Association for
Computing Machinery
INSAT Student Chapter

INSAT - National Institute of Applied Science and Technology

March 3rd 2024

Winter Cup 6.0

Problem Set

Main Contest

Problem A. Maximal String

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 256 megabytes

You are given a **binary** string of length n.
You can take two consecutive 1, remove them, and replace them with 0s. Also, you can take two consecutive 0, remove them, and replace them with 1. You can do these operations any number of times (possibly zero). You have to find the **lexicographically largest** string you can obtain.

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a \neq b$;
- In the first position k where a and b differ (from left to right), the character a_k is smaller than the character b_k

Input

The first line contains one integer t ($1 \leq t \leq 10^4$): the number of test cases.
The first line of each test case contains one integer n ($1 \leq n \leq 10^5$): the length of the binary string s.
The second line of each test case contains a binary string consisting of n characters.
It is guaranteed that the sum of n over all test cases does not exceed 2×10^5

Output

For each test case, output t: the lexicographically largest string you can obtain.

Example

standard input	standard output
6 6 000010 6 101000 6 011000 6 110101 14 01010101010100 14 10101010110101	1110 110 110 110101 1 110101

Note:

- **A binary string** is a string consisting only of zeros and ones.
- The character '0' is **smaller than** the character '1'.

Problem B. Balanced Tournament

Input file: standard input
Output file: standard output
Time limit: 1 seconds
Memory limit: 256 megabytes

Each year, Hogwarts organizes a tournament between different schools to win the Winter Cup.

There are N participating schools. Each school is represented by a champion. Each champion is selected by the Goblet of Fire, a magical object tasked with the selection. The i^{th} champion has power P_i. Dumbledore wants to make this year's edition even more exciting so he introduced two new potions. The first potion increases the power of the wizard by K while the second potion decreases it by K. Dumbledore will give each wizard at most one potion. He wants to minimize the difference of power between the strongest and the weakest wizard.

Help Dumbledore balance the tournament.

Input

The first line contains an integer T $(1 \le T \le 500)$ denoting the number of test cases.

The first line of each test case in the input contains two integers N $(1 \le N \le 10^3)$ and K $(0 \le K \le 10^9)$.

The second line contains N integers $P_1 \ldots P_N$ $(1 \le P_i \le 10^9)$ representing the original power of each wizard.

It is guaranteed that the sum of N across all test cases doesn't exceed 5×10^3

Output

Print the smallest possible difference of power.

Example

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

Problem C. Cryptocurrency

Input file: standard input
Output file: standard output
Time limit: 1 seconds
Memory limit: 256 megabytes

Many of you know *Yessine*, but very few know that he is a very successful entrepreneur. In fact, he founded his start-up `*Iceberg Corporation*` specialized in cryptocurrency a few years ago. Even though he is millionaire, he keeps going to INSAT to compete at the ICPC regionals.

As time went, *Yessine*'s schedule has gotten busy. He decided to hire his first employee. He hired his friend *Oussama* to work as a software engineer, a data scientist and a business analyst while paying him minimal wage. *Oussama* was very motivated to work so he didn't care about the money. He just wanted to solve hard problems.

One time, he was analyzing the evaluation of cryptocurrency market and came across the following problem:

Given a sequence A of N numbers denoting the prices of a certain cryptocurrency over a period of time, find the highest contiguous increase in price (HCI). The HCI is the largest possible difference between the last and first element over all **contiguous** increasing subsequences.

* A contiguous subsequence $A_i, A_{i+1}, A_{i+2}, \ldots, A_j$ is an increasing sequence if it satisfies $A_i < A_{i+1} < A_{i+2} < \ldots < A_j$

Can you solve this problem?

Input

The first line contains the number of test cases $1 \leq T \leq 100$.

The first line of each test case contains an integer N $(1 \leq N \leq 10^5)$ denoting the length of the sequence. The second line of each test case contains N integers denoting the sequence A $(0 \leq A_i \leq 10^9)$.

It is guaranteed that the sum of N over all testcases doesn't exceed 10^6.

Output

Print a single integer: the highest contiguous increase of the cryptocurrency.

Example

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

Problem D. Divisibility Game

Input file: standard input
Output file: standard output
Time limit: 1 seconds
Memory limit: 256 megabytes

As stated in a previous problem, *Yessine* is a wealthy business man who founded *Iceberg Corporation* and hired *Oussama* to work for him. The start-up grew significantly so he bought a big office and hired one additional employee who can't be anyone other than *Rami*. The three of them would meet every evening (because none of them would wake up in the morning) and they would work together. Everything was perfect, except for one thing... Sometimes, *Rami* and *Yessine* will make *Shisha* and start smoking in the office. *Oussama* doesn't really like smoking so he asked them to stop. They refused at first, but he kept insisting so they told him they will stop if he wins in a divisibility game.

Oussama and *Rami* will play on an array A of n positive integers, and they agreed on some **odd integer** k before starting the game. *Oussama* and *Rami* will make alternating moves. But this time, and unlike previous *Winter Cups*, *Oussama* insisted on going first. In each turn, the current player will play as follows:

- If every element is divisible by k, the player loses the game.
- Otherwise, the player will have to choose two elements A_i and A_j removes them and adds $A_i + A_j$ to the end of array.

It's guaranteed that the sum of the elements of the array is divisible by k.
Oussama is very good at games, but so is *Rami*, so both players play **optimally**. Help *Oussama* win before his lungs get sick from Shisha.

Input

The first line of the input contains two integers $n, k,$ with:

- $1 \leq n \leq 10^5$: The size of the array.
- $1 \leq k \leq 10^9$: An **odd** integer defining the game.

The second line contains n integers, the elements of the array $0 \leq A_1, \ldots, A_n \leq 10^9$

Output

"Oussama" if *Oussama* will win. Otherwise "Rami" if *Rami* will win.

Examples

standard input	standard output
3 5 0 2 3	Oussama
3 5 1 1 3	Rami

Note

- In the first test case, *Oussama* can select elements $A_2 = 2$ and $A_3 = 3$ and replace them with $5 = 2 + 3$. The new array will be $A' = [0, 5]$. *Rami* will not be able to make a move, and so *Oussama* will win.
- In the second test case, *Rami* will win no matter what *Oussama* makes on his first move.

Problem E. Enemies of the heir... beware

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

Harry was in detention when he started to hear a strange voice. He went out and kept following the voice until he found a message written in blood on the wall. The message said *"The Chamber of Secrets has been opened, enemies of the heir... beware"*.

There was also a sequence of N integers. *Harry* looked into the mystery and found out that there is a secret room where a giant Basilisk is living. However, he couldn't understand the written numbers. Fortunately, *Moaning Myritle* was passing by. She told him that the sequence of numbers on the wall is a **prefix function** used to open the door to the chamber. In fact, *any* sequence of numbers that has the given prefix function will open the door.

The prefix function of an array a of size n is a sequence $P_1, P_2...P_n$, where P_i is the maximum value of k such that $k < i$ and $A[1..k] = A[i - k + 1..i]$
($A[l..r]$ denotes a sub-array of an array A from a position l to a position r, inclusive).
In other words, it's the **longest proper** prefix of the string $A[1..i]$ that is equal to its suffix of the same length. Note that the empty array is both a common prefix and suffix of **all arrays.**

Finally, $A[i..j]$ is the empty array if $i > j$.

For example, for the array $A = [1, 2, 1, 3, 1, 2, 1]$, the values of the prefix function in positions $1, 2, ..., 7$ are equal to $P = [0, 0, 1, 0, 1, 2, 3]$.

Help *Harry* find a valid sequence of numbers.

Input

Each test contains multiple test cases. The first line contains the number of test cases $1 \le T \le 10^4$. Description of the test cases follows.

The first line of each test case contains a single number $1 \le n \le 10^5$: The length of the array P.

The second line of each test case contains n integers $0 \le P_1, P_2, \ldots, P_n \le n$.

It is guaranteed that the sum of the values of n for all test cases does not exceed 2×10^5.

Output

If such an array exists, output n integers $0 \le A_1, \ldots, A_n \le n$. Otherwise, output -1.

Example

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

Problem F. Find The Tree?

Input file: standard input
Output file: standard output
Time limit: 1 seconds
Memory limit: 256 megabytes

"Master has given Dobby a tree, Dobby is free"

Harry has opened the Chamber of Secrets and defeated the Basilisk inside using the *Sword of Griffyndor*. However, he won't rest until he frees his house-elf friend *Dobby* from his evil master *Lucius*. To do so, *Harry* will give *Dobby* a special gift to free him, a tree! However, he should give him a very specific tree, or else it won't count as a gift. He doesn't know what the tree looks like, but for each node in the tree, he knows the furthest node from it.

A tree, **by definition**, is a connected graph with n nodes and $n-1$ edges. Formally, for each node $u \in \{1, \ldots, n\}$, we know a node v such that :

$$\text{dist}(u, v) = \max_{1 \le w \le n} \text{dist}(u, w)$$

Where $\text{dist}(u, v)$ is the minimum number of edges that have to be traversed to get from u to v.
Harry wants to know if it possible for such a tree to exist. Help him free *Dobby*.

Input

The first line in the input contains one integer n: the number of vertices in the tree ($2 \le n \le 10^5$).

The second line contains n integer v_i : the furthest node from node i ($1 \le v_i \le n$).

Output

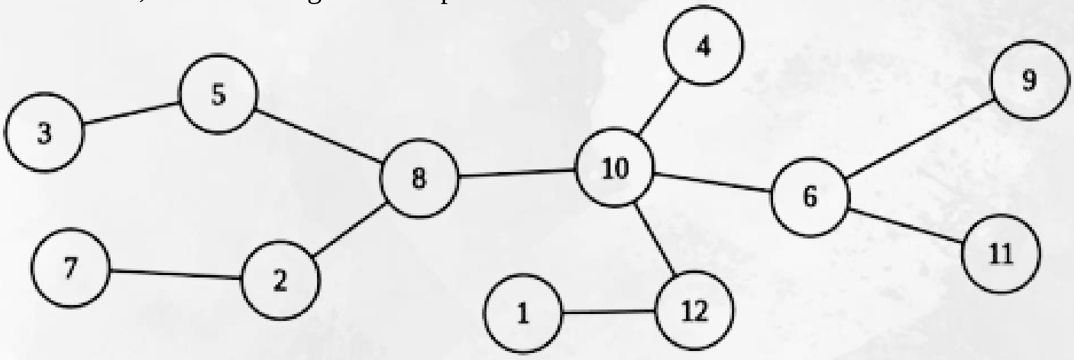
Print "YES" if Harry can construct the tree and "NO" otherwise.

Examples

standard input	standard output
5 5 2 1 3 2	NO
12 7 1 9 7 11 3 1 9 7 3 3 3	YES
2 2 1	YES

Note

In the second test case, the following tree is a possible construction:



Problem G. Grid Crash

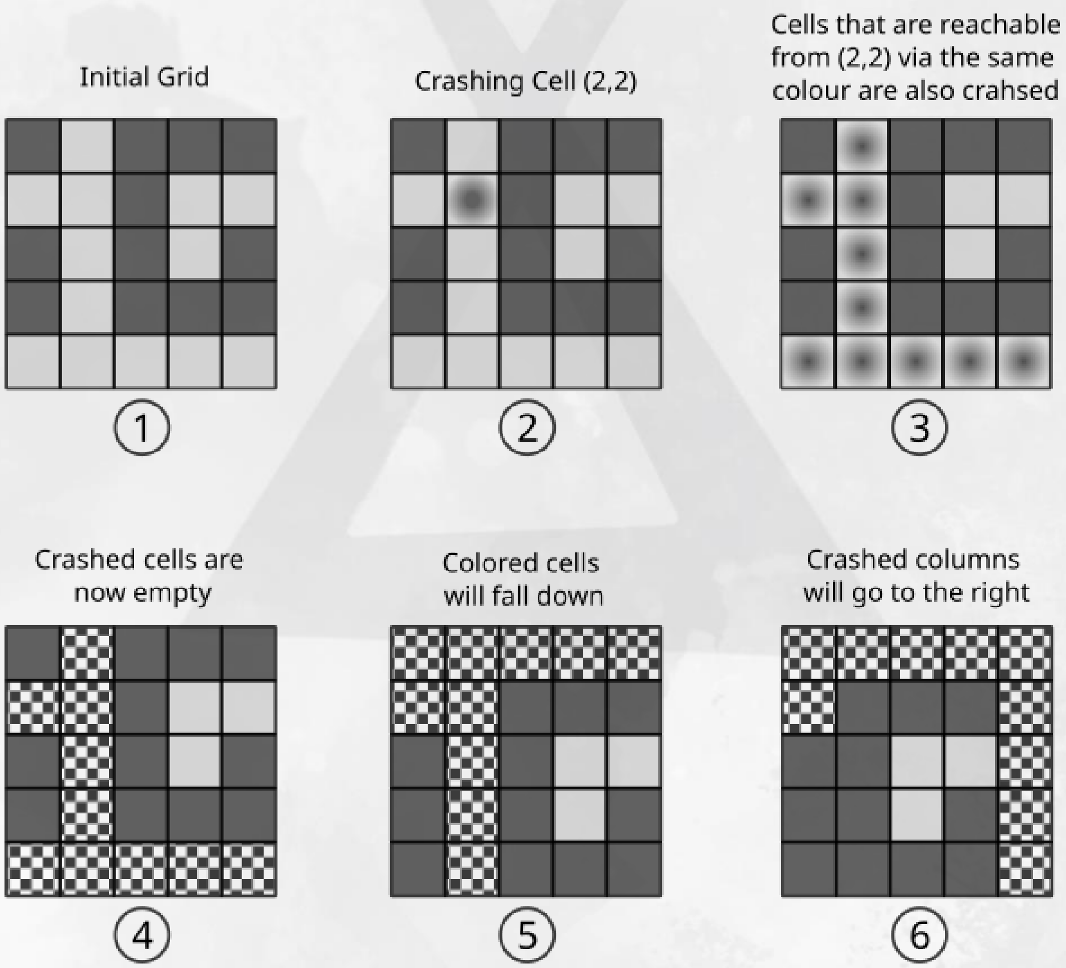
Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 384 megabytes

Since getting his job at *Iceberg Corporation*, *Rami* has been ignoring his tasks and playing a new game called *Grid Crash*. It is a game played on a grid G of size $n \times m$. Each cell in the grid is either **black**, **white**, or **empty.**

Initially, each cell (i, j) in the grid contains some color $c_{i,j}$.
Now the game is played as follow:

- *Rami* selects some cell (i, j) with color $c_{i,j}$ and removes the color.
- This induces a **chain reaction**, on which adjacent cells of (i, j) **with the same color** are also removed. Also, the adjacent cells of those that have color $c_{i,j}$ are removed, and so on.
- After that, each **non empty** cell $(a-1, b)$ that is on top of an **empty cell** (a, b) will swap their content. That is $(a-1, b)$ will be empty, and (a, b) will contain $c_{a-1,b}$. This will be repeated until all empty cells will be **on top** of colorful cells.
- After that, if we have a column C_i of empty cells, then we swap its content with C_{i+1}. This will be repeated until empty columns will be a the rightmost.
- *Rami* will repeat step 1 until the grid has no remaining color.

To illustrate how the game works, we will show the following example of a move on a grid:



Now, *Rami* wants to finish the game in the **fewest** moves possible. Please help him.

Input

1. The first line contains two integer $1 \leq n, m \leq 5$: The dimensions of the grid.
2. The following n lines contains each m characters $c_{i,j} \in \{B, W\}$: The $(i, j)^{th}$ cell of the grid.

Output

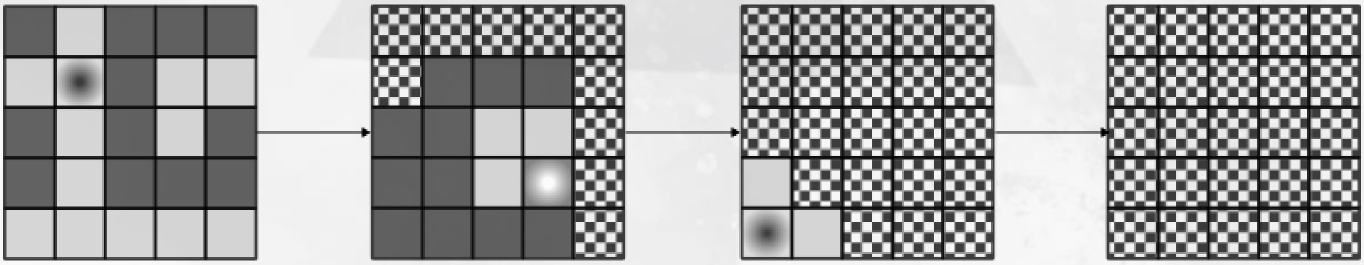
A single integer r: the minimum number of moves to win the game.

Examples

standard input	standard output
5 5 BWBBB WWBWW BWBWB BWBBB WWWWW	3
5 5 WWWWW WBBBW WWWWW WBBBW WWWWW	2
5 5 WBBBW BWBWB BBWBB BWBWB WBBBW	3

Note

The following example shows a sequence of moves that solves the first game (test case 1) in 3 moves. It can be shown that it is optimal:



Problem H. Hide the money

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

As *Iceberg Corporation* kept growing and making significant profit, *Yessine* went from millionaire to billionaire. He got really greedy and decided to hide his money from the government to avoid taxes.

To do so, he decided to put all his money in bags and bury them in *Sfax*. As you may know from previous *Winter Cups*, *Sfax* is an $N \times M$ rectangular grid. *Yessine* has K bags of money. He will hide at most one bag per grid cell. However, he won't place them in random cells. Instead, he will bury a bag in a cell if it maximizes sum of Manhattan distances from all the other cells in *Sfax*.

Formally, he wants to maximize S defined as:

$$S = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{K} D(i, j, k)$$

Where $D(i, j, k)$ denotes the Manhattan distance between cell (i,j) and the k^{th} piece. The Manhattan distance between (x, y) and (x', y') is $|x' - x| + |y' - y|$.

Help *Yessine* hide the money before the government capture him.

Input

The first line contains an integer $1 \leq T \leq 100$ denoting the number of test cases.

Each test case contains three integers N, M, K such that $1 \leq N, M, K \leq 2 \times 10^4$ and $K \leq N \times M$.

Output

Print a single integer S: the maximum possible sum.

Example

standard input	standard output
4 2 2 1 2 2 2 3 3 6 4 4 12	4 8 102 512

Note

It is guaranteed that **within the given constraints**, for each test case, the result will not exceed 10^{18}.

Problem I. Inclusion And Diversity

Input file: standard input
Output file: standard output
Time limit: 1 seconds
Memory limit: 256 megabytes

Iceberg Corporation is desired by many working students thanks to its smart ads strategies. Each year, *Yessine* will get a very large pool of (unpaid) internship applicants. Usually, *Yessine* will only select beautiful girls without reading any resume. However, this year, he has heard that companies with *diversity and inclusion* are more likely to attract investors. So, he decided to include people from different ethnicities.

To select candidates, *Yessine* has prepared a list of m minority groups. For each minority, each candidate either belongs or doesn't belong to that minority group.

To maximize diversity, *Yessine* will select candidates in such a way that no applicant is **more diverse** than another one. However, if two or more candidates are **equally diverse**, then *Yessine* will only select one of them.

A candidate A is more diverse than B if the set of minority groups of B is included in the set of minority groups of A. Two candidates are equally diverse if they belong to **exactly** the same minority groups.

Now, 2024 is the year of diversity and inclusion, and *Yessine* wants to attract more investors. For that reason, he will **maximize** the number of interns. But still, he will keep in his selection criteria both conditions stated above.

The application pool is very large, and it contains **every possible set of minority groups**. What is the maximal number of applications *Yessine* can accept under both criteria. Since the answer can be large, output it modulo $\boldsymbol{M = 998'244'353}$

Input

The input contains one integer $1 \le m \le 4 \times 10^3$: The number of minority groups.

Output

Output one integer, representing the desired answer modulo $998'244'353$.

Examples

standard input	standard output
1	1
2	2
3	3
1936	794915107
2024	109427260

Note

In the second test case, we have two minority groups. Let's say they are {Asian, Hispanic}. We have 4 possible profiles in total: {}, {Asian}, {Hispanic}, {Asian, Hispanic}. *Yessine* can select at most 2 candidates under his criteria, the first belongs to minority groups {Asian}, and the second belongs to minority groups {Hispanic}. Remember that the existence of candidates belonging to any **mix** of these minorities is **guaranteed.**

For the third test case, let {Asian, Hispanic, Middle Eastern} be the set of minority groups. *Yessine* can select 3 candidates with the following respective profiles: {Asian}, {Hispanic}, {Middle Eastern}. It can be proven that he cannot select more under his criteria.

Problem J. New Language

Input file: standard input
Output file: standard output
Time limit: 1 seconds
Memory limit: 256 megabytes

While at *Iceberg Corporation*, *Oussama* worked on a high-frequency cryptocurrency trading system. He tried to implement it using multiple programming languages but they were all slow. Therefore, he decided to make his own programming language and use it internally in the company. He prepared the specifications and sent them to *Rami* who will write a compiler for that. While doing this task, *Rami* encountered a problem: sometimes, *Rami* will take some N elements of different sizes and try to store them in memory. The i^{th} element has size 2^{S_i}.

Computer systems are very complicated. You can think of the memory as a bloc of cells. The cells are numbered 0,1,2,3, ... and so on (To ease things, you may assume that the memory is infinite). Each cell can store one unit of storage, and the storage starts from position 0. For example, if you were storing 3 pieces of sizes $2, 4, 1$ the first element will occupy cells $0, 1$, the second will occupy cells $2, 3, 4, 5$ while the third will occupy cell 6. The smallest cell occupied by an element is called *memory address*. In the previous example, the address at which we stored our elements are 0,2,6. Now here is the catch: in reality, computers only store elements of size 2^{S_i} in addresses divisible by 2^{S_i}. This is done to make access faster. For example, elements of size 2 can only be stored in addresses $0, 2, 4, 6, \ldots$ while elements of size 4 can occupy addresses $0, 4, 8, 12, \ldots$ So in the previous example, the placements of the elements isn't valid. What a computer will do in reality is place the element 2 at address 0, leave cells 2 and 3 empty and place the element of size 4 at address 4 then place element of size 1 at address 8, occupying a total of 9 units of storage. Notice that we wasted 2 unites of storage so we can *align* the element of size 4 with address 4. This problem is called *Memory alignment*. (You thought this was a programming contest, not a computer architecture course, didn't you?)

While trying to optimize the runtime performance of the new programming language, he wrote an algorithm to optimize the placement of elements in memory. In fact, the algorithm *won't take elements in order* but will rearrange them in some order to **minimize** the total space occupied by them. *Oussama* wanted to test *Rami* 's algorithm so he asks you to write an algorithm that outputs the **minimum** possible space occupied by the elements.

Input

The first line contains the number of test cases $1 \leq T \leq 10^3$

The first line of each test case contains only one integer $1 \leq N \leq 10^3$.

The second line of each test case contains N integers $0 \leq S_1, \ldots, S_N \leq 20$.

Please note that the size of element i is 2^{S_i}.

Output

Print a single integer: the minimum possible memory occupied by the elements.

Example

standard input	standard output
3 3 0 0 0 3 1 2 2 3 1 2 4	3 10 22

Problem K. Unique Disk Identifier

Input file: standard input
Output file: standard output
Time limit: 2.5 seconds
Memory limit: 768 megabytes

Iceberg corporation has made many breakthroughs in many areas, such as cryptography and artificial intelligence, where they were able to register many patents.

One of them is a powerful automated personnel identification system that does not suffer from any practical security exploits. It was carefully designed by its prominent engineers: *Yessine*, *Rami* and *Oussama*. This system identifies each person with his unique disk.

In fact, each disk is composed of n sectors, where the i^{th} sector is divided in A_i isometric chunks. Each chunk has its own color from a list of K available colors.

To have login access, each person will insert his disk into a disk scanner, that identifies the person and gives or denies him access. The scanner is smart enough to detect a flipped disk, and correctly realign any sector by rotating it.

The security of this system relies heavily in the number of different configurations of disks. Calculate the number of different disk configurations. As the number can be very big, you only have to calculate it modulo $M = 998244353$.

Two configurations are equivalent if you can get one from the other by rotating each sector by **any angle**, and flipping the whole disk around **any axis** any number of times.

Input

- The first line contains two integers $1 \leq n \leq 10^5$ and $1 \leq K \leq 10^9$ representing respectfully the number of sectors and the number of colors.
- The second line contain n integers $1 \leq A_1, \dots, A_n \leq 5 \times 10^5$ representing the number of chunks for each sector.

Output

One integer R, equal to the number of distinct colorings modulo $M = 998244353$.

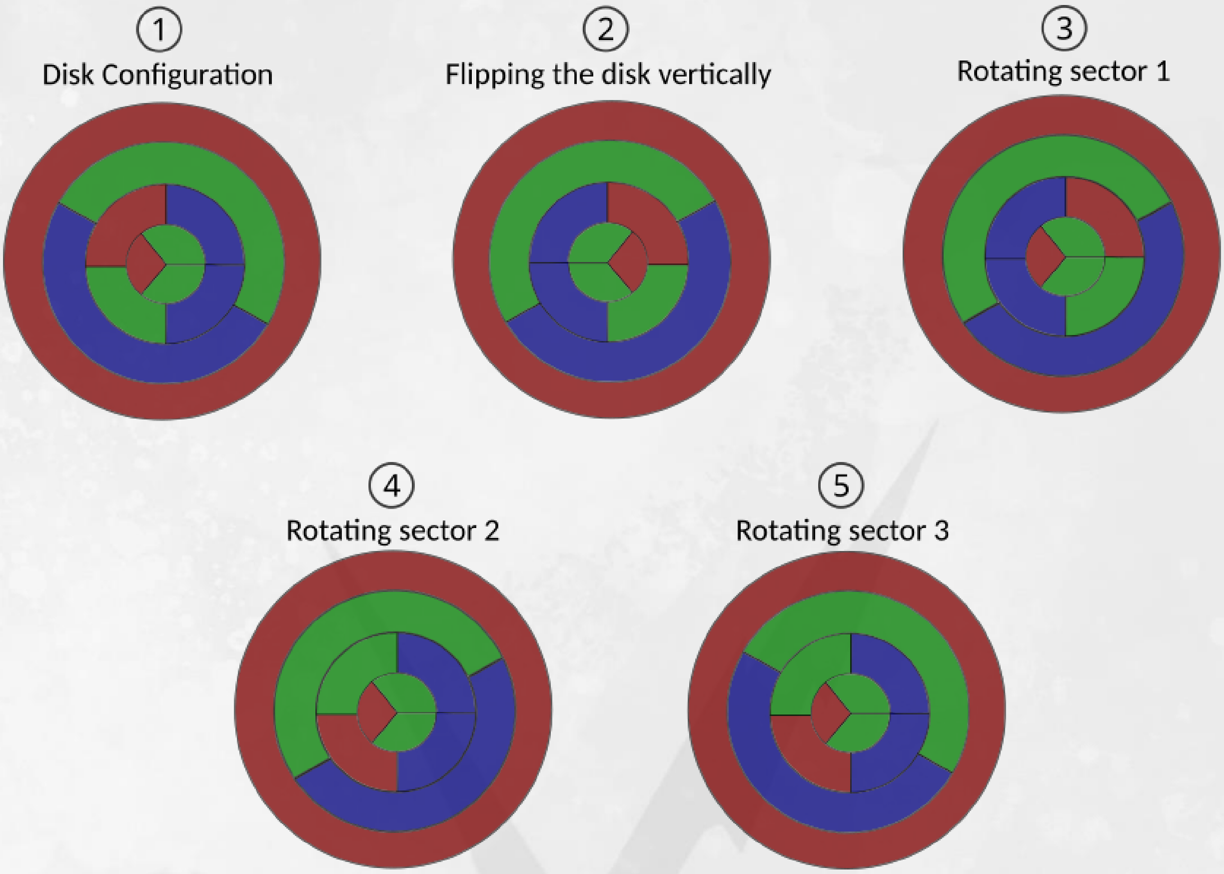
Examples

standard input	standard output
4 3 3 4 2 1	3834
2 2 4 2	18

Note

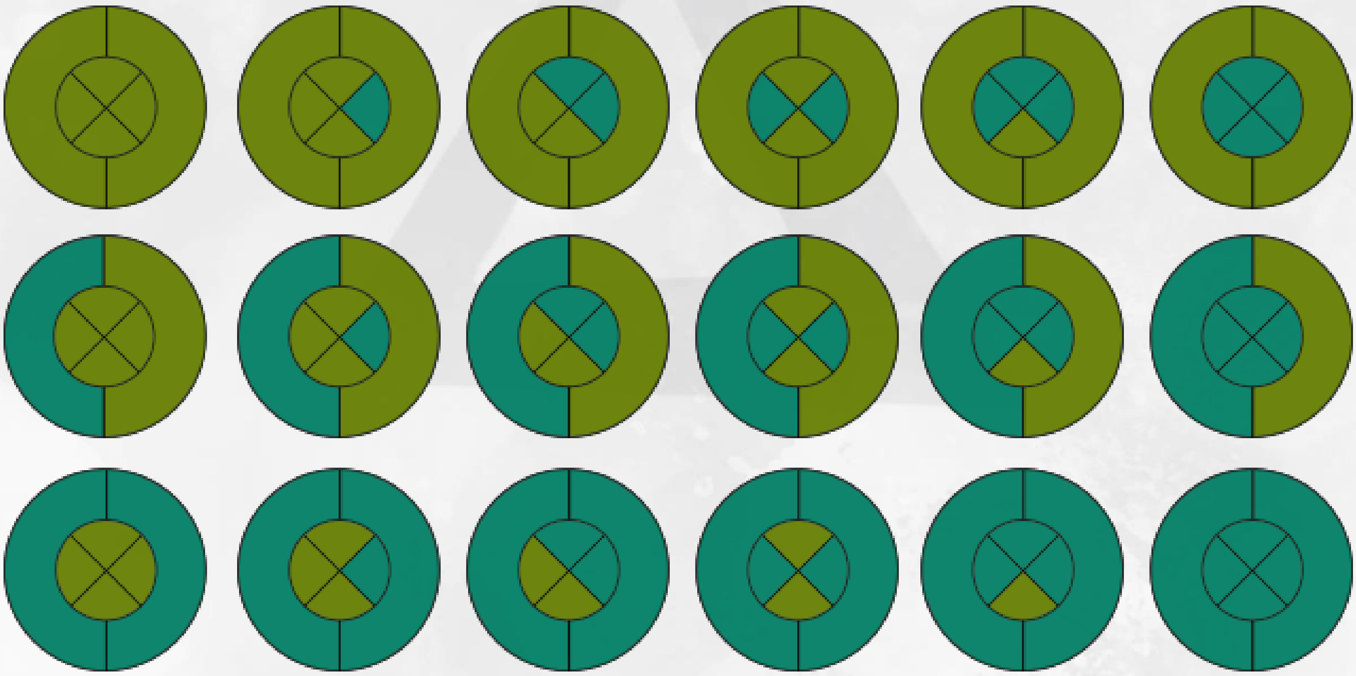
The first test case corresponds to a system designed as follows:

- The system supports $K = 3$ available colors.
- The disk is composed of 4 sectors.
- $A = [3, 4, 2, 1]$ is the array of chunks, starting from the innermost sector.



The example above corresponds to five configurations of the same disk. In the system, they will be detected by the scanner as **the same** disk. It can be shown that there 3834 **distinct** disks in total.

For the second test case, we can enumerate all distinct disks as follows:



Problem L. LCM Guess

Input file: standard input
Output file: standard output
Time limit: 12 seconds
Memory limit: 256 megabytes

This is an interactive problem.

There is a **hidden** permutation P of length n and you have to find it. For this you have to choose **two different** integers i and j and the jury will give you lcm(P_i, P_j).

You can make no more than $2n + 100$ queries in order to guess P.

A permutation is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, [3, 1, 2, 5, 4] is a permutation, but [1, 2, 1] is not a permutation (1 appears twice in the array) and [2, 1, 4] is also not a permutation ($n = 3$, but 4 is in the array).

Input

One integer $3 \leq n \leq 10^5$: The size of the **hidden** permutation P. Then, the interaction will start.

Interaction Protocol

After reading the length of the permutation n, the interaction protocol is as follows:

1. To get lcm(P_i, P_j), output the query in format ? i j where $1 \leq i \neq j \leq n$. You can make at most $2n + 100$ queries.
2. When you find P, output P in format ! P_1 P_2 ... P_n. Printing the permutation is not counted as one of $2n + 100$ operations.
3. In the first occurrence of a query **not conforming** to the given format, you will receive −1 and you should exit immediately to get a Presentation error verdict.

Example

standard input	standard output
3	
	? 1 2
3	
	? 1 3
2	
	? 2 3
6	
	! 3 1 2

Note

After printing a query or the answer, do not forget to output a the end of line and flush the output. Otherwise, you will get **Idleness limit exceeded**. To do this, use:

- **fflush(stdout)** or **cout.flush()** in C++
- **fflush(stdout)** in C
- **System.out.flush()** in Java
- **stdout.flush()** in Python

Problem M. Modular Universe

Input file: standard input
Output file: standard output
Time limit: 2.5 seconds
Memory limit: 256 megabytes

In the modular universe, everything is modular, even coordinates!

For example, if someone has coordinates (x, y) in the normal universe, he will have coordinates $(x \bmod n, y \bmod m)$ in some modular universe, where (n, m) is the modularity of that universe.

Now, $n \times m$ people live in the (n, m)-modular universe. They are usually found on different positions in the universe, but there is always a day called the singularity day, where all the population will gather at the same coordinate $(0, 0)$. This is a very special holiday as **only** singularity days have this property.

Once the singularity day finishes, for each $k \in \{0, \ldots, nm - 1\}$, the k^{th} person will go to the point $((k \text{ div } m) \bmod n, k \bmod m)$ on the next day, and from there on, that person will continue her journey to $((2k \text{ div } m) \bmod n, 2k \bmod m)$, $((3k \text{ div } m) \bmod n, 3k \bmod m)$, etc... in the subsequent days.

Now, the most important phenomenon in history is the modular big bang, which is the **only** day in history on which all modular universes have a singularity day **at the same moment.**

You are from universe $(0, 0)$, where you don't have modular coordinates, and you are asked to inquire more intel about the patterns of universe (n, m). You will do that by inspecting the number of people residing in the same point (x, y) during some day d.

Your supervisors shared to you some important intel, which is the day on which the modular big bang occurred. In fact, it **is denoted** as day 0 by your agency's convention. He sent you also Q queries, where each one is of the form 'n m x y d'. For each one of these queries, you have to find the number of people residing in point (x, y) in day d in the (n, m)-modular universe. Good luck with that!

Input

- The first line contains an integer $1 \le Q \le 10^5$ denoting the number of queries.
- Each of the following Q lines contains each 5 integers n, m, x, y, d where:
  - $1 \le n, m \le 10^9$: The modularity of the universe.
  - $0 \le x < n$ and $0 \le y < m$: The coordinates of the inspected position.
  - $0 \le d \le 10^9$: The day of inspection, with respect to the modular Big Bang.

Output

For each query, output one integer r denoting the number of persons residing at position (x, y) during day d in the universe (n, m).

Example

standard input	standard output
5 1 1 0 0 1 2 2 0 0 1 10 20 0 0 30 10 20 3 1 16 10 20 0 0 5	1 1 100 0 25

Note

In the queries, the time is represented with respect to the modular Big Bang. In fact, day d means d days since the Big Bang. In particular, day 0 is the day of the modular Big Bang.

Problem N. Infinite money glitch

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

Rami got tired of working for a low salary. He asked for a raise but *Yessine* told him that the best he can do is buy him a new Shisha. *Rami* got angry and quit the company. Now he wants to get rich quickly. So, he decided to use the knowledge he acquired at *Iceberg Corporation* and become a cryptocurrency trader.

In fact, the cryptocurrency market has n currencies. Also, there are m pairs of currencies that can be exchanged. For each pair (u, v), there is a rate $r_{u,v}$ that represents the exchange rate of the currency u to the currency v. But that exchange is not free, there is a fee $f_{u,v}$ that is deducted from the exchanged amount according to the following formula:

$$\text{amount}(v) = r_{u,v} \times (\text{amount}(u) - f_{u,v})$$

Rami is broke, but he can still borrow up to x amount of money of currency 0 from *Oussama*. He wants to know the **minimal integer** amount of money he should spend so that he can end up with **strictly more money** by only converting currencies and make a win.

Rami can exchange currencies as many times as he wants. But the only condition is the final amount must be in the currency 0.

Formally, let z_1 be the amount that *Rami* has borrowed from *Oussama* **expressed in currency** 0, and let z_2 the amount that he got after the series of conversions, and **expressed in currency** 0 **as well**. *Rami* will make a win if and only if $\Delta z = z_2 - z_1 > 0$.

Input

The first line contains three integers n, m, x, with $1 \le n, m \le 1000$ and $0 \le x \le 10^9$
The next m lines contain 4 numbers each $u_i, v_i, r_{u_i,v_i}, f_{u_i,v_i}$ with:

- $u_i \to v_i$: denotes an exchange from currency u_i to currency v_i, where $1 \le u_i \ne v_i \le n$.
- $0.01 \le r_{u_i,v_i} \le 100$: A real denoting the exchange rate of the currency u_i to the currency v_i.
- $0 \le f_{u_i,v_i} \le 1000$: A real denoting the fee deducted from the exchanged amount.

The following **is guaranteed**:

- Along any trajectory $\Pi = (p_1, \ldots, p_r)$ of length $r \le n$,the product of all rates along the path is less than 10^{14} :

$$\prod_{i=1}^{r} r_{p_i,p_{i+1}} \le 10^{14}$$

- There is **at most** one edge between any pair of currencies.
- Both r_{u_i,v_i}, f_{u_i,v_i} are represented in the form a.b where a, b are integers and $0 \le b \le 99$.

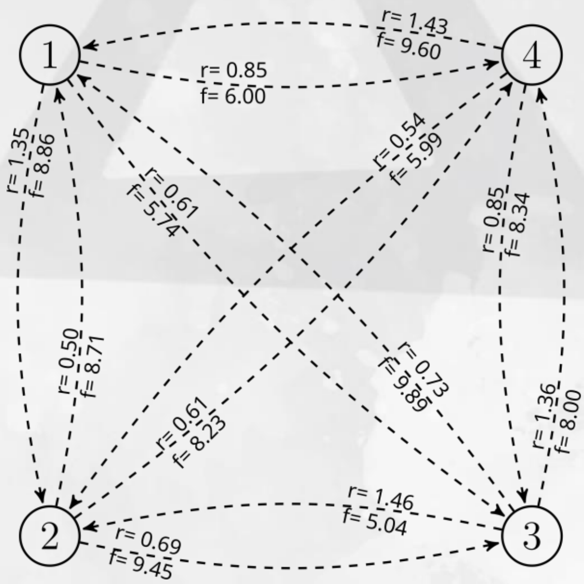
Output

Output M: The minimum **integer** amount he needs to borrow so that he can make a win. If it is impossible to win any amount of money or *Rami* needs to borrow more than x, output -1

Examples

standard input	standard output
2 2 1000 1 2 1.01 10.00 2 1 1.01 10.00	-1
3 4 5000 1 2 0.70 1.00 2 1 1.60 1.00 1 3 0.80 5.00 3 1 2.00 5.00	23
5 5 1000 1 2 1.00 1.00 2 3 1.00 1.00 3 4 1.00 1.00 4 1 1.00 1.00 1 5 10.00 0.00	-1
4 12 1000000000 1 3 0.61 5.74 1 2 1.35 8.86 1 4 0.85 6.00 2 4 0.61 8.23 2 3 0.69 9.45 2 1 0.50 8.71 3 2 1.46 5.04 3 4 1.36 8.00 3 1 0.73 9.89 4 3 0.85 8.34 4 2 0.54 5.99 4 1 1.43 9.60	72

Note



The graph above is a representation of the third test case. *Rami* needs to borrow only 72 to win money. By doing exchanges along the trajectory $1 \to 2 \to 3 \to 4 \to 1$. He will finish with $\sim 72.415768568 > 72$. It can be proven that in this case, 72 is the smallest amount that will make a win.