Russian Code Cup 2017 - Finals [Unofficial Mirror, Div. 1 Only Recommended, Teams Allowed]

A. Set Theory

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Masha and Grisha like studying sets of positive integers.

One day Grisha has written a set A containing n different integers a_i on a blackboard. Now he asks Masha to create a set B containing n different integers b_j such that all n^2 integers that can be obtained by summing up a_i and b_j for all possible pairs of i and j are different.

Both Masha and Grisha don't like big numbers, so all numbers in A are from 1 to 10^6, and all numbers in B must also be in the same range.

Help Masha to create the set B that satisfies Grisha's requirement.

Input

Input data contains multiple test cases. The first line contains an integer t — the number of test cases ($1 \le t \le 100$).

Each test case is described in the following way: the first line of the description contains one integer n — the number of elements in A ($1 \le n \le 100$).

The second line contains n integers a_i — the elements of A ($1 \le a_i \le 10^6$).

Output

For each test first print the answer:

- `NO`, if Masha's task is impossible to solve, there is no way to create the required set B.
- `YES`, if there is the way to create the required set. In this case the second line must contain n different positive integers b_j — elements of B ($1 \le b_j \le 10^6$). If there are several possible sets, output any of them.

Example

input
3 3 1 10 100 1 1 2 2 4
output
YES 1 2 3 YES 1 YES 1 2

B. Similar Words

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Let us call a non-empty sequence of lowercase English letters a *word*. *Prefix* of a word x is a word y that can be obtained from x by removing zero or more last letters of x.

Let us call two words *similar*, if one of them can be obtained from the other by removing its first letter.

You are given a set S of words. Find the maximal possible size of set of non-empty words X such that they satisfy the following:

- each word of X is prefix of some word from S;
- X has no similar words.

Input

Input data contains multiple test cases. The first line of the input data contains an integer t — the number of test cases. The descriptions of test cases follow.

The first line of each description contains an integer n — the number of words in the set S ($1 \le n \le 10^6$). Each of the following n lines contains one non-empty word — elements of S. All words in S are different.

It is guaranteed that the total length of all words in one input data doesn't exceed 10^6.

Output

For each test case print one line that contains one integer m — the maximal number of words that X can contain.

Example

input
2 3 aba baba aaab 2 aa a
output
6 1

C. Eleventh Birthday

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

It is Borya's eleventh birthday, and he has got a great present: n cards with numbers. The i-th card has the number a_i written on it. Borya wants to put his cards in a row to get one greater number. For example, if Borya has cards with numbers 1, 31, and 12, and he puts them in a row in this order, he would get a number 13112.

He is only 11, but he already knows that there are $n!$ ways to put his cards in a row. But today is a special day, so he is only interested in such ways that the resulting big number is divisible by eleven. So, the way from the previous paragraph is good, because $13112 = 1192 \times 11$, but if he puts the cards in the following order: 31, 1, 12, he would get a number 31112, it is not divisible by 11, so this way is not good for Borya. Help Borya to find out how many good ways to put the cards are there.

Borya considers all cards different, even if some of them contain the same number. For example, if Borya has two cards with 1 on it, there are two good ways.

Help Borya, find the number of good ways to put the cards. This number can be large, so output it modulo 998244353.

Input

Input data contains multiple test cases. The first line of the input data contains an integer t — the number of test cases ($1 \le t \le 100$). The descriptions of test cases follow.

Each test is described by two lines.

The first line contains an integer n ($1 \le n \le 2000$) — the number of cards in Borya's present.

The second line contains n integers a_i ($1 \le a_i \le 10^9$) — numbers written on the cards.

It is guaranteed that the total number of cards in all tests of one input data doesn't exceed 2000.

Output

For each test case output one line: the number of ways to put the cards to the table so that the resulting big number was divisible by 11, print the number modulo 998244353.

Example

input
4 2 1 1 3 1 31 12 3 12345 67 84 9 1 2 3 4 5 6 7 8 9

output
2 2 2 31680

D. Masha and Cactus

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Masha is fond of cacti. When she was a little girl, she decided to plant a tree. Now Masha wants to make a nice cactus out of her tree.

Recall that *tree* is a connected undirected graph that has no cycles. *Cactus* is a connected undirected graph such that each vertex belongs to at most one cycle.

Masha has some additional edges that she can add to a tree. For each edge she knows which vertices it would connect and the *beauty* of this edge. Masha can add some of these edges to the graph if the resulting graph is a cactus. *Beauty* of the resulting cactus is sum of beauties of all added edges.

Help Masha find out what maximum *beauty* of the resulting cactus she can achieve.

Input

The first line of the input data contains two integers n and m — the number of vertices in a tree, and the number of additional edges available ($3 \le n \le 2 \cdot 10^5$; $0 \le m \le 2 \cdot 10^5$).

Let us describe Masha's tree. It has a root at vertex 1. The second line contains $n - 1$ integers: $p_2, p_3, \ldots, p_n$, here p_i — is the parent of a vertex i — the first vertex on a path from the vertex i to the root of the tree ($1 \le p_i < i$).

The following m lines contain three integers u_i, v_i and c_i — pairs of vertices to be connected by the additional edges that Masha can add to the tree and *beauty* of edge ($1 \le u_i, v_i \le n$; $u_i \ne v_i$; $1 \le c_i \le 10^4$).

It is guaranteed that no additional edge coincides with the edge of the tree.

Output

Output one integer — the maximum *beauty* of a cactus Masha can achieve.

Example

input
7 3 1 1 2 2 3 3 4 5 1 6 7 1 2 3 1

output
2

E. Satellites

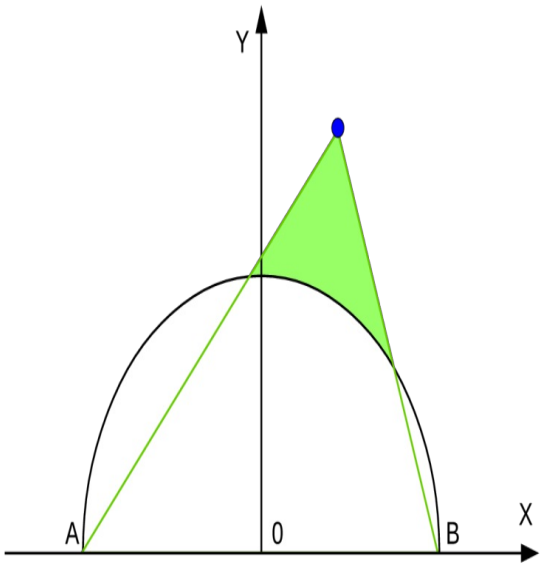
time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Real Cosmic Communications is the largest telecommunication company on a far far away planet, located at the very edge of the universe. RCC launches communication satellites.

The planet is at the very edge of the universe, so its form is half of a circle. Its radius is r, the ends of its diameter are points A and B. The line AB is the edge of the universe, so one of the half-planes contains nothing, neither the planet, nor RCC satellites, nor anything else. Let us introduce coordinates in the following way: the origin is at the center of AB segment, OX axis coincides with line AB, the planet is completely in $y > 0$ half-plane.

The satellite can be in any point of the universe, except the planet points. Satellites are never located beyond the edge of the universe, nor on the edge itself — that is, they have coordinate $y > 0$. Satellite antennas are directed in such way that they cover the angle with the vertex in the satellite, and edges directed to points A and B. Let us call this area the satellite *coverage area*.

The picture below shows coordinate system and coverage area of a satellite.

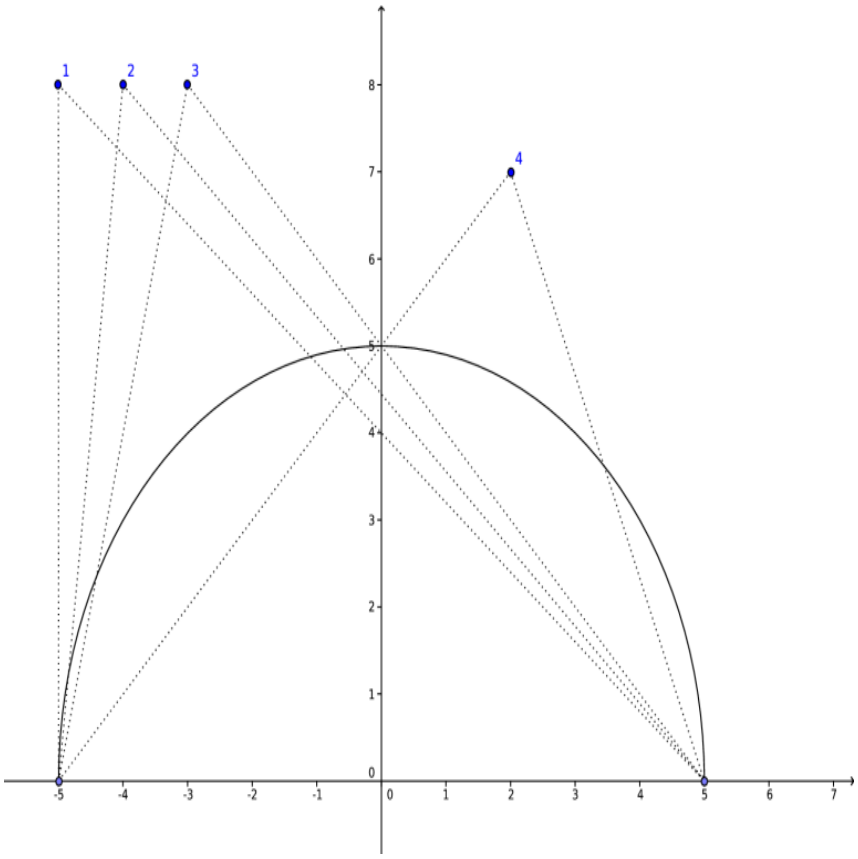


When RCC was founded there were no satellites around the planet. Since then there have been several events of one of the following types:

1. `1 x y` — launch the new satellite and put it to the point (x, y). Satellites never move and stay at the point they were launched. Let us assign the number i to the i-th satellite in order of launching, starting from one.
2. `2 i` — remove satellite number i.
3. `3 i j` — make an attempt to create a communication channel between satellites i and j. To create a communication channel a repeater is required. It must not be located inside the planet, but can be located at its half-circle border, or above it. Repeater must be in coverage area of both satellites i and j. To avoid signal interference, it must not be located in coverage area of any other satellite. Of course, the repeater must be within the universe, it must have a coordinate $y > 0$.

For each attempt to create a communication channel you must find out whether it is possible.

Sample test has the following satellites locations:



Input

The first line of input data contains integers r and n — radius of the planet and the number of events ($1 \le r \le 10^9$, $1 \le n \le 5 \cdot 10^5$).

Each of the following n lines describe events in the specified format.

Satellite coordinates are integer, the satisfy the following constraints $|x| \le 10^9$, $0 < y \le 10^9$. No two satellites that simultaneously exist can occupy the same point. Distance from each satellite to the center of the planet is strictly greater than r.

It is guaranteed that events of types 2 and 3 only refer to satellites that exist at the moment. For all events of type 3 the inequality $i \ne j$ is satisfied.

Output

For each event of type 3 print «YES» on a separate line, if it is possible to create a communication channel, or «NO» if it is impossible.

Example

input
5 8 1 -5 8 1 -4 8 1 -3 8 1 2 7 3 1 3 2 2 3 1 3 3 3 4
output
NO YES YES

F. To Play or not to Play

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya and Petya are playing an online game. As most online games, it has hero progress system that allows players to gain experience that make their heroes stronger. Of course, Vasya would like to get as many experience points as possible. After careful study of experience points allocation, he found out that if he plays the game alone, he gets one experience point each second. However, if two players are playing together, and their current experience values differ by at most C points, they can boost their progress, and each of them gets 2 experience points each second.

Since Vasya and Petya are middle school students, their parents don't allow them to play all the day around. Each of the friends has his own schedule: Vasya can only play during intervals $[a_1;b_1]$, $[a_2;b_2]$, ..., $[a_n;b_n]$, and Petya can only play during intervals $[c_1;d_1]$, $[c_2;d_2]$, ..., $[c_m;d_m]$. All time periods are given in seconds from the current moment. Vasya is good in math, so he has noticed that sometimes it can be profitable not to play alone, because experience difference could become too big, and progress would not be boosted even when played together.

Now they would like to create such schedule of playing that Vasya's final experience was greatest possible. The current players experience is the same. Petya is not so concerned about his experience, so he is ready to cooperate and play when needed to maximize Vasya's experience.

Input

The first line of input data contains integers n, m and C — the number of intervals when Vasya can play, the number of intervals when Petya can play, and the maximal difference in experience level when playing together still gives a progress boost ($1 \le n, m \le 2 \cdot 10^5$, $0 \le C \le 10^{18}$).

The following n lines contain two integers each: a_i, b_i — intervals when Vasya can play ($0 \le a_i < b_i \le 10^{18}$, $b_i < a_{i+1}$).

The following m lines contain two integers each: c_i, d_i — intervals when Petya can play ($0 \le c_i < d_i \le 10^{18}$, $d_i < c_{i+1}$).

Output

Output one integer — the maximal experience that Vasya can have in the end, if both players try to maximize this value.

Examples

input
2 1 5 1 7 10 20 10 20
output
25

input
1 2 5 0 100 20 60 85 90
output
125