Special Spud Stalls

Input File: *standard input*
Output File: *standard output*

Time limit	Memory limit
1 second	512 MB

Statement

In Spudheart City, there are S spud stalls positioned at S intersections numbered 1 to N. Stall i is located on intersection i. Some of these intersections may be connected to each other by a network of R roads. Two stalls are considered adjacent if there is a road that connects the two intersections they are on.

The i-th stall currently sells p_i varieties of potatoes.

A stall will not attract any customers unless it is special. A stall is special if the number of potato varieties it sells modulo M is different to all adjacent stalls.

It is in the interest of the Spudheart City council to ensure each stall can attract customers. The council has the power to multiply the number of potato varieties a stall sells by a constant C. However, this is expensive and some stalls may complain if another stall receives too much support from the council. Therefore, the council may use their powers **at most once per stall**.

Is it possible for the council to use their powers so all spud stalls will attract customers. If so, which stalls should the council use their powers on?

Input

The first line of input contains the four space separated integers S, R, M, C. S lines follow, the i-th of which contains the integer p_i. R more lines follow, the i-th of which contains the integers a_i and b_i, indicating there exists a road connecting intersections a_i and b_i.

Output

If it is not possible for all stalls to attract customers, output -1. Otherwise, the first line of output should contain a single integer: the number of times the council used their powers. On the next line, list the stalls that the council used their powers on in any order. If there are multiple valid lists, output any of them.

Sample Input 1

```
6 7 10 5
3
12
2
0
3
5
1 2
1 3
1 5
2 3
2 4
3 6
5 6
```

Sample Output 1

```
2
1 3
```

Sample Input 2

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

Sample Output 2

```
-1
```

Sample Input 3

```
5 4 2 1
1
4
1
5
0
2 1
3 2
4 5
2 4
```

Sample Output 3

```
0
```

Explanation

See the included file "interactive.py"

Constraints

- $2 \le S \le 10^6$
- $0 \le R \le 10^6$
- $2 \le M \le 10^9$
- $0 \le C \le 10^9$
- $0 \le p_i \le 10^9$ for all i
- $1 \le a_i, b_i \le S$ for all i
- $a_i \ne b_i$ for all i
- There are no duplicate roads

Subtasks

- For Subtask 1 (15 points), Every stall is adjacent to every other stall. In other words, the intersections and roads form a complete graph.
- For Subtask 2 (17 points), $S \le 16$.
- For Subtask 3 (24 points), $M = 3, C = 2$ and $p_i = 1$ or 2 for all i.
- For Subtask 4 (29 points), $R = S - 1$ and each intersection is reachable from every other intersection by a sequence of 1 or more roads. In other words, the intersections and roads form a tree.
- For Subtask 5 (15 points), No additional constraints apply.

Note: You may want to avoid deep recursion to prevent stack overflow. Additionally, you may want to use a 64-bit integer type for intermediate calculations.