A. Game of Division

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

You are given an array of integers $a_1, a_2, \ldots, a_n$ of length n and an integer k.

Two players are playing a game. The first player chooses an index $1 \le i \le n$. Then the second player chooses a different index $1 \le j \le n, i \ne j$. The first player wins if $|a_i - a_j|$ is not divisible by k. Otherwise, the second player wins.

We play as the first player. Determine whether it is possible to win, and if so, which index i should be chosen.

The absolute value of a number x is denoted by $|x|$ and is equal to x if $x \ge 0$, and $-x$ otherwise.

Input

Each test contains multiple test cases. The first line of input contains a single integer t ($1 \le t \le 100$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers n and k ($1 \le n \le 100$; $1 \le k \le 100$) — the length of the array and the number k.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 100$) — the elements of the array a.

Output

For each test case, if it is impossible for the first player to win, print "NO" (without quotes).

Otherwise, print "YES" (without quotes) and on the next line the appropriate index $1 \le i \le n$. If there are multiple solutions, print any of them.

You can output each letter in any case (lowercase or uppercase). For example, the strings "yEs", "yes", "Yes" and "YES" will be recognized as a positive answer.

Standard Input	Standard Output
7 3 2 1 2 3 4 2 1 2 4 5 5 3 10 7 3 4 5 5 3 1 31 15 55 36 2 1 17 17 2 2 17 18 1 3 6	YES 2 NO YES 3 NO NO YES 2 YES 1

Note

In the first test case, the first player can choose $a_2 = 2$. Then:

- If the second player chooses $a_1 = 1$, the resulting difference is $|2 - 1| = 1$ which is not divisible by $k = 2$.
- If the second player chooses $a_3 = 3$, the resulting difference is $|2 - 3| = 1$ which is not divisible by $k = 2$.

In the second test case:

- If the first player chooses $a_1 = 1$ and then the second player chooses $a_4 = 5$, the resulting difference is $|1 - 5| = 4$ which is divisible by $k = 2$.
- If the first player chooses $a_2 = 2$ and then the second player chooses $a_3 = 4$, the resulting difference is $|2 - 4| = 2$ which is divisible by $k = 2$.
- If the first player chooses $a_3 = 4$ and then the second player chooses $a_2 = 2$, the resulting difference is $|4 - 2| = 2$ which is divisible by $k = 2$.
- If the first player chooses $a_4 = 5$ and then the second player chooses $a_1 = 1$, the resulting difference is $|5 - 1| = 4$ which is divisible by $k = 2$.

In any case, the second player wins.

,

B. Paint a Strip

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

You have an array of **zeros** $a_1, a_2, \ldots, a_n$ of length n.

You can perform two types of operations on it:

1. Choose an index i such that $1 \le i \le n$ and $a_i = 0$, and assign 1 to a_i;
2. Choose a pair of indices l and r such that $1 \le l \le r \le n$, $a_l = 1$, $a_r = 1$, $a_l + \ldots + a_r \ge \lceil \frac{r-l+1}{2} \rceil$, and assign 1 to a_i for all $l \le i \le r$.

What is the minimum number of operations of the **first type** needed to make all elements of the array equal to one?

Input

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

The only line of each test case contains one integer n ($1 \le n \le 10^5$) — the length of the array.

Note that there is no limit on the sum of n over all test cases.

Output

For each test case, print one integer — the minimum number of needed operations of **first type**.

Standard Input	Standard Output
4 1 2 4 20	1 2 2 4

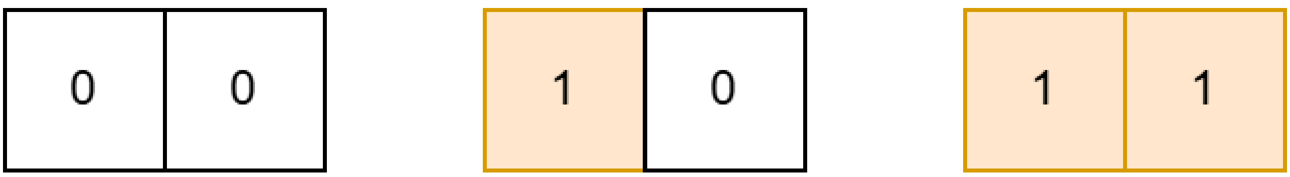
Note

In the first test case, you can perform an operation of the 1st type with $i = 1$.

In the second test case, you can perform the following sequence of operations:

1. Operation of 1st type, $i = 1$. After performing this operation, the array will look like this: $[1, 0]$.
2. Operation of 1st type, $i = 2$. After performing this operation, the array will look like this: $[1, 1]$.

The sequence of operations in the second test case

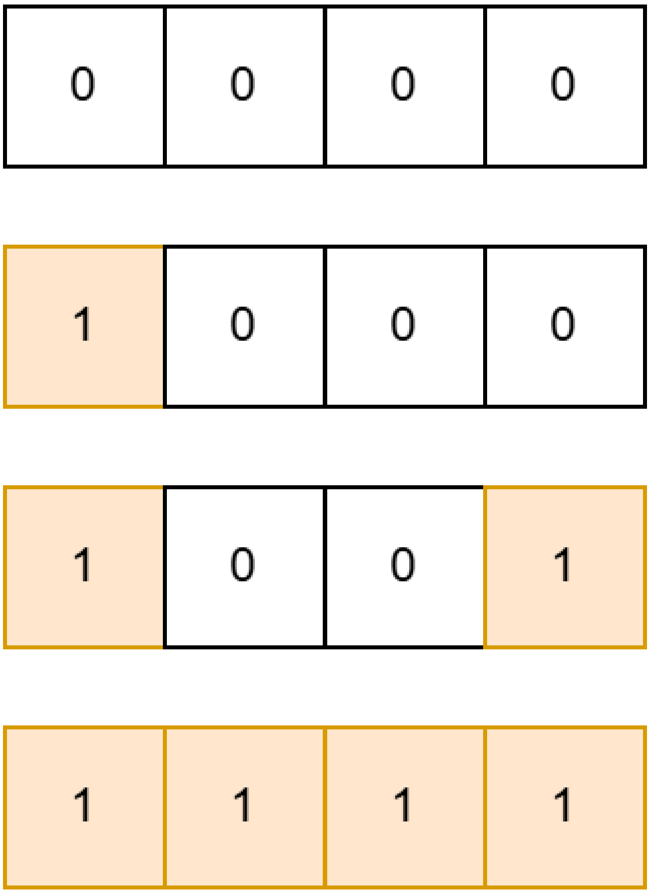


In the third test case, you can perform the following sequence of operations:

1. Operation of 1st type, $i = 1$. After performing this operation, the array will look like this: $[1, 0, 0, 0]$.

2. Operation of 1st type, $i = 4$. After performing this operation, the array will look like this: $[1, 0, 0, 1]$.
3. Operation of 2nd type, $l = 1$, $r = 4$. On this segment, $a_l + \ldots + a_r = a_1 + a_2 + a_3 + a_4 = 2$, which is not less than $\lceil \frac{r-l+1}{2} \rceil = 2$. After performing this operation, the array will look like this: $[1, 1, 1, 1]$.

The sequence of operations in the third test case



,

C. Ordered Permutations

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Consider a permutation* $p_1, p_2, \ldots, p_n$ of integers from 1 to n. We can introduce the following sum for it†:

$$S(p) = \sum_{1 \le l \le r \le n} \min(p_l, p_{l+1}, \ldots, p_r)$$

Let us consider all permutations of length n with the maximum possible value of $S(p)$. Output the k-th of them in lexicographical‡ order, or report that there are less than k of them.

---

*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).

† For example:

- For the permutation $[1,2,3]$ the value of $S(p)$ is equal to $\min(1) + \min(1,2) + \min(1,2,3) + \min(2) + \min(2,3) + \min(3) = 1+1+1+2+2+3 = 10$
- For the permutation $[2,4,1,3]$ the value of $S(p)$ is equal to $\min(2) + \min(2,4) + \min(2,4,1) + \min(2,4,1,3) + \min(4) + \min(4,1) + \min(4,1,3) + \min(1) + \min(1,3) + \min(3) = 2+2+1+1+4+1+1+1+1+3 = 17$.

‡ An array a is lexicographically smaller than an array b if and only if one of the following holds:

- a is a prefix of b, but $a \ne b$; or
- in the first position where a and b differ, the array a has a smaller element than the corresponding element in b.

Input

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

The only line of each test case contains two integers n and k ($1 \le n \le 2 \cdot 10^5$; $1 \le k \le 10^{12}$) — the length of the permutation and the index number of the desired permutation.

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

Output

For each test case, if there are less than k suitable permutations, print -1.

Otherwise, print the k-th suitable permutation.

Standard Input	Standard Output

6 3 2 3 3 4 11 4 6 6 39 7 34	1 3 2 2 3 1 -1 2 4 3 1 -1 2 3 4 5 7 6 1

Note

Let us calculate the required sum for all permutations of length 3 (ordered lexicographically):

Permutation	Value of $S(p)$
$[1, 2, 3]$	10
$[1, 3, 2]$	10
$[2, 1, 3]$	9
$[2, 3, 1]$	10
$[3, 1, 2]$	9
$[3, 2, 1]$	10

In the first test case, you have to print the second suitable permutation of length 3. Looking at the table, we see that it is the permutation $[1, 3, 2]$.

In the second test case, you have to print the third suitable permutation of length 3. Looking at the table, we see that it is the permutation $[2, 3, 1]$.

D. Non Prime Tree

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given a tree with n vertices.

You need to construct an array $a_1, a_2, \ldots, a_n$ of length n, consisting of **unique** integers from 1 to $2 \cdot n$, and such that for each edge $u_i \leftrightarrow v_i$ of the tree, the value $|a_{u_i} - a_{v_i}|$ is not a prime number.

Find any array that satisfies these conditions, or report that there is no such array.

Input

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

The first line of each test case contains a single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of vertices in the tree.

The next $n-1$ lines contain the edges of the tree, one edge per line. The i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$; $u_i \neq v_i$), denoting the edge between the nodes u_i and v_i.

It's guaranteed that the given edges form a tree.

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

Output

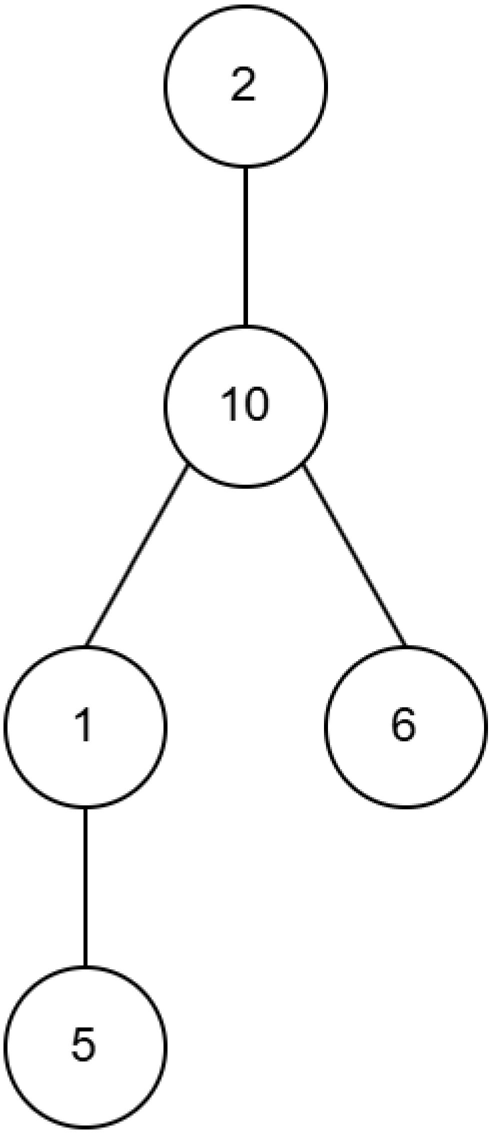
For each test case, if an array that satisfies the conditions exists, print its elements $a_1, a_2, \ldots, a_n$. Otherwise, print -1.

Standard Input	Standard Output
2 5 1 2 2 3 2 4 3 5 7 1 2 1 3 2 4 3 5 3 6 3 7	2 10 1 6 5 8 7 12 1 4 6 3

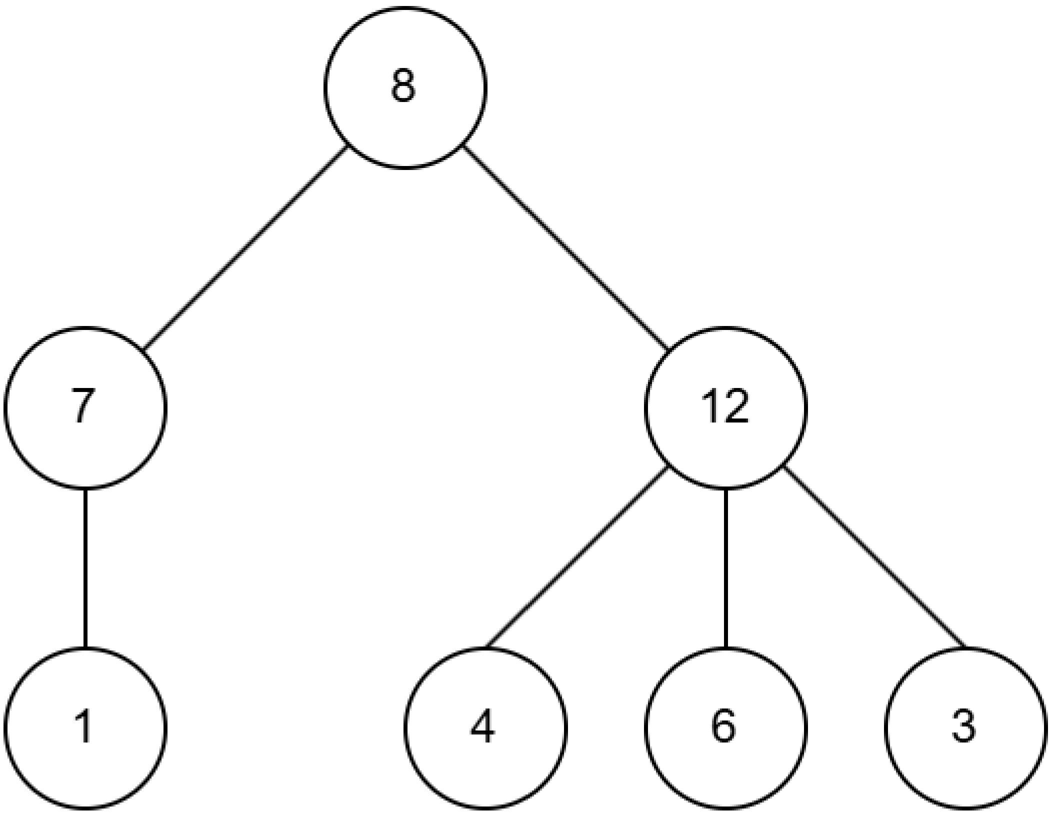
Note

The possible answers are shown below. Instead of the vertex numbers, the corresponding elements of the array a are written in them.

The image of the tree in the first test case



The image of the tree in the second test case



E. Control of Randomness

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given a tree with n vertices.

Let's place a robot in some vertex $v \neq 1$, and suppose we initially have p coins. Consider the following process, where in the i-th step (starting from $i = 1$):

- If i is odd, the robot moves to an adjacent vertex in the direction of vertex 1;
- Else, i is even. You can either pay one coin (if there are some left) and then the robot moves to an adjacent vertex in the direction of vertex 1, or not pay, and then the robot moves to an adjacent vertex chosen **uniformly at random**.

The process stops as soon as the robot reaches vertex 1. Let $f(v, p)$ be the minimum possible expected number of steps in the process above if we spend our coins optimally.

Answer q queries, in the i-th of which you have to find the value of $f(v_i, p_i)$, modulo* $998\,244\,353$.

---

* Formally, let $M = 998\,244\,353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^3$). The description of the test cases follows.

The first line of each test case contains two integers n and q ($2 \le n \le 2 \cdot 10^3$; $1 \le q \le 2 \cdot 10^3$) — the number of vertices in the tree and the number of queries.

The next $n - 1$ lines contain the edges of the tree, one edge per line. The i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$; $u_i \neq v_i$), denoting the edge between the nodes u_i and v_i.

The next q lines contain two integers v_i and p_i ($2 \le v_i \le n$; $0 \le p_i \le n$).

It's guaranteed that the given edges form a tree.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^3$.

It is guaranteed that the sum of q over all test cases does not exceed $2 \cdot 10^3$.

Output

For each test case, print q integers: the values of $f(v_i, p_i)$ modulo $998\,244\,353$.

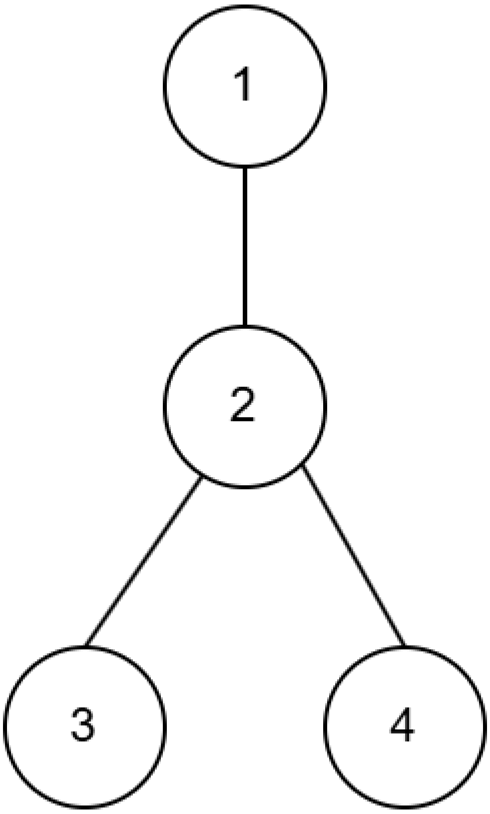
Formally, let $M = 998\,244\,353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Standard Input	Standard Output

2 4 4 1 2 2 3 2 4 2 0 3 0 4 0 3 1 12 10 1 2 2 3 2 4 1 5 5 6 6 7 6 8 6 9 8 10 10 11 10 12 6 0 9 0 10 0 11 0 3 1 7 1 10 1 12 1 12 2 11 12	1 6 6 2 4 9 8 15 2 3 6 9 5 5

Note

The tree in the first test case:



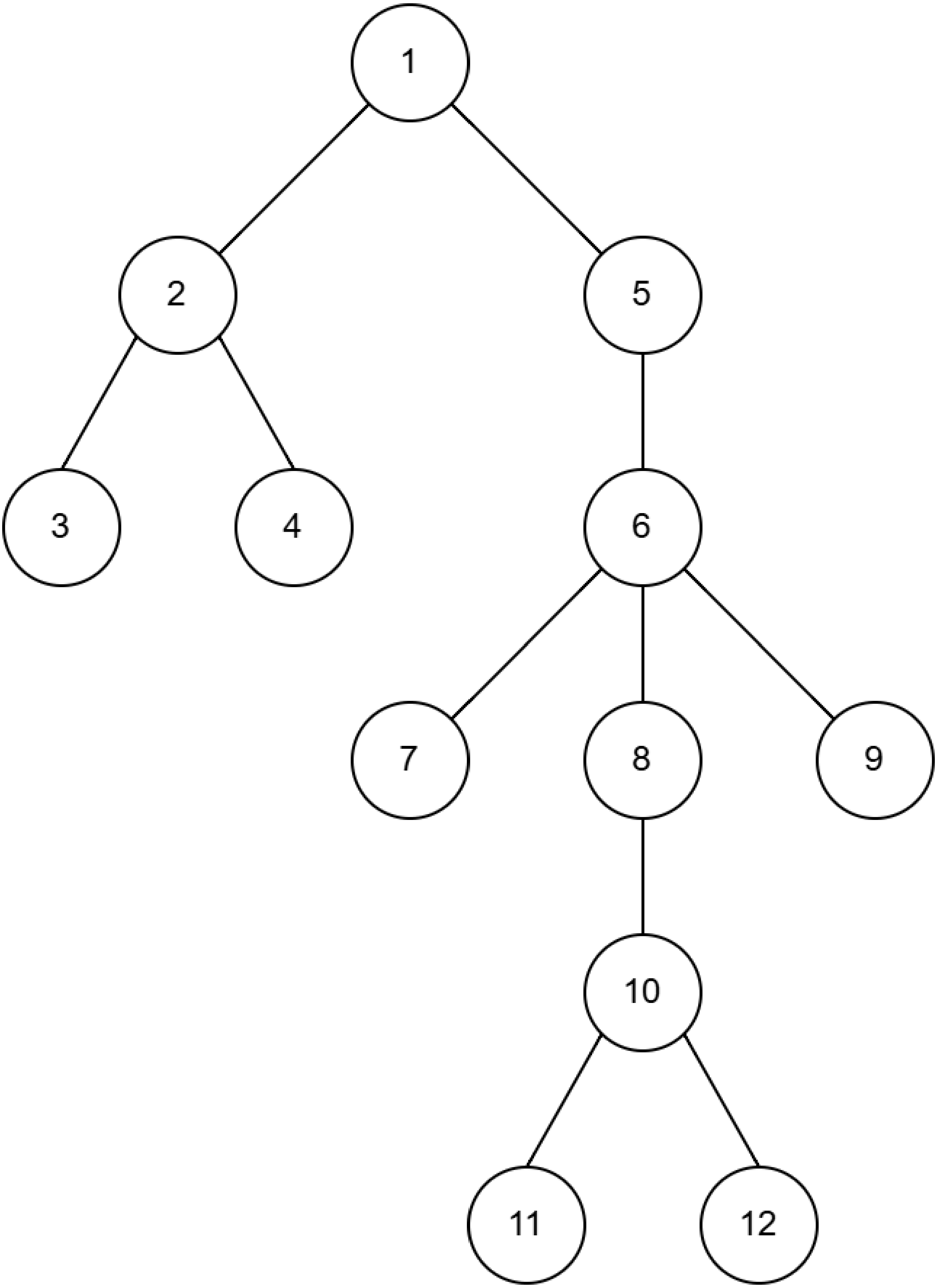
In the first query, the expected value is equal to 1, since the robot starts moving from vertex 2 to vertex 1 in the first step and the process stops.

Let's calculate the expected value in the second query (x is the number of steps):

- $P(x < 2) = 0$, the distance to vertex 1 is 2 and the robot cannot reach it in fewer steps.
- $P(x = 2) = \frac{1}{3}$, since there is only one sequence of steps leading to $x = 2$. This is $3 \rightarrow_1 2 \rightarrow_{0.33} 1$ with probability $1 \cdot \frac{1}{3}$.
- $P(x \bmod 2 = 1) = 0$, since the robot can reach vertex 1 by only taking an even number of steps.
- $P(x = 4) = \frac{2}{9}$: possible paths $3 \rightarrow_1 2 \rightarrow_{0.67} [3, 4] \rightarrow_1 2 \rightarrow_{0.33} 1$.
- $P(x = 6) = \frac{4}{27}$: possible paths $3 \rightarrow_1 2 \rightarrow_{0.67} [3, 4] \rightarrow_1 2 \rightarrow_{0.67} [3, 4] \rightarrow_1 2 \rightarrow_{0.33} 1$.
- $P(x = i \cdot 2) = \frac{2^{i-1}}{3^i}$ in the general case.

As a result, $f(v, p) = \sum\limits_{i=1}^{\infty} i \cdot 2 \cdot \frac{2^{i-1}}{3^i} = 6$.

The tree in the second test case:


1
2
5
3
4
6
7
8
9
10
11
12

F. Number of Cubes

```
Input file:     standard input
Output file:    standard output
Time limit:     5 seconds
Memory limit:   256 megabytes
```

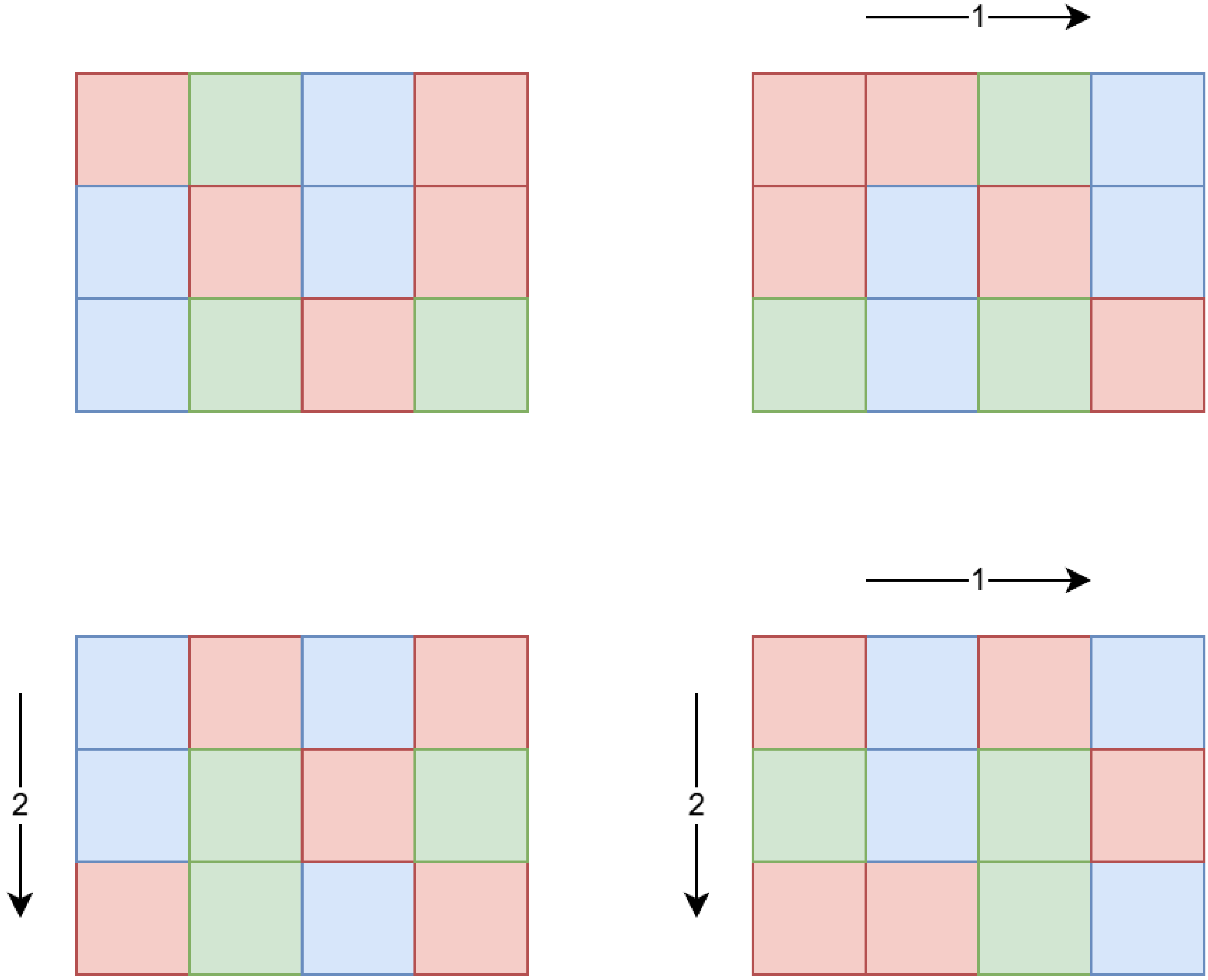
Consider a rectangular parallelepiped with sides a, b, and c, that consists of unit cubes of k different colors. We can apply cyclic shifts to the parallelepiped in any of the three directions any number of times*.

There are d_i cubes of the i-th color ($1 \le i \le k$). How many different parallelepipeds (with the given sides) can be formed from these cubes, no two of which can be made equal by some combination of cyclic shifts?

---

*On the image:

- Top left shows the top view of the original parallelepiped. Lower layers will shift in the same way as the top layer.
- Top right shows the top view of a parallelepiped shifted to the right by 1.
- Bottom left shows the top view of a parallelepiped shifted down by 2.
- Bottom right shows the top view of a parallelepiped shifted to the right by 1 and down by 2.



Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

The first line of each test case contains four integers: a, b, c, and k ($1 \le a, b, c \le 3 \cdot 10^6$; $a \cdot b \cdot c \le 3 \cdot 10^6$; $1 \le k \le 10^6$) — three sides of the parallelepiped and the number of colors of unit cubes.

The second line of each test case contains k integers $d_1, d_2, \ldots, d_k$ ($1 \le d_1 \le d_2 \le \ldots \le d_k \le 3 \cdot 10^6$) — the elements of the array d: the number of cubes of a given color.

It is guaranteed that in each test case the sum of the elements of the array d is equal to $a \cdot b \cdot c$.

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

Output

For each test case, print one integer — the number of different parallelepipeds modulo $998\,244\,353$.

Standard Input	Standard Output
6 1 1 1 1 1 6 1 1 3 1 2 3 12 1 1 3 2 4 6 3 3 1 2 3 6 2 3 3 2 6 12 72 60 96 4 17280 86400 120960 190080	1 10 1160 12 1044 231490207

Note

In the first test case, there is only one parallelepiped, which consists of one unit cube.

Possible parallelepipeds in the second test case