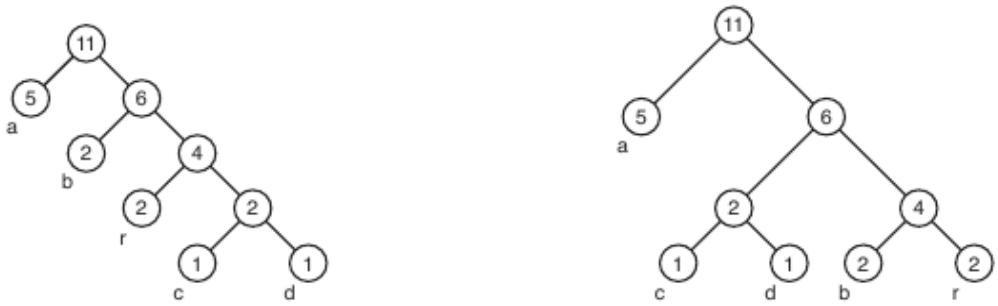
Inverting Huffman

https://vjudge.net/problem/UVA-12676

Static Huffman coding is an encoding algorithm used mainly for text compression. Given a text of certain size made of N different characters, the algorithm chooses N codes, one for each different character. The text is compressed using these codes. To choose the codes, the algorithm builds a binary rooted tree having N leaves. For $N \geq 2$ the tree can be built as follows.

1. For each different character in the text build a tree containing just a single node, and assign to it a weight coincident with the number of occurrences of the character within the text.
2. Build a set s containing the above N trees.
3. While s contains more than one tree:
   (a) Choose $t_1 \in s$ with minimum weight and remove it from s.
   (b) Choose $t_2 \in s$ with minimum weight and remove it from s.
   (c) Build a new tree t with t_1 as its left subtree and t_2 as its right subtree, and assign to t the sums of the weights of t_1 and t_2.
   (d) Include t into s.
4. Return the only tree that remains in s.

For the text "`abracadabra`", the tree produced by the above procedure can look like the one on the left of the following picture, where each internal node is labeled with the weight of the subtree rooted at that node. Notice that the obtained tree can also look like the one on the right of the picture, among others, because at steps 3a and 3b the set s may contain several trees with minimum weight.



For each different character in the text, its code depends on the path that exists, in the final tree, from the root to the leaf corresponding to the character. The length of the code is the number of edges in that path (which is coincident with the number of internal nodes in the path). Assuming the tree on the left was built by the algorithm, the code for "`r`" has length 3 while the code for "`d`" has length 4.

Your task is, given the lengths of the N codes chosen by the algorithm, find the minimum size (total number of characters) that the text can have so as the generated codes have those N lengths.

Input

The input file contains several test cases, each of them as described below.

The first line contains an integer N ($2 \leq N \leq 50$) representing the number of different characters that appear in the text. The second line contains N integers L_i indicating the lengths of the codes chosen by Huffman algorithm for the different characters ($1 \leq L_i \leq 50$ for $i = 1, 2, \ldots, N$). You may assume that there exists at least one tree, built as described, that produces codes with the given lengths.

Output

For each test case, output a line with an integer representing the minimum size (total number of characters) that the text can have so as the generated codes have the given lengths.

Sample Input

```
2
1 1
4
2 2 2 2
10
8 2 4 7 5 1 6 9 3 9
```

Sample Output

```
2
4
89
```