Monsters

Problem Statement

Tom is playing a TCG (trading card game). Each monster card is characterised by its power level, a positive integer. Note that multiple different monsters can have the same power level.

The game has multiple rounds (starting from round 1). On round *k* Tom searches his hand of monster cards from left to right. If he finds a pair of cards, both with power *k* (the leftmost such pair), he discards the second card of the pair. Then, he merges its power into the first card, raising its power to 2*k*.

The game stops when Tom can no longer upgrade his hand, in any possible future round. Can you help Tom figure out what his final hand will be?

Note: For the purposes of this lab, you may not use Java API TreeSet, TreeMap, or any other implementations of binary search trees.

Input

The first line of input consists of a single integer N ($1 <= N <= 2 \times 10^5$), the number of cards. The second line of input consists of N integers c_1 to c_n ($1 <= c_i <= 10^9$), representing the values on the cards, going from left to right.

Output

In the first line, print out the length of the final hand of cards. Then, in the second line, print out the values on the remaining cards, going left to right.

Sample Input 1

```
8
4 3 1 2 2 1 1 8
```

Sample Output 1

```
4
16 3 2 1
```

Sample Input 2

```
7
1 2 3 5 8 13 100000
```

Sample Output 2

```
7
1 2 3 5 8 13 100000
```