

The 41<sup>st</sup> Annual ACM  
International Collegiate Programming Contest  
Asia Regional – Daejeon  
Nationwide Internet Competition



## 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.

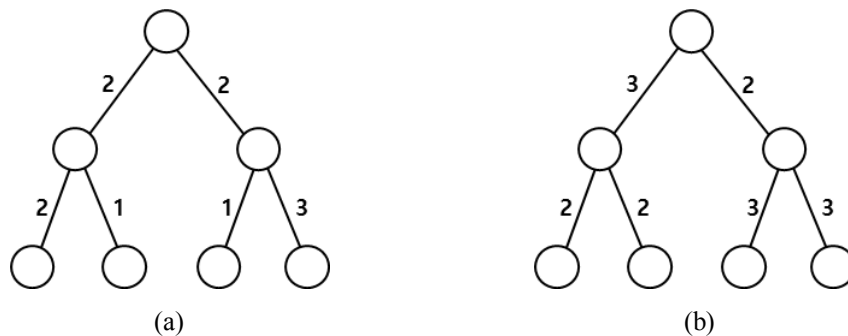


Figure 1. Illustration of increasing edge weights.

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

The 41<sup>st</sup> Annual ACM  
International Collegiate Programming Contest  
Asia Regional – Daejeon  
Nationwide Internet Competition



## Problem A

### 이진 트리

Time Limit: 1 Second

각 에지에 양수인 가중치가 부여된 높이  $k$ 인 포화이진트리가 주어져 있다. 높이  $k$ 인 포화이진트리는  $2^k$ 개의 리프를 포함하여  $(2^{k+1} - 1)$ 개의 노드를 가진다. 루트에서 어떤 리프까지의 거리는 루트에서 그 리프까지의 경로상에 있는 모든 에지들의 가중치를 더한 값이다. 이 문제에서는, 어떤 에지들의 가중치를 증가시켜서 루트에서 모든 리프까지의 거리가 같도록 하고, 또한 에지 가중치들의 총합을 최소화 하려고 한다.

예를 들어, 그림 1(a)에 있는 높이 2 인 포화이진트리를 살펴보자. 에지 옆에 있는 수는 그 에지의 가중치를 나타낸다. 이 경우에 대한 답이 그림 1(b)에 나타나 있다. 즉, 루트에서 모든 리프까지의 거리가 5 이고, 에지 가중치들의 총합은 이 경우에 가능한 최소값인 15 이다.

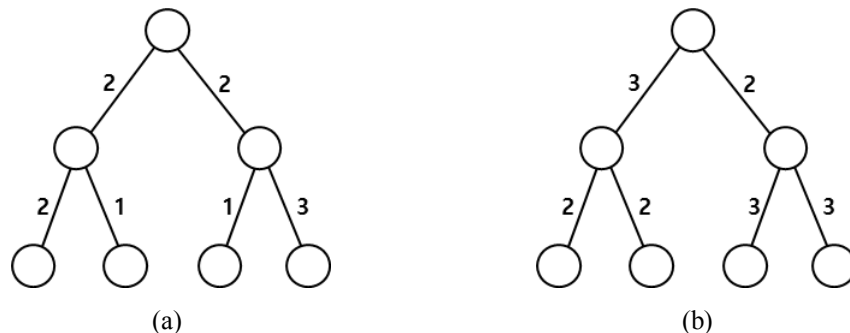


그림 1. 에지 가중치를 증가시키는 예.

포화이진트리에 들어 있는 모든 에지들의 가중치가 주어졌을 때, 어떤 에지들의 가중치를 증가시켜서 루트에서 모든 리프까지의 거리가 같게 하면서 에지 가중치들의 총합이 최소가 되도록 하는 프로그램을 작성하시오.

#### 입력(Input)

입력 데이터는 표준입력을 사용한다. 입력의 첫째 줄에는 포화이진트리의 높이를 나타내는 양의 정수  $k$  ( $1 \leq k \leq 20$ )가 주어진다. 두 번째 줄에는 모든 에지들의 가중치가 주어진다. 에지들의 가중치는 루트에서 가까운 레벨에 있는 것부터, 같은 레벨에 있는 경우는 왼쪽부터 오른쪽의 순서로 주어진다. 각 에지의 가중치는 1 이상 1,000 이하인 정수로 주어진다.

#### 출력(Output)

출력은 표준출력을 사용한다. 에지들의 가중치를 증가시킨 다음에 얻어지는 트리에 있는 모든 에지들의 가중치들의 총합을 한 줄에 출력한다. 어떤 에지의 가중치는 경우에 따라 1,000 이상의 값으로 증가될 수도 있다는 점에 유의하시오.

다음은 네 개의 테스트 데이터에 대한 입력과 출력의 예이다.

입력 예제 1(Sample Input 1)	출력 예제 1(Output for the Sample Input 1)
2 2 2 2 1 1 3	15
입력 예제 2(Sample Input 2)	출력 예제 2(Output for the Sample Input 2)
1 1 1000	2000
입력 예제 3(Sample Input 3)	출력 예제 3(Output for the Sample Input 3)
3 1 2 1 3 2 4 1 1 1 1 1 1 1 1	27
입력 예제 4(Sample Input 4)	출력 예제 4(Output for the Sample Input 4)
2 1 1000 1 1 1000 1000	5001