

Problem A. The World Is Full of Colors

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

The world is full of colors.

When creating an artwork, painters often use the technique of **complementary colors** to make their works more visually striking. In the **RGB** color model, **complementary colors** have a clear special property when represented as color codes.

Specifically, in the **RGB** color model, a color code consists of a **#** followed by the hexadecimal representations of its **R** (red), **G** (green), and **B** (blue) values concatenated together. The values of **R**, **G**, and **B** all range from $[0, 255]$. For example, the color code **#1f1e33** represents a very dark purple. Its **R**, **G**, and **B** values in hexadecimal are **1f**, **1e**, **33**, and in decimal are 31, 30, 51. In particular, if any of the hexadecimal representations of **R**, **G**, or **B** has fewer than two digits, a leading 0 must be added. For example, the color code for black is **#000000**.

We define two colors to be **complementary colors** if and only if their **R**, **G**, and **B** values satisfy that each corresponding pair sums to 255.

Whalica landed on a colorless planet during her interstellar voyage. She wanted to bring more vitality and visual impact to the planet by adding color to it, so she first thought of a color. Can you provide the **complementary color** of that color?

Input

The first line contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The next t lines each contain a string s — the color code of the color Whalica came up with.

It is guaranteed that all input color codes are valid **RGB** color codes.

Output

For each test case, output a string — the color code of the **complementary color** of the given color.

Note: The output is case-sensitive. When printing the color code, please use lowercase letters rather than uppercase letters.

Example

standard input	standard output
2	#e0e1cc
#1f1e33	#eebaeb
#114514	

Problem B. Whalica's Permutation Construction

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Whalica has a mysterious permutation* p of length n . Let

$$S_k = \sum_{i=1}^k p_i$$

If for all $1 \leq i \leq n$, S_i is divisible by p_i , then the permutation p is called a **good permutation**.

Checking whether a permutation is a **good permutation** is far too easy, and Whalica would never enjoy doing something that simple. Therefore, Whalica has specially prepared another problem for you:

Given a positive integer n , you need to construct a permutation p of length n such that it is a **good permutation**. You need to determine whether such a permutation exists. Moreover, if such a permutation exists, output any one permutation that satisfies the condition.

Input

The first line contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases.

Each of the next t lines contains an integer n ($1 \leq n \leq 5 \times 10^5$) — the length of the permutation you need to construct.

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

Output

For each test case, the first line contains a string indicating whether such a permutation can be constructed. If it is possible, output YES; otherwise, output NO.

If such a permutation can be constructed, output any one permutation that satisfies the conditions on the second line.

Note that the output is case-sensitive.

Example

standard input	standard output
3	YES
1	1
3	YES
4	2 1 3
	YES
	3 1 4 2

*A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

Problem C. Shichieichou's Guidance

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

"The so-called Shichieichou (Seven-Shadow Butterflies) are the fragments of human memory..."

"The 'memories' of those who left behind attachments or regrets during their lifetime exist in this world in the form of butterflies."

"...Even though they are just memories, these are things that should not remain in this world."

"Someone must guide them back to where they truly belong."

"This is the Yamanomatsuri (Mountain Festival), which has been managed by the Sorakado family for generations."

"...There is one specific memory that I must find, no matter what..."

To fulfill the duties of the Sorakado family and to find that indispensable memory, Sorakado Ao needs to guide the Shichieichou scattered across the mountain toward the "Mayoi no Tachibana" (The Lost Mandarin Orange) during the Yamanomatsuri.

The mountain paths of Torishirojima can be represented as a tree[†] with n nodes, rooted at node 1. **It is guaranteed that every leaf node in the tree has the same depth (i.e., the distance to the root).**

Due to the rugged mountain paths and the deep night, Ao's movement must follow these rules:

1. Ao starts at time 0 from **any leaf node** of her choice. Her goal is to reach the root node 1, where the "Mayoi no Tachibana" is located.
2. Ao can spend 1 unit of time to move from her current node to its **parent node**.
3. During the journey, Ao may use Inari's power **at most once**: she can spend 1 unit of time to take a side path and move from her current non-root node at depth d to **any** node at depth $d - 1$.
4. When Ao is at a node, she can successfully guide all Shichieichou at that location that have not yet disappeared (i.e., those with the current time $\leq t_i$).

To allow as many Shichieichou as possible to return, please tell Ao the maximum number of butterflies she can guide.

Input

Each test contains multiple test cases. The first line contains an integer T ($1 \leq T \leq 10^4$), representing the number of test cases.

For each test case:

- The first line contains two integers n, m ($2 \leq n \leq 2 \times 10^5$, $0 \leq m \leq 2 \times 10^5$), representing the number of nodes in the tree and the number of Shichieichou, respectively.
- The next $n - 1$ lines each contain two integers u_i, v_i ($1 \leq u_i, v_i \leq n$), representing an edge in the tree.

[†]A tree with n vertices is an undirected connected graph with n vertices and $n - 1$ edges. A rooted tree is a tree in which one of the vertices is special and is called the root.

- The next m lines each contain two integers a_i, t_i ($1 \leq a_i \leq n, 0 \leq t_i \leq n$), representing the location and the disappearance time of the i -th Shichieichou.

It is guaranteed that the sum of n over all test cases and the sum of m over all test cases do not exceed 2×10^5 . It is also guaranteed that the given edges form a tree, and that all leaf nodes have the same depth.

Output

For each test case, output a single integer representing the maximum number of Shichieichou Ao can guide.

Example

standard input	standard output
3	2
3 2	2
1 2	3
1 3	
3 0	
1 1	
6 3	
1 2	
1 3	
2 4	
2 5	
3 6	
6 1	
2 1	
1 1	
8 5	
1 2	
1 3	
2 4	
3 6	
4 5	
6 7	
6 8	
5 2	
4 2	
6 8	
3 1	
1 5	

Note

In the first test case, the tree is rooted at node 1, and nodes 2 and 3 are both leaves. Starting from node 3, one Seven-Shadow Butterfly can be guided at time $t = 0$. Then, moving to node 1, another butterfly can be guided at time $t = 1$. Therefore, the answer is 2.

Problem D. Whalica's Lottery Game

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Whalica has n fans who are invited to join Whalica's lottery game.

Whalica stipulates that only non-empty strings consisting only of the characters **a, c, h, i, l, w** can be winning strings. Before the game starts, each fan writes down a valid guess string s_i .

There are q events in total. For the i -th event, one of the following happens:

1. Whalica conducts a draw. Whalica provides a valid winning string T_i , and then reads out each character of this string from left to right. Let $s[l, r]$ denote the string formed by concatenating the characters $s_l, s_{l+1}, \dots, s_{r-1}, s_r$ from s in order. When Whalica reads the j -th character $T_{i,j}$, every fan k such that $s_k[1, j] = T_i[1, j]$ (i.e., the first j characters of the fan's guess string match the first j characters of the winning string) receives 1 Whalica coin. **Formally, the number of Whalica coins fan k gets in this draw is $\text{LCP}(s_k, T_i)$, where $\text{LCP}(s, t)$ denotes the length of the longest common prefix of strings s, t .**
2. Fans modify their strings. To prevent the game from becoming monotonous, Whalica requires **every** fan to change their guess string. However, the fans have secretly expressed their affection for Whalica in their strings and are unwilling to make significant changes. Specifically, they do not change the length of their guess strings, but they perform one replacement on each character of their guess string, following the cyclic rule: $w \rightarrow h \rightarrow a \rightarrow l \rightarrow i \rightarrow c \rightarrow w$. For example, if a fan's string is **whalica**, after modification it becomes **halicwl**.

Whalica is too lazy and just wants to rest. Can you help her compute how many Whalica coins she needs to pay out for each draw event?

Input

The first line contains two integers n, q ($1 \leq n, q \leq 10^5$) — the number of fans and the number of events.

Each of the next n lines contains a fan's initial guess string s_i ($1 \leq |s_i| \leq 10^5$).

Each of the next q lines contains an integer $op_i \in \{1, 2\}$ — the event type.

- If $op_i = 1$, it is a draw event. The same line also contains a string T_i ($1 \leq |T_i| \leq 10^5$) — the winning string.
- If $op_i = 2$, it is a modification event.

It is guaranteed that the total sum of $|s_i|$ and the total sum of $|T_i|$ do not exceed 10^5 , and that the character set of all strings is a subset of **{a, c, h, i, l, w}**.

Output

For each draw event, output the total number of Whalica coins Whalica needs to pay out.

Example

standard input	standard output
3 3	5
awa	7
awhlica	
alicawhalica	
1 awc	
2	
1 licwlaawwl	

Problem E. Eight-thousand-year Wait

Input file: `standard input`
Output file: `standard output`
Time limit: 4 seconds
Memory limit: 1024 megabytes

To commemorate the eight-thousand-year wait, Iroha and Kaguya will edit an echo sequence inside the virtual space “Tsukuyomi”. Whenever two echoes are perfectly in sync, the stage will light up with a flash. They want to know: how many echo sequences can make the stage flash exactly k times?

More specifically, they are given an array a of length n .

For a sequence b of length m , define its number of flashes as

$$f(b) = |\{i | 1 \leq i \leq m - 1, b_i = b_{i+1}\}|$$

that is, the number of times two adjacent elements in the sequence are equal.

You need to compute, among the 2^n subsequences[‡] of a , how many subsequences b have exactly k flashes (i.e. $f(b) = k$). Since the answer may be very large, print it modulo 998244353.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n, k ($1 \leq n \leq 2 \times 10^5$, $0 \leq k \leq 200$) — the length of the array a and the desired number of flashes.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$).

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

Output

For each test case, output a single integer — the number of subsequences whose number of flashes is k , taken modulo 998244353.

Example

standard input	standard output
2	5
4 1	6
1 2 2 1	
3 0	
1 1 2	

Note

In the first test case, $a = [1, 2, 2, 1]$, and the 16 subsequences are as follows:

- $b = []$, the number of flashes is 0.
- $b = [a_1] = [1]$, the number of flashes is 0.
- $b = [a_1, a_2] = [1, 2]$, the number of flashes is 0.
- $b = [a_1, a_2, a_3] = [1, 2, 2]$, the number of flashes is 1.

[‡]A sequence b is a subsequence of a sequence a if b can be obtained from a by the deletion of several (possibly, zero or all) elements.

- $b = [a_1, a_2, a_3, a_4] = [1, 2, 2, 1]$, the number of flashes is 1.
- $b = [a_1, a_2, a_4] = [1, 2, 1]$, the number of flashes is 0.
- $b = [a_1, a_3] = [1, 2]$, the number of flashes is 0.
- $b = [a_1, a_3, a_4] = [1, 2, 1]$, the number of flashes is 0.
- $b = [a_1, a_4] = [1, 1]$, the number of flashes is 1.
- $b = [a_2] = [2]$, the number of flashes is 0.
- $b = [a_2, a_3] = [2, 2]$, the number of flashes is 1.
- $b = [a_2, a_3, a_4] = [2, 2, 1]$, the number of flashes is 1.
- $b = [a_2, a_4] = [2, 1]$, the number of flashes is 0.
- $b = [a_3] = [2]$, the number of flashes is 0.
- $b = [a_3, a_4] = [2, 1]$, the number of flashes is 0.
- $b = [a_4] = [1]$, the number of flashes is 0.

Therefore, there are 5 subsequences whose number of flashes is 1.

Problem F. Magicology

Input file: `standard input`
Output file: `standard output`
Time limit: 4 seconds
Memory limit: 1024 megabytes

I've always loved the word magic. It has a way of making people happy, of putting a smile on their faces.

— Anisphia Wynn Palettia

To push magicology toward a more rigorous discipline, Anisphia and Euphyllia build a mana circuit network at the detached palace. The network forms a tree[§] with n vertices labeled $1, 2, \dots, n$, and vertex i stores a mana value a_i .

Root the tree at vertex 1. For a vertex u , let $\text{Subtree}(u)$ be the set of vertices in the subtree of u (including u itself and all of its descendants[¶]).

Define one operation f on an array $x = (x_1, x_2, \dots, x_n)$ by $y = f(x)$, where

$$y_u = \sum_{v \in \text{Subtree}(u)} x_v$$

They apply the operation exactly k times: let $x^{(0)} = a$, and $x^{(t)} = f(x^{(t-1)})$ for all $1 \leq t \leq k$.

Output $x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}$ modulo 998244353.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 1000$). The description of the test cases follows.

The first line of each test case contains two integers n, k ($1 \leq n \leq 5000, 0 \leq k \leq 10^{18}$).

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i < 998244353$).

The next $n - 1$ lines of each test case contain two integers u_i, v_i ($1 \leq u_i, v_i \leq n$) — two vertices that the i -th edge of the tree connects.

It is guaranteed that the given edges form a tree, and it is guaranteed that the sum of n over all test cases does not exceed 5000.

Output

For each test case, print n integers $x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}$ in one line, each taken modulo 998244353.

[§]A tree with n vertices is an undirected connected graph with n vertices and $n - 1$ edges. A rooted tree is a tree in which one of the vertices is special and is called the root.

[¶]The descendants of vertex u are all vertices v such that $v \neq u$ and on the shortest path from the root to v , vertex u is encountered.

Example

standard input	standard output
3	38 2 21 4 5
5 2	408356129 643450937 4 42549043 5 9
1 2 3 4 5	60 50 30
1 3	
3 4	
3 5	
1 2	
6 10000000000000000000	
3 1 4 1 5 9	
1 2	
2 3	
2 4	
4 5	
4 6	
3 1	
10 20 30	
1 2	
3 2	

Note

In the first test case, $a = [1, 2, 3, 4, 5]$, so

$$x^{(0)} = a = [1, 2, 3, 4, 5]$$

$$x^{(1)} = f(x^{(0)}) = [x_1^{(0)} + x_2^{(0)} + x_3^{(0)} + x_4^{(0)} + x_5^{(0)}, x_2^{(0)}, x_3^{(0)} + x_4^{(0)} + x_5^{(0)}, x_4^{(0)}, x_5^{(0)}] = [15, 2, 12, 4, 5]$$

$$x^{(2)} = f(x^{(1)}) = [x_1^{(1)} + x_2^{(1)} + x_3^{(1)} + x_4^{(1)} + x_5^{(1)}, x_2^{(1)}, x_3^{(1)} + x_4^{(1)} + x_5^{(1)}, x_4^{(1)}, x_5^{(1)}] = [38, 2, 21, 4, 5]$$

Problem G. True Blue

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

Retinal cone cells (Cone Cells) are mainly concentrated in the fovea of the macula and are responsible for color vision and high-resolution vision. Due to differences in their light absorption spectra, their distribution ratios are significantly asymmetric:

L-cones (long-wavelength / red): about 60% – 64%.

M-cones (medium-wavelength / green): about 30% – 32%.

S-cones (short-wavelength / blue): about 2% – 10%, and they are absent in the very center of the fovea.

This distribution explains why the human eye is most sensitive to green light (covered jointly by L and M cones), while its spatial resolution for pure blue is relatively low.

yyz likes the color blue. To find the most beautiful shade of blue, she keeps collecting it using magic. For a beautiful photo, yyz will first transform it into a blue Hilbert curve.

Specifically, there are n color blocks arranged from left to right. The i -th block contains an amount of blue element b_i .

When yyz starts casting magic, she will give an incantation $l\ r\ x$, which means: extract all blocks in the array interval $[l, r]$ that satisfy $b_i \geq x$. After a block is extracted once, it will lose **all** of its blue elements.

Given q magic casts, please tell yyz how many blue elements can be extracted each time.

Input

The first line contains two integers n, q ($1 \leq n, q \leq 2 \times 10^5$) — the number of blocks and the number of queries.

The second line contains n integers $b_1, b_2, \dots, b_n$ ($1 \leq b_i \leq 10^7$) — the amount of blue elements in the i -th block.

Each of the next q lines contains three integers l, r, x ($1 \leq l \leq r \leq n, 1 \leq x \leq 10^7$) — a query interval and the threshold value.

Output

Output q lines. For each query, output one integer: the total amount of blue elements collected in the i -th magic operation.

Example

standard input	standard output
3 4	70
86 91 70	91
3 3 56	86
2 3 1	0
1 1 76	
1 2 52	

Note

yanami ~

Problem H. Whalica's Mysterious Set

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Whalica has a set S , and S is initially empty. In addition, a parameter n is given. You need to perform the given q operations in order. There are four types of operations, and the format and meaning of each operation are as follows:

Operation 1: **1 x** Given an integer x ($1 \leq x \leq n$), insert x into the set S . In particular, if S already contains the element x , then ignore this operation.

Operation 2: **2 x** Given an integer x ($1 \leq x \leq n$), delete x from the set S . In particular, if S does not contain the element x , then ignore this operation.

Operation 3: **3** Query the smallest **positive integer** that does not appear in the set S .

Operation 4: **4** Query the **weight** of the set S . Here, the definition of the **weight** of the set S is as follows: let the number of integers in the range $[1, n]$ that do not appear in the set S be k , and let these k elements in increasing order be $a_1, a_2, a_3, \dots, a_k$, then the **weight** of the set S is:

$$\sum_{i=1}^k a_i \cdot 2^i$$

For each operation 4, since the result may be very large, please take the result modulo 998244353.

Input

The first line contains two integers n, q ($1 \leq n, q \leq 10^5$) — the given parameter and the number of queries.

Next come q queries, one per line. Please refer to the statement for the format and meaning.

Output

For each operation 3, output an integer, indicating the smallest **positive integer** that does not appear in the set S .

For each operation 4, output an integer, indicating the weight of the set S , with the result taken modulo 998244353.

Example

standard input	standard output
5 7	2
1 1	128
1 2	
1 2	
2 2	
2 3	
3	
4	

Note

Explanation for the example:

Whalica Cup (Round 2)
China, February, 15, 2026

Operation No.	Operation Type	Operation Result	Elements in the Set
1	Operation 1	Insert element 1	{1}
2	Operation 1	Insert element 2	{1, 2}
3	Operation 1	The set S already contains element 2, ignore this operation	{1, 2}
4	Operation 2	Delete element 2	{1}
5	Operation 2	The set S does not contain element 3, ignore this operation	{1}
6	Operation 3	Query the smallest positive integer that does not appear in the set S , the result is 2	{1}
7	Operation 4	Query the weight of the set S , the result is $2 \times 2^1 + 3 \times 2^2 + 4 \times 2^3 + 5 \times 2^4 = 4 + 12 + 32 + 80 = 128$	{1}

Problem I. Fireflies Lead the Way

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

At night, there are n firefly lamps lit along the riverbank. The i -th lamp is located at coordinate x_i , and they are arranged from left to right along the river, i.e.,

$$x_1 < x_2 < \cdots < x_n$$

XqAiyee wants to walk from the 1-st lamp to the n -th lamp.

When she is at the i -th lamp ($1 \leq i \leq n-1$), she can choose:

1. Stroll along the bank. Walk along the riverbank to the $(i+1)$ -th lamp, consuming stamina a_i .
2. Fireflies lead the way. Choose an index j ($i+1 \leq j \leq n$), and follow the fireflies to the j -th lamp, consuming stamina $b_i + (x_j - x_i)^2$.

Since the fireflies will get tired, XqAiyee can use “Fireflies lead the way” at most k times.

Please determine the minimum stamina required for XqAiyee to reach the n -th lamp.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 2 \times 10^5$). The description of the test cases follows.

The first line of each test case contains two integers n, k ($2 \leq n \leq 2 \times 10^5$, $0 \leq k \leq 200$) — the number of firefly lamps and the maximum number of times “Fireflies lead the way” can be used.

The second line of each test case contains n integers $x_1, x_2, \dots, x_n$ ($-10^5 \leq x_1 < x_2 < \cdots < x_n \leq 10^5$) — the coordinates of the lamps.

The third line of each test case contains $n-1$ integers $a_1, a_2, \dots, a_{n-1}$ ($0 \leq a_i \leq 10^5$).

The fourth line of each test case contains $n-1$ integers $b_1, b_2, \dots, b_{n-1}$ ($0 \leq b_i \leq 10^5$).

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

Output

For each test case, output a single integer — the minimum stamina required for XqAiyee to reach the n -th lamp.

Example

standard input	standard output
2	23
5 1	20
0 2 3 7 10	
3 2 100 2	
1 5 0 4	
4 2	
-3 0 4 5	
2 100 2	
0 1 0	