

[⚡ Try our new interface for solving problems](#)TRY ([HTTPS://BASECAMP.EOLYMP.COM/EN/PROBLEMS/4003](https://basecamp.eolymp.com/en/problems/4003))

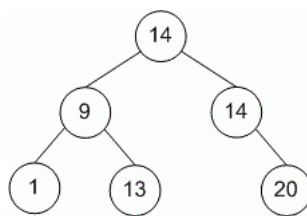
Problems (/en/problems) › Sum of Left Leaves

[Statement \(/en/problems/4003\)](/en/problems/4003)[Solutions \(/en/problems/4003/submissions\)](/en/problems/4003/submissions)[Statistics \(/en/problems/4003/statistics\)](/en/problems/4003/statistics)[Discussion \(/en/problems/4003/discussion\)](/en/problems/4003/discussion)

Sum of Left Leaves

Given an array of integers. Create a Binary Search Tree from these numbers. If the inserted value equals to the current node, insert it to the **right** subtree.

Write method **SumLeft** that returns the sum of all left leaves in a tree.



Write the code according to the next interface:

```
class TreeNode
{
public:
    int val;
    TreeNode *left;
    TreeNode *right;
    TreeNode(int x) : val(x), left(NULL), right(NULL) {}
};

class Tree
{
public:
    TreeNode *head;
    Tree() : head(NULL) {};
    void Insert(int val); // Вставка числа val в Бинарное Дерево Поиска
    int SumLeft(void); // Вернуть сумму всех левых листьев в дереве
};
```

You can create (use) additional methods if needed.

Input

The first line contains number **n** ($1 \leq n \leq 100$). The second line contains **n** integers.


Output



Create the Binary Search Tree from input data. Print the sum of all left leaves in a tree. If the tree does not contain left leaves, print **0**.

⌚ Time limit **1** second

📄 Memory limit **128** MiB

Input example #1 📄

6
14 9 1 14 20 13
Output example #1 
1

-
-  Author Mykhailo Medvediev
 -  Source C++
-

[< CHEATING AWAY!\(/EN/PROBLEMS/4002\)](/en/problems/4002) [IS THERE A CYCLE? > \(/EN/PROBLEMS/4004\)](/en/problems/4004)