



Congratulations! You passed!

TO PASS 80% or higher

Keep Learning

GRADE
100%

Week 2 Challenge

LATEST SUBMISSION GRADE

100%

1. Your job is to implement the function "int count(Node *n)" that computes and returns the number of nodes in the binary tree that has n as its root.

5 / 5 points

When you write your code below, assume that a class type called "Node" has already been defined for you; you cannot change the class definition. This class type has two child pointers ("left", "right"). There is also a constructor Node() that initializes the children to nullptr and a destructor that deallocates the subtree's memory recursively.

```
1 - /*****
2 You may assume that the following Node class has already
3 been defined for you previously:
4
5 * class Node {
6 public:
7     Node *left, *right;
8     Node() { left = right = nullptr; }
9     ~Node() {
10         delete left;
11         left = nullptr;
12         delete right;
13         right = nullptr;
14     }
15 };
16
17 You may also assume that iostream has already been included.
18
19 Implement the function "int count(Node *n)" below to return
20 an integer representing the number of nodes in the subtree
21 of Node n (including Node n itself).
22
23 *****/
24
25 int count(Node *n) {
26     // Implement count() here.
27
28     if( nullptr == n )
29     {
30         // if visit empty subtree, return 0
31         return 0;
32     }
33     else
34     {
35         // node count of node n
36         // = node count of left sub tree + node count of right sub tree + 1 (self)
37         return count(n->left) + count( n->right ) + 1;
38     }
39 }
40
41 //end of function count(Node *n)
42
43
44
45 int main() {
46     Node *n = new Node();
47     n->left = new Node();
48     n->right = new Node();
49     n->right->left = new Node();
50     n->right->right = new Node();
51     n->right->right->right = new Node();
52
53     // This should print a count of six nodes
54     std::cout << count(n) << std::endl;
55
56     // Deleting n is sufficient to delete the entire tree
57     // because this will trigger the recursively-defined
58     // destructor of the Node class.
59     delete n;
60     n = nullptr;
61
62     return 0;
63 }
```

Run

Reset



Correct

You correctly counted a tree of 47 nodes.