A. Alice's Adventures in "Chess"

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

Alice is trying to meet up with the Red Queen in the countryside! Right now, Alice is at position $(0, 0)$, and the Red Queen is at position (a, b). Alice can only move in the four cardinal directions (north, east, south, west).

More formally, if Alice is at the point (x, y), she will do one of the following:

- go north (represented by `N`), moving to $(x, y + 1)$;
- go east (represented by `E`), moving to $(x + 1, y)$;
- go south (represented by `S`), moving to $(x, y - 1)$; or
- go west (represented by `W`), moving to $(x - 1, y)$.

Alice's movements are predetermined. She has a string s representing a sequence of moves that she performs from left to right. Once she reaches the end of the sequence, she repeats the same pattern of moves forever.

Can you help Alice figure out if she will ever meet the Red Queen?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 500$). The description of the test cases follows.

The first line of each test case contains three integers n, a, b ($1 \le n, a, b \le 10$) — the length of the string and the initial coordinates of the Red Queen.

The second line contains a string s of length n consisting only of the characters `N`, `E`, `S`, or `W`.

Output

For each test case, output a single string "`YES`" or "`NO`" (without the quotes) denoting whether Alice will eventually meet the Red Queen.

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

Standard Input	Standard Output
6 2 2 2 NE 3 2 2 NNE 6 2 1 NNEESW 6 10 10 NNEESW 3 4 2 NEE 4 5 5 NEWS	YES NO YES YES YES NO

Note

In the first test case, Alice follows the path $(0,0) \underset{\texttt{N}}{\rightarrow} (0,1) \underset{\texttt{E}}{\rightarrow} (1,1) \underset{\texttt{N}}{\rightarrow} (1,2) \underset{\texttt{E}}{\rightarrow} (2,2)$.

In the second test case, Alice can never reach the Red Queen.

,

B. Alice's Adventures in Permuting

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

Alice mixed up the words transmutation and permutation! She has an array a specified via three integers n, b, c: the array a has length n and is given via $a_i = b \cdot (i-1) + c$ for $1 \le i \le n$. For example, if $n = 3$, $b = 2$, and $c = 1$, then $a = [2 \cdot 0 + 1, 2 \cdot 1 + 1, 2 \cdot 2 + 1] = [1, 3, 5]$.

Now, Alice really enjoys permutations of $[0, \ldots, n-1]^*$ and would like to transform a into a permutation. In one operation, Alice replaces the maximum element of a with the $\operatorname{MEX}^\dagger$ of a. If there are multiple maximum elements in a, Alice chooses the leftmost one to replace.

Can you help Alice figure out how many operations she has to do for a to become a permutation for the first time? If it is impossible, you should report it.

---

*A permutation of length n is an array consisting of n distinct integers from 0 to $n-1$ in arbitrary order. **Please note, this is slightly different from the usual definition of a permutation.** For example, $[1, 2, 0, 4, 3]$ is a permutation, but $[0, 1, 1]$ is not a permutation (1 appears twice in the array), and $[0, 2, 3]$ is also not a permutation ($n = 3$ but there is 3 in the array).

†The MEX of an array is the smallest non-negative integer that does not belong to the array. For example, the MEX of $[0, 3, 1, 3]$ is 2 and the MEX of $[5]$ is 0.

Input

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

The only line of each test case contains three integers n, b, c ($1 \le n \le 10^{18}$; $0 \le b, c \le 10^{18}$) — the parameters of the array.

Output

For each test case, if the array can never become a permutation, output -1. Otherwise, output the minimum number of operations for the array to become a permutation.

Standard Input	Standard Output
7 10 1 0 1 2 3 100 2 1 3 0 1 3 0 0 1000000000000000000 0 0 1000000000000000000 1000000000000000000 1000000000000000000	0 1 50 2 -1 -1 1000000000000000000

Note

In the first test case, the array is already $[0, 1, \ldots, 9]$, so no operations are required.

In the third test case, the starting array is $[1, 3, 5, \ldots, 199]$. After the first operation, the 199 gets transformed into a 0. In the second operation, the 197 gets transformed into a 2. If we continue this, it will take exactly 50

operations to get the array $[0, 1, 2, 3, \ldots, 99]$.

In the fourth test case, two operations are needed: $[1, 1, 1] \to [0, 1, 1] \to [0, 2, 1]$.

In the fifth test case, the process is $[0, 0, 0] \to [1, 0, 0] \to [2, 0, 0] \to [1, 0, 0] \to [2, 0, 0]$. This process repeats forever, so the array is never a permutation and the answer is -1.

,

C. Alice's Adventures in Cutting Cake

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

Alice is at the Mad Hatter's tea party! There is a long sheet cake made up of n sections with tastiness values $a_1, a_2, \ldots, a_n$. There are m creatures at the tea party, excluding Alice.

Alice will cut the cake into $m+1$ pieces. Formally, she will partition the cake into $m+1$ subarrays, where each subarray consists of some number of adjacent sections. The tastiness of a piece is the sum of tastiness of its sections. Afterwards, she will divvy these $m+1$ pieces up among the m creatures and herself (her piece can be empty). However, each of the m creatures will only be happy when the tastiness of its piece is v or more.

Alice wants to make sure every creature is happy. Limited by this condition, she also wants to maximize the tastiness of her own piece. Can you help Alice find the maximum tastiness her piece can have? If there is no way to make sure every creature is happy, output -1.

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 three integers n, m, v ($1 \le m \le n \le 2 \cdot 10^5$; $1 \le v \le 10^9$) — the number of sections, the number of creatures, and the creatures' minimum requirement for tastiness, respectively.

The next line contains n space separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the tastinesses of the sections.

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

Output

For each test case, output the maximum tastiness Alice can achieve for her piece, or -1 if there is no way to make sure every creature is happy.

Standard Input	Standard Output
7 6 2 1 1 1 10 1 1 10 6 2 2 1 1 10 1 1 10 6 2 3 1 1 10 1 1 10 6 2 10 1 1 10 1 1 10 6 2 11 1 1 10 1 1 10 6 2 12 1 1 10 1 1 10	22 12 2 2 2 0 -1

6 2 12 1 1 1 1 10 10	

Note

For the first test case, Alice can give the first and second section as their own pieces, and then take the remaining $10 + 1 + 1 + 10 = 22$ tastiness for herself. We can show that she cannot do any better.

For the second test case, Alice could give the first and second section as one piece, and the sixth section as one piece. She can then take the remaining $10 + 1 + 1 = 12$ tastiness for herself. We can show that she cannot do any better.

For the seventh test case, Alice cannot give each creature a piece of at least 12 tastiness.

,

D. Alice's Adventures in Cards

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

Alice is playing cards with the Queen of Hearts, King of Hearts, and Jack of Hearts. There are n different types of cards in their card game. Alice currently has a card of type 1 and needs a card of type n to escape Wonderland. The other players have one of each kind of card.

In this card game, Alice can trade cards with the three other players. Each player has different preferences for the n types of cards, which can be described by permutations* q, k, and j for the Queen, King, and Jack, respectively.

A player values card a more than card b if for their permutation p, $p_a > p_b$. Then, this player is willing to trade card b to Alice in exchange for card a. Alice's preferences are straightforward: she values card a more than card b if $a > b$, and she will also only trade according to these preferences.

Determine if Alice can trade up from card 1 to card n subject to these preferences, and if it is possible, give a possible set of trades to do it.

---

* 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).

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 an integer n ($2 \le n \le 2 \cdot 10^5$) — the number of card types.

The next three lines contain the preferences of the Queen, King, and Jack respectively. Each of these lines contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$) — a permutation corresponding to the player's preferences.

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

Output

For each test case, on the first line output a single string "YES" or "NO" (without the quotes) denoting whether Alice can trade up to card n.

If the first line was "YES", then on the next line output k — the number of trades Alice will make. On the next k lines output space separated a character $c \in \{\texttt{q}, \texttt{k}, \texttt{j}\}$ and integer x, denoting that Alice trades with player c to get card x. It must be the case that on the k'th line, $x = n$. If there are multiple solutions, print any of them.

You can output this answer in any case (upper or lower). For example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as positive responses. The same goes for the character c denoting the player in the trade (Q, K, J will all be accepted alongside their lowercase variants).

Standard Input	Standard Output
2 3	YES 2

1 3 2 2 1 3 1 2 3 4 2 3 1 4 1 2 3 4 1 4 2 3	k 2 q 3 NO

Note

In the first testcase, Alice can trade with the King to get card 2. She can then trade with the Queen to get card 3.

In the second testcase, even though Alice can trade with the Queen to get card 3, with the King to get card 2, and then with the Jack to get card 4, this is <u>not</u> a valid solution since it doesn't respect Alice's preferences. We can show that there is no way for Alice to get to card 4.

E. Alice's Adventures in the Rabbit Hole

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

Alice is at the bottom of the rabbit hole! The rabbit hole can be modeled as a tree* which has an exit at vertex 1, and Alice starts at some vertex v. She wants to get out of the hole, but unfortunately, the Queen of Hearts has ordered her execution.

Each minute, a fair coin is flipped. If it lands heads, Alice gets to move to an adjacent vertex of her current location, and otherwise, the Queen of Hearts gets to pull Alice to an adjacent vertex of the Queen's choosing. If Alice ever ends up on any of the non-root leaves† of the tree, Alice loses.

Assuming both of them move optimally, compute the probability that Alice manages to escape for every single starting vertex $1 \le v \le n$. Since these probabilities can be very small, output them modulo $998\,244\,353$.

Formally, let $M = 998\,244\,353$. It can be shown that the exact 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 $x \cdot q \equiv p \pmod{M}$.

---

* A *tree* is a connected simple graph which has n vertices and $n-1$ edges.

† A *leaf* is a vertex that is connected to exactly one edge.

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 i-th of the next $n-1$ lines contains two integers x_i and y_i ($1 \le x_i, y_i \le n$ and $x_i \neq y_i$) — the edges 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 $2 \cdot 10^5$.

Output

For each test case, output n integers on one line — the probabilities of Alice escaping starting from vertex $1, 2, \ldots, n$. Since these probabilities can be very small, output them modulo $998\,244\,353$.

Standard Input	Standard Output
2 5 1 2 1 3 2 4 3 5 9 1 2 2 3 4 5	1 499122177 499122177 0 0 1 499122177 0 332748118 166374059 0 443664157 720954255 0

5 6 7 8 8 9 2 4 5 7	

Note

For the first test case:

1. Alice escapes from the root (vertex 1) by definition with probability 1.
2. Alice immediately loses from vertices 4 and 5 since they are leaves.
3. From the other two vertices, Alice escapes with probability $\frac{1}{2}$ since the Queen will pull her to the leaves.

,

F. Alice's Adventures in Addition

```
Input file:     standard input
Output file:    standard output
Time limit:     3 seconds
Memory limit:   32 megabytes
```

Note that the memory limit is unusual.

The Cheshire Cat has a riddle for Alice: given n integers $a_1, a_2, \ldots, a_n$ and a target m, is there a way to insert $+$ and $\times$ into the circles of the expression

$$a_1 \circ a_2 \circ \cdots \circ a_n = m$$

to make it true? We follow the usual order of operations: $\times$ is done before $+$.

Although Alice is excellent at chess, she is not good at math. Please help her so she can find a way out of Wonderland!

Input

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

The first line of each test case contains two integers n, m ($1 \le n \le 2 \cdot 10^5$; $1 \le m \le 10^4$) — the number of integers and the target, respectively.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^4$) — the elements of the array a.

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

Output

For each test case, output "YES" without quotes if it is possible to get the target by inserting $+$ or $\times$ and "NO" otherwise.

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

Standard Input	Standard Output
6 5 4 2 1 1 1 2 5 5 2 1 1 1 2 5 6 2 1 1 1 2 5 7 2 1 1 1 2 5 8 2 1 1 1 2 5 6 2 0 2 2 3	YES YES YES YES NO YES

Note

Possible solutions for the first four test cases are shown below.

$$2 \times 1 + 1 \times 1 \times 2 = 4$$
$$2 \times 1 + 1 + 1 \times 2 = 5$$
$$2 \times 1 + 1 + 1 + 2 = 6$$
$$2 + 1 + 1 + 1 + 2 = 7$$

It is impossible to get a result of 8 in the fifth test case.