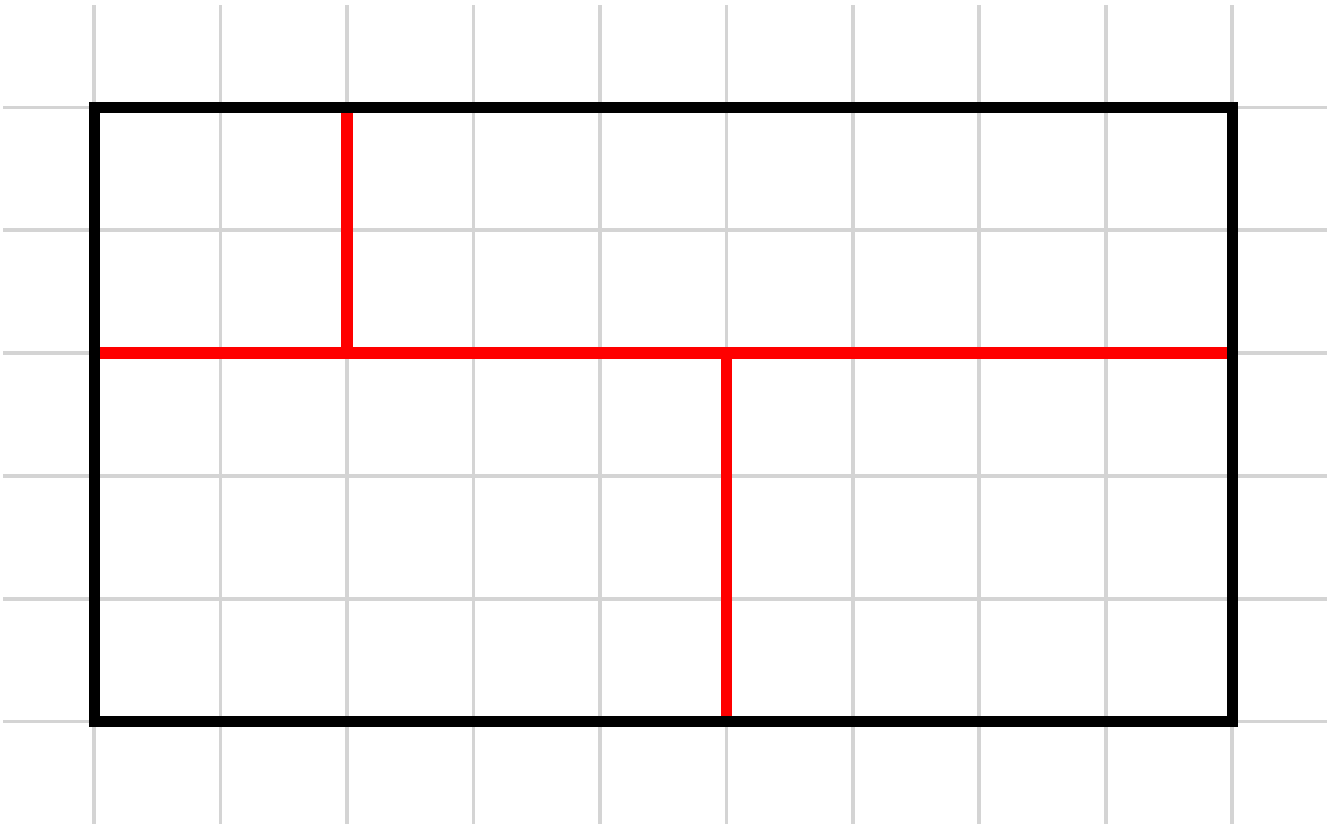
Problem A. Floor Tiles in a Park

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

Grammy is enjoying her holiday in Pigeland City Park. She was interested in the floor tiles in the park. After careful examination, she found out that each of the floor tiles is a $W \times H$ rectangle grid with vertical and/or horizontal colored segments on it. The colored segments have ends on grid points, and they split the rectangle into exactly k subrectangles.

For instance, the following illustration shows a floor tile with $W = 9$, $H = 5$, $k = 4$.



Grammy wants to know the number of different floor tiles satisfying the condition. Please tell her the answer. Since the answer may be very large, you should output the number modulo 998 244 353.

Note that two floor tiles are considered different if and only if a grid line is colored in one tile but not in the other. If two tiles can turn into the same by rotation or reflection, they may still be considered as different tiles.

Input

The only line contains three integers W, H, k ($1 \le W, H \le 10^9$, $1 \le k \le \min(5, W \cdot H)$).

Output

Output a single integer denoting the number of different floor tiles modulo 998 244 353.

Examples

standard input	standard output
2 3 5	7
4 3 5	307
6 372065168 5	114514

Problem B. Efficient Amplification

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

Without the amplifier, the effective signal coverage of a wireless network base station is a circle. The power consumption of the wireless base station depends on the radius of the circle.

The directed amplifier can extend the coverage of the effective signal p times in a specific direction without increasing the power consumption of the wireless base station. That is: the coverage area of the wireless base station using the amplifier is an ellipse, and its power consumption depends on the length of the minor semiaxis.

You have a wireless network station with already installed amplifier placed **at the origin**, and you **can not** change the direction of the amplification (which coincides with the x axis).

Given the locations of several network users on the plane, please choose a minimal length of the minor semiaxos of the coverage ellipse, so, with the help of the amplifier, all users can receive signals.

Input

The first line of the input contains one integer n — the number of network users ($1 \le n \le 5 \cdot 10^4$).

Each of the following n lines contains two integers x and y — the coordinates of the user ($-2 \cdot 10^8 \le x, y \le 2 \cdot 10^8$).

The last line contains one integer p — the amplification rate ($1 \le p \le 100$).

Output

Print the minimal length of the minor semiaxis with absolute error 10^{-3} or better.

Examples

standard input	standard output
2 2 0 -2 0 3	0.66666666666667

Problem C. Oscar's Round Must Have a Constructive Problem

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

Grammy has a sequence A of length n.

Please find a permutation P such that $P_i \neq A_i$ for all i.

Input

There are multiple test cases.

The first line contains a single integer T ($1 \leq T \leq 100\,000$), denoting the number of test cases.

For each test case:

The first line contains a single integer n ($1 \leq n \leq 100\,000$).

The second line contains n integers $A_1, A_2, \ldots, A_n$ ($1 \leq A_i \leq n$).

It is guaranteed that the sum of n does not exceed $500\,000$.

Output

For each test case:

If the permutation does not exist, output "NO" on a single line.

Otherwise, output "YES" one the first line, then output n integers one the second line, denoting the permutation $P_1, P_2, \ldots, P_n$.

Example

standard input	standard output
3 3 3 3 3 3 3 2 1 6 1 1 4 5 1 4	NO YES 1 3 2 YES 4 5 1 2 3 6

Problem D. Chess

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

Given the chessboard $m \times n$, with the leftmost bottom cell painted in black (and usuall chess coloring, i.e. the cells are colored in black or white such that two cells sharing the side have distinct colors).

Count the difference between the number of the black cells and the number of the white cells on board.

Input

Input contains two integers m and n ($1 \le m, n \le 10^{18}$).

Output

Print one integer — the difference between the number of the black cells on board and the number of the white cells on board.

Examples

standard input	standard output
1 1	1
8 8	0

Problem E. Ternary Search

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

Recently, Grammy has learned ternary search in Tony's class. She can find the peak value in an array using this algorithm when the array is unimodal. Here, we say that an array $a_1, a_2, \ldots, a_n$ is *unimodal* if and only if it satisfies one of the following conditions:

- There exists an index k $(1 \le k \le n)$ such that $a_1 < a_2 < \ldots < a_k > a_{k+1} > \ldots > a_n$.
- There exists an index k $(1 \le k \le n)$ such that $a_1 > a_2 > \ldots > a_k < a_{k+1} < \ldots < a_n$.

As the tutor of Grammy, Tony wants to examine whether Grammy fully understands what he taught in class, so he leaves n tasks for Grammy to try ternary search. The tasks are as follows.

Initially, there is an empty array. Each task appends a **distinct** number at the right end of the array, and Grammy should do ternary search on it. However, due to Tony's carelessness, the array may not be unimodal after some addition. Since Tony has already gone to sleep, Grammy has to solve the problem by herself.

For each task, before Grammy tries ternary search on it, some operations should be performed to make it unimodal. In each operation, Grammy can swap the values of a_i and a_{i+1} for some i $(1 \le i < n)$. Grammy is a lazy girl, and she thinks that if she has to perform too many operations, she would instead wait for Tony to wake up and solve the problem. For each task, she wonders what is the least possible number of operations she has to perform to make the array unimodal. Can you help her?

Input

The input contains only a single case.

The first line contains a single integer n $(1 \le n \le 200\,000)$, denoting the number of tasks. The i-th line of the following n lines contains one integer a_i $(1 \le a_i \le 1\,000\,000\,000)$, denoting the number appended in the i-th task.

It is guaranteed that a_i are pairwise distinct.

Output

The output contains n lines. Each line contains one integer, denoting the answer to the i-th task.

Example

standard input	standard output
9 11 4 5 14 1 9 19 8 10	0 0 0 0 2 3 3 6 7

Problem F. Candies

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

Grammy has a circular array $a_1, a_2, \ldots, a_n$. You can do the following operations several (possibly zero) times in any order:

- Choose two adjacent positions with the same number, and erase them.
- Choose two adjacent positions such that the numbers on these positions add up to a special number x, and erase them.

After each time you do an operation successfully, Grammy will give you a candy. Meanwhile, the remaining parts of the array will be concatenated. For example, after deleting the third and fourth element of the array, the second element and the fifth element will become adjacent.

Find the maximum number of candies you can get.

Two positions u and v ($u < v$) are *adjacent* if and only if $u + 1 = v$ or $u = 1$ and $v = L$, where L is the length of the remaining array.

Input

The first line contains two integers n and x ($1 \le n \le 10^5$, $1 \le x \le 10^9$) denoting the length of the array and the special number x.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) denoting the numbers in the circular array.

Output

Output an integer denoting the maximum number of candies you can get.

Examples

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

Problem G. Regular Expression

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

Grammy has recently been interested in regular expressions while focusing on cases where the alphabet consists of characters from '`a`' to '`z`'. Today she asks NIO some questions. Each question gives string A, asking the minimum length of an expression matching string A according to the matching rules, and also the number of such shortest expressions.

To learn detailed rules about how regular expressions match strings, you can refer to `https://en.wikipedia.org/wiki/Regular\_expression`.

Here we only consider characters from '`a`' to '`z`' and special characters '`.`', '`?`', '`*`', '`+`', '`|`', '`(`', '`)`'. It is assumed that the asterisk, the question mark and the plus sign have the highest priority, then follows concatenation and then alternation. Parentheses can be used to change the priority. For example, "`a(b|c)`" can match "ab" and "ac". Parentheses may be omitted when they don't change the priority. For example, "`(ab)c`" can be written as "`abc`", and "`a|(b(c*))`" can be written as "`a|bc*`".

Here are some examples of matching:

- (or): "`gray|grey`" can match "gray" or "grey".
- (question mark): "`colou?r`" matches both "color" and "colour".
- (asterisk): "`ab*c`" matches "ac", "abc", "abbc", "abbbc", and so on.
- (plus sign): "`ab+c`" matches "abc", "abbc", "abbbc", and so on, but not "ac".
- (wildcard): "`a.b`" matches any string that contains an "a", then any single character, and then "b"; and "`a.*b`" matches any string that contains an "a", and then the character "b" at some later point. More precisely, "ab" can be matched by "`a.*b`" but not by "`a.b`".
- (concatenation): Consider expression R = "`(ab|c)`" matching {"ab", "c"}, and expression S = "`(d|ef)`" matching {"d", "ef"}. Then, (RS) = "`((ab|c)(d|ef))`" matches {"abd", "abef", "cd", "cef"}.

Input

The input contains only a single case.

The first line contains a single integer Q ($1 \le Q \le 100\,000$) denoting the number of questions. The i-th line of the following Q lines contains one string A consisting of lowercase English letters ($1 \le |A| \le 200\,000$) denoting the string A of the i-th question. It is guaranteed that $\sum |A| \le 1\,000\,000$.

Output

For each question, output a single line containing two integers: the minimum length of a matching expression and the number of matching expressions of such length. Note that the answers may be extremely large, so please print them modulo 998 244 353.

Example

standard input	standard output
2 a ab	1 2 2 6

Problem H. Kingdom Of Towers

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

In the Kingdom of Towers, there are n cities and $(n-1)$ bidirectional roads between cities. The i-th road is between the a_i-th and b_i-th cities. It is guaranteed that cities are connected.

In the i-th city, there is a defense tower with power p_i. The tower protects all cities with a road directly connected to city i. However, the tower in city i does not protect city i itself.

The famous monster Bytezilla would like to destroy all defense towers. When it tries to destroy the tower in city i, any not-destroyed tower protecting city i will deal damage whose value equals to its power to Bytezilla.

Find out the minimum total damage Bytezilla will receive if it chooses the order to destroy the towers optimally.

Input

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

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 10^4$).

The i-th of the last $(n-1)$ lines contains 2 integers a_i, b_i ($1 \le a_i, b_i \le n$).

Output

Print an integer that denotes the minimum total damage.

Examples

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

Problem I. Least Divisor

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

This is an interactive problem.

The jury have an integer n between 2 and 10^6 written on the paper. You do not know that integer, but you may use the queries of the following format: `?` x — add the integer x between 0 and 2022, inclusively, to n, and tell the least divisor of the **current** value of n, greater, than 1.

Note that all additions are cumulative.

Your task is to guess the initial integer for no more than 60 queries.

Interaction Protocol

The interaction is started by your program.

To ask the query, send the line `?` x, where $0 \le x \le 2022$.

Then you receive one integer — the least divisor of the value of n after increasing by x, that is greater, than 1.

When you are ready to print the answer, send the line `!` n — the **initial** value of n. This action is not counted as the query. The answer can be printed only once.

The interactor is **not adaptive**, i.e. the integer n is generated once before the interactor starts.

Don't forget to print the end-of-line and flush the output (using the `flush` function of your programming language) after each query or the final output. Otherwise your program may receive Idleness Limit error. There are some flush samples: for C/C++ (stdio): `fflush(stdout);`, for C++ (iostream): `std::cout.flush();`, for Java: `System.out.flush();` and for Python: `sys.stdout.flush()`.

Examples

standard input	standard output
7 3 19	? 1 ? 2 ? 8 ! 2022

Note

Note that int he example the contestant did not had enough reason to tell the answer, but he was lucky. Please do not assume that your luck will be the same.

Problem J. Melborp Lacissalc

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

Grammy has a favorite number k. She thinks that all the numbers divisible by k are good.

For each array containing only numbers from 0 to $k-1$, Grammy defines its *goodness* as the number of non-empty consecutive subarrays that sum to a good number.

Please count the number of arrays of length n such that their goodness is t. Since the answer can be enormous, output the answer modulo 998 244 353.

Input

A single line contains three integers n, k, t ($1 \le n, k \le 64$, $0 \le t \le \frac{n(n+1)}{2}$).

Output

Output a single integer denoting the answer modulo 998 244 353.

Examples

standard input	standard output
2 5 1	12
7 10 15	2016
46 50 171	645560469

Problem K. Great Party

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

Grammy joined a great party.

There is an interesting game at the party. There are n piles of stones on the table. The i-th pile has a_i stones in it. Two players participate in the game and operate the stones in turn.

In each player's turn, the player will do the following two steps:

1. Select a **non-empty** pile of stones, select a positive amount of stones to remove from it.
2. Keep the remaining stones in the pile still **or** merge them all into another **non-empty** pile of stones.

Those who cannot operate lose the game.

Now, Grammy has q questions. For each question, she asks you how many sub-segments of $[l, r]$ satisfy that if the piles in the segment are taken out alone for the game, the first player will win.

Input

The first line contains two integers n and q ($1 \le n, q \le 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$).

The i-th of the next q lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$).

Output

The output contains q lines. Each line contains a single integer, denoting the answer to the question.

Examples

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

Problem L. Maximum Range

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

Grammy has a simple connected undirected graph. Each of the edges has a value written on it. Please choose a simple cycle for her such that the values written on the cycle have the maximum possible range.

The *range* of a cycle is the difference between the maximum value and the minimum value written on it.

A cycle $i_1 - e_1 - i_2 - e_2 - \cdots - i_k - e_k - i_1$ (e_j is some edge connecting vertices i_j and $i_{j \bmod k+1}$ in the graph) is simple if and only if each **edge** appears at most once in it.

To prove that you really found the cycle, you need to output the vertices on the cycle in order.

It is guaranteed that there is at least one cycle in the graph.

Input

The first line contains two integers n and m ($3 \le n \le m \le 10^5$) denoting the number of vertices and the number of edges in the graph.

In each of the next m lines, there are three integers u, v, w ($1 \le u, v \le n$, $-10^9 \le w \le 10^9$, $u \ne v$), indicating that there is an edge between vertex u and vertex v having value w written on it. It is guaranteed that the graph is connected, and there is at most one edge between each pair of vertices.

Output

On the first line, output a single integer denoting the maximum range of a simple cycle in the graph.

On the second line, output a single integer k denoting the number of edges in the cycle. It is not hard to find out that the number of edges is equal to the number of vertices in the cycle.

On the last line, output k integers, denoting the vertices on the cycle in order. Note that these vertices can be repeated since only edges cannot be visited multiple times.

If there are multiple solutions, output any one of them.

Example

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

Note

In the first sample, the cycle 1-2-5-4-3-1 has the maximum range of 5, since the maximum value on the cycle is 3, and the minimum value on the cycle is -2, so the maximum range of a cycle is $3 - (-2) = 5$. It can be shown that there are no cycles with a range larger than 5.