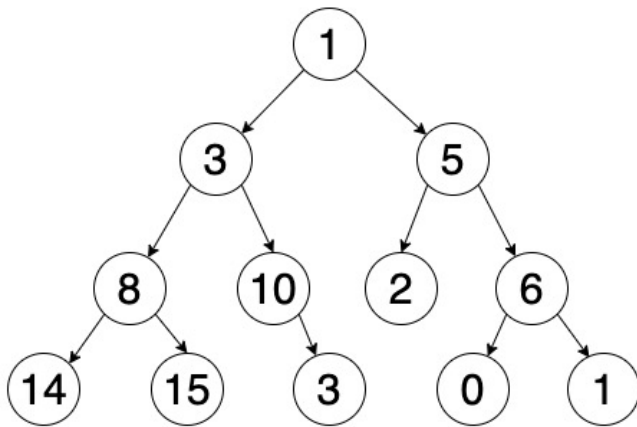


Яндекс.Практикум, Python-разработчик: Спринт 15. Деревья.

А. Бриллианты

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

У Евлампии есть брошь с бриллиантами в виде дерева. В каждом узле дерева есть какое-то количество бриллиантов. Помогите выяснить, какое максимальное количество бриллиантов есть в одном узле.



Ответ : 15

Формат ввода

На вход подается корень дерева.

Python:

```
class Node:
    def __init__(self, value, left=None, right=None):
        self.value = value
        self.right = right
        self.left = left
```

Ваша функция должна иметь сигнатуру `solution(Node) -> int`.

C++:

```
struct Node{
    int value;
    const Node* left = nullptr;
    const Node* right = nullptr;
};
int Solution(const Node* root);
```

Нужно подключить `solution_tree.h`

Go:

```
type TNode struct {
value int
left *TNode
right *TNode
}
```

Ваша функция должна иметь сигнатуру `func Solution(*TNode) int`.

JS:

```
class CNode {
  constructor(value) {
    this.value = value;
    this.left = null;
    this.right = null;
  }
}
```

Ваша функция должна иметь сигнатуру `solution :: CNode -> Number`.

Java:

Файл должен содержать `public class Solution` с функцией

```
public static int treeSolution(Node head)
```

```
class Node {
  int value;
  Node left;
  Node right;

  Node(int value) {
    this.value = value;
    right = null;
    left = null;
  }
}
```

Формат вывода

Функция должна вернуть максимальное число бриллиантов в узле дерева.

Примечания

Решение нужно отправлять в виде файла с расширением соответствующем вашему языку программирования. Нужно выбирать компилятор `make`.

Для Java файл должен называться `Solution.java`

Для остальных языков назовите файл с решением `brilliant_solution.ext`, заменив `ext` на нужное расширение.