2024 Canadian Computing Olympiad
Day 1, Problem 1
Treasure Hunt

Time Limit: 4 seconds

Problem Description

Perry the Pirate is sailing the seven seas! He has a map consisting of N islands connected by a network of M sea routes. The i-th sea route connects islands a_i and b_i and costs c_i coins to traverse in either direction. As it turns out, fighting off sea monsters can be quite expensive. In search of his next big plunder, Perry has scouted out each of the N islands and has determined that the i-th island contains a treasure chest with v_i coins inside.

It remains for him to plan out his next journey. He decides that he will sail through some (possibly empty) path of sea routes starting at island x and ending at island y. At the end of his journey, he will open the chest at island y and collect his well-earned booty.

There is one small problem though: Perry doesn't know what island he's currently on! Thus, for every possible starting island x, he would like to know the maximum possible number of coins he can earn out of all journeys starting at island x. Can you help him compute these values? You may assume Perry has enough coins to traverse any path of sea routes he chooses; he only cares about the net profit of his next journey.

Input Specification

The first line of input contains two space-separated integers N and M.

The second line of input contains N space-separated integers $v_1, v_2, \ldots, v_N$ ($0 \le v_i \le 10^9$).

The next M lines each contain three space-separated integers a_i, b_i ($1 \le a_i, b_i \le N$), and c_i ($0 \le c_i \le 10^9$).

It is guaranteed that there is at most one sea route between any pair of islands and each sea route connects two distinct islands.

Marks Awarded	Bounds on N	Bounds on M	Additional constraints
5 marks	$1 \le N \le 3\,000$	$0 \le M \le 3\,000$	None
5 marks	$1 \le N \le 10^6$	$0 \le M \le 10^6$	For all i, either $c_i = 0$ or $c_i = 10^9$
7 marks	$1 \le N \le 10^6$	$0 \le M \le 10^6$	Exactly one path of sea routes between any pair of islands
8 marks	$1 \le N \le 10^6$	$0 \le M \le 10^6$	None

Output Specification

Output N lines, where the x-th line contains the maximum possible net profit (in coins) of any journey starting at island x.

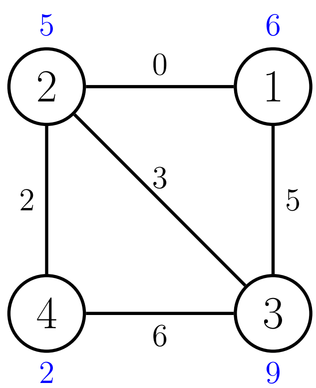
Sample Input

```
4 5
6 5 9 2
1 2 0
3 2 3
4 3 6
1 3 5
2 4 2
```

Output for Sample Input

```
6
6
9
4
```

Explanation of Output for Sample Input



For the first and third islands, it is best to just stay and open the chest on the island itself.

For the second island, Perry can travel to the first island and open the chest there. This has a net profit of $6 - 0 = 6$ coins and is the best possible net profit.

For the fourth island, Perry can to travel to the second and then the third island and open the chest there. This has a net profit of $9 - 2 - 3 = 4$ coins and is the best possible net profit.

2024 Canadian Computing Olympiad
Day 1, Problem 2
Pizza Party

Time Limit: 4 seconds

Problem Description

Troy is fleshing out the details of his latest initiative, HackCCO! Everyone knows that the biggest appeal of any hackathon is the free food. As such, to ensure the unparalleled success of HackCCO, Troy ordered a comically large cart stacked with N pizzas where the i-th pizza from the top of the cart has flavour a_i.

After the pizza cart arrives, Troy needs to arrange them into some number of stacks on a long table. To do this, he takes the pizzas one-by-one from the top of the cart and moves them onto the table, each time either placing the pizza on top of another stack of pizzas or forming a new stack on the table.

The N hungry HackCCO participants are lined up to get pizza from the table, one-by-one. Troy knows that the i-th person in line has a favourite pizza flavour of b_i. When the i-th person walks up to the table, if they see any pizzas of their favourite flavour at the top of any stack they will take any one of them at random. Otherwise, they won't take anything and will leave the table hungry.

Of course, hungry coders are not happy coders, so Troy doesn't want anyone to leave the table hungry. Thus, he asks you to help him find an arrangement of pizzas on the table such that it is possible for nobody to leave hungry. Furthermore, out of all such arrangements, Troy wants you to find one that creates the smallest number of stacks on the table (after all, tables can only get so long). Help him find such an arrangement or determine that it's impossible!

Input Specification

The first line of input contains a single integer N.

The second line of input contains N space-separated integers $a_1, \ldots, a_N$ $(1 \leq a_i \leq N)$.

The third line of input contains N space-separated integers $b_1, \ldots, b_N$ $(1 \leq b_i \leq N)$.

Marks Awarded	Bounds on N	Additional constraints
3 marks	$1 \leq N \leq 10^6$	$1 \leq a_i, b_i \leq 2$
4 marks	$1 \leq N \leq 5\,000$	All a_i are distinct
5 marks	$1 \leq N \leq 10^6$	All a_i are distinct
6 marks	$1 \leq N \leq 5\,000$	None
7 marks	$1 \leq N \leq 10^6$	None

[Post-CCO edit: Subtasks 4 and 5 may not be solvable efficiently. CCO competitors were judged only on subtasks 1-3.]

Output Specification

If it is impossible to arrange the pizzas as desired, output `-1`.

Otherwise, your output should consist of three lines. On the first line output K, the minimum number of stacks required. On the second line output N space-separated integers $c_1, \ldots, c_N$ ($1 \leq c_i \leq K$), indicating that the i-th pizza should be placed on stack c_i. On the third line output N space-separated integers $d_1, \ldots, d_N$ ($1 \leq d_i \leq K$), indicating that the i-th person in line takes their pizza from the d_i-th stack. This stack must have a pizza of flavour b_i at the top when the i-th participant walks up to get their pizza.

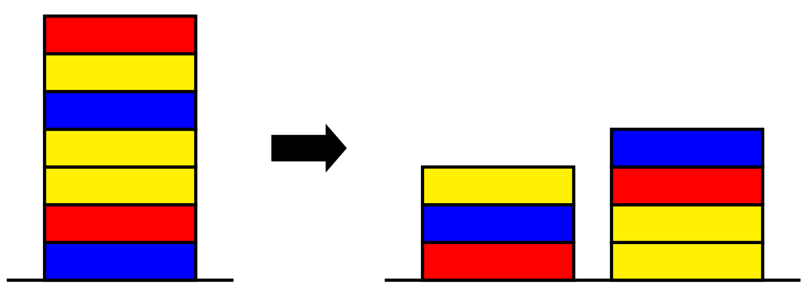
Sample Input 1

```
7
1 2 3 2 2 1 3
2 3 1 2 3 2 1
```

Output for Sample Input 1

```
2
1 2 1 2 1 2 2
1 2 2 2 1 2 1
```

Explanation of Output for Sample Input 1



An illustration of the arrangement of pizzas on the table is shown above where red, yellow, and blue boxes represent pizzas of flavours 1, 2, and 3 respectively.

Sample Input 2

```
2
1 2
1 1
```

Output for Sample Input 2

```
-1
```

2024 Canadian Computing Olympiad
Day 1, Problem 3
Summer Driving

Time Limit: 6 seconds

Problem Description

In Ontario, there are N cities numbered from 1 to N. There are $N-1$ roads numbered from 1 to $N-1$, where the i-th road connects city u_i and city v_i. It is possible to travel from any city to any other city using these roads.

Alice and Bob are travelling together, starting at city R. To make their driving experience more interesting, they devise the following game.

Alice and Bob will alternate turns, starting with Alice. On Alice's turn, she must drive along **exactly** A distinct roads that they have never traversed before in either direction. On Bob's turn, he must drive along **up to** B distinct roads (possibly zero), but some of these roads may have been traversed before.

Eventually, it will be Alice's turn, but it will be impossible for her to drive along exactly A distinct roads that they have never used before. When this happens, the game enters a final phase before Alice does any more driving. In this final phase, Bob drives along **up to** B distinct roads (possibly zero) that they have **never traversed before** in either direction.

Alice wants to end up in a city with as large a number as possible, while Bob wants to end up in a city with a small number. What is the city that Alice and Bob end their journey in when they both play optimally?

Input Specification

The first line of input contains four space-separated integers, N, R, A, and B ($1 \leq R, A, B \leq N$).

The next $N-1$ lines of input each contain two space-separated integers u_i and v_i ($1 \leq u_i, v_i \leq N, u_i \neq v_i$), describing a road.

Marks Awarded	**Bounds on N**	**Additional Constraints**
1 mark	$2 \leq N \leq 300\,000$	$A \leq B$
4 marks	$2 \leq N \leq 300\,000$	There are at most two roads connected to any city, and at most one road connected to city R.
5 marks	$2 \leq N \leq 300$	No additional constraints.
3 marks	$2 \leq N \leq 3\,000$	No additional constraints.
4 marks	$2 \leq N \leq 100\,000$	$B \leq 10$
6 marks	$2 \leq N \leq 100\,000$	No additional constraints.
2 marks	$2 \leq N \leq 300\,000$	No additional constraints.

Output Specification

Output the city that Alice and Bob end their journey in, assuming they both play optimally.

Sample Input 1

```
9 6 2 1
1 3
1 6
2 4
2 5
2 7
3 9
4 6
4 8
```

Output for Sample Input 1

```
2
```

Explanation of Output for Sample Input 1

On Alice's first turn, she has three options: Drive to city 2, 3, or 8.

If Alice drives to city 2, Bob can choose not to drive on any roads in his turn. Then, it will be impossible for Alice to drive along 2 distinct roads that were never traversed before, so the game enters the final phase. Bob can choose not to drive on any roads during this final phase, ending in city 2.

If Alice drives to city 3, Bob can choose to drive 1 road to city 1. Then, it will be impossible for Alice to drive along 2 distinct roads that were never traversed before, so the game enters the final phase. Bob's only option is to not drive on any roads during this final phase, ending in city 1.

If Alice drives to city 8, Bob can choose to drive 1 road to city 4. Then, Alice can drive to either city 5 or 7. In both cases, Bob can then drive 1 road to city 2. Alice can no longer drive along 2 distinct roads that were never traversed before, so the game enters the final phase. Bob can choose not to drive on any roads during this final phase, ending in city 2.

In all cases, Bob can force the game to end in city 1 or 2. Bob cannot force the game to end in city 1, so under optimal play, the game ends in city 2.

Sample Input 2

```
7 2 3 2
2 7
7 3
3 1
1 4
```

```
4 5
5 6
```

Output for Sample Input 2

```
3
```