Problem A. Bootfall

Input file:	`bootfall.in`
Output file:	`bootfall.out`
Time limit:	1 second
Memory limit:	256 megabytes

Tima and his N friends love to play *Bootfall*. *Bootfall* — is a sport game for $N+1$ players. Each player has a strength, which can be represented as a positive integer number. Game consists of $N+1$ rounds, in each round one of the players will record the round on video and rest of N players divides into two teams, such that each player will be assigned to one of the teams and both teams are non-empty. Strength of the team is sum of the strengths of all players in team. Also, each player will be recorder in **exactly one** round.

Round called *draw* if exists division into two teams with **equal** strength, also whole game called *friendly* if **all rounds** are *draw*. Each of N friends are already informed Tima about their strength, and now Tima can assign himself any valid value of strength.

Tima know the strengths of all N players and he will choose some value, such that game can be *friendly*. Help him to find all possible strength.

Input

First line of input contains one positive integer number N ($1 \le N \le 500$) — the number of friends of Tima. Second line of input contains N positive integer numbers $a_1, a_2, \ldots, a_N$ ($1 \le a_i \le 500$; $1 \le i \le N$) separated with space, a_i — strength of i-th person.

Output

First line of output must contain one integer number K — number of possible strength for Tima. If there is no possible strength for Tima, then print only "0" (without quotes), otherwise on second line of output print K positive integer numbers separated by space — all possible strength values for Tima in **increasing** order.

Scoring

This problem consists of six subtasks:

1. $1 \le N \le 12$, $1 \le a_i \le 200$, for all $1 \le i \le N$. Score 6 points.
2. $1 \le N \le 30$, $1 \le a_i \le 20$, for all $1 \le i \le N$. Score 7 points.
3. $1 \le N \le 100$, $1 \le a_i \le 100$, for all $1 \le i \le N$. Score 15 points.
4. $1 \le N \le 270$, $1 \le a_i \le 270$, for all $1 \le i \le N$. Score 16 points.
5. $1 \le N \le 350$, $1 \le a_i \le 350$, for all $1 \le i \le N$. Score 21 points.
6. $1 \le N \le 500$, $1 \le a_i \le 500$, for all $1 \le i \le N$. Score 35 points.

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

Examples

bootfall.in	bootfall.out
4 1 3 1 5	1 3
6 3 5 7 11 9 13	4 1 3 17 19
3 2 2 2	0
4 200 200 200 200	2 200 600

Note

Notes to first sample test.
Let us show, that if Tima selects strength 3 then the game can be *friendly*.
— When Tima will record game, to make round *draw* other gamers may be divided as follows : $(1, 3, 1)$ in the first team, and (5) in second.
— When friend 1 will record game, others may be divided as follows: $(1, 5)$ in the first team, $(3, 3)$ in the second.
— When friend 2 will record game, others may be divided as follows: $(1, 1, 3)$ in the first team, (5) in the second.
— When friend 3 will record game, others may be divided as follows: $(3, 3)$ in the first team, $(1, 5)$ in the second.
— When friend 4 will record game, others may be divided as follows: $(1, 3)$ in the first team, $(1, 3)$ in the second.
If Tima selects strength not equal to 3, then the game cannot be *friendly*.

Problem B. Money

Input file:	`money.in`
Output file:	`money.out`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

AlanashKO loves money. On the eve of the New Year he was given N banknotes. Nominal value of each banknote is a positive integer. While playing, AlanashKO lay out all banknotes in a row and numbered them from the left to right from 1 to N. Then he decided to sort all banknotes in **nondecreasing** order. For this AlanashKO doing following procedure: at first, he divides banknotes into one or more **disjoint** subsegments and **each** banknote belongs to some subsegment. Then, all subsegments in order from the left to right in turn are inserted into new row, i.e. at the first left-most subsegment (first subsegment) is inserted, then the next left-most inserted and so on. Each subsegment is inserted either between any two banknotes or at the one of the two ends of the current new row. The order of banknotes within the subsegment is not changed when inserted.
AlanashKO wants to minimize the number of subsegments so that he could finally sort the banknotes in **nondecreasing** order. Help him to find this value.

Input

The first line contains single positive integer N ($1 \le N \le 10^6$) — the number of banknotes. The next line contains N positive integers a_i ($1 \le a_i \le 10^6$) — the value of i-th banknote.

Output

In the single line print single integer — the minimal number of subsegments allowing AlanashKO to sort banknotes.

Scoring

This task includes four subtasks:

1. $N \le 8$. Score 9 points.
2. $N \le 20$. Score 16 points.
3. $N \le 300$. Score 20 points.
4. $N \le 10^6$. Score 55 points.

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

Example

money.in	money.out
6 3 6 4 5 1 2	3

Note

Subsegment is consecutive sequence.
Let us consider sample test:
The minimal answer is division into 3 subsegments: `|3 6|4 5|1 2|` (sticks — subsgments borders)
After first step: initial row is `|4 5|1 2|`, new row is `|3 6|`.
On second step subsegment `|4 5|` inserted between 3 and 6.
After second step: initial row `|1 2|`, new row is: `|3 4 5 6|`.
Then, subsegment `|1 2|` inserted at the beginning of new row with a result `|1 2 3 4 5 6|`.

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 $10000_2 = 0_2 = 0_{10}$.