Problem I
Hidden Tree

Input: Standard Input
Time Limit: 20 seconds

Consider a binary tree whose leaves are assigned integer weights. Such a tree is called *balanced* if, for every non-leaf node, the sum of the weights in its left subtree is equal to that in the right subtree. For instance, the tree in the following figure is balanced.

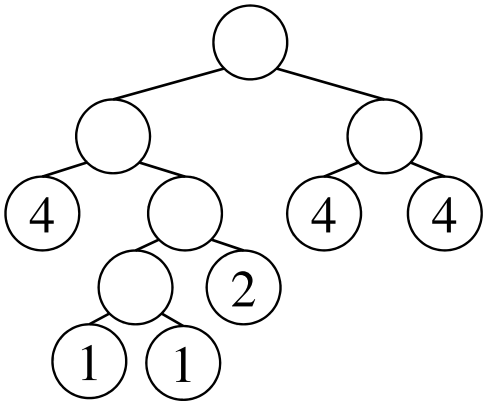


Figure I.1. A balanced tree

A balanced tree is said to be *hidden* in a sequence A, if the integers obtained by listing the weights of all leaves of the tree from left to right form a subsequence of A. Here, a subsequence is a sequence that can be derived by deleting zero or more elements from the original sequence without changing the order of the remaining elements.

For instance, the balanced tree in the figure above is hidden in the sequence 3 4 1 3 1 2 4 4 6, because 4 1 1 2 4 4 is a subsequence of it.

Now, your task is to find, in a given sequence of integers, the balanced tree with the largest number of leaves hidden in it. In fact, the tree shown in Figure I.1 has the largest number of leaves among the balanced trees hidden in the sequence mentioned above.

Input

The input consists of multiple datasets. Each dataset represents a sequence A of integers in the format

N
$A_1\ A_2\ \ldots\ A_N$

where $1 \leq N \leq 1000$ and $1 \leq A_i \leq 500$ for $1 \leq i \leq N$. N is the length of the input sequence, and A_i is the i-th element of the sequence.

The input ends with a line consisting of a single zero. The number of datasets does not exceed 50.

Output

For each dataset, find the balanced tree with the largest number of leaves among those hidden in A, and output, in a line, the number of its leaves.

Sample Input

```
9
3 4 1 3 1 2 4 4 6
4
3 12 6 3
10
10 9 8 7 6 5 4 3 2 1
11
10 9 8 7 6 5 4 3 2 1 1
8
1 1 1 1 1 1 1 1
0
```

Output for the Sample Input

```
6
2
1
5
8
```