of restrooms, number of requests for women's restroom, number of requests for men's restroom respectively.

In the next $w + m$ lines you are given descriptions of requests, first about women's restrooms, then about men's restrooms. Description of one request consists of two integers l_i, r_i ($1 \le l_i \le r_i \le n$).

Output

In the first line output string ¡¡Yes¿¿ (without quotes), if the way to satisfy all requests exists and ¡¡No¿¿ (without quotes), if it is impossible. If answer is yes, then output in the second line string consisting of n zeros and ones, describing possible way of assigning restrooms to be men's (1) and women's (0).

Examples

standard input	standard output
3 1 1 1 1 3 3	Yes 001
3 1 1 1 1 1 1	No
1 3 0 1 1 1 1 1 1	Yes 0

Problem G. Flying Doors

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

Artem and Kostya are playing a following game. Artem has n doors with free ceiling. The floors of the doors are fixed on zero height, ceiling could change. Each door has two parameters a_i, b_i: a_i is the initial height of the ceiling, and b_i is the velocity of the ceiling. If $b_i > 0$, then the ceiling is moving up b_i meters per second. If $b_i < 0$, then the ceiling is moving down $-b_i$ meters per second until it becomes zero, after that it stops. And if $b_i = 0$, the ceiling is always fixed at height a_i.

The game proceeds as follows: Artem places these n doors at points on $0X$-axis with coordinates $1, 2, \ldots, n$ in some order. Kostya is located at point $x = 0$ just before the game. He chooses some height $h \geq 0$ and velocity $v > 0$ (here we consider Kostya as a point). Then Artem shoots the starting pistol, and the doors' ceilings begin to move and simultaneously Kostya starts his journey on constant height h with constant velocity v in the positive direction of $0X$-axis. Kostya must fly through **all the doors**. So, if Kostya flies through some coordinate i and the ceiling of the door at coordinate i is strictly down Kostya, then Kostya loses.

To make the game more exciting, Kostya decided to choose a randomized strategy. He chooses the height h as a uniformly distributed on $[0; 10^9]$ real number. Then he chooses some $v > 0$ which allows him to win, if it is possible. For each arrangement of doors consider the probability of Kostya's win if he uses this strategy. Artem wants to place doors in such order that this probability is minimized. Help Artem to find this minimal probability. To make the answer format more convenient, output the desired probability multiplied by 10^9.

Input

The first line of input contains one positive integer n ($1 \leq n \leq 5 \cdot 10^5$) — the number of doors Artem has.

Next n lines contain description of doors, each of them contain two integers a_i, b_i ($1 \leq a_i \leq 10^9, |b_i| \leq 10^9$), denoting the initial height and the velocity of the ceiling of the i-th door.

Output

In the only line output a real number — the minimal probability of Kostya's win, multiplied by 10^9. The answer will be considered correct if the absolute or relative error is less than 10^{-6}. Note that this rule applies to the output value, not to the desired probability.

Examples

standard input	standard output
3 1 2 3 -2 2 1	1.5000000000
1 1 2	1000000000.0000000000

Problem H. MIPT Campus

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds (5 seconds for Java)
Memory limit:	256 mebibytes

A part of Dolgoprudny that belongs to MIPT is very convenient for students. Long Pervomayskaya street divides Dolgoprudny into two parts: university (study) buildings are on the left side of this street, and dormitories are on the right side. Assume $0X$-axis is collinear to that street. The width of Pervomayskaya street and distances from the buildings to the street are negligible, so you may consider this model as one-dimentional.

There are exactly n students of MIPT that attend the classes. Every morning each of them goes from his dormitory to his study building, crossing the Pervomayskaya street. i-th student lives in dormitory with x-coordinate a_i and studies in the building with x-coordinate b_i. Students can cross the street only at crosswalks. The are m crosswalks perpendicular to the $0X$-axis which are located in points with x-coordinates $c_1, c_2, \ldots, c_m$. While going to the study building, students always choose the shortest way to reach it.

Recently the Youth Committee of MIPT has concluded that the number of crosswalks is not enough for students. They lobbied the initiative to make a new crosswalk at any point of Pervomayskaya street. So now they want to make a new crosswalk in such a way that the sum of distances walked by all the students on their way from their dormitories to the study buildings is minimized. Your task is to find this minimum value.

Input

The first line contains an integer n ($1 \le n \le 5 \cdot 10^5$) — the number of students attending classes at MIPT.

Each of the next n lines contain two integers a_i, b_i ($1 \le a_i, b_i \le 10^9$) — the coordinates of the dormitory and the study building of i-th student.

The next line contains an integer m ($0 \le m \le 5 \cdot 10^5$) — the number of crosswalks.

The last line contains m distinct integers $c_1, c_2, \ldots, c_m$ ($1 \le c_i \le 10^9$) — the coordinates of crosswalks.

Output

Output one integer in a single line — the minimal sum of distance students have to travel after the building of a new crosswalk. It can be proven that the answer is always an integer.

Examples

standard input	standard output
2 5 10 20 30 0	35
2 5 10 10 20 1 2	15
2 5 10 20 30 1 11	17

Problem I. Tickets

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds (5 seconds for Java)
Memory limit:	512 mebibytes

There are n cities in Berland, which are enumerated from 1 to n. City number 1 is a capital of Berland. Cities are connected by $n-1$ trains, i-th of them connects city a_i with city b_i. Berland train system allows you to get from every city to any other, possibly, using more than one train.

There are m types of the tickets: i-th of them can be bought in city v_i for w_i berland dollars and allows to travel from v_i to any city x such that the distance from v_i to x is less or equal to k_i. The distance is measured in trains used during the travel.

Your task is to find minimal cost to get from some cities to capital.

Input

In the first line you are given three integers n, m, q ($1 \le n \le 10^5$, $0 \le m \le 10^5$, $1 \le q \le 10^5$) – number of cities in Berland, number of different type of tickets, number of queries correspondingly.

Each of next $n-1$ lines contain integers a_i and b_i ($1 \le a_i, b_i \le n$) – cities connected by i-th train.

In the next m lines types of tickets are described: i^{th} contains three integers v_i ($1 \le v_i \le n$) – city where one could buy and use it, k_i ($1 \le k_i \le n-1$) – maximum distance one could travel using this ticket, w_i ($0 \le w_i \le 10^9$) – price of the ticket.

In the next q lines are described queries: i^{th} of them contains integer q_i ($1 \le q_i \le n$) – city, from which you want to calculate cost to get to capital.

Output

For every query output single line with integer – required cost to get from city to capital. If it's impossible to get to the capital using such tickets, print ¡¡Impossible¿¿ (without quotes).

Examples

standard input	standard output
5 4 5 1 2 2 3 3 4 3 5 5 2 10 3 1 40 4 3 1 2 1 100 1 2 3 4 5	0 100 41 1 11
2 0 2 1 2 1 2	0 Impossible

Note

In the first sample, to get to the capital from the city 5, we need to perform the following actions:

– Buy a ticket with type 1 for 10 dollars and travel to the city 4.

– Buy a ticket with type 3 for 1 dollar and travel to the capital city 1.

Note that it's also possible to travel to the cities 3 and 2 using a ticket with type 1 (the distance between 5 and 3 is one train, while the distance between 5 and 2 is two trains, which is less or equal to 2), and get to the capital using tickets 2 or 4, but it would be a lot more expensive to travel to the capital this way.

Problem J. Total control

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

ByteCity is the capital of Byteland. It can be described as a convex polygon with n vertices, surrounded by walls.

The mayor of Bytecity decided to upgrade the weapons that the army of ByteCity is using. If new guns have a firing range d ($d \geq 0$) then the mayor of the city would consider all terrain inside the city and all terrain on the distance no further than d from the city walls as loyal.

Pride of the mayor will be satisfied if loyal area will be at least S. What is the minimum value of guns' firing range he need to buy for his army?

Input

In the first line you are given two integers n and S ($3 \leq n \leq 5 \cdot 10^4, 1 \leq S \leq 10^{13}$) — number of vertices in city polygon and needed area of loyal terrain.

In each of the next n lines you are given two integers x and y ($-10^6 \leq x, y \leq 10^6$) — coordinates of polygon vertices.

It is guaranteed that these n points are vertices of convex polygon given in counterclockwise order.

Output

Output a single number — minimum fire range of guns. Your answer considered will be correct if its absolute or relative error doesn't exceed 10^{-6}.

Example

standard input	standard output
4 2 0 0 1 0 1 1 0 1	0.21402387849518847

Problem K. Mbius

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

For any positive integer n, let's define Mbius function $\mu(n)$. It has values in $\{-1, 0, 1\}$ depending on the factorization of n into prime factors:

- $\mu(x) = 1$ if x is a square-free positive integer with an even number of prime factors.
- $\mu(x) = -1$ if x is a square-free positive integer with an odd number of prime factors.
- $\mu(x) = 0$ if x has is divisible by some squared prime factor.

For example, $\mu(1) = 1, \mu(2) = -1, \mu(6) = 1, \mu(12) = 0$.

You are given two arrays a and b, consisting of positive integers.

Let k_y be the number of pairs $(i, j), 1 \le i \le n, 1 \le j \le m$ such that $\mu(a_i \cdot b_j)$ is equal to y.

Your task is to calculate k_{-1}, k_0 and k_1.

Input

The first line of input contains two integers n, m ($1 \le n, m \le 2 \cdot 10^5$) — the sizes of arrays a and b.

The second line contains n integers a_i ($1 \le a_i \le 10^6$) separated by spaces.

The third line contains m integers b_i ($1 \le b_i \le 10^6$) separated by spaces.

Output

Output three integers k_{-1}, k_0, k_1 separated by spaces in a single line, where $k_y = |\{(i, j) : \mu(a_i \cdot b_j) = y\}|$.

Example

standard input	standard output
6 4 1 2 3 4 5 6 2 3 5 7	6 9 9