A. Random Game

time limit per test: 15 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

Daniel and sszcdjr are playing a game after playing Genshin Impact. The game is played on a sequence a of length n. Initially, all elements in a are equal to 0. They will do the following operations m times on it:

1. Daniel selects one segment $[l_i, r_i]$ $(1 \le l < r \le n)$;
2. Then, sszcdjr chooses a real number x such that $0 \le x \le 1$ **uniformly at random**;
3. For all $l_i \le j \le r_i$, set $a_j := a_j + x$.

However, sszcdjr thinks it is unfair for him. So he adds a rule: for any two segments $[l_i, r_i]$ and $[l_j, r_j]$ $(i \ne j)$, one of the following condition must holds:

- The two segments are completely disjoint. More formally, either $l_i < r_i < l_j < r_j$ or $l_j < r_j < l_i < r_i$;
- One of the two segments are inside another. More formally, either $l_i < l_j < r_j < r_i$ or $l_j < l_i < r_i < r_j$.

Note that no two l_i, r_i is the same. That is, $l_i \ne l_j$, $r_i \ne r_j$, $l_i \ne r_j$ for all $1 \le i \ne j \le n$.

Let s be the maximum number in a after the operations. Piggy is crazy about probabilities, so he gives you t queries. In each query, he gives you two numbers P, Q, and asks you the probability that $P \le s \le Q$. You just need to tell him the answer modulo $998\,244\,353$.

Input

The first line of input contains three integers n, m, t $(1 \le n \le 10^6$, $1 \le m \le 200, 1 \le t \le 5 \times 10^5$ $)$ — the length of the sequence, the number of operations and the number of queries.

Then m lines follow, the i-th line contains two integers l_i, r_i $(1 \le l_i \le r_i \le n)$ — the segment in the i-th operation.

In the following t lines, each line contains two integers p, q $(0 \le p < q \le 10^7)$ — the two numbers piggy give you in the query are equal to $P = p \cdot 10^{-6}$ and $Q = q \cdot 10^{-6}$. That is, $0 \le P < Q \le 10$.

Output

For each query print an integer — the probability that $P \le s \le Q$, modulo $998\,244\,353$.

Formally, let $M = 998\,244\,353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Example

input

```
6 3 5
1 6
2 3
4 5
0 1000000
1000000 2000000
200000 800000
0 1500000
114 514
```

output

```
332748118
665496236
143747187
540715692
586651392
```

B. Tree Climber

time limit per test: 5 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

Given a rooted tree with n nodes. Each edge has a weight and initially all weights are 0.

You need to perform m operations. In each operation you are given three integers l, r, x, which means you need to find the smallest connected component in the tree that contains all nodes in the range $[l, r]$, and add x to the weights of edges in this connected component.

After all operations are performed, output the weights of all edges.

Input

The first line contains two integers n, m.

The following $n-1$ lines each contain two integers u, v representing that there's an edge between u and v on the tree.

The following m lines each contains three integers l, r, x representing an operation.

Output

Output $n-1$ lines. The i-th line contains an integer representing the final weight of the i-th edge in the input order.

Example

input

```
6 3
1 2
1 3
2 4
2 5
5 6
3 5 1
```

```
4 6 10
1 2 100
```

output

```
101
1
11
11
10
```

Note

For the first operation, the smallest connected component contains vertices $1, 2, 3, 4, 5$.

For the second operation, the smallest connected component contains vertices $2, 4, 5, 6$.

For the third operation, the smallest connected component contains vertices $1, 2$.

$1 \le n \le 3 \times 10^5$, $0 \le m \le 10^6$, $1 \le u, v \le n, u \ne v$, $1 \le l \le r \le n$, $1 \le x \le 10^9$.

C. Beautiful sequence

time limit per test: 1.3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Students from Antarctic Penguin Language School are studying sequence.

We call a sequence $A = (A_1, A_2, \dots, A_n)$ of length n beautiful when all the following conditions are true:

- $A_1 \le A_2 \le \dots \le A_n$.
- For each A_i $(1 \le i \le n)$, $0 \le A_i < 2^{30}$.
- For each $1 \le i < n$, $A_i \oplus A_{i+1} = B_i$, where B is a given sequence, and $\oplus$ represents bitwise XOR.

You need to find the k-th smallest lexicographically legal beautiful sequence.

Input

Each test contains multiple test cases. The first line contains the number of test cases T. The description of the test cases follows.

The first line contains two integers n, k $(1 \le n \le 10^6, 1 \le k \le 2^{30})$.

The second line contains $n - 1$ integers $B_1, B_2, \dots, B_{n-1}$ $(0 \le B_i < 2^{30})$.

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

Output

For each test case, output your operations in the following format.

If the number of beautiful sequences is less than k, output one line containing an integer -1.

Otherwise, output one line containing n integers $A_1, A_2, \ldots, A_n$ separated by spaces — the k-th smallest lexicographically legal beautiful sequence.

Example

input
3 3 3 1 2 5 1 1 2 1 2 2 1 0
output
8 9 11 -1 0 0

D. Game on Tree

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob are playing a game. Initially, there is a set of string S. Alice and Bob take turns to play, with Alice going first. Each turn, a player can choose a string s from S, remove it from S, and select a character c which appears in s. Then, they remove all occurences of character c from s and divide s into several substrings along these removed positions. They add all non-empty substrings to the set S. The player who cannot make any move loses.

You are now given a tree with n nodes. Each node contains a character. Let $\mathrm{path}(u, v)$ represent the string formed by concatenating the characters on the shortest path from node u to node v (including nodes u and v). There are q queries, and for each query, given nodes u and v, you need to determine the winner when $S = \{\mathrm{path}(u, v)\}$. If Alice wins, also output the number of ways to make the first move.

Input

The first line contains two positive integers, n and q.

The second line contains a string of length n, where the i-th character represents the character at node i.

The next $n - 1$ lines each contain two positive integers, u and v, describing an edge in the tree.

Following that, there are q lines, each containing two positive integers, u and v, representing a set of queries.

Output

Output q lines in total. For each set of queries, if Alice wins, output 'Alice' followed by the number of ways to make the first move, separated by a space. Otherwise, output 'Bob'.

Examples

input

```
5 5
01001
2 1
3 1
4 3
5 4
4 5
4 3
1 5
2 5
5 1
```

output

```
Bob
Alice 1
Bob
Alice 1
Bob
```

input

```
15 15
446397013450369
2 1
3 2
4 3
5 4
6 5
7 5
8 5
9 5
10 5
11 10
12 7
13 6
14 2
15 6
1 13
10 5
7 13
4 2
3 10
9 3
8 3
9 4
2 7
6 5
1 6
4 5
14 3
13 1
9 12
```

output

```
Alice 1
Bob
Bob
Alice 3
Bob
Alice 1
Bob
Alice 1
Alice 5
Bob
Alice 5
Bob
Alice 1
Alice 1
Alice 3
```

Note

$1 \leq n \leq 5 \times 10^4$, $1 \leq q \leq 10^5$, $\Sigma = [0, 9] \cap \mathbf{Z}$。

E. Star wars

time limit per test: 0.8 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

According to a certain statistic, if I keep answering `NO`*, the probability of completely correct judgment in a battle is 45%.*

—— CSP-S 2022 T3

The new round of Star Wars has begun.

In this round of Star Wars, our side has established n bases in the universe. At the beginning of the war, all bases had zero troops and there were no wormholes. A wormhole connects two different bases and can be bidirectional.

You need to help the commander-in-chief write a program to answer some questions based on the battlefield situation.

Input

We use last to represent the answer of the last 4 operation and initialize $\text{last} = 0$.

The first line contains two integers n ($1 \leq n \leq 3 \times 10^5$) and q ($1 \leq q \leq 3 \times 10^5$), which represent the number of bases and the number of instructions that the commander-in-chief will give you.

Next q lines, each line contains one instruction. There are four types of instructions:

- `1 x y` Our side added a wormhole connecting x and y.
- `2 x y` The enemy destroyed a wormhole connecting x and y. It is guaranteed that this wormhole existed before this operation.
- `3 x c` Base x increased c ($0 \leq c \leq 10^9$) troops.

- `4 x` Find the sum of troops that can be reached from base x through at most one wormhole.

You need to transform the input x, y, c into $x \oplus (\text{last mod } 2^{30}), y \oplus (\text{last mod } 2^{30}), c \oplus (\text{last mod } 2^{30})$ to get the real data, where $\oplus$ means binary XOR.

Output

For each 4 operation, output a line representing the answer.

Example

input

```
5 10
1 5 1
3 5 10
3 4 100
1 1 4
4 1
2 111 107
1 106 107
3 106 10
4 107
4 214
```

output

```
110
210
210
```

F. Counting sequences

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Students from Antarctic Penguin Language School are counting sequences.

You are given three integers n, m, k, You need to calculate the number of sequence $A = (A_1, A_2, \dots, A_n)$ that satisfies both of the following conditions:

- For each $A_i (1 \le i \le n)$, $0 \le A_i < 2^m$.
- We let sequence B be the sequence obtained by rotating sequence A once,

$$\sum_{i=1}^{n} cnt_1(A_i \oplus B_i) = k$$

Where $\oplus$ represents bitwise XOR, one rotation operation will change $(A_1, A_2, \dots, A_{n-1}, A_n)$ to $(A_2, A_3, \dots, A_n, A_1)$. $cnt_1(x)$ represents the number of 1s in the binary representation of x.

Input

One line containing 3 integers n, m, k ($2 \le n < 998244353$, $1 \le m \le 10^8$, $1 \le k \le 5 \times 10^4$).

Output

Output one line containing an integer, representing the answer $\bmod 998244353$.

Examples

input
2 1 2
output
2

input
2 2 2
output
8

input
2 3 2
output
24

Note

In the first example, A can be $(0,1)$ or $(1,0)$.

In the second example, A can be $(0,1)$, $(0,2)$, $(1,0)$, $(1,3)$, $(2,0)$, $(2,3)$, $(3,1)$, $(3,2)$.

G. Cyperation

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given a circular array a of length n and a positive integer k, you can perform the following operation any number of times (including zero):

Select two indices i and j ($1 \le i < j \le n$) such that $\min(j-i, n+i-j) = k$, which means the minimum distance between i and j on the circular array is k, and subtract 1 from both a_i and a_j.

Determine if it is possible to transform the entire array into 0. If possible, output `YES`; otherwise, output `NO`.

You are given T test cases. Find the answer for each of them.

Input

The first line contains a positive integer T ($1 \le T \le 10^6$) , denoting the number of test cases. The first line of each test case contains two positive integers n and k ($2 \le n \le 10^6, 1 \le k \le n$), indicating the length of the array a and the minimum distance of i and j for each index selection. The second line consists of n non-negative integers a_i $(0 \le a_i \le 10^9)$.

It is guaranteed that $2 \le \sum n \le 10^6$.

Output

For each test case, output one string per line, either YES or NO, indicating whether it is possible or not.

Example

input

```
5
5 2
3 0 5 1 3
6 3
5 0 0 3 0 0
5 5
1 2 3 4 5
6 1
0 0 1 0 0 1
7 2
1 1 0 1 1 1 1
```

output

```
YES
NO
NO
NO
YES
```

Note

For the first test case, perform these operations: $(1,3),(1,3),(1,4),(3,5),(3,5),(3,5)$.

For the third test case, you can't do any operation, so it's impossible to transform the entire array into 0.

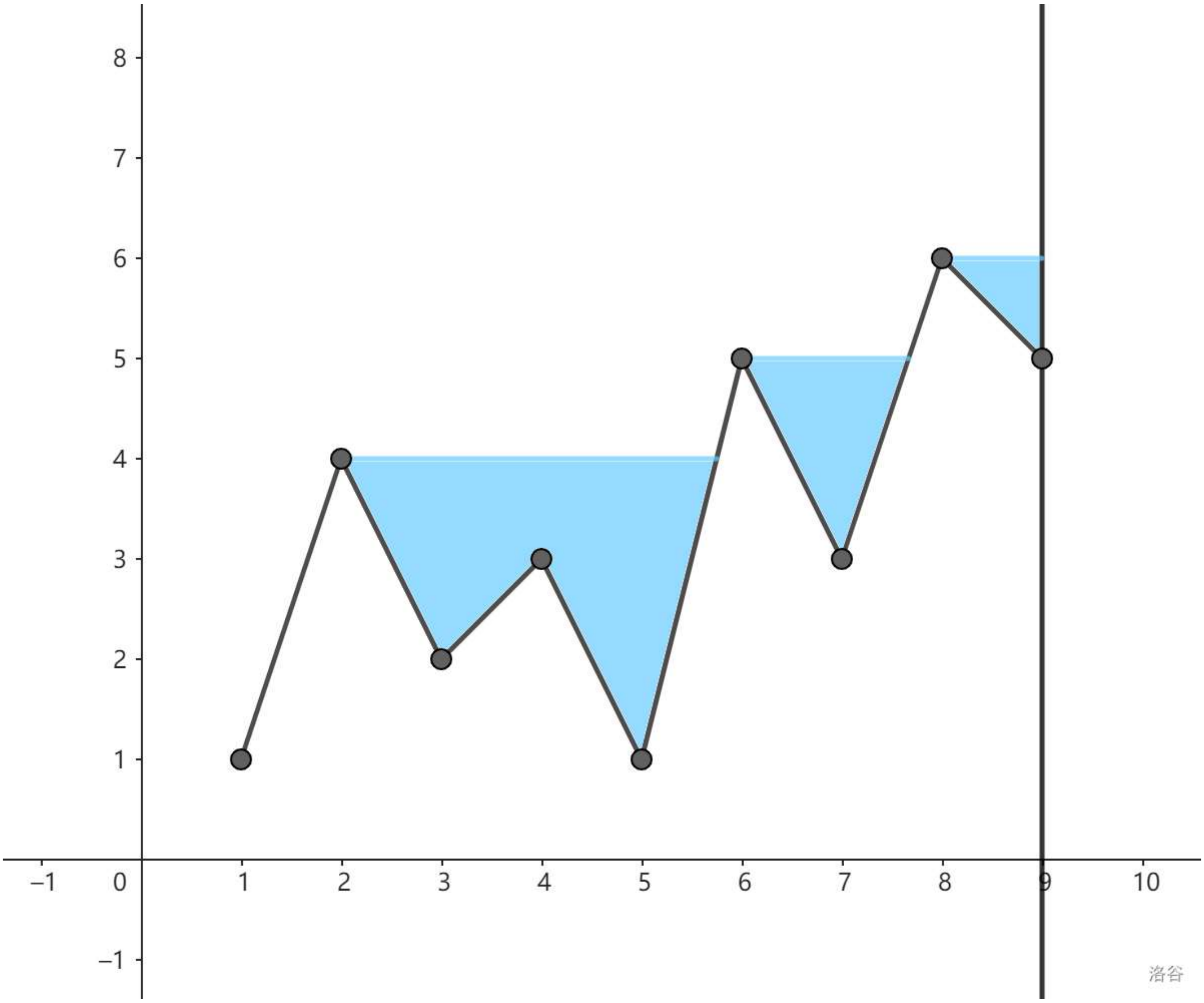
H. Mountain View

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You enjoy the mountain view, especially on a rainy day.

For a positive integer sequence $b_1, b_2, \cdots, b_k (k \ge 2)$, you define its **beauty** as following:

Suppose there is a broken line connecting $(1, b_1), (2, b_2), \cdots, (k, b_k)$ one by one on the Cartesian plane, representing a mountain's outline. To the immediate right of (k, b_k) is a vertical cliff whose height can be assumed infinite (but to the left of $(1, b_1)$ there isn't). Small ponds form on rainy days, and the **beauty** is the maximum ponding area on this 2D graph.



Now you have an integer sequence $a_1, a_2, \cdots, a_n$. Support two operations:

- `1 x y` — Set a_x to y.
- `2 l r` — Print the **beauty** of $a_l, a_{l+1}, \cdots, a_r$.

Input

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

The second line contains n integers: the initial values of $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le 10^9$).

The next q lines each contain three integers, representing the operations. It is guaranteed that $1 \le x \le n$, $1 \le y \le 10^9$, $1 \le l < r \le n$.

Output

Print the answer for each query. The answer may not be an integer, so your output will be considered correct if its absolute or relative error does not exceed 10^{-9}.

Example

input

```
9 5
1 4 2 3 1 5 3 6 5
2 1 9
```

```
1 5 3
2 4 6
2 2 7
2 1 8
```

output

```
7.791666666667
0.000000000000
4.750000000000
5.416666666667
```

Note

The blue area of the image in the statement demonstrates the first query of the first sample.

I. We Love Strings

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Students from Antarctic Penguin Language School love strings.

01 is the root of every thing. And strings in OI is the root of every thing. To show your almost manic passion for strings, you have to solve this problm.

We have n strings s_i which only contain letters 01?. ? means it can match either 0 or 1.

You need to find how many 01 strings are there which matches at least one of these strings. Since the answer may be enormous, you should output the answer modulo 998244353.

Input

The first line contains one integer $n(1 \leq n \leq 400)$.

The next n lines, the i-th line contains a string $s_i(1 \leq \sum |s_i| \leq 400)$.

Output

Output contains one integer, the answer modulo 998244353.

Example

input

```
3
10?
?0?
?11
```

output

```
6
```

J. Is it a tree?

time limit per test: 3 seconds

memory limit per test: 512 megabytes
input: standard input
output: standard output

Students from Antarctic Penguin Language School are counting connected components.

You are given a tree and a cactus with n vertices. The vertices is numbered from 1 to n. We define $f(i,j)$ as if we only keep those vertices in the cactus whose number appears on the path from i to j in the tree and the edges between them, how many connected components will be left. Vertex 1 is the root. For all $1 \le x \le n$, you need to calculate:

$$\sum_{i \in S_x} \sum_{j \in S_x} [i \le j] f(i,j)$$

in which S_x means the set of vertices in the subtree of vertex x. A cactus is a simple undirected graph whose nodes are in at most one simple cycle.

Input

The first line contains two integers, $n, m (1 \le n \le 5 \times 10^5, 1 \le m \le 10^6)$ — the number of nodes and the number of edges in the cactus.

The next $n-1$ lines, each line contains two integers, describing an edge in the tree.

The next m lines, each line contains two integers, describing an edge in the cactus.

Output

Output n lines, the i-th line contains an integer, the answer to $x = i$.

Example

input

```
6 7
1 2
1 3
2 4
2 5
3 6
1 2
2 3
3 1
1 4
4 5
5 6
6 4
```

output

```
30
9
4
1
1
1
```

K. Set

time limit per test: 10 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Sylvy has an array a of length n. Glacy wants you to calculate $\sum_{S\subseteq\{x|x\in Z,1\le x\le n\}} |S| \left(\min_{x\in S} a_x\right) \left(\max_{x\in S} a_x\right) \left(\bigoplus_{x\in S} a_x\right)$ module 998244353, where $\bigoplus$ denotes the bitwise operator xor.

Input

The input is given from Standard Input in the following format:

$$n$$
$$a_1\ a_2\ \cdots\ a_n$$

Output

Print the answer as an integer.

Examples

input
4 0 1 2 3
output
72

input
8 1 2 4 8 16 32 64 128
output
46631937

Note

$$n \le 1000000$$
$$0 \le a_i \le 2147483647$$

L. Misaka Mikoto's dynamic KMP problem

time limit per test: 1 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

Misaka Mikoto has a pattern string s. Let $|s|$ be the length of s, s_i be the i-th character of s, and $s[l \sim r]$ be the substring from the l-th character to the r-th character, that is, $s_l s_{l+1} \cdots s_r$.

In this problem, all characters are represented by an integer.

You need to perform m operations. There are two types of operations:

`1 x c`: change s_x to c.

`2 t`: given a text string t, find the number of times s occurs as a substring in t and the longest border of s.

The longest border of s means the largest number $r(1 \le r < |s|)$ that $s[1 \sim r]$ is the same as $s[(|s| - r + 1) \sim |s|]$. If such r doesn't exist, let the longest border of s be 0.

Let's call the operation 2 'query'.

Given two integers b, p, let x_i, y_i be the answer to the first and the second question of the i-th query, you only need to output $\left(\sum\ x_i \times y_i \times b^i\right) \bmod p$.

For some reason, the operations are encrypted. Let z be the value of $x_i \times y_i$ of the last query (if there aren't any queries before, let z be 0). For operation 1, you are given two integers x', c', which means $x = (x' \oplus z) \bmod |s| + 1$ and $c = c' \oplus z$. For operation 2, you are given an integer n and n integers $t'_1 \sim t'_n$, which means $|t| = n$ and $t_i = t'_i \oplus z$.

Input

The first line contains four integers $|s|, m, b, p$.

The second line contains $|s|$ integers, which represent the string s.

The i-th line of the following m lines contains the i-th operation: $1\ x'\ c'$ or $2\ n\ t'_1\ t'_2\ \cdots\ t'_n$.

Output

One integer that represents the answer.

Example

input
3 3 10 1000000007 1 2 1 2 5 1 2 1 2 1 1 3 3 2 4 3 3 3 3
output
420

Note

For all test cases,

$1 \le m, |s|, n \le 10^6$.

$\sum n \le 2 \times 10^6$, where $\sum n$ represents the sum of n of all queries.

$1 \le b, p, s_i, c, t_i \le 2^{31} - 1$.

$0 \le x', c', t'_i \le 10^{12}$.

M. Writing books

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bandycar is writing page numbers for T books. The page numbers of all books start from 1. The i-th book has n_i pages. Bandycar wants to know, for each book, how many digits does he need to write.

Input

The first line contains an integer T. The i-th line in the following T lines contains an integer n_i.

Output

Output T lines, the i-th line contains the answer to the i-th book.

Example

input

```
2
15
1000
```

output

```
21
2893
```

Note

For the first book, page 1 to 9 each has one digit; page 10 to 15 each has two digits. So the answer is $1 \times 9 + 2 \times 6 = 21$.

$1 \leq T \leq 10^5, 1 \leq n_i \leq 10^9$.