

题目描述：

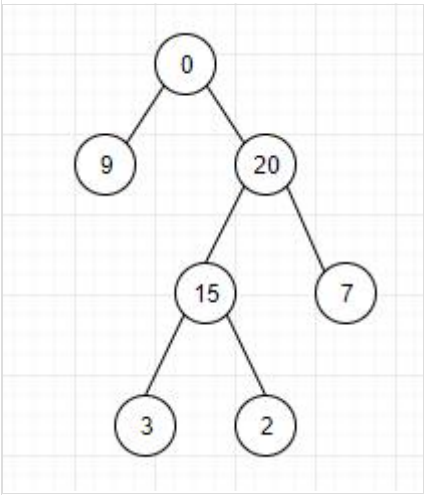
给定一个二叉树，每个节点上站着一个人，节点数字表示父节点到该节点传递悄悄话需要花费的时间。
初始时，根节点所在位置的人有一个悄悄话想要传递给其他人，求二叉树所有节点上的人都接收到悄悄话花费的时间。

输入描述：

给定二叉树

0 9 20 -1 -1 15 7 -1 -1 -1 -1 3 2

注：-1 表示空节点



输出描述：

返回所有节点都接收到悄悄话花费的时间 38

示例 1

输入：

0 9 20 -1 -1 15 7 -1 -1 -1 -1 3 2

输出：

38

```
import java.util.Scanner;
import java.util.List;
import java.util.ArrayList;
import java.util.LinkedList;
```

// 注意类名必须为 Main, 不要有任何 package xxx 信息

```
public class Main {  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
        List<Integer>list = new ArrayList<>();  
        list.add(-1);  
        int index = 1;  
        int result = 0;  
  
        while(in.hasNextInt()) {  
            int curNum = in.nextInt();  
            if(list.get(index/2)!=-1) {  
                curNum = list.get(index/2) + curNum;  
            }  
  
            list.add(curNum);  
            index++;  
            result = Math.max(result, curNum);  
        }  
        System.out.println(result);  
    }  
}
```