

## Problem D. Dull game

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:           **1 second**  
Memory limit:        **256 megabytes**

Nim is a game in which two players take turns removing objects from distinct heaps. On each turn, a player must remove at least one object, and may remove any number of objects provided they all come from the same heap. The player who takes the last item wins.

You are given heaps with sizes  $a_0, a_1, \dots, a_n$ . Find the heaps subsequence  $S$  which satisfies the following requirements:

- If we optimally play the “Nim” game using heaps from this subsequence, the first player loses.
- $a_0 \in S$ .

You should also process  $m$  queries for modifying the sequence of heaps: the  $i$ -th query  $p_i, x_i$  means that from now on  $a_{p_i} = x_i$ . After each request, it is necessary to find the subsequence  $S_i$  that satisfies the requirements described above.

It is guaranteed that initially, as well as after each modification query, the number of matching subsequences  $S$  heaps will be **exactly one**.

### Input

The first line of input data contains two space-separated integers  $n$  and  $m$ .

The second line of input data contains  $n + 1$  space-separated integers  $a_0, a_1, \dots, a_n$ .

The following  $m$  lines contain modification queries descriptions,  $i$ -th line contains two space-separated integers  $t_i, x_i$ . To calculate  $p_i$  you have to know the answer for **the previous** question. Namely, if you printed numbers  $k, b_1, b_2, \dots, b_k$  as the answer of the previous question, then  $p_i = t_i \oplus k \oplus b_1 \oplus b_2 \oplus \dots \oplus b_k$ .

$$2 \leq n \leq 1000$$

$$0 \leq m \leq 1000$$

$$0 \leq a_0, a_1, \dots, a_n < 2^n$$

$$0 \leq p_i \leq n$$

$$0 \leq x_i < 2^n$$

### Output

In the first line print integer  $k$  – the size of subsequence  $S$ .

In the second line print the sequence of space-separated integers  $b_1 < b_2 < \dots < b_k$  where  $S = a_{b_1}, a_{b_2}, \dots, a_{b_k}$ .

After each  $i$ -th query print the subsequence  $S_i$  in the same format.

## Examples

standard input	standard output
3 1 5 6 2 7 0 3	3 0 2 3 3 0 1 2
3 2 1 2 3 4 0 2 2 6	3 0 1 2 2 0 1 3 0 1 3