The 11th China Collegiate Programming Contest Finals

Contest Session

April 26, 2026

This problem set contains 13 problems and spans 19 pages.

ID	English Title	Chinese Title
A	White Night	白夜
B	Yearning for Yonder 2	对远方的向往 2
C	Four Kubic Theorem 2	四太祖先父母定理 2
D	Call You With Your Name	连名带姓
E	Call You With Your Name 2	连名带姓 2
F	Many Many Heads 2	很多很多头 2
G	Puzzle: The Artisan of Glimmith	谜题：格里米斯工匠
H	Strange Sorting 2	奇怪的排序 2
I	Not a Work of Idol 2	不是偶像的大作 2
J	Slay the Spire 2	杀戮尖塔 2
K	Sequence Operation	序列操作
L	Spirited Away	神隐
M	Rectangle Reversal 3	矩形反转 3

Problem A. White Night

Little Cyan Fish has a matrix A with n rows and m columns in his hand. Each element in the matrix can be Cyan or White. We use the character `C` to represent Cyan, and the character `W` to represent White. For convenience, Little Cyan Fish denotes the element in the i-th row and j-th column ($1 \le i \le n$, $1 \le j \le m$) of the matrix as $A_{i,j}$.

Little Cyan Fish can perform the following operation any number of times:

- Choose a pair of vertically adjacent or horizontally adjacent cells $A_{i,j}$ and $A_{k,l}$. That is, $|i-k|+|j-l|=1$.
- Swap $A_{i,j}$ and $A_{k,l}$.

Little Cyan Fish wants to transform matrix A into another given matrix B. Of course, Little Cyan Fish guarantees that the number of Cyan elements in matrix A is equal to the number of Cyan elements in the final required matrix B, so there must exist an operation scheme that satisfies Little Cyan Fish's requirement.

You need to help Little Cyan Fish calculate the minimum number of operations required to complete his requirement.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \le T$), indicating the number of test cases.

For each test case, the first line of the input contains two integers n and m ($1 \le n \le 10^5$, $1 \le m \le 6$), representing the number of rows and columns of the matrix.

The next n lines, each containing a string of length m (only containing characters `C` or `W`), representing each row of matrix A.

The next n lines, each containing a string of length m (only containing characters `C` or `W`), representing each row of matrix B. It is guaranteed that the number of character `C`s in matrix A and matrix B are the same (naturally, the number of character `W`s will also be the same).

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, output a single line with an integer, representing the minimum number of operations required for Little Cyan Fish to transform matrix A into matrix B.

Example

standard input	standard output
2 2 2 CW WC WC CW 5 3 WWC WCW CWC CCC CCC CCC CCC CCC CWW WWW	2 16

Problem B. Yearning for Yonder 2

This is an interactive problem.

> Thanks for your hard work, darling. Little Cyan Fish doesn't know how much of his code words can be conveyed, but still thanks you for walking to this point with Little Cyan Fish.
> So many things, so many absurd things happened, Little Cyan Fish is tired, very tired... In the past year, Little Cyan Fish felt several times that he was going to save himself, but after all, he was not strong enough to cut the strings on his body. Little Cyan Fish always imagines that in that faraway place, there is another self who can clearly see the map of his life.
>
> — "Longing for the Distance" · The 10th China Collegiate Programming Contest Final

Well, maybe, you can try, how difficult this will be. Little Cyan Fish feels that his life is a permutation of unknown shape. People always say the world is full of randomness, but why is Little Cyan Fish always so unlucky...? Aha! Maybe randomness is just this profound! So, Little Cyan Fish decides that this permutation of his will be randomly generated in the following way:

- Generate a random permutation $p_1, p_2, \ldots, p_n$: chosen uniformly at random from all $n!$ possible permutations of 1 to n.
- If $p_1 > p_n$, then reverse the entire permutation.

You cannot directly observe the structure of the permutation, but Little Cyan Fish has granted you a superpower: asking! Each time, you can interactively query the value of the distance between two numbers on the permutation modulo 3. Specifically, each time you can choose two numbers u, v ($1 \le u, v \le n$, $u \neq v$), the permutation will tell you the value of the distance between these two numbers modulo 3 (i.e., finding two indices i, j satisfying $p_i = u, p_j = v$, the permutation will answer $|i - j| \bmod 3$).

Now, Little Cyan Fish wants you to try and see what you can get. You need to help Little Cyan Fish determine this permutation in no more than $25n$ queries.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \le T \le 1\,000$), indicating the number of test cases.

For each test case, first you need to read an integer n ($3 \le n \le 10^4$).

Interaction Protocol

Next, the interaction begins. You can make no more than $25n$ queries in each test case. To make a query, you need to output a single line "`?` u v" ($1 \le u, v \le n$, $u \neq v$), describing a query. Then, you need to read the result from standard input.

To provide your answer, you need to output "`!` $p_1\, p_2 \cdots p_n$". You output must satisfy $p_1 < p_n$. Outputting the answer will not be counted towards the $25n$ queries limit. After you output the answer, you need to read the next test case immediately, or exit your program immediately.

After outputting a query, **do not** forget to output a newline character and flush the output stream. To do this, you can use `fflush(stdout)` or `cout.flush()` in C++, `System.out.flush()` in Java, `flush(output)` in Pascal, and `stdout.flush()` in Python.

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

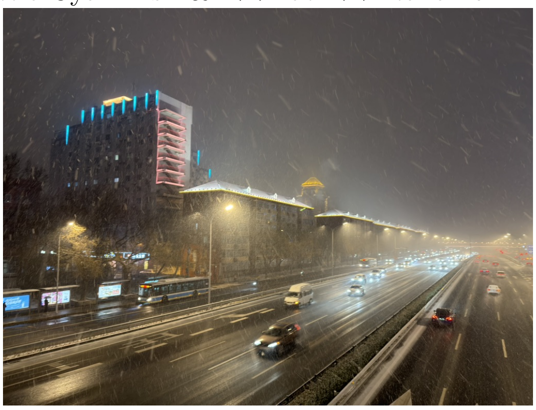
In this problem, it is guaranteed that the interactor is **non-adaptive**. That is, the permutations are determined **randomly as the problem stated** before the interaction process. They will not change according to your queries. There are a total of 30 test cases (including the sample test case).

Example

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

Note

Figure 1: Little Cyan Fish & <?> at <?> taken on March 4, 2026



Problem C. Four Kubic Theorem 2

Little Cyan Fish is a student studying at the Powerful Kubic University (PKU). In 2023, Little Cyan Fish took the course Introduction to the Kubic Theory taught by Prof. Kubic. After proving the Four Kubic Theorem, Little Cyan Fish became a teaching assistant for this course. In the final exam of this course, Little Cyan Fish prepared the following interesting little problem:

- Given a prime p and four integers a_1, a_2, a_3, a_4 between 1 and $p-1$.
- Solve the equation $a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 \equiv m \pmod{p}$, where $x_i \geq 0$.

This problem is too simple for you, who have taken Introduction to the Kubic Theory. Therefore, Little Cyan Fish gives you another four integers b_1, b_2, b_3, b_4. Little Cyan Fish wants your solution to minimize the value of $b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4$ while satisfying the above equation.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \leq T \leq 10^4$), indicating the number of test cases.

For each test case, the first line of the input contains two integers p and m ($2 \leq p \leq 1.01 \times 10^9$, $0 \leq m < p$, guaranteed that p is a prime).

The next line contains four integers a_1, a_2, a_3, a_4 ($1 \leq a_1, a_2, a_3, a_4 < p$).

The next line contains four integers b_1, b_2, b_3, b_4 ($1 \leq b_1, b_2, b_3, b_4 \leq 10^9$).

It is guaranteed that the sum of $\lceil \sqrt{p} \rceil$ over all test cases does not exceed 2^{17}.

Output

For each test case, output a single line containing one integer, indicating the minimum value of $b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4$.

Example

standard input	standard output
3 101 99 1 2 3 4 5 6 7 8 998244353 114514 1919 811 123 777 1000000000 1000000000 1000000000 1000000000 1000000007 767336601 142205992 920557330 725753607 763861942 1 1 1 1	199 76000000000 187

Problem D. Call You With Your Name

Little Cyan Fish likes string theory very much. Today, Little Cyan Fish invites you to study human nicknames with him.

In Little Cyan Fish's universe, human nicknames can all be represented as a string containing only lowercase Latin letters (`a` to `z`). For example, "`qingyu`", "`xiuga`" are human nicknames, but "`Abacde`" is not a human nickname.

As a human, your current nickname is the string s_0. Little Cyan Fish offers humans n kinds of renaming opportunities, the i-th renaming opportunity can be represented by a positive integer ℓ_i. You can execute these renaming operations in any order, but each renaming operation must be executed exactly once.

If you choose to use the i-th type of renaming operation, then your name s will become the first ℓ_i characters of s^ω, where s^ω denotes the string obtained by copying the string s infinitely many times. For example, if your current name is $s =$ `ad`, then after executing the renaming operation with $\ell = 5$, your name will become $s' =$ `adada`

Little Cyan Fish wants human nicknames to become as lexicographically large as possible. Now, Little Cyan Fish gives s_0 and $\ell_1, \ell_2, \ldots, \ell_n$. You need to find some renaming order $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$, p_i is a permutation of length n), such that after renaming in the order of $\ell_{p_1}, \ell_{p_2}, \ldots, \ell_{p_n}$, the final string obtained has the largest possible lexicographical order.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \le T$), indicating the number of test cases.

For each test case, the first line of the input contains a string s_0 ($1 \le |s_0| \le 10^5$, s_0 only contains lowercase Latin letters), indicating your initial nickname.

The next line contains an integer n ($1 \le n \le 10^5$), indicating the number of renaming opportunities.

The next line contains n integers $\ell_1, \ell_2, \ldots, \ell_n$ ($1 \le \ell_i \le 10^9$), indicating the parameters of each renaming opportunity.

It is guaranteed that the sum of $|s_0|$ over all test cases does not exceed 10^5, and the sum of n over all test cases does not exceed 10^5.

Output

For each test case, output a single line containing n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$, p_i is a permutation of length n), indicating your renaming order.

Example

standard input	standard output
3 qingyu 2 7 3 abacde 2 8 11 abcabadefdghiajkd 6 11 4 5 14 1919 810	2 1 1 2 3 6 2 1 4 5

Problem E. Call You With Your Name 2

Little Cyan Fish likes string theory very much. Today, Little Cyan Fish invites you to study human nicknames with him.

In Little Cyan Fish's universe, human nicknames can all be represented as a string containing only lowercase Latin letters (`a` to `z`). For example, "`qingyu`", "`xiuga`" are human nicknames, but "`Abacde`" is not a human nickname.

Little Cyan Fish considers a name s to be Lyndon if and only if for each proper suffix[1] t of s, the lexicographical order of s is strictly smaller than t. For example, "`abacde`" is a Lyndon string, but "`qingyu`" is not a Lyndon string (because the original string is not lexicographically smaller than the proper suffix `ingyu`).

Now, Little Cyan Fish gives you a human nickname s, you need to calculate how many different $1 \le l \le r \le |s|$ exist such that $s[l \dots r]$ is Lyndon.

Input

There are multiple test cases. The first line of the input contains a single integer T $(1 \le T)$, indicating the number of test cases.

For each test case, the input contains a single line with a string s $(1 \le |s| \le 2 \times 10^5)$, indicating the human nickname.

It is guaranteed that the sum of $|s|$ over all test cases does not exceed 2×10^5.

Output

For each test case, output a single line containing one integer, indicating the answer.

Example

standard input	standard output
6 aaa qingyu littlecyanfish abacde abcdefghbidajkad abcdeabdfag	3 9 27 16 58 32

[1]A proper suffix is a non-empty suffix that is not equal to the original string. For example, "`qqoj`" has 3 proper suffixes, which are "`j`", "`oj`", and "`qoj`".

Problem F. Many Many Heads 2

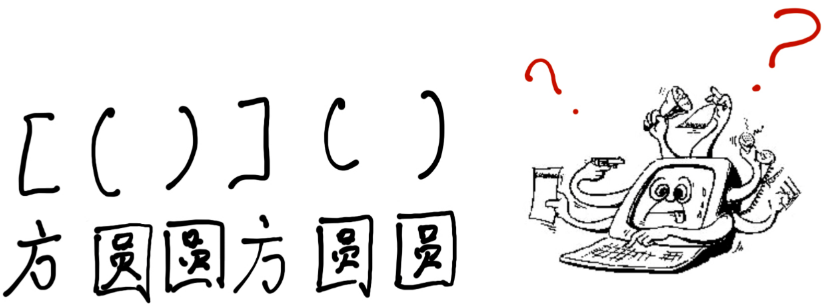
Multi-Heads Cup, or MHC for short, is a worldwide programming contest made for the participants with **many many heads**. In 2023, the chief judge of this event, Little Cyan Fish, solved a tricky authentication problem using many types of brackets.

That was the first time Little Cyan Fish saw contestants with many many heads. And this time, Little Cyan Fish brings another bracket problem to his best friend who has many many heads. Little Cyan Fish has n types of brackets in his hand, and each type of bracket is divided into left brackets and right brackets. For convenience, we use $\texttt{L}^i$ to denote the i-th type of left bracket, and $\texttt{R}^i$ to denote the i-th type of right bracket.

"Hey, don't forget," Little Cyan Fish thought, "I have introduced to you what a balanced bracket sequence is before!" To ensure you understand the concept of a balanced bracket sequence, Little Cyan Fish prepared the following formal definition of a balanced bracket sequence:

- ε (an empty string) is a balanced bracket sequence.
- If A is a balanced bracket sequence, (A) is also a balanced bracket sequence.
- If A and B are balanced bracket sequences, then AB is also a balanced bracket sequence.

For example, "`()`", "`()()`" and "`(())`" are balanced bracket sequences, but "`)(`", "`(`" and "`))`" are not.



Now, Little Cyan Fish gives you a bracket sequence S of length n, containing n types of brackets. Unfortunately, Little Cyan Fish forgot the types of brackets at some positions, and also forgot the directions of brackets at these positions. The memory of these positions has become blurred and is represented by `?` by Little Cyan Fish.

Little Cyan Fish is very curious, for all $1 \le l \le r \le n$, how many pairs of (l, r) correspond to substrings $S[l \dots r]$, such that there exists a way to fill `?` with some type of bracket in some direction, such that for each $1 \le i \le n$, we have:

- Extract all brackets of type i (i.e. all $\texttt{L}^i$ and $\texttt{R}^i$), the resulting bracket sequence (containing only the i-th type of bracket) is a balanced bracket sequence.

For example, if we use "()" to denote the first type of bracket and "[]" to denote the second type of bracket, the bracket string `([?)` satisfies the above condition, because we can replace `?` with `]`. `(?)]` also satisfies the above condition, because we can replace `?` with `[`.

Little Cyan Fish wants you to calculate the number of all valid pairs (l, r).

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \le T$), indicating the number of test cases.

For each test case, the first line of the input contains an integer n ($1 \le n \le 2 \times 10^5$), indicating the length of the bracket string.

The next line contains n integers $x_1, x_2, \ldots, x_n$ ($-n \le x_i \le n$). These n integers describe the information of the bracket string, where:

- If $x_i > 0$, then the i-th position represents the x_i-th type of left bracket (i.e. $\texttt{L}^{x_i}$).
- If $x_i < 0$, then the i-th position represents the $-x_i$-th type of right bracket (i.e. $\texttt{R}^{-x_i}$).
- If $x_i = 0$, it means Little Cyan Fish forgot the information at this position (i.e. `?`).

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

Output

For each test case, output a single line containing one integer, indicating the answer.

Example

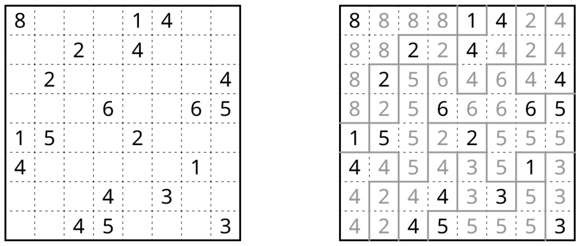
standard input	standard output
4 4 1 2 -1 -2 4 1 0 -2 0 6 1 2 3 -3 -2 -1 8 1 0 0 3 0 0 0 -2	1 3 3 14

Problem G. Puzzle: The Artisan of Glimmith

After practicing Fillomino skills, Little Cyan Fish decides to train more on the game "The Artisan of Glimmith".

Little Cyan Fish first introduces the rules of Fillomino to you:

> For a rectangle of m rows and n columns, divide the grid into several regions (the cells of each region must be orthogonally connected), such that any two adjacent regions have different areas.
> Some cells in the grid already have numbers filled in. The number indicates the area of the region it belongs to.



An example of a Fillomino puzzle. Source: Puzzle GP 2022 R4

Little Cyan Fish now gives you a Fillomino board, and he wants to construct any valid solution for this Fillomino puzzle.

Of course, this problem is too difficult for Little Cyan Fish, so you only need to solve the case where $m = 1$. In other words, given a row of n cells, you want to divide these cells into several segments (each segment completely contains several consecutive cells), requiring that:

- Any two adjacent segments have different lengths;
- Some cells contain some numbers, and these numbers indicate the number of cells in the segment to which this cell belongs.

Little Cyan Fish asks you to determine whether there exists a valid solution for the board. If it exists, you also need to output any valid solution.

Input

There are multiple test cases. The first line of the input contains a single integer T $(1 \le T)$, indicating the number of test cases.

For each test case, the first line of the input contains an integer n $(1 \le n \le 10^6)$.

The next line contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le n)$. If $a_i = 0$, it means no number is filled in the i-th cell. Otherwise, it means a_i is filled in the i-th cell.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, if there does not exist any valid scheme to fill the numbers, then output a single line "`No`".

Otherwise, the first line of the output contains the string "`Yes`". The next line outputs n integers, representing the length of the segment where each cell is located in the scheme you constructed. If there are multiple schemes that meet the requirements, you can output any of them.

Example

standard input	standard output
7 3 0 0 0 4 2 0 0 0 4 0 2 0 0 6 0 3 3 0 3 0 10 2 0 0 3 0 0 4 0 0 1 6 0 0 1 1 0 0 6 1 2 3 4 5 6	Yes 2 2 1 No Yes 1 2 2 1 No Yes 2 2 3 3 3 4 4 4 4 1 No No

Problem H. Strange Sorting 2

Little Cyan Fish and Little Kevinfish are playing a game of sorting sequences. Little Kevinfish has a tree T with n vertices, where the vertices are numbered with integers from 1 to n.

For a sequence A consisting of integers from 1 to n, Little Kevinfish defines a swap operation as:

- Choose indices i, j, such that the vertices numbered A_i and A_j are directly connected by an edge in T.
- Swap the positions of A_i and A_j.

Little Kevinfish asks Little Cyan Fish the following question:

- For a given constant m, for each $\ell = 1, 2, \ldots, m$, solve the following problem:
  - Consider a sequence A of length ℓ consisting of integers from 1 to n (there are n^ℓ such sequences in total), how many sequences A can be transformed into a sequence that is monotonically non-decreasing through some number of the above swap operations.

Please help Little Cyan Fish answer Little Kevinfish's question. Since the answer may be very large, you only need to output the answer modulo $10^9 + 7$.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \le T$), indicating the number of test cases.

For each test case, the first line of the input contains two integers n and m ($1 \le n \le 200$, $1 \le m \le 10^5$).

The next $(n-1)$ lines each contain two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), indicating an edge connecting vertex u_i and v_i. It is guaranteed that these $(n-1)$ edges form a valid tree.

It is guaranteed that the sum of n over all test cases does not exceed 200, and the sum of m over all test cases does not exceed 10^5.

Output

For each test case, output a single line containing m integers. The i-th ($1 \le i \le m$) integer represents the answer when $\ell = i$, modulo $10^9 + 7$.

Example

standard input	standard output
3 3 4 1 2 2 3 4 2 1 2 1 3 3 4 1 10	3 8 23 70 4 13 1 1 1 1 1 1 1 1 1 1

Problem I. Not a Work of Idol 2

Little Cyan Fish has a tree $G = (V, E)$ with n vertices. The vertices of the tree are numbered with positive integers from 1 to n. The i-th ($1 \le i \le n-1$) edge connects vertex u_i and v_i.

Little Cyan Fish wants you to assign a positive integer p_i from 1 to n to each vertex i, satisfying the following requirements:

- For each $1 \le i < j \le n$, $p_i \ne p_j$. In other words, $p_{1\dots n}$ forms a permutation of length n.
- For each edge $(u, v) \in E$ on the tree, we have $p_u + p_v \le n + 1$.

Little Cyan Fish wants you to calculate how many permutations p satisfy this condition. As usual, since this problem is *Not a Work of Idol*, Little Cyan Fish does not want you to output the answer modulo a large prime. Therefore, please output the answer modulo **4**.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \le T$), indicating the number of test cases.

For each test case, the first line of the input contains an integer n ($1 \le n \le 10^6$), indicating the number of vertices in the tree.

The next $(n-1)$ lines each contain two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), indicating an edge connecting vertex u_i and v_i. It is guaranteed that these $(n-1)$ edges form a valid tree.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output a single line containing one integer, indicating the answer modulo **4**.

Example

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

Problem J. Slay the Spire 2

Little Cyan Fish is playing *Slay the Spire II* and chooses the character Silent.

Little Cyan Fish now needs to fight a Battle Friend. Initially, the Poison level of the Battle Friend and your Accelerant level are both 0. For the next n turns, the following events will occur in sequence each turn:

1. Play Deadly Poison: Increase the Battle Friend's Poison level by x_i.
2. Choose whether to play Accelerant: Choose whether to increase your Accelerant level by 1.
3. Trigger Poison effect: Let your Accelerant level be t, repeat the following event t times.
   - Let the Battle Friend's Poison level be x. If $x > 0$, deal x damage to the Battle Friend and decrease the Poison level by 1; otherwise, if $x = 0$, nothing happens.

Note that both Poison and Accelerant levels are retained across turns and do not disappear.

Little Cyan Fish wants to maximize the total damage dealt to the Battle Friend over the n turns, output the answer.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \le T$), indicating the number of test cases.

For each test case, the first line of the input contains an integer n ($1 \le n \le 5\,000$).

The next line contains n integers $x_1, x_2, \ldots, x_n$ ($0 \le x_i \le 10^7$), indicating the Poison levels increased for the Battle Friend in each turn.

It is guaranteed that the sum of n over all test cases does not exceed 10^6, and the sum of n^2 over all test cases does not exceed 10^8.

Output

For each test case, output a single line containing one integer, indicating the maximum total damage dealt to the Battle Friend over all turns.

Example

standard input	standard output
4 5 1 1 0 3 7 3 0 3 2 9 9 9 8 2 4 4 3 5 3 13 1 1 4 5 1 4 1 9 1 9 8 1 0	33 11 750 725

Problem K. Sequence Operation

Little Cyan Fish has a non-negative integer sequence $a_1, a_2, \ldots, a_n$ of length n in his hand. Little Cyan Fish can perform the following operation any number of times:

- Choose an integer $1 \le i \le n$.
- Change a_i to $\operatorname{mex}_{1 \le k < i} a_k$.

Here, for a set of non-negative integers S, the notation $\operatorname{mex} S$ is defined as the smallest non-negative integer that does not appear in the set S. For example, $\operatorname{mex}\{0, 1, 2, 4, 6\} = 3$, $\operatorname{mex}\{1, 2, 3\} = 0$, $\operatorname{mex} \varnothing = 0$.

Little Cyan Fish wants to make the value of a_n as large as possible through some number of operations (possibly 0). Little Cyan Fish wants you to calculate:

1. What is the maximum possible value of a_n after any number of operations;
2. On the basis of satisfying 1, what is the minimum number of operations required to achieve this maximum value.

Input

There are multiple test cases. The first line of the input contains a single integer T $(1 \le T)$, indicating the number of test cases.

For each test case, the first line of the input contains an integer n $(1 \le n \le 10^6)$, representing the length of the sequence.

The next line contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output a single line with two integers. The first integer represents the maximum value of a_n, and the second integer represents the minimum number of operations required.

Example

standard input	standard output
4 2 0 114514 3 0 1 0 4 3 1 2 0 6 6 5 4 3 2 1	114514 0 2 1 3 2 5 3

Problem L. Spirited Away

You know a freak in Forgotten Hill. Today, the freak wants you to solve the following problem:

- Given a 32-dimensional vector space $\mathbb{F}_2^{32}$ over the finite field $\mathbb{F}_2$ and n elements $a_1, a_2, \ldots, a_n$ in $\mathbb{F}_2^{32}$. Find $\left|\left\{\text{span}_{\mathbb{F}_2}\{a_l, a_{l+1}, \ldots, a_r\} \mid 1 \le l \le r \le n\right\}\right|$.

The freak once told Little Cyan Fish how to solve his problem. However, the related memories have been blurred by the freak. Therefore, Little Cyan Fish hopes you can solve the freak's problem. If you are not familiar with the related notations, Little Cyan Fish describes the above problem in natural language as:

- Given a sequence of non-negative integers $a_1, a_2 \ldots a_n$ of length n $(0 \le a_i < 2^{32})$.
- Define the value of $B(l, r)$ as: choose any number of numbers (can be 0) from $a_l, a_{l+1} \ldots a_r$, and XOR them all together, the set of all possible XOR values obtained. In other words:

$$B(l, r) = \left\{ \bigoplus_{i=l}^{r} c_i a_i \;\middle|\; c_i \in \{0, 1\} \right\}$$

- where $\oplus$ denotes the bitwise XOR operation. For example, $4 \oplus 6 = 2$, $1 \oplus 2 = 3$.
- Find how many essentially different sets $B(l, r)$ there are for all $1 \le l \le r \le n$.

Input

There are multiple test cases. The first line of the input contains a single integer T $(1 \le T)$, indicating the number of test cases.

For each test case, the first line of the input contains an integer n $(1 \le n \le 10^5)$.

The next line contains n integers a_i $(0 \le a_i < 2^{32})$.

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, output a single line containing one integer, indicating the number of essentially different $B(l, r)$ sets, i.e., $\left|\left\{\text{span}_{\mathbb{F}_2}\{a_l, a_{l+1}, \ldots, a_r\} \mid 1 \le l \le r \le n\right\}\right|$.

Example

standard input	standard output
5 3 1 2 3 4 10 12 17 33 6 1 0 9 8 7 3 8 9012 91829 9819 78 237 862 7672 2 10 0 1 2 4 8 16 32 64 128 256	4 10 12 36 46

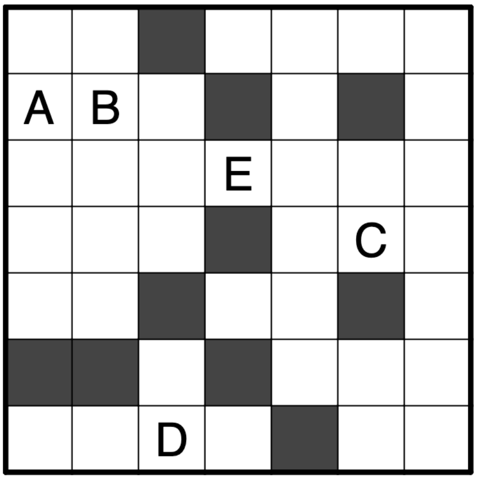
Note

For the first sample test case, valid sets include: $\{0, 1\}$, $\{0, 2\}$, $\{0, 3\}$, $\{0, 1, 2, 3\}$.

Problem M. Rectangle Flip 3

Little Cyan Fish has a rectangle with n rows and m columns, where the rows and columns are both indexed starting from 0. The cell at the i-th row and j-th column can be represented as (i, j) $(0 \le i \le n-1, 0 \le j \le m-1)$. Initially, all cells are present (not deleted).

Little Cyan Fish calls two cells (x_a, y_a) and (x_b, y_b) adjacent if and only if $|x_a - x_b| + |y_a - y_b| = 1$. In other words, these two cells share a common boundary (also known as 4-adjacent). And for two cells $p = (x_a, y_a)$ and $q = (x_b, y_b)$, if they can be connected together through a sequence of adjacent undeleted cells, then cell p and q are said to be connected (4-connected).



For example, consider the above situation (black cells represent deleted cells, and white cells represent undeleted cells), cell A and cell B are adjacent. Cell A is connected to cell B and cell C, but cell A and cell D are not connected.

Now, Little Cyan Fish is going to perform some operations in his dream. In each operation, Little Cyan Fish will choose a cell (x, y) and delete the cell (x, y).

Little Cyan Fish is very curious about how many remaining undeleted cells in the current rectangle are cut vertices after each deletion. Specifically:

- For a certain cell $r = (x_r, y_r)$, if there exist two cells $p = (x_p, y_p)$ and $q = (x_q, y_q)$, such that p and q are connected in the grid, but are not connected after deleting cell r, then r is called a cut vertex.

For example, in the figure above, cell E is a cut vertex. Because for cell A and cell C, they are connected before cell E is deleted, but are no longer connected after cell E is deleted.

However, since Little Cyan Fish's dream is very magical, you cannot predict his subsequent operations. Therefore, you need to answer all these operations online. Specifically, the given (x, y) in the i-th $(2 \le i \le q)$ operation will depend on the returned result of the $(i-1)$-th operation. Please help Little Cyan Fish answer all queries under this restriction.

Input

The first line of the input contains three integers n, m, and q $(2 \le n, m \le 500, 1 \le q \le n \cdot m)$.

The next q lines each contain two integers x'_i, y'_i $(0 \le x'_i < n, 0 \le y'_i < m)$, indicating the encrypted coordinates of the deleted cell (x_i, y_i). If the answer to the previous query was preans, then the true $x_i = (x'_i + \text{preans}) \bmod n$, $y_i = (y'_i + \text{preans}) \bmod m$. Initially, we consider $\text{preans} = 0$.

It is guaranteed that the decrypted (x_i, y_i) are pairwise distinct. That is, the same cell will not be deleted multiple times.

Output

For each operation, output a single line containing one integer, indicating the number of cut vertices after this deletion operation.

Example

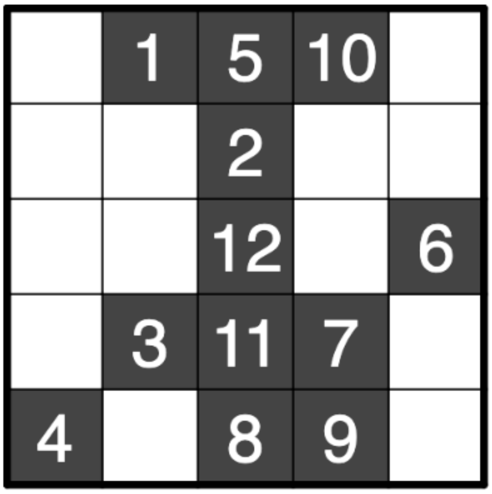
standard input	standard output
5 5 12 0 1 0 1 1 4 2 3 4 1 2 4 1 1 4 2 2 1 4 2 1 0 0 0	1 2 2 6 5 7 10 7 6 7 7 4

Note

The decrypted data:

5 5 12
0 1
1 2
3 1
4 0
0 2
2 4
3 3
4 2
4 3
0 3
3 2
2 2

Below is the order in which each cell is painted black.


1
5
10
2
12
6
3
11
7
4
8
9