The 2nd Universal Cup

Stage 24: Chongqing

February 24-25, 2024

This problem set should contain 12 problems on 18 numbered pages.

Problem A. Code Congestion

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Little Beth, to commemorate the discontinued Code Jam, prepared a "Code Congestion Memorial Competition." Beth's friend, Little Mho, also came to watch, so Beth wants to predict Mho's score.

The competition lasts for T seconds, with n problems. The score for the i-th problem $(1 \le i \le n)$ is a_i, and Beth predicts that Mho will need t_i seconds to complete it.

There are two types of problems: results visible and results invisible. The results of the invisible problems are only known after the competition ends, while the results of the visible problems are known immediately after submission. Beth has not yet determined the type of each problem.

Mho will first complete all the results visible problems in order from the smallest index to the largest index, then complete all the results invisible problems in the same order. Mho will spend t_i seconds to complete the i-th problem, and a submission will be made on the i-th problem if and only if the total time spent on the i-th problem and all previous problems does not exceed T.

Since Mho's submissions are always correct (AC), Beth wants to know, for all 2^n ways to determine the types of n problems, the sum of the total scores Mho can get. Since the answer can be very large, you need to take the answer modulo 998244353.

Input

The first line of the input contains two integers n, T ($1 \le n \le 200$, $1 \le T \le 3 \times 10^5$,), representing number of problems and competition time, respectively.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le 3 \times 10^5$), representing the score of each problem.

The third line contains n integers $t_1, t_2, \cdots, t_n$ ($1 \le t_i \le T$), representing the time Mho takes to solve each problem.

Output

Output one line containing an integer, representing the sum of the total scores Mho can get for all ways to determine the types of n problems, modulo 998244353.

Example

standard input	standard output
3 3 2 3 4 1 2 2	40

Problem B. Tree and Permutation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given an integer n, an undirected tree with n nodes, and two distinct nodes s, t on the tree where each edge has a length of 1. Nodes are numbered with integers from 1 to n. Let $\text{dist}(u, v)$ denote the distance between nodes u and v (i.e., the number of edges on the simple path between them). You are required to find a permutation $p_1, p_2, \cdots, p_n$ of numbers from 1 to n that satisfies the following two conditions:

- $p_1 = s, p_n = t$;
- For $d_i = \text{dist}(p_i, p_{i+1})$ where $1 \le i \le n - 1$, the permutation should minimize $\oplus_{i=1}^{n-1} d_i$, where $\oplus$ denotes the bitwise XOR operation.

If there are multiple permutations that satisfy the conditions, output any one of them.

Input

This problem has multiple test cases. The first line inputs a positive integer T ($T \ge 1$) indicating the number of test cases.

For each test case, the first line inputs three positive integers n, s, t ($2 \le n \le 5 \times 10^4$, $1 \le s, t \le n$, $s \ne t$). The following $n - 1$ lines each contain two positive integers u, v ($1 \le u, v \le n$, $u \ne v$), indicating that there is a direct undirected road connection (i.e., an edge on the tree) between locations u and v.

It is guaranteed that the sum of n over all test cases does not exceed 5×10^5

Output

For each test case, output a line with n positive integers $p_1, p_2, \cdots, p_n$, ensuring it is a permutation of 1 to n with $p_1 = s, p_n = t$, and $\oplus_{i=1}^{n-1} d_i$ is minimized.

Examples

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

Problem C. Secret Poems

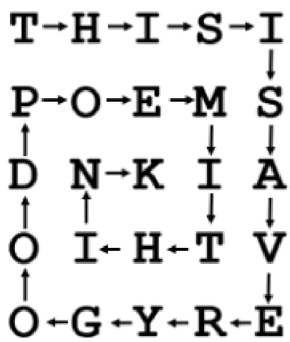
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The Yongzheng Emperor (13 December 1678 – 8 October 1735), was the fifth emperor of the Qing dynasty of China. He was a very hard-working ruler. He cracked down on corruption and his reign was known for being despotic, efficient, and vigorous.

Yongzheng couldn't tolerate people saying bad words about Qing or him. So he started a movement called "words prison". "Words prison" means literary inquisition. In the famous Zhuang Tinglong Case, more than 70 people were executed in three years because of the publication of an unauthorized history of the Ming dynasty.

In short, people under Yongzheng's reign should be very careful if they wanted to write something. So some poets wrote poems in a very odd way that only people in their friends circle could read. This kind of poems were called secret poems.

A secret poem is a $N \times N$ matrix of characters which looks like random and meaning nothing. But if you read the characters in a certain order, you will understand it. The order is shown in the left figure:



Following the order indicated by arrows, you can get "THISISAVERYGOODPOEMITHINK", and that can mean something.

But after some time, poets found out that some Yongzheng's secret agent called "Mr. blood dripping" could read this kind of poems too. That was dangerous. So they introduced a new order of writing poems as shown in the right figure. And they wanted to convert the old poems written in old order as figure1 into the ones in new order. Please help them.

Input

The first line of the input is an integer N ($1 \le N \le 100$), indicating that a poem is a $N \times N$ matrix which consist of capital letters.

Then N lines follow, each line is an N letters string. These N lines represent a poem in old order.

Output

Output N lines, each line is an N letters string, indicating the converted poem.

Examples

standard input	standard output
5 THSAD IIVOP SEOOH RGETI YMINK	THISI POEMS DNKIA OIHTV OGYRE
2 AB CD	AB DC
4 ABCD EFGH IJKL MNOP	ABEI KHLF NPOC MJGD

Problem D. Segment Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Recently, Little Y learned how to maintain a sequence using a segment tree and support the operation of interval summation.

The following provides the definition of the segment tree in this problem, which might differ from the segment tree you are familiar with.

- A segment tree is a rooted binary tree, with each node corresponding to an interval $[l, r)$ on the sequence, where the root node corresponds to $[0, n)$.
- For each node, if the interval $[l, r)$ it represents satisfies $r - l = 1$, then it is a leaf node; otherwise, there exists an integer m $(l < m < r)$, such that its left child represents the interval $[l, m)$ and its right child represents the interval $[m, r)$.
  - It can be noted that the shape of the segment tree depends on the choice of the division point m for each non-leaf node.
- In the problem of interval summation, for the sequence $a_0, a_1, \ldots, a_{n-1}$, each node $[l, r)$ of the segment tree maintains the value of $(a_l + a_{l+1} + \cdots + a_{r-1})$.

Little J has an array of length N, $A_0, A_1, \ldots, A_{N-1}$, and he does not know any number in A, but he has a segment tree that maintains the interval sum of A. The segment tree is given by $X_1, X_2, \ldots, X_{N-1}$, where X_i is the division point of the i-th non-leaf node in the preorder traversal of the segment tree. For example, if $N = 5, X = [2, 1, 4, 3]$, then the nodes contained in the segment tree's preorder traversal are $[0, 5), [0, 2), [0, 1), [1, 2), [2, 5), [2, 4), [2, 3), [3, 4), [4, 5)$.

Little J has M intervals $[L_1, R_1), [L_2, R_2), \ldots, [L_M, R_M)$, and he wants to know, among all the subsets of the 2^{2N-1} segment tree nodes, how many subsets S satisfy the following condition:

- If the values maintained by all nodes in S are known, then the sum of each interval $[L_i, R_i)$ can be uniquely determined.

For example, if $[0, 1), [1, 2)$ are known, then the sum of $[0, 2)$ can be determined; conversely, if $[0, 1), [0, 2)$ are known, then the sum of $[1, 2)$ can also be determined. However, if only $[0, 2), [2, 4)$ are known, then the sum of $[0, 3)$ or $[1, 2)$ cannot be determined.

Since the answer can be very large, you need to output the answer modulo $998,244,353$.

Input

The first line of the input contains two integers N and M ($2 \le N \le 2 \times 10^5$, $1 \le M \le \min\left\{\frac{N(N+1)}{2}, 2 \times 10^5\right\}$).

The second line of the input contains $N - 1$ integers $X_1, X_2, \cdots, X_{N-1}$ ($1 \le X_i \le N - 1$, the sequence X_i describes a valid segment tree).

The following M lines describes the values of L_i and R_i ($0 \le L_i < R_i \le N$). The i-th line of these lines contains two integers L_i and R_i.

It is guaranteed that $(L_i, R_i) \neq (L_j, R_j)$ for all $i \neq j$.

Output

Output a single line with a single integer, indicating the answer modulo $998\,244\,353$.

Examples

standard input	standard output
2 1 1 0 2	5
2 1 1 1 2	5
5 2 2 1 4 3 1 3 2 5	193
10 10 5 2 1 3 4 7 6 8 9 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10	70848

Problem E. Game of Strings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Little P and Little B like to play games, and they found Little Skip. Little Skip introduced them to the following game:

- There is a string S containing lowercase letters, and at the start of the game, it is given by skip as a string S_0. The game scores Little P and Little B, with their initial scores both being 0.
- Little P and Little B take turns operating on this string, with Little P going first. Each player can perform the following operation during their turn:
  - Choose a non-empty prefix of S (which can be S itself), earn a score equal to the number of occurrences of this prefix in S, and then remove this prefix from S.
- If S becomes empty after a certain operation, the game ends.

To help you better understand the rules of the game, consider the following example:

- Initially, $S_0 = ababa$;
- Little P chooses the prefix a of $ababa$, earning 3 points, and S becomes $baba$;
- Little B chooses the prefix ba of $baba$, earning 2 points, and S becomes ba;
- Little P chooses ba, earning 1 point, and the string becomes empty, ending the game. Finally, Little P earns 4 points, and Little B earns 2 points.

Little P aims to maximize the score of Little P minus the score of Little B, while Little B aims to minimize this value. They want to know, assuming both sides are extremely smart, what the value of the score of Little P minus the score of Little B will be.

Input

The first line of the input contains a string S_0 made up of lowercase letters. It is guaranteed that $1 \le |S_0| \le 10^6$.

Output

Output a single line contains a single integer, representing under the premise of both sides being extremely smart, the value of the game's end score difference between Little P and Little B.

Examples

standard input	standard output
`ababa`	`2`
`letitrotwillwinworldfinals`	`4`

Problem F. Pangu and Stones

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

In Chinese mythology, Pangu is the first living being and the creator of the sky and the earth. He woke up from an egg and split the egg into two parts: the sky and the earth.

At the beginning, there was no mountain on the earth, only stones all over the land.

There were N piles of stones, numbered from 1 to N. Pangu wanted to merge all of them into one pile to build a great mountain. If the sum of stones of some piles was S, Pangu would need S seconds to pile them into one pile, and there would be S stones in the new pile.

Unfortunately, every time Pangu could only merge successive piles into one pile. And the number of piles he merged shouldn't be less than L or greater than R.

Pangu wanted to finish this as soon as possible.

Can you help him? If there was no solution, you should answer `0`.

Input

The first line of each case contains three integers N, L, and R as above mentioned ($2 \le N \le 100$, $2 \le L \le R \le N$).

The second line of each case contains N integers $a_1, a_2, \cdots, a_N$ ($1 \le a_i \le 1\,000$), indicating the number of stones of pile 1, pile 2, ..., pile N.

Output

You should output the minimum time(in seconds) Pangu had to take. If it was impossible for Pangu to do his job, you should output `0`.

Examples

standard input	standard output
`3 2 2` `1 2 3`	`9`
`3 2 3` `1 2 3`	`6`
`4 3 3` `1 2 3 4`	`0`

Problem G. Mirrors

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given an integer sequence of length n, $h_1, h_2, \ldots, h_n$, find the number of pairs (u, v) that satisfy all of the following conditions:

- $1 \le u < v \le n$, and u, v are integers;
- There exists a **positive real number** L and a sequence of length $(v - u + 1)$, $r_u, r_{u+1}, \cdots, r_v$, satisfying all of the following conditions:
  - For all $u \le i \le v$, let $h'_i = 2L - h_i$, then $r_i \in \{h_i, h'_i\}$;
    - Specifically, when $h_i = h'_i$, then $r_i = h_i$;
  - For all $u \le i < v$, $r_i < r_{i+1}$.

Input

The first line of the input contains a positive integer n ($2 \le n \le 5 \times 10^5$), representing the number of pillars.

The second line contains n positive integers $h_1, h_2, \ldots, h_n$ ($1 \le h_i \le 10^{12}$), representing the heights of the pillars.

Output

Output a single line contains a single integer, representing the number of pairs (u, v).

Example

standard input	standard output
4 1 3 2 4	6

Problem H. Game of Votes

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 megabytes

There are n people, numbered from 1 to n. The i-th person, for $2 \le i \le n$, dislikes one person f_i ($1 \le f_i < i$), while the first person dislikes no one.

One day, the n people participate in a voting game, which consists of n rounds. At the start of the game, no one has been voted out. In each round of the game, the following process occurs:

1. Each person i who has not been voted out starts with a_i votes.
2. Then, for each person i who has not been voted out and dislikes someone whose disliked person f_i has not been voted out, i will cast b_i votes for f_i.
3. Finally, the person among those not yet voted out who has the highest number of votes is voted out. If there are multiple people with the highest number of votes, the one with the largest number is voted out.

Votes are tallied independently in each of the n rounds of the game.

Before the game starts, q events occur, which are of the following two types:

1. Given p, x, y, modify (a_p, b_p) to (x, y);
2. Xiao Ming wants to know, given two people c, d, if a game were to be played at this moment, which of the two would be voted out first.

As Xiao Ming's friend, can you help him?

Input

The first line contains two positive integers n and q ($1 \le n, q \le 2 \times 10^5$), representing the number of people and the number of events that occurred.

The second line contains $(n-1)$ integers $f_2, f_3, \cdots, f_n$ ($1 \le f_i < i$).

The third line contains n integers $a_1, a_2, \cdots, a_n$ ($0 \le a_i \le 10^8$).

The fourth line contains n integers $b_1, b_2, \cdots, b_n$ ($0 \le b_i \le 10^8$).

The following q lines, each line contains three or four integers describing an event. The first positive integer *op* indicates the type of event.

- If $op = 1$, then the next three integers p, x, y follow ($0 \le x, y \le 10^8$, $1 \le p \le n$), indicating that (a_p, b_p) is modified to (x, y).
- If $op = 2$, then the next two positive integers c, d follow ($1 \le c, d \le n$, $c \ne d$), and you need to determine if a game were played at this moment, who among c and d would be voted out first.

Output

For each event with $op = 2$, output one line with a single character. Output "0" if c is voted out first, otherwise output "1".

Example

standard input	standard output
5 8 1 2 3 2 1 3 2 1 0 0 4 1 0 0 2 1 3 1 1 0 3 2 2 5 1 1 2 2 2 4 3 2 5 4 2 5 1 2 2 1	0 0 1 1 1 1

Problem I. Cats and Fish

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

There are many homeless cats in PKU campus. They are all happy because the students in the cat club of PKU take good care of them. Li lei is one of the members of the cat club. He loves those cats very much. Last week, he won a scholarship and he wanted to share his pleasure with cats. So he bought some really tasty fish to feed them, and watched them eating with great pleasure. At the same time, he found an interesting question:

There are m fish and n cats, and it takes c_i minutes for the i-th cat to eat out one fish. A cat starts to eat another fish (if it can get one) immediately after it has finished one fish. A cat never shares its fish with other cats. When there are not enough fish left, the cat which eats quicker has higher priority to get a fish than the cat which eats slower. All cats start eating at the same time. Li Lei wanted to know, after x minutes, how many fish would be left.

Input

The first line of the input contains three integers m, n, and x ($0 < m \le 5\,000$, $1 \le n \le 100$, $0 \le x \le 1\,000$).

The second line contains n integers $c_1, c_2, \cdots, c_n$, c_i means that it takes the ith cat c_i minutes to eat out a fish ($1 \le c_i \le 2\,000$).

Output

Output a single line contains two integers p and q, meaning that there are p complete fish(whole fish) and q incomplete fish left after x minutes.

Examples

standard input	standard output
2 1 1 1	1 0
8 3 5 1 3 4	0 1
4 5 1 5 4 3 2 1	0 3

Problem J. Shell Sort

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

Shell sort is an excellent sorting algorithm, which can be regarded as a kind of group insertion sort. Next, we will briefly introduce this algorithm:

Assume we need to sort an array $A_{0\ldots n-1}$ of length n in ascending order. First, we need to determine an integer m and a decreasing sequence d of length m with the last number being 1 as the step sequence, and then perform m rounds of operations.

For the i-th round of operation, let $t = d_i$, and then consider dividing A into as evenly as possible t groups. Specifically, we choose to group those positions that have the same modulo t, and then perform insertion sort within each group.

```
void insert_sort(vector<int> &v) {
        int n = v.size();
        for (int i = 0; i < n; i++) {
                for (int j = i; j && v[j] < v[j - 1]; j--){
                        swap(v[j], v[j - 1]);
                        swap_count++;
                }
        }
}

void work() {
        for (int i = 0; i < t; i++) {
                vector<int> v;
                for (int j = i; j < n; j += t) v.push_back(A[j]);
                insert_sort(v);
                for (int j = i, k = 0; j < n; j += t, k++) A[j] = v[k];
        }
}

void shell_sort() {
        swap_count = 0;
        for (int i = 1; i <= m; i++) {
                t = d[i];
                work();
        }
}
```

The `work` function represents one round of operation with parameter $t = d[i]$.

Given two integers n, m, and a step sequence d of length m, you need to calculate the maximum number of array element swaps, that is, the maximum value of the variable `swap_count`, after running the `shell_sort` function for all permutations of lengths n. Also, you need to give the number of permutations that can achieve this maximum value.

The answers need to be modulo $10^9 + 7$.

Input

The first line of the input contains two integers n and m ($2 \le n \le 30$, $1 \le m \le 10$).

The second line of the input contains m integers, where the i-th integer represents d_i. It is guaranteed that $1 \le d_i \le 10$, $d_m = 1$, and $d_i > d_{i+1}$ for all $1 \le i \le m-1$.

Output

Output a single line contains two integers, representing the maximum number of swaps and the number of permutations that achieve this maximum number of swaps, respectively. The answers need to be modulo $10^9 + 7$.

Example

standard input	standard output
5 2 2 1	7 2

Problem K. Stacks

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

There are n stacks, numbered from 1 to n. There are also m operations, which come in three types:

- 1 l r x y, meaning to push x copies of y onto each stack numbered within the interval $[l, r]$.
- 2 l r w, meaning to perform the pop operation w times on each stack numbered within the interval $[l, r]$. Here, the pop operation means that if the stack is empty, do nothing; otherwise, pop the top element.
- 3 k p q, meaning to query the sum of elements from the p-th to the q-th, starting from the bottom of the stack numbered k. If the i-th element does not exist in the stack, it is considered to be 0.

Help me to process all the m operations.

Input

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

The following m lines describe an operation each, in the form of:

- 1 l r x y ($1 \le l \le r \le n$, $1 \le x, y \le 10^5$), to push x copies of y onto each stack numbered within the interval $[l, r]$.
- 2 l r w ($1 \le l \le r \le n$, $1 \le w \le 10^{10}$), to perform the pop operation w times on each stack numbered within the interval $[l, r]$.
- 3 k p q ($1 \le k \le n$, $1 \le p \le q \le 10^{10}$), to query the sum of elements from the p-th to the q-th, starting from the bottom of the stack numbered k.

Output

For each query, output a single line contains a single integer, representing the answer.

Example

standard input	standard output
4 8 1 1 3 3 2 1 2 4 2 1 3 1 2 4 3 2 2 4 2 2 3 1 2 1 2 2 3 1 1 1 3 2 2 3	4 5 2 2

Problem L. Caught in the Middle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given a string s of length n containing only the characters `L` and `R`. Alice and Bob are planning to play a game using the string.

Alice and Bob will take turns operating on string s, with Alice going first.

In each operation, assuming the current remaining string is s. If s is an empty string, then the operator loses the game. Otherwise, the operator can choose an integer i from $1, 2, \cdots, |s|$. If $s_i = \texttt{L}$, then the remaining string after the operation is $s_1 s_2 \cdots s_{i-1}$; if $s_i = \texttt{R}$, then the remaining string after the operation is $s_{i+1} s_{i+2} \cdots s_{|s|}$.

Both are extremely intelligent, so they will always adopt the best strategy. And you, an ordinary onlooker participating in ~~PKUWC~~ Universal Cup, want to know the winner of this game.

Input

There are multiple test cases in a single test file.

The first line of the input contains a single integer T, indicating the number of test cases. For each test case:

The first line of the input contains a single integer n ($1 \le n \le 10^6$).

The second line of the input contains a string s of length n that only contains `L` and `R`, representing the initial string of the game.

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

Output

For each test case, output a single line `Alice` or `Bob`, indicating the winner of the game.

Example

standard input	standard output
3 5 LRLLR 6 RLRLRL 1 L	Alice Bob Alice