

Problem A

Binary Tree

Time Limit: 1 Second

We are given a full binary tree of height k , where each edge has a positive weight. The full binary tree of height k has $(2^{k+1} - 1)$ nodes containing 2^k leaves. The distance from the root to a leaf is the sum of weights of all the edges on the path from the root to the leaf. In this problem, we would like to increase the weights of certain edges so that all root-to-leaf paths have the same distance and keep the sum of all the edge weights as small as possible.

For example, consider a full binary tree of height 2 in Figure 1(a). A number alongside an edge indicates the weight of the edge. The solution of this instance is shown in Figure 1(b). That is, all root-to-leaf paths have the same distance 5, and the sum of all the edge weights is 15 which is the minimum possible value in this case.

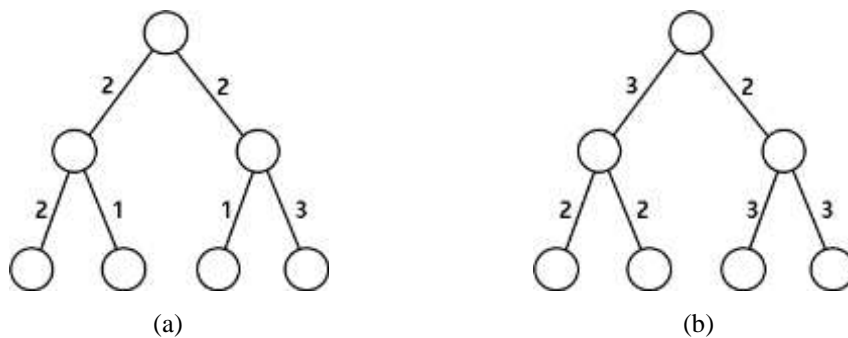


Figure 1. Illustration of increasing edge weights.

Given the weights of all edges in a full binary tree, you are to write a program that increases the weights of certain edges so that all root-to-leaf paths have the same distance and the total edge weights is as small as possible.

Input

Your program is to read from standard input. The input begins with a line containing a positive integer k ($1 \leq k \leq 20$) which represents the height of a full binary tree. In the second line, all the edge weights are given. The edge weights are given by level-by-level, and left-to-right order in a level. The weight of any edge is between 1 and 1,000, inclusive.

Output

Your program is to write to standard output. Print exactly one line which contains the total edge weights of the tree obtained after increasing edge weights. Note that the weight of a certain edge can be increased to a value larger than 1,000 in some cases.

The following shows sample input and output for four test cases.

Sample Input 1	Output for the Sample Input 1
2 2 2 2 1 1 3	15
Sample Input 2	Output for the Sample Input 2
1 1 1000	2000
Sample Input 3	Output for the Sample Input 3
3 1 2 1 3 2 4 1 1 1 1 1 1 1 1	27
Sample Input 4	Output for the Sample Input 4
2 1 1000 1 1 1000 1000	5001