Problem A. Adding Integers

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

You are given integers n and k.

For a positive integer q, $f(q)$ is defined as the sum of $\binom{n}{a_1} \cdot \binom{a_1}{a_2} \cdot \ldots \cdot \binom{a_{q-1}}{a_q}$ for all integer sequences $(a_1, a_2, \ldots, a_q)$ that satisfy the condition $n \geq a_1 \geq a_2 \geq \ldots \geq a_q \geq 0$.

Calculate the value $\sum_{q=1}^{k} f(q)$ modulo $998\,244\,353$.

Here, $\binom{A}{B}$ is the binomial coefficient: the number of ways to select B distinct items from A distinct items.

Input

The first line of input contains an integer t: the number of test cases ($1 \leq t \leq 10^5$).

Each of the following t lines contains two integers: n and k ($0 \leq n \leq 10^9$; $1 \leq k \leq 2 \cdot 10^5$).

The total sum of k over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print the value of the sum modulo $998\,244\,353$.

Example

standard input	*standard output*
4 2 2 0 1 271 818 141 42	13 1 812506614 405709861

Problem B. Bottles

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

An immortal elf got $e + p + w$ bottles from the mage.

The first e bottles contain elixir. If the elf drinks it, she becomes immune to all poison effects, present or future.

The other p bottles, numbered from 1 to p, contain different poisons. Each poison has a delayed effect: poison bottle i has a delay of $t + i - 0.5$ days. If the elf drinks from poison bottle i, the respective delay passes, and she didn't yet drink any elixir in her life (doesn't matter if it's before or after drinking the poison), she dies. Poisons act independently: the delay for each poison is not changed by drinking other poisons.

The remaining w bottles contain water. When the elf drinks it, nothing happens.

At the same time every morning, the elf chooses one non-empty bottle with equal probability and drinks it. If all bottles are empty, she does nothing.

Find the probability that the elf will be alive $10^{10^{10}}$ days after the first day she starts drinking bottles. Remember that the elf is immortal, so she won't die from anything other than poison.

Input

The first line of input contains four integers: e, p, w, and t ($1 \leq e, p, w, t \leq 10^5$).

Output

It can be proven that the probability is a rational number. Represent it as p/q where p and q are coprime integers, and print the integer $p \cdot q^{-1}$ modulo 998 244 353. You may assume that q will be coprime with 998 244 353.

Examples

standard input	*standard output*
`1 1 2 1`	`249561089`
`1 1 1 42`	`1`
`2 2 2 2`	`987152750`

Note

For the three examples, the answers in rational form are: 3/4, 1/1, and 83/90.

Problem C. Counting Orthogonal Pairs

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

A regular polygon with n vertices has n edges and $\frac{n(n-1)}{2} - n$ diagonals. Consider the set of all these items. It contains $\frac{n(n-1)}{2}$ line segments.

Calculate how many pairs of segments from this set satisfy the following conditions:

- the segments have a common endpoint (which is a vertex of the regular polygon),
- the segments are orthogonal.

Input

The first line of input contains an integer t: the number of test cases ($3 \le t \le 10^5$).

Each of the following t lines contains an integer n: the number of vertices in the regular polygon ($3 \le n \le 10^9$).

Output

For each test case, print the answer on a separate line.

Example

standard input	*standard output*
3 5 4 10836006	0 4 58709502180012

Problem D. Divine Tree

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

Consider a weighted tree with a gold or bronze coin placed at each vertex. Such a tree is called a *Divine Tree* if the following process is possible:

1. Zero or more times repeat the following action: select a pair of vertices that are directly connected by an edge, and swap the coins placed in those vertices.
2. Delete at most one edge from the tree. This operation may be done only once after all operations of type 1 are performed.
3. After the operation of type 2, the tree is divided into at most two trees, and for each resulting tree, the vertices contain the coins of the same metal.

The cost of each operation of type 1 is equal to the weight of the chosen edge. The cost of the Divine Tree is defined as the minimal total cost of all operations of type 1 required to transform the tree.

You are given a tree with n vertices, where n is odd. You may assume that the given tree is a Divine Tree. Let the i-th edge have weight w_i.

The tree grows in the following way: q growing events happen. In the j-th event, one of the edges e_j is chosen, and its weight is increased by d_j. The effect of growth is permanent.

Your task is to print the cost of the Divine Tree after each event.

Input

The first line of input contains an integer n: the number of vertices in the given Divine Tree ($3 \le n < 10^5$; n is odd).

The second line contains a string of length n consisting of capital letters `G` and `B`. If the i-th letter in the string is `G`, the vertex i initially contains a gold coin, and if it is `B`, the vertex contains a bronze coin.

Each of the following $n-1$ lines contains three integers, u_i, v_i, and w_i, which mean that the i-th edge connects vertices u_i and v_i and has weight w_i ($1 \le u_i, v_i \le n$; $u_i \ne v_i$; $0 \le w_i \le 10^5$). You may assume that the given graph is a Divine Tree.

Then follows a line containing an integer q: the number of events ($1 \le q \le 10^5$).

Each of the following q lines contains two integers e_j and d_j: the number of the growing edge and the weight increase, respectively ($1 \le e_j \le n-1$; $1 \le d_j \le 10^5$).

Output

Print q lines. On the j-th of these lines, print the cost of the Divine Tree after the j-th event.

Examples

standard input	*standard output*
5 BGGGG 1 3 0 2 1 0 5 2 0 2 4 0 5 2 1 1 3 4 4 3 10 1 2	0 1 1 3 5
5 GBBGB 3 2 0 2 1 0 1 4 0 1 5 1000 4 4 1 3 1 2 1 1 1	0 1 3 4
7 GGBBBBG 1 5 101 2 5 101 3 5 100 3 6 100 4 6 100 7 6 100 6 6 1 6 1 6 1 5 3 3 3 6 12345	301 302 303 303 306 711

Problem E. Experiments With Divine Trees

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

Consider a tree with a gold or bronze coin placed at each vertex. Such a tree is called a *Divine Tree* if the following process is possible:

1. Zero or more times repeat the following action: select a pair of vertices that are directly connected by an edge, and swap the coins placed in those vertices.
2. Delete at most one edge from the tree. This operation may be done only once after all operations of type 1 are performed.
3. After the operation of type 2, the tree is divided into at most two trees, and for each resulting tree, the vertices contain the coins of the same metal.

You are given a tree with n vertices without the coins placed at the vertices. There are 2^n ways to place coins such that exactly one coin is placed at each vertex. How many of them satisfy both of the following conditions?

- The tree is a Divine Tree.
- We can choose a leaf and remove it along with its coin, so that the new tree is a Divine Tree as well.

Because the answer may be too large, print it modulo $998\,244\,353$.

Input

The first line of input contains an integer n: the number of the vertices in the tree ($2 \le n \le 10^5$).

Each of the following $n-1$ lines contains two integers u_i and v_i and defines one edge ($1 \le u_i, v_i \le n$; $u_i \ne v_i$).

You may assume that the given graph is a tree.

Output

Print the answer modulo $998\,244\,353$.

Examples

standard input	*standard output*
3 1 3 3 2	8
4 1 2 2 3 2 4	10
7 1 2 1 4 1 5 1 6 2 3 2 7	84

Problem F. Fruit Tea

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

Appropriate temperature changes are essential for good fruit tea. Artemis has been taught a recipe for delicious tea.

The recipe is represented by a sequence of non-negative integers $a = a_0, a_1, a_2, \ldots, a_n, a_{n+1}$ of length $n+2$. When brewing the tea, the temperature at each moment i must be equal to a_i.

Raising the temperature is hard work. The cost of a recipe a is defined by $f(a) = \sum_{i=0}^{n} \max(0, a_{i+1} - a_i)$.

Artemis has forgotten the recipe she was taught. All she remembers is that $a_0 = a_{n+1} = 0$ and that the cost was k.

How many possible recipes satisfy these constraints? As this number may be very large, find it modulo 998 244 353.

Two recipes are different when there is a moment i ($0 \le i \le n+1$) such that the values of a_i in the two recipes are different.

Input

The first line of input contains two integers: n and k ($1 \le n \le 2 \cdot 10^5$; $0 \le k \le 2 \cdot 10^5$).

Output

Print the number of possible recipes modulo 998 244 353.

Examples

standard input	standard output
3 3	31
42 0	1
314 159	734464844

Problem G. Gold Coins

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

Consider a box containing a grid with r rows numbered from 1 to r and c columns numbered from 1 to c. Each square of the grid either is empty or contains a single gold coin.

The state of the box is denoted by a two-dimensional binary matrix s. The value $s_{x,y}$ is 0 if the cell in the x-th row and y-th column is empty, or 1 if it contains a single gold coin.

Consider the following method to distinguish between the coins:

- Let $a_{x,y}$ be defined as the sum of $s_{i,j}$ for all integer pairs (i, j) satisfying $i = x$ and $1 \leq j \leq y$.
- Let $b_{x,y}$ be defined as the sum of $s_{i,j}$ for all integer pairs (i, j) satisfying $1 \leq i \leq x$ and $j = y$.
- If the cell in the x-th row and y-th column contains a gold coin, label the coin with the pair $(a_{x,y}, b_{x,y})$.

This method could result in multiple gold coins having the same label, and the coins could not be distinguished. Therefore, we can add some gold coins before labeling them. More formally, we can perform the following operation zero or more times: select a pair (x, y) such that $s_{x,y} = 0$, and set $s_{x,y} \leftarrow 1$.

What is the minimum non-negative number of coins that should be added so that no two coins have the same label?

Input

The first line of input contains two integers: r and c ($1 \leq r, c \leq 300$). Each of the following r lines contains c integers $s_{i,j}$. The j-th integer in the i-th of these lines is 0 if there is no gold coin initially at $s_{i,j}$, and 1 otherwise.

Output

Print a single integer: the minimum number of gold coins that should be added so that all the gold coins are labeled differently.

Examples

standard input	standard output
2 3 1 0 0 0 1 1	1
4 4 0 1 1 1 1 1 0 0 1 1 1 1 1 0 1 0	2
7 7 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	18

Problem H. Heroes and Illusions

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

In one well-known MOBA game, n heroes are aligned in a row. However, some of them may be illusions.

The observer counted the number of real heroes from ℓ-th to r-th inclusively for every pair of ℓ and r ($1 \le \ell \le r \le n$), and recorded their evenness. Her $\frac{n(n+1)}{2}$ records show that there were k intervals that contained an odd number of real heroes. How many possible hero alignments are there? The answer may be too large, so print it modulo 998 244 353.

Two alignments are considered different if, for some i, the i-th hero from the left is real in one alignment and an illusion in the other.

Input

The first line of input contains an integer t: the number of test cases ($1 \le t \le 10^5$).

Each of the following t lines contains two integers: n and k ($1 \le n \le 10^5$; $0 \le k \le n(n+1)/2$).

Output

Print the number of possible hero alignments modulo 998 244 353.

Examples

standard input	standard output
1 5 9	10
4 3 4 6 12 6 11 12 30	3 35 0 286

Problem I. Interesting Permutations

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	1024 mebibytes

A permutation $p = (p_1, p_2, \ldots, p_{10^5})$ of length 10^5 is considered *interesting* if it may be constructed in the following way.

- Consider points $1, 2, \ldots, 10^5$ arranged along the coordinate line.
- Choose one of those points as the current point.
- There is a sequence p that is initially empty.
- Repeat the following operation until the length of p is 10^5: let x be the number corresponding to the current point. If x is not in p, add x to the end of p. Then move to one of the points whose distance to x is less than or equal to k.

The distance between points i and j is $|i - j|$.

Your task is to answer the queries of the following form. You are given three integers: n, ℓ, and r. We can pick any interesting permutation p, and then construct $s = (s_1, s_2, \ldots, s_n)$: the permutation created by removing elements larger than n from p. Among the possible s, find the number of permutations such that $\ell \le s_1 \le r$.

Since the number of such permutations may be too large, print it modulo 998 244 353.

Input

The first line of input contains two integers, k and q: the maximum distance and the number of queries, respectively ($1 \le k \le 10^5$; $2 \le q \le 2 \cdot 10^5$). Each of the following q lines contains one query: three integers n, ℓ, and r ($1 \le n \le 10^5$; $1 \le \ell \le r \le n$).

Output

Output q lines. On the i-th of these lines, print the answer to the i-th query.

Examples

standard input	standard output
2 4 4 1 3 3 1 1 9 3 7 1 1 1	16 2 27160 1
256 2 3 1 2 65536 1024 32768	4 517264494

Problem J. Jumping Game

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

The rules of the game are simple.

There is a chessboard with r rows and c columns with a chess knight on it. The cell at the i-th row from the top and the j-th column from the left is called square (i, j). Initially, the knight is placed on square (r_s, c_s).

Annapurna and Brahma alternately take the following action, starting with Annapurna:

- Move the knight onto one of the squares on the board that the knight has never visited since the beginning of the game. Remember that knights can move from (x_1, y_1) to (x_2, y_2) if and only if $(x_1 - x_2)^2 + (y_1 - y_2)^2$ is 5.

The player who cannot move the knight loses the game, and their opponent is declared the winner. Determine whether Annapurna or Brahma will win if both play optimally.

Input

The first line of input contains an integer t: the number of test cases ($2 \le t \le 2 \cdot 10^5$).

Each of the following t lines contains four integers, r, c, r_s and c_s: the number of rows and columns of the board, as well as the starting row and column for the knight, respectively ($1 \le r, c \le 10^9$; $1 \le r_s \le r$; $1 \le c_s \le c$).

Output

Output t lines. On the i-th line, print the name of the winner for the i-th test case: `Annapurna` or `Brahma`.

Example

standard input	*standard output*
2 6 6 6 6 7 19 7 3	Annapurna Brahma

Problem K. Kangaroo On Graph

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

You are given a weighted directed graph consisting of n vertices and m edges, with vertices numbered from 1 to n and edges numbered from 1 to m. The j-th ($1 \le j \le m$) edge goes from vertex u_j to vertex v_j ($u_j < v_j$), and its weight is w_j.

Also, k triplets of integers are given. The i-th ($1 \le i \le k$) triplet is (a_i, b_i, c_i) ($a_i < b_i < c_i$).

Kangaroo starts at vertex 1 and goes to vertex n by repeatedly moving along an edge. In addition, for all i ($1 \le i \le k$), if the kangaroo moves from vertex a_i to vertex b_i directly, then it must next move to a vertex other than vertex c_i.

Determine whether it is possible for the kangaroo to reach vertex n. If it is possible, also calculate the minimum sum of the weights of the edges on the kangaroo's path.

Input

The first line of input contains two integers n and m: the number of vertices and edges in the graph, respectively ($3 \le n \le 2 \cdot 10^5$; $0 \le m \le 2 \cdot 10^5$).

The j-th of the following m lines contains three integers, u_j, v_j, and w_j: the starting and the ending point of the i-th edge and its weight, respectively ($1 \le u_j < v_j \le n$; $(u_i, v_i) \ne (u_j, v_j)$ for $i \ne j$; $1 \le w_j \le 10^9$).

Then follows a line containing an integer k: the number of forbidden triples ($0 \le k \le 2 \cdot 10^5$).

Each of the following k lines contains three integers: a_i, b_i, and c_i ($1 \le a_i < b_i < c_i \le n$). You may assume that both edges (a_i, b_i) and (b_i, c_i) exist in the graph.

Output

If vertex n is unreachable, print -1. Otherwise, print the minimum sum of the weights of the edges on the kangaroo's path.

Examples

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

Problem L. Low Cost Set

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

Consider an integer sequence c of length $2k-1$. Also consider k intervals $[\ell_i, r_i)$. Here, ℓ_i and r_i satisfy $\ell_i < r_i$, and each integer between 1 and $2k$ appears exactly once as an end of an interval.

Given this sequence, create a set s of intervals. Each interval $[\ell, r)$ in the set must satisfy $1 \le \ell < r \le 2k$. Additionally, for all $i = 1, 2, \ldots, k$, the set s has to satisfy at least one of the two following conditions:

- $[\ell_i, r_i) \in s$,
- there exists an integer x ($\ell_i < x < r_i$) such that $[\ell_i, x) \in s$ and $[x, r_i) \in s$.

The *cost* of the set s is defined as the sum of $c_\ell + c_{\ell+1} + \ldots + c_{r-1}$ for all intervals $[\ell, r)$ included in s.

Find the minimum cost of a set that satisfies all the conditions.

Input

The first line of input contains an integer k: the number of intervals ($1 \le k \le 100$).

The i-th of the following k lines contains two integers ℓ_i and r_i: the left (included) and the right (excluded) end of the i-th interval ($1 \le \ell_i < r_i \le 2k$, each integer between 1 and $2k$ can be found in those k lines exactly once).

The last line contains $2k-1$ integers: the sequence c_i ($1 \le c_i \le 10^9$).

Output

Print the minimum cost of the set that satisfies the condition.

Examples

standard input	*standard output*
3 1 4 2 6 3 5 1 2 3 5 8	27
5 3 10 1 5 7 8 4 9 2 6 9 9 8 2 4 4 3 5 3	82