Problem F. Graph Stabilization

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

Consider an undirected graph with n vertices, where the i-th vertex has the integer a_i written on it. Two vertices are neighbours if there exists an edge connecting them.

We call a vertex i *stable* if it has a neighbour j such that at least one of the following conditions holds:

- $a_j < a_i$;
- $a_j \oplus a_i = k$, where $\oplus$ denotes the bitwise XOR operator.

If a vertex i does not have any such neighbours, then i is *unstable*.

The process of stabilization of the graph is denoted as follows:

1. If there are no unstable vertices in the graph, end the process. Otherwise, go to step 2.
2. Let v be the unstable vertex with the minimum index. Remove v and all edges incident to it from the graph, then go to step 1.

Let the *stability* of the graph be the number of vertices left in the graph after it is stabilized.

You are given an array a, an integer k, and an undirected graph, where each edge has a cost associated with it. You may remove any edges from the graph (possibly none, possibly all of them). Your goal is to remove edges in such a way that:

- the resulting graph's stability is the maximum possible;
- the total cost of removed edges is the maximum possible (among all ways which result in the maximum possible stability).

Input

The first line contains three integers n, m, k ($1 \le n \le 100$; $0 \le m \le \frac{n(n-1)}{2}$; $1 \le k \le 127$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 127$).

Then n lines follow, the i-th of them contains three integers x_i, y_i, c_i ($1 \le x_i, y_i \le n$; $x_i \ne y_i$; $1 \le c_i \le 10^9$) — the endpoints of the i-th edge and its cost.

Additional constraint on the input: every pair of vertices is connected by at most one edge.

Output

First, print one integer e ($0 \le e \le m$) — the number of edges you remove. Then, print e distinct integers from 1 to m in any order — the indices of those edges. If there are multiple optimal answers, you may print any of them.

Examples

standard input	standard output
5 6 1 0 1 2 2 2 1 2 1 1 3 2 2 3 3 3 4 4 3 5 5 4 5 6	4 3 4 5 6
8 10 3 34 5 6 33 4 7 1 4 1 3 11 4 1 15 8 6 10 5 7 1 5 6 10 5 2 5 2 6 2 3 2 4 5 3 1 4 2 2	5 2 4 6 7 9