The 2024 CCPC National Invitational Contest (Changchun)

The 17th Jilin Provincial Collegiate Programming Contest

Contest Session

May 18, 2024

东北师范大学
NORTHEAST NORMAL UNIVERSITY

Problem List

A	Eminor Array
B	Dfs Order 0.5
C	Fibonacci Sum
D	Parallel Lines
E	Connected Components
F	Best Player
G	Platform Game
H	Games on the Ads 2: Painting
I	The Easiest Problem
J	Lone Trail
K	String Divide II
L	Recharge

This problem set should contain 12 (twelve) problems on 17 (seventeen) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Problem A. Eminor Array

Input file: `standard input`
Output file: `standard output`

Gew is looking for "Eminor" sequences $[a_1, a_2, \ldots, a_m]$ which have the following properties:

- The sequence is not empty ($m \geq 1$);
- $1 \leq a_i \leq 2^n - 1$;
- The array is strictly increasing ($a_i < a_{i+1}$, for each $i \leq m - 1$);
- There are no three consecutive elements with their bitwise XOR equal to zero ($a_i \oplus a_{i+1} \neq a_{i+2}$, for each $i \leq m - 2$. Here $\oplus$ denotes the bitwise XOR operation).

Now, Gew is curious about how many "Eminor" sequences there are. Since there may be a large number of "Eminor" sequences, you only need to output the answer modulo 998 244 353.

Input

The input contains a single integer n ($1 \leq n \leq 10^6$).

Output

Output a single integer, denoting the number of "Eminor" sequences, modulo 998 244 353.

Examples

standard input	standard output
1	1
2	6
3	91

Note

For the second testcase, the following are 6 possible "Eminor" sequences.

- $[1]$
- $[2]$
- $[3]$
- $[1, 2]$
- $[1, 3]$
- $[2, 3]$

Irrelevent: Originating from an incorrect problem reading https://codeforces.com/gym/102956/problem/C XD

Problem B. Dfs Order 0.5

Input file:	`standard input`
Output file:	`standard output`

Shuishui has a rooted tree of n vertices numbered from 1 to n, where vertex 1 is the root. The value of the i-th vertex is a_i.

Now we start the depth-first search at the root. Because sons of a node can be iterated in arbitrary order, multiple possible depth-first orders exist. We define the value of a depth-first order to be adding up the value of vertices which appear in the depth-first order at an even index. Shuishui wonders among all possible depth-first orders of the given tree, what the maximum value is.

Following is a pseudo-code for the depth-first search on a rooted tree. After calling MAIN(), `dfs_order` will be the depth-first search order.

Algorithm 1 An implementation of depth-first search

procedure DFS(vertex x)
  Append x to the end of `dfs_order`
  for each son y of x **do** ▷ Sons can be iterated in arbitrary order.
    DFS(y)
  end for
end procedure
procedure MAIN()
  Let `dfs_order` be a global variable
  `dfs_order` $\leftarrow$ empty list
  DFS(1)
end procedure

Following is a pseudo-code for calculating the value of a depth-first order. Calling CALC(`dfs_order`) will return the value of it.

Algorithm 2 An implementation of calculating the value of a depth-first order

procedure CALC(`dfs_order`) ▷ `dfs_order` is a 1-based array of length n
  $s \leftarrow 0$
  $p \leftarrow 2$
  while $p \le n$ **do**
    $u \leftarrow \texttt{dfs\_order}_p$
    $s \leftarrow s + a_u$
    $p \leftarrow p + 2$
  end while
  return s
end procedure

Input

The input contains multiple testcases.

The first line contains a single integer t ($1 \le t \le 2 \times 10^5$), denoting the number of testcases.

For each testcase:

The first line contains a single integer n ($1 \le n \le 2 \times 10^5$), denoting the number of vertices in the tree.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), denoting the values of these vertices.

Each of the next $n-1$ lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n, u_i \ne v_i$), denoting an edge of the tree.

It is guaranteed that the given edges form a tree.

It is guaranteed that the sum of n over all test cases does not exceed 2×10^5.

Output

For each testcase, output a single integer in one line, denoting the answer.

Example

standard input	standard output
3 2 1 2 1 2 4 1 1 222 222 1 2 1 3 2 4 6 1 5 4 6 1 1 6 1 4 5 4 2 1 4 1 3	2 444 15

Note

In the first testcase of the example, the `dfs_order` is unique, which is $[1, 2]$, and its value is $a_2 = 2$. Note that the `dfs_order` is 1-based.

In the second testcase of the example, the `dfs_order` with the maximum value is $[1, 3, 2, 4]$, and its value is $a_3 + a_4 = 444$.

Problem C. Fibonacci Sum

Input file:	`standard input`
Output file:	`standard output`

Let $f(i)$ denote the i-th number in the famous Fibonacci Sequence. Formally:

$$f(i) = \begin{cases} 1 & \text{if } i \le 2, \\ f(i-1) + f(i-2) & \text{if } i > 2. \end{cases}$$

Let $g(x)$ denote the count of 1's in the binary representation of number x. For example, $g(5) = g(101_{(2)}) = 2$, $g(15) = g(1111_{(2)}) = 4$.

Your task is to calculate the following value:

$$\sum_{i=1}^{n} f\left(g(i)\right) \bmod 10^9 + 7$$

Input

The input contains a string s ($1 \le |s| \le 10^7$) consisting of only '`0`' or '`1`', denoting the number n **in binary form**.

It's guaranteed that the input contains no leading 0's.

Output

The output contains a single integer, denoting the answer modulo $10^9 + 7$.

Examples

standard input	standard output
`1`	`1`
`10`	`2`
`11`	`3`

Note

It can be calculated that $f\left(g(1_{(2)})\right) = 1$, $f\left(g(10_{(2)})\right) = 1$, $f\left(g(11_{(2)})\right) = 1$.

Problem D. Parallel Lines

Input file: `standard input`
Output file: `standard output`

Once upon a time, there were k parallel lines in a two-dimensional plane and n points on these lines. It is known that there were **at least two** points on each line.

Now you are given these n points, and your task is to find those k parallel lines.

Input

The first line contains two integers n, k ($2 \le n \le 10^4, 1 \le k \le \min(50, \frac{n}{2})$), denoting the number of points and parallel lines.

The i-th of the next n lines contains two integers x_i, y_i ($1 \le x_i, y_i \le 10^9$), denoting the coordinates of the i-th point.

It is guaranteed that n points are pairwise distinct (i.e. $\forall 1 \le i < j \le n$, either $x_i \ne x_j$ or $y_i \ne y_j$ holds).

Output

The output contains k lines.

In the i-th line, first output an integer m_i ($2 \le m_i \le n$), denoting the number of points on the i-th parallel line. Then output m_i integers $x_1, x_2, \ldots, x_{m_i}$, denoting the indices of points on the i-th line.

Your output should satisfy that each point appears and only appears on one line, and k lines are parallel and different.

It is guaranteed there is a valid solution to distribute the n points onto k parallel lines. If multiple solutions exist, output any of them.

Example

standard input	standard output
4 2 1 3 2 5 4 7 5 9	2 3 4 2 1 2

Problem E. Connected Components

Input file: `standard input`
Output file: `standard output`

There are n kingdoms on the continent of Eminor numbered from 1 to n. Each kingdom has two attribute values represented by a_i and b_i.

Kingdom i and j ($i < j$) are connected by an undirected road when $a_i - a_j \leq i - j \leq b_i - b_j$, or $a_j - a_i \leq j - i \leq b_j - b_i$.

Gew wants to know how many connected components are in this continent.

Input

The first line contains a single integer n ($1 \leq n \leq 10^6$).

The i-th of the next n lines contains two integers a_i, b_i ($-10^9 \leq a_i, b_i \leq 10^9$)

Output

Output a single integer, denoting the number of connected components.

Examples

standard input	standard output
5 1 -4 3 -2 5 0 7 2 9 4	5
2 1 2 2 1	1
5 5 4 3 3 2 5 3 4 4 5	2

Problem F. Best Player

Input file: `standard input`
Output file: `standard output`

The 17th Culinary Combat Professional Contest (CCPC) has ended, and the event organizers will select the best player of this competition.

There are n players numbered from 1 to n who participated in this competition and played a total of m 1 vs 1 duels:

- The opponents in the i-th duel are players a_i and b_i;
- Each duel consists of a first half and a second half:
  - In the first half of the duel, the score of player a_i is x_i, and the score of player b_i is y_i;
  - In the second half of the duel, the exact scores of the two players a_i and b_i are uncertain now, but the sum of the scores is z_i;
  - A player's score in a duel equals to the sum of the first half score and the second half score. In other words, in the i-th duel, the possible scores for a_i and b_i are $x_i + p_i$ and $y_i + q_i$ respectively, where $0 \le p_i, q_i \le z_i, p_i + q_i = z_i$.

Note that all scores are non-negative integers, and each player has participated in at least one duel.

A player's final score is: the **maximum** score achieved in duels he participated. A player will win the best player award if and only if his final score is **strictly** larger than any other player's final score.

Due to the uncertainty of the scores in the second half of the m duels, the best player may also be different. Please find all the players who could possibly be the best player.

Input

The input contains multiple testcases.

The first line contains a single integer t ($1 \le t \le 10^5$), denoting the number of testcases.

For each testcase:

The first line contains two integers n, m ($2 \le n \le 2 \times 10^5$, $1 \le m \le 2 \times 10^5$), denoting the number of players and duels.

The i-th of the next m lines contains 5 integers a_i, b_i, x_i, y_i, z_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$, $0 \le x_i, y_i, z_i \le 10^5$), of which meanings are described above. It is guaranteed that each player participates in at least one duel.

It is guaranteed that neither the sum of n nor the sum of m over all test cases exceeds 2×10^5.

Output

For each testcase:

The first line output a single integer k, denoting the number of players who could possibly be the best player.

The second line output k integers listed in **ascending order**, denoting the player numbers who could possibly be the best player. Specially, when $k = 0$ either output an empty line or not will be considered correct.

Example

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

Note

In the first testcase of the example, all three players could possibly be the best player. Player 1 can be the best player only in the following case:

- In the first duel, player 1 scores $2 + 6 = 8$, player 2 scores $3 + 0 = 3$;
- In the second duel, player 2 scores $6 + 1 = 7$, player 3 scores $6 + 1 = 7$.

In this case, player 1's final score is 8, player 2's final score is $\max(3, 7) = 7$, and player 3's final score is 7, so player 1 is the best player.

Problem G. Platform Game

Input file: `standard input`
Output file: `standard output`

Haha is playing a platform game.

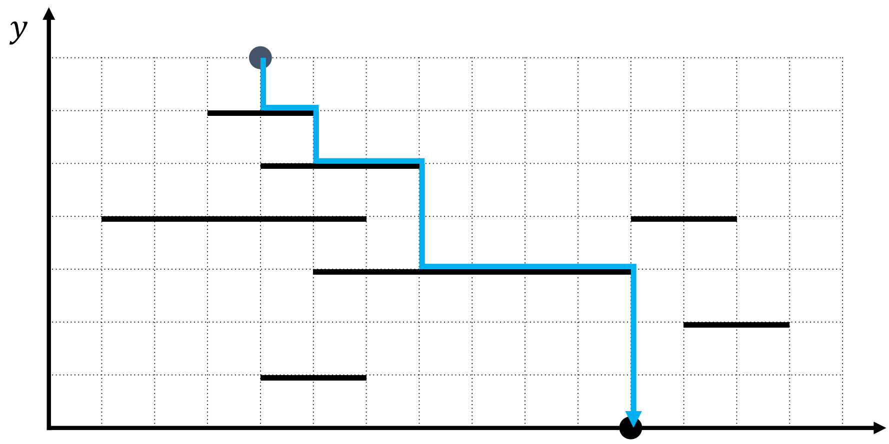
One level of the game is about a robot falling among the platforms on a 2D map. Let's consider the 2D map as a coordinate system where the ground is the x-axis. Above the ground, there are n platforms numbered from 1 to n. All platforms are parallel to the ground, and the i-th platform can be viewed as a segment extending from (l_i, y_i) to (r_i, y_i), where the thickness of the platform is negligible. In the game, a robot initially spawns at point (s_x, s_y). It'll move according to the following rules:

- If the robot is on a platform, it will move toward the right until it leaves the platform. Note that we consider upon leaving the platform i, the x-coordinate of the robot will be r_i (i.e. the right-border of the platform).
- If the robot is not on any platform, it'll keep falling vertically until it either lands on a platform or finally lands on the ground.

Note that the robot can land on the i-th platform if and only if its x-coordinate is between l_i and r_i **exclusively** (i.e. $l_i < x < r_i$).

It's guaranteed that the robot is not on any platform initially.

Now, Haha is curious about at which position the robot will land on the ground.



An example of how the robot will move among the given platforms.

Input

The input contains multiple testcases.

The first line contains a single integer t ($1 \le t \le 2 \times 10^5$), denoting the number of testcases.

For each testcase:

The first line contains a single integer n ($1 \le n \le 2 \times 10^5$), denoting the number of platforms.

The i-th of the following n lines contains three integers l_i, r_i, y_i ($1 \le l_i < r_i \le 10^9$, $1 \le y_i \le 10^9$), denoting the position of the i-th platform.

The last line contains two integers s_x, s_y ($1 \le s_x, s_y \le 10^9$), denoting the initial position of the robot.

It is guaranteed that no two platforms overlap with each other. Two platforms overlap when they share at least one common point.

It is guaranteed that the sum of n over all test cases does not exceed 2×10^5.

Output

For each testcase, output a single integer in one line, denoting the answer.

Example

standard input	standard output
2 7 4 6 1 12 14 2 5 11 3 1 6 4 11 13 4 4 7 5 3 5 6 4 7 1 2 4 2 2 5	11 2

Note

The first testcase of the example is illustrated as the picture above.

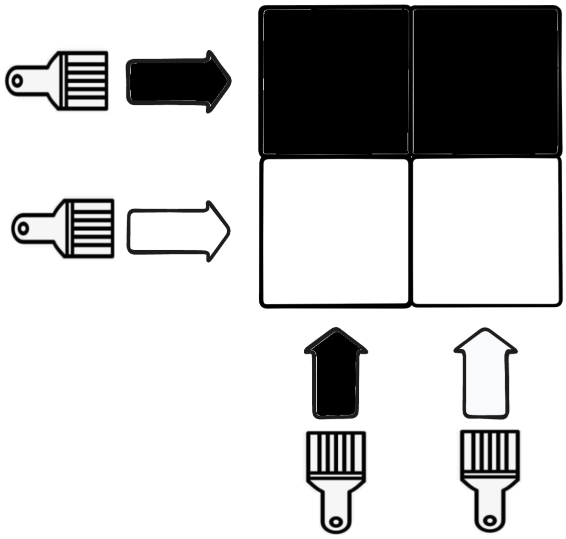
For the second testcase of the example, the robot will keep dropping without landing on any platform, until finally landing on the ground with the x-coordinate unchanged.

Problem H. Games on the Ads 2: Painting

Input file: `standard input`
Output file: `standard output`

Tired of those fake games on advertisements? Now try this one!

You are given an $n \times n$ grid. For each row and each column, there's a painting brush associated with it, totally $2n$ brushes. When you choose to use a brush, it will paint every cell in its corresponding row (or column) to the brush's designated color. In the game, there are n colors numbered from 1 to n. The designated colors of brushes corresponding to the rows form a permutation of $[1, 2, \ldots, n]$. Similarly, the colors of brushes corresponding to the columns form a permutation of $[1, 2, \ldots, n]$.



Brushes and the goal pattern of the example.

The grid is initially empty with no color in any cell. The goal of the game is to paint the grid into the given pattern.

You are not satisfied with determining whether the goal can be achieved. You want to know, if you use each brush to paint exactly once, out of all $(2n)!$ possibilities, how many of them result in the goal pattern? Since the answer could be large, output it modulo $998\,244\,353$.

Input

The first line contains a single integer n ($1 \le n \le 20$), denoting the size of the grid.

The second line contains n integers $p_1, p_2, \cdots, p_n$ ($1 \le p_i \le n$), where p_i denotes the color of the brush corresponding to the i-th row. It is guaranteed that p is a permutation of $[1, 2, \ldots, n]$.

The third line contains n integers $q_1, q_2, \cdots, q_n$ ($1 \le q_i \le n$), where q_i denotes the color of the brush corresponding to the i-th column. It is guaranteed that q is a permutation of $[1, 2, \ldots, n]$.

The following n lines each contains n integers, denoting an $n \times n$ matrix c ($1 \le c_{i,j} \le n$), where $c_{i,j}$ denotes the color of the cell in the i-th row and j-th column of the goal pattern.

Output

Output a single integer, denoting the number of solutions modulo $998\,244\,353$.

Example

standard input	standard output
2 1 2 1 2 1 1 2 2	6

Note

For the example, if we number the 4 brushes from top to bottom and then from left to right, the 6 ways of the order to achieve the goal pattern are as follows.

- $3, 2, 4, 1$
- $3, 4, 1, 2$
- $3, 4, 2, 1$
- $4, 1, 3, 2$
- $4, 3, 1, 2$
- $4, 3, 2, 1$

Problem I. The Easiest Problem

Input file:	`standard input`
Output file:	`standard output`

During class, the teacher often says “Scan the QR code to sign in now.”

Now, Gew wants to ask you how many **lowercase** English letters there are in this sentence.

Input

The input contains only one line, containing the sentence “Scan the QR code to sign in now.”

Output

Output an integer, denoting the answer to the question.

Example

standard input	standard output
`Scan the QR code to sign in now.`	`-1`

Note

The output of the example is not correct. It’s you who need to figure out the answer, and get a correct verdict.

Problem J. Lone Trail

Input file: `standard input`
Output file: `standard output`

If for millennia to come, our descendants have sailed along the stars, they shall greatly exalt her name.

In case of uncovering the "fake sky" of Terra, Kristen has constructed a space station.

The space station can be represented as a tree consisting of n nodes, numbered from 1 to n. As the launching process of the space station kicked off, the energy of the station is increasing. Initially, there are b_i units of energy at node i. At the start of each day, the energy at node i will increase by a_i.

Pure water is essential for life. As a water elf, Muelsyse needs to transport pure water to each node in the space station.

You need to process k operations, which are of two types:

1. At the end of day x, the station's mode is changed. Consequently, two **directly connected** nodes u and v will be chosen. a_u (i.e. daily energy increase of node u) will decrease by w, and a_v (i.e. daily energy increase of node v) will increase by w. It is guaranteed that there exists an edge connecting u and v, and $a_u \geq w$ before the operation is performed.
2. At the day x, Muelsyse needs to transport pure water to each node. Specifically, Muelsyse will select a node as the source of water, denoted as r. It costs her $dis_u \times c_u$ flowing shapes to transport pure water from the source of water to node u, where dis_u denotes the number of edges on the path from node u to r, and c_u denotes the energy units at node u. Please help her choose a suitable node as water source, so that she spends the least number of flowing shapes.

It's guaranteed that x is increasing among these operations. Formally, let x_i be the value of x in the i-th operation, then $x_i < x_{i+1}$ holds for all $1 \leq i \leq k-1$.

Input

The first line contains two integers n, k ($1 \leq n, k \leq 10^5$), denoting the number of nodes in the space station and the number of operations.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 1000$), where a_i denotes the daily energy units increase of node i.

The third line contains n integers $b_1, b_2, \ldots, b_n$ ($0 \leq b_i \leq 1000$), where b_i denotes the initial energy units of node i.

The next $n-1$ lines, each contains two integers u, v ($1 \leq u, v \leq n, u \neq v$), denoting an edge in the tree.

Next k lines contain descriptions of operations. The i-th operation has one of two types:

- "1 x_i u_i v_i w_i" ($1 \leq x_i \leq 10^9$, $1 \leq u_i, v_i \leq n$), of which meanings are described above. It is guaranteed that there exists an edge connecting u and v, and $a_{u_i} \geq w_i$ before the operation is performed.
- "2 x_i" ($1 \leq x_i \leq 10^9$), of which meaning is described above.

It's guaranteed that x is increasing among these operations.

Output

For each operation type 2, output a single integer, denoting the minimum flowing shapes Muelsyse spends.

Example

standard input	standard output
5 10 1 1 4 5 1 4 1 9 1 9 1 2 2 3 2 4 1 5 2 1 1 2 3 2 3 1 3 4 2 4 1 4 2 1 8 2 5 1 6 1 5 7 2 7 2 8 2 9 2 10	44 83 116 134 146 158

Problem K. String Divide II

Input file: `standard input`
Output file: `standard output`

Given a string $s = s_1 s_2 \dots s_n$ of length n, consisting of only lowercase English letters. For convenience, we define string $s_{[l,r]} = s_l s_{l+1} \dots s_r$, which is the substring of s from index l to r.

Given an integer k, you're curious about those substrings of s which are composed by concatenating k identical strings together. Please find out the longest length of those substrings.

Formally, you should find a set of intervals $\{[l_1, r_1], [l_2, r_2], \dots, [l_k, r_k]\}$ which satisfies the following two conditions:

- For each i $(1 \le i \le k)$, $1 \le l_i \le r_i \le n$;
- For each i $(1 \le i < k)$, $r_i + 1 = l_{i+1}$, and $s_{[l_i, r_i]} = s_{[l_{i+1}, r_{i+1}]}$.

You need to maximize $r_k - l_1 + 1$.

Input

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

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

Output

Output a single integer, denoting the answer. Specially, if there are no such substrings, output 0.

Examples

standard input	standard output
5 3 bacbc	0
7 2 ababbba	4

Note

For the second example, the set we choose is $\{[1, 2], [3, 4]\}$. It can be proved that this is the longest substring satisfying the condition.

Problem L. Recharge

Input file: `standard input`
Output file: `standard output`

The Binding of Isaac is a randomly generated action RPG shooter with heavy roguelike elements.

During the game, the player can pick up different items and cause different effects after being used, which adds a lot to the gameplay of this game. *Activated items* are a special category of items. Each activated item can be used once it is fully charged, thus a charge bar is used to denote the current charging progress. To charge an activated item, the player needs to clear the rooms: a small room fills 1 unit of the charge bar, while a large room fills 2 units of the charge bar. Specifically, for an activated item with only 1 unit uncharged, clearing a large room has the same impact as clearing a small room. Using the activated item will **empty** the charge bar.

The most powerful activated item in the game, Void, has a charge bar size of 6 units, with 5 units charged.

Now, Shuishui is holding an activated item with a charge bar size of k units. Initially, the charge bar is empty, and there are x uncleared small rooms and y uncleared large rooms on the map. Shuishui can choose to clear the uncleared rooms in arbitrary order. Please calculate the **maximum** number of times Shuishui uses the item.

Input

The input contains multiple testcases.

The first line contains an integer t ($1 \le t \le 2 \times 10^5$), denoting the number of testcases.

For each testcase, only one line contains three integers k, x, y ($1 \le k \le 10^9$, $0 \le x, y \le 10^9$).

Output

For each testcase, output a single integer in one line, denoting the answer.

Example

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