A. Dominoes on Tree

time limit per test: 2 seconds
memory limit per test: 256 megabytes

Given a tree with n nodes, rooted at node 1. At node i, there is a domino, which is a rectangle of size 1×2. This domino is divided into two squares on the left and right, containing a_i points and b_i points, respectively. The domino is undirected, meaning you can rotate it and swap the number of points in the left and right squares.

There is a group of people living on this tree, and the resident at node x can go to the subtree of x and collect all the dominoes (including the domino at node x itself). The residents want to form the fewest possible "chains" with these dominoes. You need to calculate the minimum number of "chains" that can be formed by the residents at all nodes $i \in [1, n]$ using all the dominoes in the subtree of node i. We consider each resident's operation to be **independent**, meaning that before each resident starts collecting, each node in the tree contains exactly one domino.

Here, a "chain" refers to a sequence of dominoes placed horizontally in order, ensuring that the number of points on the right side of the previous domino equals the number of points on the left side of the next domino.

Input

The first line of input contains an integer n $(1 \leq n \leq 2 \times 10^5)$, representing the number of nodes in the tree.

The next line contains n integers a_i $(1 \leq a_i \leq n)$, representing the number of points in the left square of the domino.

The following line contains n integers b_i $(1 \leq b_i \leq n)$, representing the number of points in the right square of the domino.

The next $n-1$ lines each contain two integers u, v $(1 \leq u, v \leq n, u \neq v)$, indicating that there is an edge between node u and node v. It is guaranteed that the given $n-1$ edges can connect the n nodes into a tree.

Output

Output a single line with n integers, where the i-th integer represents the minimum number of "chains" that the resident at node i can form.

Example

input
7 7 6 1 6 6 3 4 7 4 2 7 5 2 4 2 1 3 1 4 2 5 2 6 3 7 3
output

```
3 2 2 1 1 1 1
```

Note

For the sample:

- For the resident at node 2, one possible chain configuration is: `[4|6][6|7]` and `[6|5]`. Note that this configuration rotates the domino at node 2.
- For the resident at node 3, one possible chain configuration is: `[1|2][2|3]` and `[4|4]`.
- For the resident at node 1, one possible chain configuration is: `[4|4][4|6][6|7][7|7]` and `[1|2][2|3]` and `[6|5]`.

B. Base Conversion Master

time limit per test: 1 second
memory limit per test: 256 megabytes

Little T has received n arrays $A_{1\ldots n}$, where the length of the i-th array A_i is l_i. Additionally, Little T has provided two integers y and M, and wishes to determine all positive integers s within the range $[2, M]$ that satisfy the following requirements.

- Enumerate $i \in [1, n]$ in order:
  - Treat the array A_i as a number in base s. If there exists a number in the array A_i that is greater than or equal to s, it is considered not to meet the requirements.
  - Convert A_i and assign the resulting value s' to s.
- The initial value satisfies the requirements if and only if after completing the above n base conversions, $s = y$ holds.

Specifically, given an array $a_{1\ldots k}$ and an initial base s, the conversion of the array a treated as a number in base s results in:

$$s' = \sum_{i=1}^{k} a_i s^{k-i} = a_1 s^{k-1} + a_2 s^{k-2} + \cdots + a_k$$

Little T's research has found that the positive integers s that meet the conditions either do not exist or are exactly all positive integers within some interval $[l, r]$. You need to report the corresponding l, r, or determine that there are no positive integers s that satisfy the conditions.

Input

This problem contains multiple test cases. The first line contains an integer T $(1 \le T \le 10^5)$, representing the number of test cases.

For each test case:

The first line contains three integers n, y, M $(1 \le n \le 2 \times 10^5, 1 \le y \le 10^9, 2 \le M \le 10^9)$.

For the next n lines, the i-th line first contains an integer l_i $(1 \le l_i \le 2 \times 10^5)$, representing the length of the i-th array A_i. It is followed by l_i integers $A_{i,j}$ $(0 \le A_{i,j} \le 10^9)$, representing the contents of the array A_i. It is guaranteed that the first element of the array $A_{i,1}$ is not 0.

It's guaranteed that the total length of all arrays in a single input does not exceed 2×10^5.

Output

For each test case:

If there are no positive integers s that satisfy the requirements, output a line `-1 -1`;

Otherwise, if all satisfying positive integers form an interval $[l, r]$, output a line with two integers, l and r.

Example

input
3 4 39244 100 2 1 1 2 1 1 2 1 1 6 1 1 4 5 1 4 3 1000000000 1000000000 3 1 0 0 1 65536 2 15258 51712 2 27 3 4 1 0 0 0 1 27
output
5 5 257 1000000000 -1 -1

C. A Classic Stack Problem

time limit per test: 1 second
memory limit per test: 256 megabytes

For a permutation P, define $f(P)$ as follows:

```
function f(P):
    stack = []

    for element in P:
        while stack is not empty and stack.top() > element:
            stack.pop()
        stack.push(element)

    return the size of stack
```

You are given an integer n. Find the sum, mod 998244353, of $(f(P'))^3$ over all possible permutations of length n.

Input

This problem contains multiple test cases.

The first line contains a positive integer $T(1 \leq T \leq 10^5)$, representing the number of test cases.

For each test case, the first input is a positive integer n ($1 \leq n \leq 5 \cdot 10^5$).

Output

For each test case, output a single line containing an integer representing the answer.

Example

input
2 3 3741
output
53 805156151

D. A Beautiful Matrix Problem

time limit per test: 3 seconds
memory limit per test: 256 megabytes

A beautiful matrix A with n rows and n columns satisfies the following condition:

- All elements are integers between 0 and m.
- $A_{i,1} = 0$ for $1 \leq i \leq n$.
- $A_{i,j} \leq A_{i,j+1}$ for $1 \leq i \leq n$ and $1 \leq j < n$.
- $A_{i,j} + A_{k,l} \leq A_{i,l} + A_{k,j}$ for every quadruple of integers (i, j, k, l) such that $1 \leq i < k \leq n$ and $1 \leq j < l \leq n$.

You are given two integers n and m, find the number, mod 998244353, of beautiful matrices.

Input

The first line contains two integers n and m ($2 \leq n \leq 5 \cdot 10^5$, $1 \leq m \leq 5 \cdot 10^5$).

Output

Print the answer.

Examples

input
2 1
output

3

input
29912 86488
output
461289656

input
4 3
output
455

E. Array

time limit per test: 1 second
memory limit per test: 64 megabytes

Given an array a of length n with all initial elements set to 0, and an array p of length n. You need to process q operations, which are of two types:

1. Update operation: Given four integers $opt = 1$, l, r, x, for all $l \leq i \leq r$, add x to the element a_i.
2. Query operation: Given three integers $opt = 2$, l, r, compute $\sum_{i=l}^{r} a_{p_i}$.

For each query operation, output the result of the query. This problem is strictly online, and for each operation, the input parameters l, r, x must be XORed with the last output answer (do not XOR before the first query operation).

Input

The first line contains two integers n and q ($1 \leq n, q \leq 10^5$), representing the length of the array a and the number of operations.

The second line contains n integers, representing the array p ($1 \leq p_i \leq n$).

The next q lines describe an operation, formatted as one of the two types mentioned above. For all decrypted inputs, there are $1 \leq opt \leq 2$, $1 \leq l \leq r \leq n$, and $1 \leq x \leq 10^8$.

Output

For each query operation, output one line with one integer, representing the result of the query.

Example

input
6 10 3 6 5 1 2 6 1 1 3 8 1 2 3 9 1 2 5 4 1 2 3 9 2 1 3

1 33 33 38 1 32 32 35 2 35 33 1 36 34 36 2 39 35
output
34 38 81

F. Random Segment Tree

time limit per test: 5 seconds
memory limit per test: 256 megabytes

Define a *random segment tree* as a binary tree where each node represents a closed interval $[l, r]$:

- If $l = r$, then the node is a leaf node.
- If $l < r$ and the length of the interval $(r - l + 1)$ is even: let $x = \lfloor (l + r)/2 \rfloor$. The left child of the node represents $[l, x]$, and the right child represents $[x + 1, r]$.
- If $l < r$ and the length of the interval $(r - l + 1)$ is odd: let $x = (l + r)/2$. With a probability of $1/2$, the left child represents $[l, x]$ and the right child represents $[x + 1, r]$; with a probability of $1/2$, the left child represents $[l, x - 1]$ and the right child represents $[x, r]$.

Given a segment tree, define $\mathrm{cost}(x, y)$ as follows:

When querying the interval $[x, y]$, starting from the root node, suppose the current node is $[l, r]$, the process is as follows:

- If $[x, y]$ contains $[l, r]$, the query ends.
- Otherwise, if the left child node intersects with $[x, y]$, query the left child node; if the right child node intersects with $[x, y]$, query the right child node.
- The value of $\mathrm{cost}(x, y)$ is the number of nodes visited during the query process.

Given n, consider a random segment tree whose root node represents $[1, n]$. For $1 \leq i \leq 2 \cdot n$, compute the expected number of intervals $[x, y]$ where $\mathrm{cost}(x, y)$ equals i, with the result modulo 998244353.

Input

The first line contains two integers n ($1 \leq n \leq 10^5$).

Output

Print $2 \cdot n$ lines. Each line contains an integer, representing the answer modulo 998244353.

Examples

input
3
output

```
1
2
2
1
0
0
```

input
4
output
1 2 4 2 1 0 0 0

input
5
output
1 2 4 4 499122179 1 499122177 0 0 0

G. Turn Around

time limit per test: 1 second
memory limit per test: 256 megabytes

There are n people standing on a number line, each facing either left ('L') or right ('R'). A string S of length n is given, where each character is 'L' or 'R', representing the initial direction of each person. At each moment, when two adjacent people face each other (i.e., the left one faces right and the right one faces left), they will turn around simultaneously: if a person faces left, he turns right; and vice versa.

Your task is to determine the time after which no one's direction changes anymore.

Additionally, there are q modifications. Each modification gives an index x, meaning the initial direction of the x-th person is flipped (from 'L' to 'R' or vice versa). After each modification, output the answer to the above problem.

Input

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

The second line contains a string S of length n, representing the initial direction of each person.

The next q lines contain an integer x ($1 \leq x \leq n$).

Output

Print q lines. Each line contains an integer, representing the answer.

Examples

input
5 5 LRLRL 5 1 3 4 3
output
1 2 0 3 3

input
5 5 RLLLL 1 2 3 4 5
output
0 3 3 3 0

H. Knapsack Problem

time limit per test: 1 second
memory limit per test: 256 megabytes

There are n types of items, each with an unlimited quantity, and each item has two attributes: volume a_i and value b_i.

Define $f(v)$ as follows: you have a knapsack with a capacity of v, and you need to select several items to put into the knapsack such that the total value of the items is maximized, and the total volume of the items in the knapsack does not exceed its capacity.

Given these n types of items and a number V, calculate $\sum_{v=1}^{V} f(v) \bmod 998244353$.

Input

The first line contains two positive integers n ($1 \le n \le 10^5$) and V ($1 \le V \le 10^{18}$).

The next n lines each contain two positive integers a_i ($1 \le a_i \le 1500$) and b_i ($1 \le b_i \le 10^9$), representing the attributes of each type of item.

Output

Output a single number representing the answer.

Examples

input
5 10 7 9 5 3 1 1 2 4 9 10
output
105

input
10 20 4 11 10 13 6 11 10 1 5 8 9 13 7 7 13 8 3 13 7 6
output
819

I. Longest Common Substring

time limit per test: 2 seconds
memory limit per test: 256 megabytes

The *longest common substring* between two strings is the longest sequence of consecutive characters that appears in both strings.

Given a string s of length n and q queries, each query provides two parameters l and r. Calculate the length of the longest common substring between the prefix $s[1..l]$ and the suffix $s[r..n]$.

Input

The first line contains two positive integers n and q ($1 \le n, q \le 2 \cdot 10^5$) — the length of the string s and the number of query operations.

The second line contains a string s consisting of n lowercase English letters.

The following q lines each contain two positive integers l and r ($1 \leq l, r \leq n$).

Output

For each query operation, output an integer on a separate line, representing the length of the longest common substring between $s[1..l]$ and $s[r..n]$.

Example

input
10 14 aaabaaaaab 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 7 1 4 8 6 10 5 5 2 5
output
1 2 3 4 4 4 3 2 1 7 3 1 4 2

J. Game on a Tree

time limit per test: 1 second
memory limit per test: 512 megabytes

Little Q is playing a game on a tree.

The rules of the game are as follows: In each round, Little Q can choose an undeleted point on the tree and delete it along with all of its directly connected edges. The game ends when all edges have been deleted (note that not all nodes need to be deleted).

Now Little Q wants to know how many rounds the game is expected to last if the node to delete is chosen **uniformly at random** in each round. Output the result modulo 998244353.

Input

The first line contains a positive integer $n(2 \leq n \leq 5000)$, representing the number of nodes in the tree.

The next $n-1$ lines each contain two positive integers $x, y(1 \leq x, y \leq n)$, indicating that there is an edge between x and y.

Output

Output a single integer, representing the answer.

Examples

input
3 1 2 2 3
output
665496237

input
5 1 2 2 3 3 4 2 5
output
598946615

K. Maximum GCD

time limit per test: 2 seconds
memory limit per test: 256 megabytes

Given a sequence a of length n, you need to perform exactly one operation: select an interval and a non-negative integer X, and add X to all numbers in the interval, maximizing the GCD of all numbers in the sequence. Output the maximum GCD, and if it is infinite, output 0.

Input

This problem contains multiple test cases.

The first line contains a positive integer $T(1 \leq T \leq 10^5)$, representing the number of test cases.

For each test case, the first input is a positive integer $n(1 \leq n \leq 10^5)$, representing the length of the sequence.

Next, input n positive integers, representing the sequence $a(1 \leq a_i \leq 10^5)$.

It is guaranteed that $\sum n \leq 2 \cdot 10^5$.

Output

For each test case, output a single line containing an integer, representing the answer.

Example

input
5 2 114 514 1 2 5 1 2 3 5 8 5 4 3 3 3 4 5 6 1 4 7 9
output
514 0 1 4 3

L. Minimum Parenthesis String

time limit per test: 1 second
memory limit per test: 256 megabytes

Given m intervals, find a valid parenthesis string of length $2n$ that is **lexicographically smallest**, such that there is at least one left parenthesis in each interval. If there is no solution, output -1.

We define that the lexicographic order of a left parenthesis is smaller than that of a right parenthesis.

Input

This problem contains multiple test cases. The first line contains an integer T $(1 \leq T \leq 10^5)$, representing the number of test cases.

For each test case, the first line contains two positive integers n and m $(1 \leq n, m \leq 10^5)$, representing half the length of the parenthesis string to be constructed and the number of restricted intervals, respectively.

Next, there are m lines, each containing two positive integers l and r $(1 \leq l \leq r \leq 2n)$, indicating that the constructed parenthesis string must have at least one left parenthesis in the interval $[l, r]$.

It is guaranteed that $\sum n \leq 2 \cdot 10^5$ and $\sum m \leq 4 \cdot 10^5$.

Output

For each test case, output a string representing the constructed **lexicographically smallest** valid parenthesis string. If there is no solution, output -1.

Example

input
3 2 2 1 2 3 4 3 4 1 1 2 2 3 3 4 4 5 3 2 4 3 9 7 10
output
() () -1 ((((()) ()))

M. Minimum Difference

time limit per test: 1 second
memory limit per test: 256 megabytes

Given n **distinct** integers a_i, you need to separate these numbers into exactly two sets, and arrange them arbitrarily to construct two base m numbers A and B (leading zeros are allowed), such that the difference between A and B is minimized.

Input

This problem contains multiple test cases. The first line contains an integer T $(1 \leq T \leq 10^5)$, representing the number of test cases.

For each test case, the first line contains two positive integers $n, m (2 \leq n, m \leq 10^5)$, representing the number of integers and the base, respectively.

The next line contains n integers $a_i (0 \leq a_i < m)$, ensuring that these n integers are all **distinct**.

It is guaranteed that $\sum n \leq 2 \cdot 10^5$.

Output

Since the minimum difference can be very large, you only need to output the result of the difference converted to decimal modulo 998244353.

For each test case, output one integer per line, representing the result.

Example

input

```
3
4 10
3 4 5 6
7 10
1 2 3 4 5 6 7
6 16
2 3 5 7 11 13
```

output

```
7
469
124
```