2023 Canadian Computing Olympiad
Day 2, Problem 1
Flip it and Stick it

Time Limit: 1 second

Problem Description

Finn is playing a game of "Flip it and Stick it" which is abbreviated as FiSi. FiSi is a one-player game played on two strings, S and T, of `0`s and `1`s. Finn is allowed to make moves of the following form:

- Select a substring of S and reverse it, gluing the pieces of the string back together in their original order to form the new string S.

For example, Finn may take the string $S =$ `101100`, take the substring `011` starting at index 2 (assuming 1-based string indexing), and create the string $S =$ `111000` in one move.

Finn wins the game if S does **not** contain T as a substring. Your task is to help Finn determine the length of the shortest winning sequence of moves or tell him that the game cannot be won.

Input Specification

The first line of input contains the string S ($1 \leq |S| \leq 200\,000$).

The second line of input contains the string T ($1 \leq |T| \leq 3$).

In the table below, T_1 is the first bit in T, T_2 is the second bit in T, and T_3 is the third bit in T, when reading from left-to-right.

Marks Awarded	**Bounds on T**
1 mark	$\|T\| = 1$
3 marks	$\|T\| = 2, T_1 \neq T_2$
4 marks	$\|T\| = 2$
5 marks	$\|T\| = 3, T_1 \neq T_3$
5 marks	$\|T\| = 3, T_1 \neq T_2$
7 marks	$\|T\| = 3$

Output Specification

Output the minimum number of moves needed or -1 if it is impossible to win the game.

Sample Input 1

```
100110
10
```

Output for Sample Input 1

```
2
```

Explanation of Output for Sample Input 1

Finn starts with the string `100110`. He cannot avoid `10` as a substring in one move, but he can in two moves.

For example, his first move could be to reverse the substring from index 4 to index 6 (`110`) to get `100011`. Then, his second move can be to reverse the substring from index 1 to index 4 (`1000`) to get `000111`, which does not have `10` as a substring.

Sample Input 2

```
000
00
```

Output for Sample Input 2

```
-1
```

Explanation of Output for Sample Input 2

No matter how many moves Finn makes, the string S will always contain T as a substring.

2023 Canadian Computing Olympiad
Day 2, Problem 2
Travelling Trader

Time Limit: 2 seconds

Problem Description

A trader would like to make a business of travelling between cities, moving goods from one city to another in exchange for a profit. There are N cities labelled $1, \ldots, N$ and $N-1$ roads. Each road joins two cities and takes one day to traverse. It is possible to reach any city from any other city using these roads.

The i-th city can give a profit of p_i if the trader is currently in that city and chooses to do business in that city, but this profit may only be obtained once. The trader starts by doing business in city 1 and wants to travel along the roads, visiting cities to maximize their total profit. However, the trader's boss will get unhappy and lay off the trader as soon as the trader goes more than K days in a row without increasing their total profit. Note that the trader will take only one day to move between adjacent cities, regardless of whether the trader does business in either city. We would like to know the maximum profit the trader can make under this condition and a route that obtains this profit.

Input Specification

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

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.

The last line of input contains N integers $p_1, \ldots, p_N$ ($1 \leq p_i \leq 10^9$), the profits given by choosing to do business in the corresponding city.

Marks Awarded	**Bounds on N**	**Bounds on K**
2 marks	$2 \leq N \leq 200\,000$	$K = 1$
7 marks	$2 \leq N \leq 200$	$K = 2$
3 marks	$2 \leq N \leq 2\,000$	$K = 2$
4 marks	$2 \leq N \leq 200\,000$	$K = 2$
4 marks	$2 \leq N \leq 2\,000$	$K = 3$
5 marks	$2 \leq N \leq 200\,000$	$K = 3$

Output Specification

On the first line, output the maximum possible total profit.

On the second line, output M ($1 \leq M \leq N$), the number of cities the trader does business in on an optimal route.

On the third line, output M space-separated integers $x_1, \ldots, x_M$, the cities the trader does business in on an optimal route in order, starting with $x_1 = 1$.

If there are multiple possible correct outputs, any correct output will be accepted.

Sample Input 1

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

Output for Sample Input 1

```
7
2
1 3
```

Explanation of Output for Sample Input 1

On day 1, the trader starts by doing business in city 1, making a profit of 3.

On day 2, the trader moves to city 3 and does business there, making a profit of 4.

At this point, the trader cannot reach another city in which they have not done business before getting laid off, so their total profit is 7.

Sample Input 2

```
5 2
1 2
1 3
2 4
2 5
3 1 4 1 5
```

Output for Sample Input 2

```
14
5
1 4 5 2 3
```

Explanation of Output for Sample Input 2

The trader can make a profit in every city by visiting them in the order $1, 2, 4, 2, 5, 2, 1, 3$.

Note that the trader strategically delays doing business in city 2 to ensure they do not go more than 2 days without making a profit.

2023 Canadian Computing Olympiad
Day 2, Problem 3
Triangle Collection

Time Limit: 4 seconds

Problem Description

Alice has a collection of sticks. Initially, she has c_ℓ sticks of length ℓ for each $\ell = 1, \ldots, N$.

Alice would like to use her sticks to make some isosceles triangles. An isosceles triangle is made of two sticks of the same length, say ℓ, and a third stick with a length between 1 and $2\ell - 1$ inclusive. Note that the triangles must strictly obey the triangle inequality, and equilateral triangles are okay. Each stick may be used in at most one triangle. Alice would like to know the maximum number of isosceles triangles she can make with her sticks.

There are Q events that change the collection of sticks she has. The i-th event consists of two integers ℓ_i and d_i, representing that the number of sticks of length ℓ_i changes by d_i. Note that d_i may be positive, negative, or even 0, but Alice will never have a negative number or more than 10^9 sticks of each length.

Your task is to determine the maximum number of isosceles triangles Alice can make after each event if she uses her sticks optimally.

Input Specification

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

The second line of input contains N space-separated integers $c_1, c_2, \ldots, c_N$ $(0 \le c_i \le 10^9)$, representing Alice's initial collection.

The next Q lines of input each contain two space-separated integers ℓ_i and d_i $(1 \le \ell_i \le N, -10^9 \le d_i \le 10^9)$, representing an event.

Initially and after each event, the number of sticks of length ℓ is between 0 and 10^9 for all $\ell = 1, \ldots, N$.

Marks Awarded	Bounds on N, Q	Additional Constraints
5 marks	$1 \le N, Q \le 2\,000$	There are at most 2 000 sticks in total initially and after each event.
5 marks	$1 \le N, Q \le 2\,000$	No additional constraints.
5 marks	$1 \le N, Q \le 200\,000$	The number of sticks of each length is either 0, 1, or 2 initially and after each event.
5 marks	$1 \le N, Q \le 200\,000$	For each event, $\lvert d_i \rvert = 1$.
5 marks	$1 \le N, Q \le 200\,000$	No additional constraints.

Output Specification

Output Q lines each containing a single integer, the answer after each event.

Sample Input

```
4 3
3 1 4 1
3 -3
1 6
2 1
```

Output for Sample Input

```
1
3
4
```

Explanation of Output for Sample Input

After the first event, Alice can make a single triangle with sticks of lengths $(1, 1, 1)$.

After the second event, Alice can make 3 triangles with sticks of lengths $(1, 1, 1)$.

After the third event, Alice can make 3 triangles with sticks of lengths $(1, 1, 1)$ and a triangle with sticks of lengths $(2, 2, 3)$.