Problem A. Alive Fossils

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

Nowcoder Multi-University Training Camp is now holding again. As a contestant, you are a little bit curious about the problem setters. Now you have a list of the problem setters among all the n Nowcoder Multi-University Training Camps, and you are curious about the alive fossil setters, who set problems in all the n camps. Please output the alive fossil setters in the lexicographic order.

Input

The first line contains one integer n ($1 \le n \le 10^5$), denoting the number of Nowcoder Multi-University Training Camps ever held.

For each training camp:

The first line contains one integer c ($1 \le c \le 10^5$), denoting the number of problem setters.

Each of the following c lines contains a string containing lowercase letters and digits only, denoting a problem setter in the camp. No problem setters appear more than once in a training camp.

It is guaranteed the total length of strings in all training camps does not exceed 10^6.

Output

Output a single integer f in the first line, and then output all the f alive fossil setters in the lexicographic order in the following f lines.

Example

standard input	standard output
2 14 buaa bit sunsiyu uestc ezec qcjjfanclub nfls oscar114514 jianglyfan gromah calabash lzr010506 gispzjz roundgod 13 uestc bit buaa gispzjz roundgod ynoi hdu oscar114514 gromah calabash quailty nfls sjtu	9 bit buaa calabash gispzjz gromah nfls oscar114514 roundgod uestc

Problem B. Bloodline Counter

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

A tournament is a directed graph obtained by assigning a direction for each edge in an undirected complete graph.

We say a tournament is grand, if and only if the largest directed cycle in the tournament contains exactly k vertices.

Please count the number of different labeled grand tournaments of n vertices modulo $998\,244\,353$.

Input

The first line contains two integers n and k ($3 \le k \le n \le 5 \times 10^5$), as explained in the problem statement.

Output

Output an integer representing the answer modulo $998\,244\,353$.

Example

standard input	standard output
3 3	2

Problem C. Clamped Sequence II

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

For an integer sequence $a_1, a_2, \ldots, a_n$ and a positive integer d, the d-clamped value of the sequence is the maximum value of $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} |a_i - a_j|$, where $|x|$ is the absolute value of x, after appointing a range $[l, r]$ satisfying $0 \le r - l \le d$ and clamping the sequence to the range $[l, r]$.

More specifically, clamping the sequence to the range $[l, r]$ makes each element

$$a_i := \begin{cases} l, & a_i < l; \\ a_i, & l \le a_i \le r; \\ r, & a_i > r. \end{cases}$$

Both l and r are arbitrary real numbers decided by you under the given constraints. It can be shown that the maximum sum is always an integer.

Now, given an integer sequence $a_1, a_2, \ldots, a_n$ and q queries, where each query is in one of the two following formats:

- `1 x d` denotes setting a_x to d
- `2 d` denotes reporting the d-clamped value of the current sequence

Please output the answer for each reporting queries.

Input

The first line contains two integers n ($2 \le n \le 10^5$) and q ($1 \le q \le 10^5$), denoting the length of the given sequence and the number of queries.

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

Each of the following q lines contains a query, which is in one of the two following formats:

- `1 x d` ($1 \le x \le n$, $1 \le d \le 10^6$) denotes setting a_x to d
- `2 d` ($1 \le d \le 10^6$) denotes reporting the d-clamped value of current sequence

Output

For each reporting query, output one line containing a single integer, denoting the d-clamped value of the current sequence.

Example

standard input	standard output
8 3 3 1 4 1 5 9 2 6 2 3 1 2 8 2 4	46 58

Problem D. Distance on Tree

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

You are given a rooted tree with n nodes labeled from 1 to n, the root of which is the 1-st node, and the i-th node has a given value a_i for each $i = 1, 2, \ldots, n$.

Given a parameter m and then q queries, the i-th of which is represented by two numbers x_i and k_i, your task is to maximize $dis(u, v) + dis(v, w) + dis(w, u)$ by choosing three distinct nodes u, v and w in the subtree of x_i such that $a_u + a_v + a_w \equiv k_i \pmod{m}$. Here $dis(u, v)$ is the length of the shortest path between the u-th and the v-th nodes on the tree. If there is no solution for a query, the answer is 0.

Input

The first line contains three integers n, m ($1 \le n, m \le 2\,000$) and q ($1 \le q \le 2 \times 10^5$).

The second line contain n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < m$).

Following $n - 1$ lines describe the tree, each of which contains three integers u, v ($1 \le u, v \le n$) and w ($1 \le w \le 2 \times 10^5$), representing an edge with length w between the u-th and the v-th nodes.

In the following q lines, the i-th line contains two integers x_i ($1 \le x_i \le n$) and k_i ($0 \le k_i < m$), describing the i-th query.

Output

Output q lines, the i-th of which contains an integer, indicating the answer to the i-th query.

Example

standard input	standard output
5 5 10 0 1 2 3 4 1 2 1 2 3 2 2 4 3 1 5 4 1 0 1 1 1 2 1 3 1 4 2 0 2 1 2 2 2 3 2 4	12 14 16 16 20 0 10 0 0 0

Problem E. Educational Problem I

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

You are given n, m, l, r.

Count the number of integer sequence $a_1, a_2, \ldots, a_n$ satisfying the following conditions, modulo 998 244 353:

- $\forall i \in \mathbb{Z} \cap [1, n]$, $l \le a_i \le r$.
- $S_1(\sum_{i=1}^{n} a_i) \equiv S_2(\sum_{i=1}^{n} a_i) \pmod m$. Here $S_1(x)$ denotes the sum of the digits of x in decimal notation, and $S_2(x)$ denotes the sum of the square of digits of x in decimal notation. For example, $S_1(123) = 1 + 2 + 3 = 6$, $S_2(123) = 1^2 + 2^2 + 3^2 = 14$.

Input

The only line contains four integers n, m, l, r ($1 \le n, m \le 20$, $0 \le l \le r < 10^{1000}$).

Output

Output an integer representing the answer.

Examples

standard input	standard output
3 5 2 5	26
10 20 24538 82350244413529949	358849619

Problem F. Educational Problem II

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

You are given n, m, l, r and a matrix $B = (b_{i,j})_{n\times m}$.

Count the number of integer sequence $a_1, a_2, \ldots, a_n$ satisfying the following conditions, modulo $998\,244\,353$:

- $\forall i \in \mathbb{Z} \cap [1, n]$, $l \le a_i \le r$.
- $\forall i \in \mathbb{Z} \cap [1, n]$, $\exists S \subseteq \mathbb{Z} \cap [1, m]$, $a_i = \bigoplus_{j\in S} b_{i,j}$. That is, a_i is equal to the bitwise XOR sum of some subsequence of $(b_{i,1}, b_{i,2}, \ldots, b_{i,m})$, and $a_i = 0$ if the subsequence is empty.
- The number of $i \in \mathbb{Z} \cap [2, n]$ that $a_{i-1} \ne a_i$ is minimized among all the sequences satisfying the above two conditions.

Input

The first line contains four integers n, m, l, r ($2 \le n \le 10^5$, $2 \le nm^2 \le 4 \times 10^7$, $0 \le l \le r < 2^m$).

The i-th of the following n lines contains m integers $b_{i,1}, b_{i,2}, \ldots, b_{i,m}$ ($0 \le b_{i,j} < 2^m$).

Please note that $l, r, b_{i,j}$ are given in binary notation length of m.

Output

Output an integer representing the answer.

Examples

standard input	standard output
3 2 01 10 01 00 01 10 00 10	2
3 2 11 11 01 00 01 10 00 10	0
5 3 100 110 111 011 111 100 100 001 111 111 010 001 101 010 110 010 010	8

Problem G. Expected Distance

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

Given two segments that are perpendicular to each other on a two-dimension Cartesian plane, you need to pick a point uniformly at random on each of them and calculate the expected Euclidean distance between the two picked points.

Input

The input contains multiple 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 case, each of the two lines contains four integers x_1, y_1, x_2 and y_2 ($-1\,000 \le x_1, y_1, x_2, y_2 \le 1\,000$), denoting a segment that connects (x_1, y_1) and (x_2, y_2). It is guaranteed that the two endpoints of each segment do not coincide, and that the two segments are perpendicular to each other.

Output

For each case, output a single real number, indicating the expected distance between the two randomly picked points.

Your answer is acceptable if its absolute or relative error does not exceed 10^{-9}. Formally speaking, suppose that your output is a and the jury's answer is b, your output is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$.

Example

standard input	standard output
1 0 0 1 0 0 0 0 1	0.765195716464212691

Note

In the sample case, the expected distance is $\int_0^1 \int_0^1 \sqrt{x^2+y^2}\,\mathrm{d}x\,\mathrm{d}y = \frac{\sqrt{2}+\ln(1+\sqrt{2})}{3} \approx 0.765195716464212691$.

Problem H. Insert 1, Insert 2, Insert 3, ...

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

A sequence $a_1, a_2, \ldots, a_n$ consisting of positive integers is said to be *generatable* when one can obtain a by repeating the following operation on an empty sequence:

- Choose a positive integer k, and insert 1, insert 2, ..., insert k respectively into some position in the sequence while preserving their relative order. More specifically, let i_x be the index of newly inserted x in the sequence after the operation, then $i_1 < i_2 < \cdots < i_k$.

For instance, $a = (1, 1, 2, 2, 1, 1, 3, 4, 2, 3)$ is generatable. Here is one way to generate it:

$() \to (\mathbf{1}, \mathbf{2}) \to (1, 2, \mathbf{1}, \mathbf{2}, \mathbf{3}) \to (1, \mathbf{1}, 2, \mathbf{2}, 1, \mathbf{3}, \mathbf{4}, 2, 3) \to (1, 1, 2, 2, \mathbf{1}, 1, 3, 4, 2, 3)$.

You are given a sequence $A_1, A_2, \ldots, A_N$ consisting of positive integers. Find the number of pairs of integers (L, R) such that:

- $1 \le L \le R \le N$ and the contiguous subsequence $A_L, \ldots, A_R$ is generatable.

Input

The first line contains an integer N $(1 \le N \le 10^6)$.

The second line contains N integers $A_1, A_2, \ldots, A_N$ $(1 \le A_i \le N)$.

Output

Output the number of pairs of integers (L, R) satisfying the conditions.

Examples

standard input	standard output
6 1 1 2 2 3 3	8
6 1 2 1 2 1 3	11

Note

In the first sample, the 8 pairs are: $(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 2), (2, 3)$.

Problem I. Make It Square

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

Luluu is a “Square” magician from Eorzea. “Square” magicians are experts to use “Square” spells.

Every spell can be represented as a non-empty string containing only lower-case English characters. The “Square” spells are those of even length which the first half is identical to the second half. For example, `abcabc` and `aaaa` are “Square” spells, while `aaa` and `abcabd` are not.

Today, Luluu found a powerful “Square” spell from Grimoire, but unfortunately, the book page is damaged, so some parts of the spell are not readable anymore.

More specifically, the original “Square” spell is of the format $p + s + q + t$, where s and t are two constant strings, but p and q are the two unreadable parts of the spell. From some investigation, Luluu believes p and q should be of the same length that doesn't exceeds m.

Now, Luluu asks your help to calculate the number of all possible original “Square” spells. Could you help this poor magician?

Input

The first line contains a single integer m ($1 \le m \le 10^6$), the second line contains a string s and the third line contains a string t.

It is guaranteed that the length of s and t don't exceed 10^6.

Output

Output m integers in a single line, the k-th of which indicates the number of possible original “Square” spells when the length of p and the length of q are both k.

The answers could be large, so you should output them modulo 998 244 353.

Examples

standard input	standard output
3 abbab b	0 1 26
6 abbabbababbab bab	0 0 1 0 1 26
3 b acbac	0 1 26

Note

For the first sample case:

When the length of p and q are both 1, there is no valid solution to make $p + s + q + t$ a “Square” spell.

When the length of p and q are both 2, there is only one valid solution, which is ($p =$`ab`, $q =$`ab`). The “Square” spell is `ababbababb`.

When the length of p and q are both 3, there are 26 valid solutions, which are of format ($p =$ `ab?`, $q =$`?ab`), where `?` could be any single lower-case English character.

Problem J. Permutation and Primes

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

Given n, you should construct a permutation P of $\{1, 2, \ldots, n\}$ satisfying that for all $1 \le i < n$, either $P_i + P_{i+1}$ is an odd prime or $|P_i - P_{i+1}|$ is an odd prime.

If multiple solutions exist, print any of them. If no solution, print "-1" in one line.

Input

The first line contains one integer T $(1 \le T \le 10^5)$, denoting the number of test cases.

For each test case, input only one line containing one integer n $(2 \le n \le 10^5)$.

It is guaranteed that the sum of n among all test cases in one test file does not exceed 10^6.

Output

For each test case:

If solution exists, print one line containing n integers $P_1, P_2, \ldots, P_n$ $(1 \le P_i \le n, \forall 1 \le i < j \le n, P_i \ne P_j)$, denoting the permutation you construct.

If no solution, print "-1" in one line.

Example

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

Note

In the first test case, $1 + 2 = 3, 2 + 3 = 5$ are odd primes.

In the second test case, $|5 - 2| = 3, 2 + 1 = 3, |1 - 4| = 3, 4 + 3 = 7$ are odd primes.

Problem K. Scheming Furry

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

The Animals Corporation has assigned a new task with codename S to its staff!

The fox and the cat have received an unsorted matrix A with n rows and m columns. The elements of the matrix ($A_{i,j}$) are integers from 1 to $n \times m$ and are pairwise distinct. Their mission is to make the matrix sorted. Here we call matrix A *sorted*, if and only if its elements are non-decreasing in reading order (i.e., $A_{1,1} \le A_{1,2} \le \cdots \le A_{1,m} \le A_{2,1} \le A_{2,2} \le \cdots A_{2,m} \le \cdots \le A_{n,1} \le A_{n,2} \le \cdots \le A_{n,m}$).

To finish the task, the furry groupmates have a clear division of labor. Specifically, the fox can only perform the following operation:

- Choose integers i, j $(1 \le i < j \le n)$, and swap the i-th row and the j-th row of matrix A.

And the cat can only perform the following operation:

- Choose integers i, j $(1 \le i < j \le m)$, and swap the i-th column and the j-th column of matrix A.

The groupmates operate in turn (The fox goes first, the cat next, then the fox again, and so on). In their turn, one is **not allowed** to perform multiple operations or give up performing the operation.

It is a golden opportunity to develop team spirit. However, both the fox and the cat desire to win their boss's favor by becoming the first one who makes the matrix sorted after his own operation. And what's worse is that, if one knows he has no chance of meeting such desire, he will try to prevent his groupmate from being the first to sort the matrix, even if the matrix will remain unsorted forever!

You, a mysterious staff member, have seen through their charade. Assuming both the fox and the cat are clever enough, you wonder who will be the first to make the matrix sorted.

Input

The first line contains an integer T $(1 \le T \le 100)$, denoting the number of test cases.

For each test case:

The first line contains two integers n and m $(2 \le n, m \le 200)$.

Then n lines follow, the i-th of which contains m integers $A_{i,1}, A_{i,2}, \ldots, A_{i,m}$ $(1 \le A_{i,j} \le n \times m)$.

It is guaranteed that in each test case, matrix A is initially unsorted and its elements are pairwise distinct.

Output

For each test case, print "`FOX`" or "`CAT`" in one line, indicating who will be the first to make the matrix sorted. If the matrix will be unsorted forever, print "`NSFW`" in one line (indicating the boss will be irritated and fire some staff members slack in work).

Example

standard input	standard output
3 2 2 3 4 1 2 3 3 1 2 3 4 5 6 7 9 8 2 4 4 3 2 1 8 7 6 5	FOX NSFW CAT

Note

In the first test case, the fox will swap the first row and the second row. Then, the matrix will be sorted.

The second test case gives a special case: no matter how the fox and the cat perform operations, the matrix is impossible to be sorted by any of them. So it will certainly be unsorted forever.