Problem A. Decomposition

Input file:	`decomposition.in`
Output file:	`decomposition.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Consider a tree T. Let us denote *decomposition tree* of T as the rooted tree $D(T)$.

Let us choose any vertex r from the tree T. Consider the connected components of the tree T after delition of r: $S_1, S_2, \ldots, S_k$. The root of $D(T)$ is a vertex r and the children of r in $D(T)$ are $D(S_1), D(S_2), \ldots, D(S_k)$.

You are given the tree T. You have to find a decoposition tree of height less than 21. The height of the tree is the simple path of maximal length starting in root.

Input

The first line contains n — the number of vertices in the tree T ($1 \le n \le 2 \cdot 10^5$).

The next $n-1$ lines contain the edges of the tree. Each tree is described by the pair of integers v_i, u_i — the ends of an edgeконцы ребра ($1 \le v_i, u_i \le n$).

Output

Print n numbers: i-th number — the parent of the vertex i in the decomposition tree. If the vertex is a root, then print 0.

Examples

decomposition.in	decomposition.out
3 1 2 2 3	2 0 2
9 3 2 4 2 1 2 5 1 1 6 7 6 6 8 8 9	0 1 2 2 1 1 6 6 8

Problem B. Amazon River

Input file:	`amazon.in`
Output file:	`amazon.out`
Time limit:	5 seconds
Memory limit:	256 megabytes

N cities, enumerated from 1 to N, are located near the Amazon river. It is known that the Amazon forest is almost impassable and the only way to get from one city to another is to use the river. As a consequence, the connections between cities form a bidirectional tree. Unfortunately, this year the crab influenza is spreding in this region. And very fast all the hospitals except the hospital in the city 1 became overcrowded.

For the convenience of citizens you open a hotline. You need to write a program that could answer on the questions of people, considering the information of the working hospitals. You are lucky that you know all the questions in advance!

More formally, the queries could be of three different types:

- «`+` v» — the hospital in the city v could accept sick people again. It is guaranteed that exactly before this query the corresponding hospital was closed.
- «`-` v» — the hospital in the city v became overcrowded again. It is guaranteed that exactly before this query the corresponding hospital was open.
- «`?` v» — the person in the city v becomes sick. You need to tell him the distance to the nearest open hospital.

Input

The first line of the input contains one integer N — the number of cities ($1 \le N \le 300\,000$). The following $N-1$ lines contain the information about roads in the format «u v l» ($1 \le u, v \le N$, $1 \le l \le 1000$) — the road of length l between vertices u and v.

The next line contains one integer Q — the number of queries. The following Q lines contain the description of the queries in the format «c v», where c is one of symbols «`+`», «`-`» and «`?`» and v is the identifier of the city ($1 \le v \le N$).

Output

For each query of type «`?` v» you should print on a separate line one integer — the distance to the nearest city with open hospital.

Examples

amazon.in	amazon.out
5 1 2 2 2 3 3 3 4 1 3 5 4 5 ? 4 + 5 ? 3 - 1 ? 2	6 4 7

Problem C. Strings in Tree

Input file:	strings.in
Output file:	strings.out
Time limit:	3 seconds
Memory limit:	256 megabytes

You are given a tree. Each edge contains a lowercase latin letter. We take an ordered pair of vertices (u, v) and calculate the string by concatenation of the symbols on the edges in the order of the shortest oriented path from u to v.

You are also given a string s. Does there exist an ordered pair of vertices with the corresponding string equal to s?

Input

The first line of the input contain s — the string to find in the tree ($1 \le |s| \le 300\,000$).

The second line contain the integer n — the number of vertices in the tree ($1 \le n \le 300\,000$). The following $n-1$ lines contain the description of the edges. Each of these lines contains two integers v and u — the indices of the vertices connected by this edge, and the lowercase latin letter, written on the edge ($1 \le v, u \le n$).

Output

Print NO, if the string does not exist in the tree. Otherwise, print YES in the first line, and in the second line print the indices of the vertices, the path between which gives s.

Examples

strings.in	strings.out
abc 4 3 2 a 4 3 b 1 4 c	YES 1 2
abc 1	NO
zy 6 5 6 z 5 3 y 3 1 a 6 2 b 5 4 x	YES 3 6

Note

In the first example, the string abc corresponds to the path 3–4–2–1.

In the second example, the tree does not have edges.

The third example, the string zy corresponds to the path 3–1–4.

Problem D. LWDB

Input file:	lwdb.in
Output file:	lwdb.out
Time limit:	5 seconds
Memory limit:	256 megabytes

The Large Wood Database is created to securely store and paint any existing tree. Update for LWDB provides new functionality, so it is time to think over the graph theory. A weighed tree is stored in the LWDB. In the query language for LWDB Management System (LWDB MS) two types of queries are available:

1. «1 v d c» — paint all tree-vertices at the distance not exceeding d from the vertex v in color c. Initial color for any vertex is 0.
2. «2 v» — return the color of the vertex v.

It is required to prototype LWDB MS and respond to all user's queries.

Input

The first line contains an integer N ($1 \le N \le 10^5$) — the number of tree vertices. The following $N-1$ lines contain the description of branches, three numbers in each line a_i, b_i, w_i ($1 \le a_i, b_i \le N$, $a_i \ne b_i$, $1 \le w_i \le 10^4$), where i-th branch with weight w_i connects vertices a_i and b_i. The next line contains integer Q ($1 \le Q \le 10^5$) —

number of queries. Each of Q following lines contain two types of queries:

1. Numbers 1, v, d, c ($1 \le v \le N$, $0 \le d \le 10^9$, $0 \le c \le 10^9$).
2. Numbers 2, v ($1 \le v \le N$).

All numbers in input files are integers.

Output

For each query of the second type output the color of requested vertex in a separate line.

Examples

lwdb.in	lwdb.out
5 1 2 30 1 3 50 3 4 70 3 5 60 8 1 3 72 6 2 5 1 4 60 5 2 3 2 2 1 2 144 7 2 4 2 5	6 6 0 5 7

Problem E. Grisha and Festivals

Input file:	events.in
Output file:	events.out
Time limit:	9 seconds
Memory limit:	256 megabytes

Grisha has come to the planet Pandora. This planet contains n cities, connected by $n-1$ bidirectional roads of some lengths, in such a way that it is possible to travel between any pair of cities. Starting from Grisha's arrival at day 0, Pandora planet will have m festivals. It is known that i-th festival will be held in city c_i at day d_i.

He still had not decided the city in which he arrives at day 0, but he wants to visit as much festivals as possible. Could you tell him this maximal number?

Input

The first line of the input contains one integer n ($1 \le n \le 2 \cdot 10^5$) — the number of cities on Pandora.

Next $n-1$ lines contain the description of the roads, each road is defined by three integers a_i, b_i and l_i ($1 \le a_i, b_i, \le n$; $1 \le l_i \ge 10^9$) — the identifiers of the cities, connected by the road, and the length of the road in days.

The next line contains one integer m ($1 \le m \le 2 \cdot 10^5$) — the number of festivals on the planet.

Each of the next m lines contains two integers c_i и d_i ($1 \le c_i \le n$; $1 \le d_i \le 10^9$) — the identifier of the city and the day of the i-th festival.

Output

The sole line of the output should contain one integer — the maximal number of festival, that Grisha could visit.

Examples

events.in	events.out
4 1 2 1 2 3 1 2 4 3 4 1 3 2 4 3 1 4 5	3

Problem F. Snails

Input file:	snails.in
Output file:	snails.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

Two snails Masha and Petya are on the lawn with apricots and want to go back to their home. Lawns are numbered by integers from 1 to n and are connected by roads. Two lawns may be connected by many roads. It may happen, a road connects a lawn with itself. Due to reasons of hygiene, if one snail already went along the road, another one can not use this road. Help to Petya and Masha to get back home.

Input

The first line contains four integers n, m, s and t ($2 \le n \le 10^5, 0 \le m \le 10^5, s \ne t$) — the number of lawns, the number of roads, the index of a lawn with apricots, the index of a lawn with the home. Next m lines describe roads. Each line contains pair of integers (x, y), means that there is a road from lawn x to lawn y (due to the nature of snails and the space all roads are one-way).

Output

If there is a solution print YES and two lines with sequences of indices of lawns in the paths. At first output the path for Masha, then output the path for Petya. If

there is no solution print NO. If there are several solutions print any of them.

Examples

snails.in	snails.out
3 3 1 3 1 2 1 3 2 3	YES 1 3 1 2 3

Problem G. Full orientation

Input file:	orientation.in
Output file:	orientation.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

You are given an undirected graph without loops and multiedges. You have to make this graph directed in such a way, that maximal outgoing degree is minimal as possible.

Input

The first line contains two integers n and m ($1 \le n, m \le 25\,000$) — the number of vertices and edges in the graph. Next m lines contain two integers from 1 to n — the edges of the graph.

Output

The onlt line of the input should contain m integer numbers from 0 to 1, separated by space. If i-th edge is given by a pair (a, b), then zero means that after orientation the edge goes from a to b, and one means an orientation from b to a.

Examples

orientation.in	orientation.out
4 4 1 2 1 3 4 2 4 3	1 0 1 1 0
5 5 1 2 2 3 3 1 1 4 1 5	1 0 0 0 1 1

Problem H. Perspective

Input file:	perspective.in
Output file:	perspective.out
Time limit:	1 second
Memory limit:	256 Megabytes

Breaking news! A Russian billionaire has bought a yet undisclosed NBA team. He's planning to invest huge effort and money into making that team the best. And in fact he's been very specific about the expected result: the first place.

Being his advisor, you need to determine whether it's possible for your team to finish first in its division or not.

More formally, the NBA regular season is organized as follows: all teams play some games, in each game one team wins and one team loses. Teams are grouped into divisions, some games are between the teams in the same division, and some are between the teams in different divisions.

Given the current score and the total number of remaining games for each team of your division, and the number of remaining games between each pair of teams in your division, determine if it's possible for your team to score at least as much wins as any other team in your division.

Input

The first line of the input file contains N ($2 \le N \le 20$) — the number of teams in your division. They are numbered from 1 to N, your team has number 1.

The second line of the input file contains N integers w_1, w_2, ..., w_N, where w_i is the total number of games that i^{th} team has won to the moment.

The third line of the input file contains N integers r_1, r_2, ..., r_N, where r_i is the total number of remaining games for the i^{th} team (including the games inside the division).

The next N lines contain N integers each. The j^{th} integer in the i^{th} line of those contains a_{ij} — the number of games remaining between teams i and j. It is always true that $a_{ij} = a_{ji}$ and $a_{ii} = 0$, for all i $\sum_j a_{ij} \le r_i$.

All the numbers in the input file are non-negative and don't exceed 10 000.

Output

On the only line of output, print "YES" (without quotes) if it's possible for the team 1 to score at least as much wins as any other team of its division, and "NO" (without quotes) otherwise.

Examples

perspective.in	perspective.out
3 1 2 2 1 1 1 0 0 0 0 0 0 0 0 0	YES
3 1 2 2 1 1 1 0 0 0 0 0 1 0 1 0	NO

Note

Problem I. Vertex cover

Input file:	cover.in
Output file:	cover.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

You are given weighted bipartite graph. You have to find vertex cover of minimal weight. *Vertex cover* —- set of vertices which covers all edges (for each edge at least one of two edges is in the set). Weight of the vertex cover is equal to the sum of weights of all vertices in the set.

Input

The first line of the input contains one integer T — the number of tests.

Then T tests are given.

Each test is described by N ($1 \le N \le 100$) — the number of vertices of the first part. Next line contains the weights of vertices of the first part.

Then one integer M ($1 \le M \le 100$) is given — the number of vertices of the second part. Mext line contains the weights of vertices of the second line.

Later, you are given an integer E ($0 \le E \le 10^4$) — the number of edges in the graph. Each of the next E lines contains two integers u, v ($0 \le u \le N-1, 0 \le v \le M-1$) which describe an edge.

The sum of N, M and E does not exceed 30 000.

All weights are non-negative integers, not exceeding 10^9.

Output

For each test print n and m — the number of vertices in vertex cover in the left and right parts, correspondingly. The second line should contain the list of the vertices of the first part from vertex cover. The third line should contain the list of the vertices of the second part from vertex cover

If there are several optimal solutions, print any of them.

Examples

cover.in	cover.out
1 2 1 178 3 178 178 1 4 0 0 0 1 1 1 1 2	2 0 0 1

Problem J. Matan

Input file:	matan.in
Output file:	matan.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

In university of city M an experiment takes place. Tutors decide what to study during the course of lections.

One tutor studying calculus associated each theme of the course with an integer usefulness (possibly negative). Tutor wants to maximize total usefulness of all themes, though it's not easy. To make students understand anything you need to tell them some other themes since proofs of facts may use another facts as a pre-requisite.

For each theme you know the list of themes that it depends from. The cycles on dependencies are allowed, but in order to fully understand the theme you have to tell students all themes from the cycle (in the other words, the order of telling is not significant but the overall set of themes should satisfy all dependencies).

You have to find the set of themes that maximizes the total usefulness.

Input

The first line contains n ($1 \le n \le 200$), the number of themes. The second line contains n integers between -1000 and 1000, the usefulnesses of all themes.

The following n line describe dependencies between themes. The i-th line starts with k_i, the number of themes students should know in order to understand the i-th theme. After that k_i distinct integers between 1 and n follow, the indices of dependency themes.

The total number of all dependencies doesn't exceed 1800.

Output

Output the only number: the maximum total usefulness of all chosen themes.

Examples

matan.in	matan.out
4 -1 1 -2 2 0 1 1 2 4 2 1 1	2
3 2 -1 -2 2 2 3 0 0	0

Problem K. Reserve

Input file:	reserve.in
Output file:	reserve.out
Time limit:	2 seconds
Memory limit:	256 Megabytes

Protoss prepare a new attack. They have a lot of battle units to attack, but it is necessary to leave some Zealots (it is one of the most useful Protoss battle unit) on the base (as a reserve) to be sure that the base is protected from any counterattack...

One Zealot requires c_{ij} minerals to defend the base if employed from day i to day j. Attack operation proceeds for n days. Protoss commander Tryt o'Kill calculated that in day k he needs at least b_k Zealots on the base ($k = 1, 2, \ldots, n$).

The general can keep exactly b_k units on the base and pay them required number of minerals, but he wants to minimize the amount of minerals spent during all n days. The general can also keep more than b_k units on day k if this will result in smaller total amount of minerals.

Input

First line of the input file contains number of days n, $1 \le n \le 50$. Next n lines describe cost of keeping Zealots. So, j-th number in the $(i + 1)$-th line of the input file is $c_{i,i+j-1}$. All costs are non-negative and don't exceed 10 000. The last line contains b_k ($k = 1, 2, \ldots, n$). Consider $0 \le b_k \le 2000$.

All numbers in the input file are integral.

Output

In the first line write total amount of minerals General has to pay to Zealots to defend the base. Output a triangular table: j-th number of the $(i+1)$-th line should contain number of Zealots which have to defend the base from day i to day $i + j - 1$. This number shouldn't exceed 10^9. Keep in mind that Protoss General wants to minimize amount of minerals paid to Zealots.

It is guaranteed that the optimal amount of minerals doesn't exceed 10^9.

Examples

reserve.in	reserve.out
3 3 2 3 3 2 3 2 3 2	7 0 1 1 0 1 0