Problem C. Longest beautiful sequence

Input file:	`subsequence.in`
Output file:	`subsequence.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

You're given two sequences of n nonnegative integer numbers: $a_1, a_2, \ldots, a_n$ and $k_1, k_2, \ldots, k_n$. The sequence of m integer numbers $i_1, i_2, \ldots, i_m$ is called *beautiful* if it meets with following criteria:

- $1 \le i_1 < i_2 < \ldots < i_m \le n$. In other words, sequence must be increasing.
- $bitCount(a_{i_{j-1}} \text{ AND } a_{i_j}) = k_{i_j}$ for all $1 < j \le m$.

Find longest *beautiful* sequence.

Input

On first line of input given positive integer number n ($1 \le n \le 10^5$) — the length of sequences a and k. Second line of input contains n nonnegative integer numbers a_i ($0 \le a_i < 2^{20}$) — sequence a. Third line of input contains n nonnegative integer numbers k_i ($0 \le k_i \le 20$) — sequence k. Numbers in both sequences are separated by single spaces.

Output

On first line of output print out one integer number m — length of longest *beautiful* sequence. On second line print out m integers — longest *beautiful* sequence, separated by single spaces. If there is multiple solutions, print any of them.

Scoring

This problem consists of four subtasks:

1. $1 \le n \le 15$, $0 \le a_i < 2^{20}$. This subtask worths 7 points.
2. $1 \le n \le 5000$, $0 \le a_i < 2^{20}$. This subtask worths 16 points.
3. $1 \le n \le 10^5$, $0 \le a_i < 2^8$. This subtask worths 17 points.
4. $1 \le n \le 10^5$, $0 \le a_i < 2^{20}$. This subtask worths 60 points.

Each subtask will be scored only if the solution successfully passes all of the previous subtasks.

Examples

subsequence.in	subsequence.out

4
1 2 3 4
10 0 1 0
	4
	1 2 3 4

2
8 9
20 0
	1
	1

5
5 3 5 3 5
10 1 20 1 20
	2
	1 2

Note

$bitCount(x)$ — number of ones in binary representation, e.g. $bitCount(5_{10}) = bitCount(101_2) = 2$, $bitCount(0) = 0$, $bitCount(8) = 1$.

AND — is a binary operation, which takes two equal-length binary representations and performs the logical AND operation on each pair of the corresponding bits, e.g. 11_{10} AND 13_{10} = 1011_2 AND $1101_2 = 1001_2 = 9$, 7_{10} AND $16_{10} = 111_2$ AND $10000_2 = 0_2 = 0_{10}$.