Card Game

This question is graded for 0.5%!

Statement

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

The game has multiple rounds (starting from round $1, 2, 3, \ldots$). On round k, Fluffy 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 $2k$).

The game stops when Fluffy can no longer upgrade his hand, in any possible future round.

Can you help Fluffy figure out what his final hand would be?

Input

The first line of input consists of a single integer N, the number of cards. The second line of input consists of N integers $c_1\ c_2\ \ldots\ c_N$, the values on the cards, going left to right.

Constraints

- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq c_i \leq 10^9$

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.

Important Note

For the purposes of this lab, you may **not** use Java API `TreeSet`, `TreeMap`, or any other implementations of binary search trees. **Submissions which violate this rule will be given a score of 0.**

Examples

Sample Input	Expected Output
8 4 3 1 2 2 1 1 8	4 16 3 2 1
7 1 2 3 5 8 13 100000	7 1 2 3 5 8 13 100000
5 2 2 5 2 2	2 8 5

Notes

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided.

Skeleton File

You are given the skeleton file `CardGame.java`. You should see the following contents when you open the file:

```
/**
 * Name          :
 * Matric. No    :
*/

import java.util.*;

public class CardGame {
    private void run() {
        // implement your "main" method here
    }

    public static void main(String args[]) {
```

```
        CardGame runner = new CardGame();
        runner.run();
    }
}
```