E. Little Pony and Summer Sun Celebration

Twilight Sparkle learnt that the evil Nightmare Moon would return during the upcoming Summer Sun Celebration after one thousand years of imprisonment on the moon. She tried to warn her mentor Princess Celestia, but the princess ignored her and sent her to Ponyville to check on the preparations for the celebration.

Twilight Sparkle wanted to track the path of Nightmare Moon. Unfortunately, she didn't know the exact path. What she knew is the parity of the number of times that each place Nightmare Moon visited. Can you help Twilight Sparkle to restore any path that is consistent with this information?

Ponyville can be represented as an undirected graph (vertices are places, edges are roads between places) without self-loops and multi-edges. The path can start and end at any place (also it can be empty). Each place can be visited multiple times. The path must not visit more than $4n$ places.

Input

The first line contains two integers n and m ($2 \le n \le 10^5$; $0 \le m \le 10^5$) — the number of places and the number of roads in Ponyville. Each of the following m lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n$; $u_i \ne v_i$), these integers describe a road between places u_i and v_i.

The next line contains n integers: $x_1, x_2, ..., x_n$ ($0 \le x_i \le 1$) — the parity of the number of times that each place must be visited. If $x_i = 0$, then the i-th place must be visited even number of times, else it must be visited odd number of times.

Output

Output the number of visited places k in the first line ($0 \le k \le 4n$). Then output k integers — the numbers of places in the order of path. If $x_i = 0$, then the i-th place must appear in the path even number of times, else i-th place must appear in the path odd number of times. Note, that given road system has no self-loops, therefore any two neighbouring places in the path must be distinct.

If there is no required path, output -1. If there multiple possible paths, you can output any of them.

Examples

input
```
3 2
1 2
2 3
1 1 1
```
output
```
3
1 2 3
```

input
```
5 7
1 2
1 3
1 4
1 5
3 4
3 5
4 5
0 1 0 1 0
```
output
```
```

```
10
2 1 3 4 5 4 5 4 3 1
```