# Jontrona of Liakot

Limits: 3s, 512 MB

Liakot, the king of Cox's Bazar, is currently in a dire situation. His own younger sister Ayesha is conspiring with a surfer named Sohel to dethrone Liakot.

It all started from when they were mere children. Ayesha and Sohel would team up to bully poor little Liakot. Often, they would give Liakot a programming problem and laugh at his failure.

In the problem, Sohel would make a tree of $N$ nodes where the $i$-th node contains a number $A_i$. The root of the tree is the node number $1$. Then Ayesha would give Liakot $Q$ queries to answer.

In each query, Ayesha would choose two nodes of the tree $U$ and $V$ ($V$ will always be in the subtree of $U$). Ayesha takes the subtree rooted at $U$ and then from that subtree, she removes the subtree rooted at $V$.

Ayesha takes all the numbers from the tree segment she made this way and does a XOR operation with $Z$ on all of these numbers (XOR is exclusive-or operation). She chooses another number $K$ in that query, and Liakot has to answer the $K$-th smallest number among Ayesha's XOR'ed numbers.

Can you help Liakot by finding the $K$-th smallest number after XOR-ing $Z$ to all the numbers which are in the subtree rooted at $U$ but not in the subtree rooted at $V$?

## Input

Input starts with two space-separated integers $N$ and $Q$, representing respectively the number of nodes in the tree and the number of queries.

In the next line, $N$ space-separated integers are given, the $i$-th integer representing the number $A_i$ of the node $i$.

In the following $N-1$ lines, two integers $X$ and $Y$ are given representing an edge between nodes $X$ and $Y$ in the tree. It is guaranteed that the given tree will be a valid one.

Finally, in each of the last $Q$ lines of input, four space-separated integers $U, V, Z$ and $K$ will be given representing the queries.

## Constraints

$1 \leq N \leq 10^5$
$1 \leq Q \leq 10^5$
$1 \leq A_i, Z, K \leq 10^9$
$1 \leq X, Y \leq N$
$1 \leq U, V \leq N$

It is guaranteed that the tree will always be a valid one. In each query, $V$ will always be within the subtree rooted at $U$.

## Output

For each query, print the answer in a single line. If the answer does not exist for a query, print -1 instead in the corresponding line.

# Joker's GCD Test

Limits 4s, 1.0 GB

Joker has a sequence **P** of **n** integers. Each two elements in the sequence are **pairwise distinct**. He defines the strength of the sequence as the **expected value** of the greatest common divisor (**gcd**) of any **two** randomly chosen **distinct** numbers from the sequence.

Joker will perform **q** updates on the sequence. The updates will be of the form :
**1 x** : Add **x** to the sequence. It is guaranteed that **x** does not exist in the sequence.
**2 x** : Remove **x** from the sequence. It is guaranteed that **x** exists in the sequence.
To save Gotham from Joker, you must answer him the strength of the sequence after each update.

## Input

First line of input contains **n** - denoting the initial length of the sequence ($1 \le n \le 10^6$).

Second line contains **n** space separated integers - the initial elements of the sequence ($1 \le P_i \le 10^6$).

Third line contains **q** - denoting the number of updates ($1 \le q \le 10^6$).

Each of the next **q** lines describes an update of the form : either **1 x** or **2 x** ($1 \le x \le 10^6$).

There will be at least one **Add** and one **Remove** update.

## Output

For each update, output in a single line the strength of the sequence after the update.

It can be shown that the strength can be always expressed as a fraction P / Q, where P and Q are coprime integers, $P \ge 0$, $Q > 0$ and Q is co-prime with **998244353**. You should compute $\mathbf{P \cdot Q^{-1}}$ **modulo 998244353**, where $\mathbf{Q^{-1}}$ denotes the multiplicative inverse of Q modulo **998244353**.

Furqan Software

# City of Burgerland

Limits: 1s, 256 MB

Have you ever heard of The Burgerland City? The city is full of burger shops. You will surely find some shops of burgers wherever you will look around to any part of that city.

The Burgerland consists of $N$ number of burger shops. The shops are numbered from $1$ to $N$. The $i^{th}$ shop has the capacity to serve $B_i$ burgers daily. The mayor of the city wants some consecutive shops to have at least a specific serving capacity to make the city more attractive to the visitors. You are recruited as a programmer to help the mayor in this project by answering some queries.

The mayor wants to ask $M$ queries, in each query, the mayor will give you three integers $l, r$ and $cap$. He wants to know the minimum number of capacity needed to be increased so that every shop numbered between $l$ to $r$ have serving capacity at least $cap$.

Suppose, there are 4 burger shops having capacity $[1, 7, 3, 5]$. If the mayor gave you $l = 2, r = 4$ and $cap = 6$, then the answer is $4$ as it is needed to be increased the capacity of $3^{rd}$ shop by $3$ and the capacity of $4^{th}$ shop by $1$ to make all the shops between segment $(2, 4)$ having capacity at least $6$. As the $2^{nd}$ burger shop has the capacity equal to $7$ which is more than $6$, no need to increase it.

~~abcbaee~~ → abcbe
~~abebe~~ aebee

## Input

The first line of input contains two integers $N(1 \leq N \leq 10^5)$ and $M(1 \leq M \leq 5 \times 10^5)$.

The second line contains $N$ integers, the $i^{th}$ integer denotes $B_i(1 \leq B_i \leq 10^9)$ which is the capacity of production of the $i^{th}$ burger shop.

Each of the next $M$ lines contains $l, r$ and $cap$ $(1 \leq l \leq r \leq N, 1 \leq cap \leq 10^9)$.

## Output

For each query, print the minimum number of capacity is needed to be increased so that $min(B_l, B_{l+1}, .., B_r)$ will be at least $cap$ in separate lines.

# Lexicographical Smallest String

Limits 1s, 512 MB

There is a string **s** of length **n** containing lowercase letters and an integer **k**. Among the letters those appear exactly **k** times in string **s**, pick the lexicographically maximum one. Now delete any (k-1) occurrences of this letter from the string in a way so that the resultant string is lexicographically smallest.

**Note that if there is no letter which appears k times in s, no deletion takes place.**

What is the lexicographical smallest string after the deletions (possibly none)?

## Input

The first line of the input contains **n** ($1 \le n \le 100000$), the number of lowercase letters in string **s** and **k** ($0 \le k \le 10$).

The second line contains the string **s**.

## Output

Print the lexicographically smallest string after the deletions (possibly none).

# Beauty Factor

Limits 1s, 32 MB

Mehedi and Burhan are two friends. Yesterday they have participated in a class of sorting. Actually they have recently learned sorting numbers. Today was the second class of sorting. Professor invented a new type of sorting numbers. He gave this task to Mehedi and Burhan as homework.

Professor defines **the beauty of a number n as count of distinct prime factor of n**. So, beauty of 1, 2, 3, 4, 5, 6 are **0**, **1**, **1**, **1**, **1**, **2** respectively because 1 has no prime factor and 4 has only one distinct prime factor.

He sorts numbers according to beauty. Number having smaller beauty will come first than having larger beauty. If two numbers have the same beauty, then a smaller number comes first. So if he sorts first 10 natural numbers then the order will be - 1, 2, 3, 4, 5, 7, 8, 9, 6, 10. So, summation of the numbers from index 7 to 9 is 8 + 9 + 6 = 23

Mehedi and Burhan tried to solve this problem, but they were not able to solve all the cases as it seemed complex to them. Can you help them by doing the same for **T** test cases?

# Input

First line will have **T**, number of test cases.
Each test case will have **4** integers - **N**, **X**, **L**, **R**.
You have to create an array of length **N** with first **N** natural numbers starting from **X**.
So, A = [ X, X+1, X+2,...,X+N-1]. Sort them according to the above method.
Now after sorting, calculate sum of the numbers from index **L** to **R** that is A[L] + A[L+1]+...+A[R-1]+A[R].

### Constraints

$1 <= T <= 2*10^5$

$1 <= N <= 10^5$

$1 <= X <= 10^5$

$1 <= L <= R <= N$

$X + N - 1 <= 10^5$

# Output

For each test case print the sum between **L** and **R**.

# Sgt. Laugh

Limits: 2.5s, 256 MB

That's it. Sgt. Laugh has had enough. He is determined to end the evil military reign of General Cry. But Laugh isn't stupid. He knows, he needs a team to break through the walls of Cry.

Platoon "BattleCry" has $n$ soldiers. Laugh will choose $k$ of these soldiers. The platoon is lined up. Height of the soldier $i$ is represented by $h_i$. The sergeant wiil choose the soldiers in a way so that their heights form a strictly increasing sequence, i.e. if he chooses soldiers with indices $i_1, i_2, .. i_{k-1}, i_k$ where $i_1 < i_2 < i_3 .. < i_k$, condition $h_{i_1} < h_{i_2} < h_{i_3} .. < h_{i_k}$ must hold. Also since the sergeant is a little bit picky, the height difference between the tallest and shortest soldier must be minimum possible.

Help Laugh choose the squad.

## Input

First line of input contains two integers $n, k$.
The following line contains $n$ integers representing the heights of the soldiers $h_i$.

## Constraints

- $1 \le n \le 2 \times 10^5$.
- $1 \le k \le \min(n, 30)$.
- $1 \le h_i \le 10^9$.

## Output

If no such squad could be formed, print $-1$ on a line.

Otherwise, output the minimum possible difference between the tallest and shortest soldier in the chosen squad.

# Earn Your Cupcake

Limits 10s, 1.0 GB

A delicious **NxN** grid is given to you. All the lattice points of this grid contain a cupcake each **except for M** specified points. (Those M points do not contain cupcakes.)

Now, you have to randomly choose four different lattice points from this grid. If your four chosen points make a rectangle and each of the four points contains a cupcake, then you will earn a cupcake for yourself.

Calculate the probability of winning a cupcake in the given grid.

## Input

First line contains two integers N (1 <= N <= $10^3$) and M (0 <= M <= 50)

Each of next M lines contains two integers $X_i$(0 <= $X_i$ <= N) and $Y_i$(0 <= $Y_i$ <= N) describing the ith point where the cupcake is missing. (each of these points are distinct)

## Output

Print one number expressing the probability of earning a cupcake in following format. If the answer is **P/Q**, then print the number **P** $\cdot$ **$Q^{-1}$ modulo $10^9$+7**, where $Q^{-1}$ denotes the multiplicative inverse of Q modulo $10^9$+7.

# Restore Missing Values

Limits 1s, 512 MB

Consider an array of '**N**' integers which is indexed from '1' to 'N'.
Exactly '**Q**' queries numbered from '1' to 'Q' are applied on them in increasing order.
For the 'i'-th query, an arbitrary range **[L, R]** of the array is selected **(1 <= L <= R <= N)** and each value from index 'L' to index 'R' gets changed to 'i'. **It is guaranteed that every index of the array is covered by at least one range.**

You are given an array of 'N' integers where value of the array is from '0' to 'Q'.
Value '0' means the value is missing which can be restored to any integer from '1' to 'Q'.
Value '**I**' **(1 <= i <= Q)** means 'i' was stored in the array while 'i'-th query was applied.
Your task is to check if the given array can be acquired after applying the 'Q' queries.

If it is possible to acquire the given array, restore the missing values of the array.
Array restoration may be possible with multiple solution. If multiple solutions are available, you have to restore the missing values such that the array is lexicographically minimum.

## Input

The first line of input file contains an integer 'T' (1 <= T <= 60) denoting the test case of input.
Each test case consists of two lines.
First line contains two integers 'N' and 'Q' (1 <= N, Q <= 100000),
where 'N' is the size of the array and 'Q' is the number of queries.
Second line of each test case contains 'N' integers 'A1', 'A2', ... 'An' (0 <= Ai <= Q).

## Output

Print 'YES' with the case number if array can be restored.
And then in the second line print the 'N' integers of the array.
Print 'NO' with the case number if array can't be restored.

NSTU ICE Fest Regional Programming Contest 2020 Toph

# Noman's Birthday

Limits 1s, 512 MB

Today is our friend Noman's birthday. For celebration of his birthday, we have already used **n** baloons to decorate the walls of the party room. These baloons are of two colors- red and green. After sometime, we came to know that Noman liked baloons arranged in such a manner that **no adjacent two** baloons have the same color.

So, we decided that if our arrangement does not match with Noman's choice, we will change it. We want to make Noman happy tomorrow.

The arrangement of n baloons is represented as a sequence of '0' and '1', where '0' represents red color and '1' represents green color. For example, if n = 5 and the sequence is "01101" it implies that first and fourth baloons are colored with red and others are colored with green.

Your task is to say if the sequence of baloons matches with Noman's choice or not.

## Input

The first line consists of an integer n(1<=n<=100), the number of baloons. The next line represents the arrangement of n ballons as a sequence where a '0' means red baloon and a '1' means green baloon.

## Output

If the sequence of baloons matches with Noman's choice then print the line **"No change needed"** else print **"Change needed"** (without quotation).