Problem A. Iron Warrior

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are playing a card game. Initially, your draw pile is empty, and you have 4 cards in your hand: Rage, Shrug It Off, Pommel Strike, and Body Slam. Each time you play a card from your hand, you put it into the draw pile after calculating its effects. The effects of each card are as follows:

Rage: Skill card; costs 0 energy. After you play an attack card, you gain 5 block (the effect is stackable, meaning if you have played Rage twice, you will gain 10 block after each attack card).

Shrug It Off: Skill card; costs 1 energy. Gain 11 block and draw one card.

Pommel Strike: Attack card; costs 1 energy. Deal 10 damage and draw two cards.

Body Slam: Attack card; costs 0 energy. Deal damage equal to the amount of block (damage is dealt first, then the block gained from Rage is calculated).

Each time you draw cards, if the number of cards drawn is greater than or equal to the size of the draw pile, you draw all cards into your hand; otherwise, you randomly draw the specified number of cards from the draw pile into your hand. The question is: if you initially have n energy, what is the maximum damage you can guarantee to deal (regardless of how the cards are drawn)?

Input

The first line of input contains a single integer n ($1 \le n \le 10^{18}$).

Output

Output a line containing a single integer: the maximum damage that can be dealt.

Examples

standard input	*standard output*
1	20
3	72
4	105

Note

In the first test case, you can play the following cards in order:

1. Play Rage. Draw pile: Rage; Hand: Shrug It Off, Pommel Strike, Body Slam.
2. Play Body Slam, dealing 0 damage, and gaining 5 block (because of Rage). Draw pile: Rage, Body Slam; Hand: Shrug It Off, Pommel Strike.
3. Play Pommel Strike, costing 1 energy, drawing Rage and Body Slam from the draw pile, dealing 10 damage, and gaining 5 block (because of Rage). Draw pile: Pommel Strike. Hand: Rage, Shrug It Off, Body Slam.
4. Play Body Slam, dealing 10 damage, and gaining 5 block (because of Rage). Draw pile: Pommel Strike, Body Slam. Hand: Rage, Shrug It Off.

In the second test case, you can play: Rage, Body Slam, Pommel Strike, Shrug It Off, Body Slam, Pommel Strike, Body Slam.

In the third test case, you can play: Rage, Shrug It Off, Body Slam, Pommel Strike, Shrug It Off, Body Slam, Pommel Strike, Body Slam.

Problem B. Little, Cyan, Fish!

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	1024 mebibytes

There is a chessboard with n columns and n rows, totaling $n \times n$ squares. Columns and rows are numbered starting from 1, and the coordinates of the square in the i-th column and the j-th row are marked as (i, j). Now, you need to perform q operations on this chessboard.

There are three types of operations:

- Mark a horizontal line with a *Little Sign*. Specifically, given two squares (x_1, y_1) and (x_2, y_2), ensuring that $y_1 = y_2$, mark all squares between these two squares (including the two squares) with a *Little Sign*.
- Mark a vertical line with a *Cyan Sign*. Specifically, given two squares (x_1, y_1) and (x_2, y_2), ensuring that $x_1 = x_2$, mark all squares between these two squares (including the two squares) with a *Cyan Sign*.
- Mark a diagonal line with a *Fish Sign*. Specifically, given two squares (x_1, y_1) and (x_2, y_2), ensuring that $x_2 - x_1 = y_2 - y_1$, mark all squares between these two squares (including the two squares) with a *Fish Sign*.

Now you want to know the sum of $a_x \cdot b_y$ for all squares (x, y) on the chessboard that have all three types of markings (*Little*, *Cyan*, *Fish*) after q coloring operations, modulo 998 244 353.

Input

The first line of input contains two positive integers, n and q: the size of the chessboard and the number of coloring operations ($1 \le n, q \le 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 998\,244\,353$).

The third line contains n integers $b_1, b_2, \ldots, b_n$ ($0 \le b_i < 998\,244\,353$).

Each of the next q lines contains four positive integers: x_1, y_1, x_2, and y_2 ($1 \le x_1, x_2, y_1, y_2 \le n$; $(x_1, y_1) \ne (x_2, y_2)$), where x_1, y_1, x_2, y_2 represent the four parameters of the coloring operation. It is guaranteed that each given operation has one of the three types described above.

Output

Output a line containing a single integer: the answer to the problem modulo 998 244 353.

Example

standard input	*standard output*
5 3 1 2 3 4 5 1 2 3 4 5 1 3 5 3 3 5 3 1 5 5 1 1	9

Problem C. Currency

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

There is a $2 \times n$ grid, and you need to walk from the top-left corner $(1, 1)$ to the bottom-right corner $(2, n)$. Each edge has a weight, and there are m additional constraints. Each constraint is described by three integers, i, j, and c. What they mean is: if you traverse both edges $(1, i)$ to $(1, i + 1)$ and $(2, j)$ to $(2, j + 1)$, you incur an additional cost of c.

Find the minimum value of the sum of the edge weights plus the costs incurred.

Input

The first line contains two integers, n and m ($1 \le n \le 500$; $1 \le m \le 1000$).

The next three lines describe the weights of each edge:

The first of these lines contains $n - 1$ integers, where the i-th integer represents the weight of the edge from $(1, i)$ to $(1, i + 1)$.

The second of these lines contains n integers, where the i-th integer represents the weight between $(1, i)$ and $(2, i)$.

The third of these lines contains $n - 1$ integers, where the i-th integer represents the weight of the edge from $(2, i)$ to $(2, i + 1)$.

All edge weights are positive and do not exceed 10^9.

Each of the following m lines contains three integers, i, j, and c, representing a constraint ($1 \le i, j < n$; $1 \le c \le 10^9$).

Output

Print a line containing a single integer: the answer to the problem.

Example

standard input	*standard output*
5 2 2 3 5 2 6 1 2 1 1 1 2 4 2 1 4 4 2 3 1	13

Problem D. Widely Known Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	1024 megabytes

This problem might be well-known in some countries, but how do other countries learn about such problems if nobody poses them?

You are given a string s of length n consisting of lowercase English letters. The characters of s are numbered from 1 to n.

You are also given m patterns, where the i-th pattern is $s[\mathrm{tl}_i \dots \mathrm{tr}_i]$ (inclusive).

Now, there are q queries. Each query is denoted by two endpoints, ql_i and qr_i. For each query, you need to find how many patterns appear **at least once** in $s[\mathrm{ql}_i \dots \mathrm{qr}_i]$ (inclusive).

Input

The first line of input contains three integers: n, m, and q ($1 \le n \le 4 \cdot 10^5$; $1 \le m, q \le 10^6$).

The next line of input contains a string s of length n.

The next m lines of the input describe the patterns. The i-th of these lines contains two integers, tl_i and tr_i ($1 \le \mathrm{tl}_i \le \mathrm{tr}_i \le n$).

The next q lines of the input describe the queries. The i-th of these lines contains two integers, ql_i and qr_i ($1 \le \mathrm{ql}_i \le \mathrm{qr}_i \le n$).

Output

Output q lines. The i-th of these lines should contain a single integer: the answer for the i-th query.

Example

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

Problem E. Light Drinking and Low Singing

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	1024 mebibytes

There is a binary sequence of length n, and you need to perform q operations. The operations are as follows:

1. For the interval $[\ell, r]$, simultaneously change each adjacent pair `01` within the interval into `10`.
2. For the interval $[\ell, r]$, simultaneously change each adjacent pair `10` within the interval into `01`.
3. Query the number of `1`s in the interval $[\ell, r]$.

Input

The first line contains two integers, n and q ($1 \le n \le 2 \cdot 10^6$; $1 \le q \le 2.5 \cdot 10^5$).

The next line contains a binary string of length n: the initial binary sequence.

Each of the following q lines contains three integers, t, ℓ, and r: the type and bounds of an operation ($1 \le t \le 3$; $1 \le \ell \le r \le n$).

Output

For each operation of type 3, output a line with a single integer: the answer to the query.

Example

standard input	*standard output*
10 10 0011101100 2 5 9 3 2 5 1 1 10 1 1 5 3 4 6 1 2 5 2 4 9 3 5 10 3 2 7 3 1 8	2 1 3 3 4

Problem F. Trash Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given an $n \times n$ grid where each cell is either black (0) or white (1).

Count the number of rectangular regions that satisfy the following condition: All white cells within the region can be exactly covered by non-overlapping 2×2 squares, and no black cells are covered. Each 2×2 square must lie entirely within the region and consist of four white cells.

Input

The first line contains a positive integer n, the size of the grid ($1 \le n \le 300$).

Each of the next n lines contains a string of length n composed of digits '0' and '1'. These lines represent the rows of the grid.

Output

Print a single integer: the number of valid rectangular regions.

Example

standard input	*standard output*
4 0110 0110 1111 1111	17

Note

Please note that regions without any white cells are also counted as valid regions.

Problem G. Analysis

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

Given an undirected tree with n vertices, Little Q wants to delete all tree edges. Initially, he selects a starting vertex. In each operation, he can perform one of the following three actions:

1. Choose an adjacent undeleted edge, traverse it, and delete the edge.
2. Pay a cost of A to restore a deleted edge.
3. Pay a cost of B to teleport to any vertex.

Find the minimum total cost required to delete all tree edges.

Input

The first line contains three positive integers: n, A, and B ($1 \le n \le 5 \cdot 10^5$; $1 \le A, B \le 10^9$).

Each of the next $n-1$ lines contains two integers u and v: two vertices connected by an edge ($1 \le u, v \le n$).

Output

Output a single integer: the minimum total cost required to delete all tree edges.

Examples

standard input	*standard output*
5 100 1000 1 2 2 3 3 4 4 5	0
5 100 200 1 2 1 3 2 4 2 5	100

Problem H. Algebra

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

Construct a tree in the following classic way:

- Root the tree at vertex 1.
- For each vertex i from 2 to n, in order, select a vertex p from 1 to $i-1$ uniformly at random, and make p the parent of i.

Let the size of the subtree of vertex u be s_u, and $f_u = s_u^k$ (that is, the k-th power of s_u). For each vertex u, calculate the expected value of f_u, modulo a given prime number M.

Formally, it can be shown that, under the constraints below, each expected value can be represented as p/q where q is coprime with M. You have to output the integer value $p \cdot q^{-1} \bmod M$. Here, q^{-1} is an integer such that $q \cdot q^{-1} \bmod M = 1$.

Input

The first line of input contains three integers: n, k, and M ($1 \le n \le 10^5$; $1 \le k \le 200$; $10^8 \le M \le 10^9+7$).

It is guaranteed that M is a prime number.

Output

Output a line containing a single integer: the answer to the problem.

Examples

standard input	*standard output*
3 1 1000000007	3 500000005 1
3 2 998244353	9 499122179 1

Problem I. Twenty-two

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

Consider a sequence of length n, denoted as $a_1, a_2, \ldots, a_n$. There are q operations planned. Each operation has one of the two following types:

1. Given c, set $a_i \leftarrow \min(a_i, c)$ for all $1 \le i \le n$.
2. Given ℓ, r, and c, set $a_i \leftarrow \max(a_i, c)$ for all $\ell \le i \le r$.

We will apply each operation exactly once. However, the order of operations can be arbitrary: there are $q!$ different possible orders. For each order, applying the operations to the initial sequence $a_1, a_2, \ldots, a_n$ in that order will yield a final sequence. The question is how many different final sequences are possible, modulo 998 244 353.

Input

The first line contains three integers, n, m, and k: the length of the sequence, the number of operations of type 1, and the number of operations of type 2 ($1 \le n, m, k \le 150$).

The second line contains n integers $a_1, a_2, \ldots, a_n$: the initial sequence ($1 \le a_i \le n$).

The third line contains m integers: the parameter c for each operation of type 1 ($1 \le c \le n$).

The next k lines describe the operations of type 2. Each of these lines contains three integers, ℓ, r, and c: the parameters of the operation ($1 \le \ell \le r \le n$; $1 \le c \le n$).

Output

Output a single positive integer: the number of different final sequences modulo 998 244 353.

Example

standard input	*standard output*
5 2 2 4 1 3 5 2 2 4 1 3 3 2 5 5	6

Problem J. Loving You in My Humble Way

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

This is an output-only problem.

You need to construct a 3-uniform hypergraph with 1893 vertices and at least 13 244 edges such that the graph does not contain any of BC_2, BC_3, or BC_4.

A 3-uniform hypergraph is a pair (V, E) where each edge (element of E) is a set of **three** distinct vertices from V.

A BC_k consists of k distinct vertices $v_1, v_2, \ldots, v_k$ and k distinct edges $e_1, e_2, \ldots, e_k$ such that $v_i \in e_i \cap e_{i \bmod k+1}$.

Input

There is no input for this problem.

Output

The first line should contain a positive integer m: the number of edges.

Each of the next m lines should contain three integers: the vertices that form an edge of the hypergraph. The order of edges and the order of vertices in each edge can be arbitrary.

To get accepted, you must output a solution with $m \geq 13\,244$.

Example

standard input	standard output
<no-input>	3 1 2 3 3 4 5 5 6 7

Note

The sample output is given only to demonstrate the output format. It is a correct solution with 3 edges, but you need to construct a solution with at least 13 244 edges.

Problem K. Ying's Cup

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Given an undirected tree $T = (V, E)$ of size n, for each $k = 1, 2, \ldots, n$, determine how many permutations $a_1, a_2, \ldots, a_n$ of 1 to n have exactly k vertices as *local minima*, modulo $998\,244\,353$.

A vertex u is a *local minimum* if and only if $a_u < a_v$ for all $(u, v) \in E$. In other words, a_u is smaller than any of its neighbors.

Input

The first line contains a number n ($1 \le n \le 500$).

Each of the next $n-1$ lines contains two integers, x_i and y_i: the endpoints of an edge ($1 \le x_i, y_i \le n$).

Output

Output n lines, each containing a non-negative integer: the answers for $k = 1, 2, \ldots, n$ modulo $998\,244\,353$.

Example

standard input	*standard output*
5 1 2 1 3 2 4 2 5	28 54 38 0 0