

# Extra Assignment 3

Imagine that an employment tree represents the formal employee hierarchy. Manager nodes have child nodes for each employee that reports to them. Each of these employees can, in turn, have child nodes representing their respective reportees. Each node can have a maximum of 2 children. Each node in the tree contains an integer representing the number of months the employee has spent at the company. Team tenure is computed as the average tenure of the manager and all the company employees working below the manager. The oldest team has the highest team tenure.

Write an algorithm to find the manager of the team with the highest tenure. An employee must have child nodes to be a manager.

## Input

The input consists of an argument, CEO, a node representing the root node of the employee hierarchy.

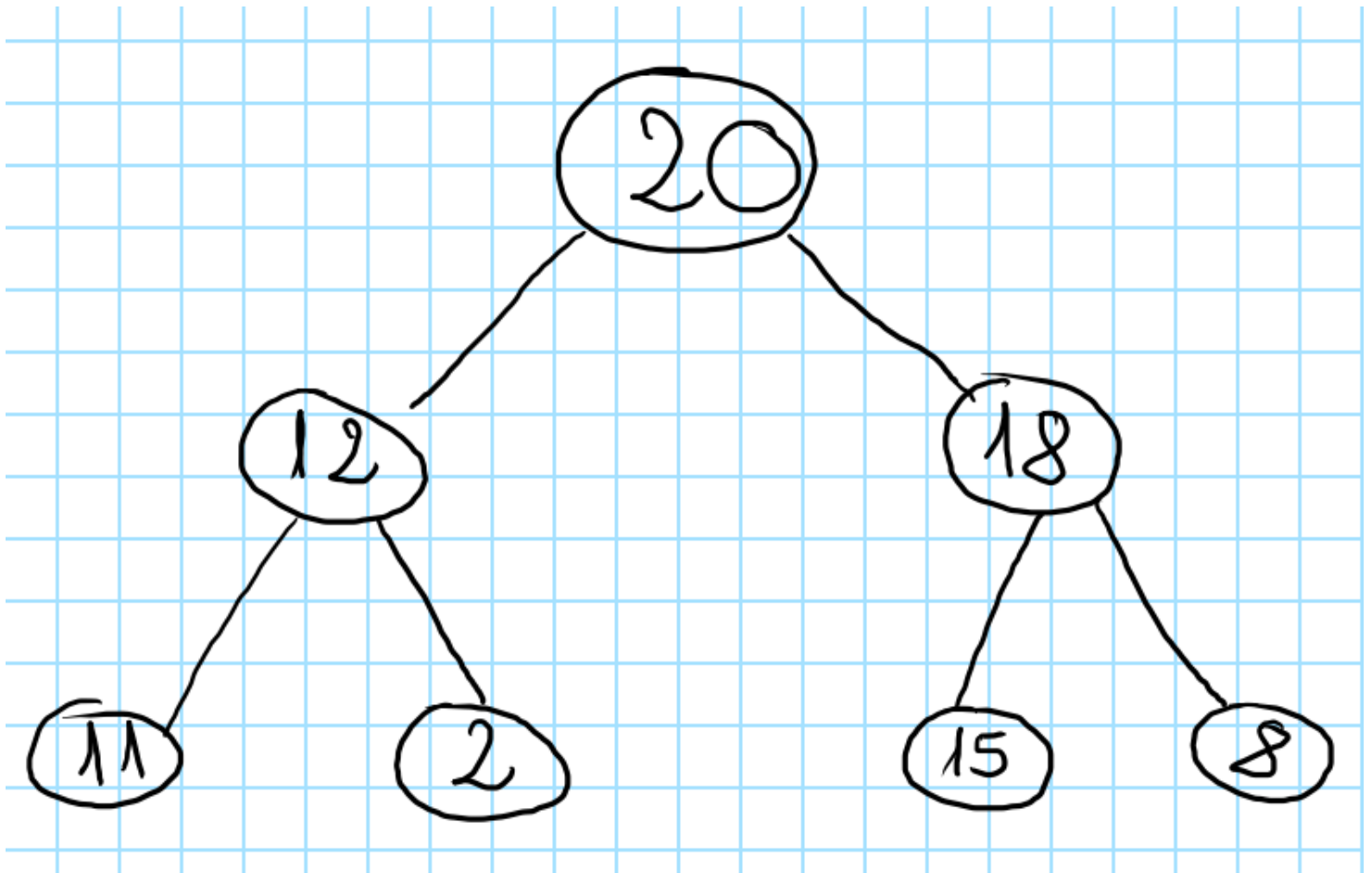
## Output

Return the node which has the oldest team.

## Node

There will be at least one child node in the tree and there will be no ties.

## Example



## Explanation

There are three managers in the team with the following tenures:

- 20 ->  $(20 + 12 + 18 + 11 + 2 + 15 + 8)/7 = 12.2857142857$
- 12 ->  $(12 + 11 + 2)/3 = 8.33333333333$
- 18 ->  $(18 + 15 + 8)/3 = 13.6666666667$

The node that has value 18 will be returned.