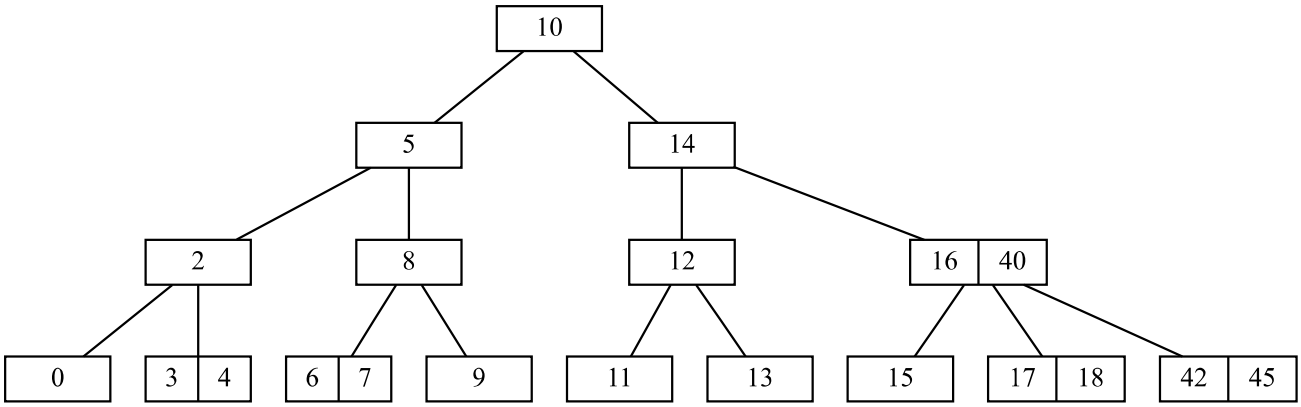
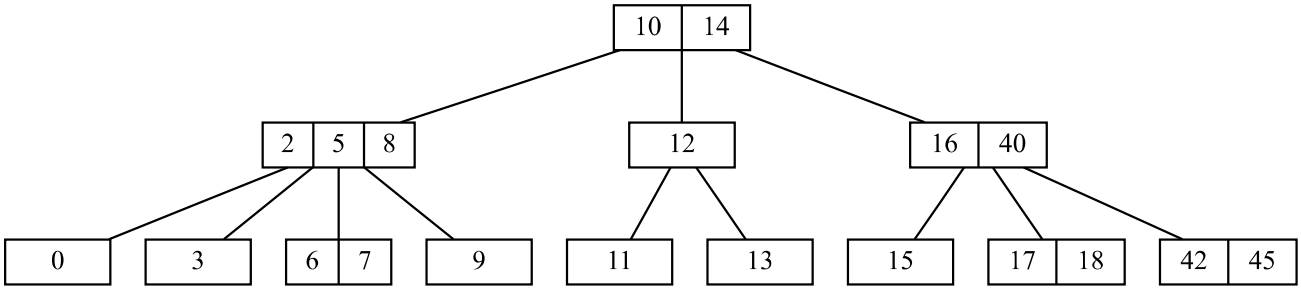
In [1]:
```
from btree_deletion import delete
from algopy.btree import BTree, display, fromlist
BTree.degree = 2
```

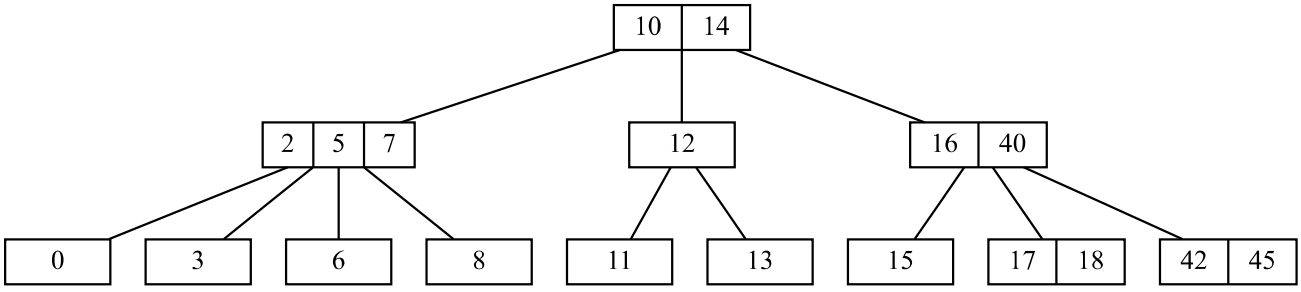
In [2]:
```
B = fromlist('(<10>(<5>(<2>(<0>)(<3,4>))(<8>(<6,7>)(<9>)))(<14>(<12>(<11>)(<13>))(<16,40>(<15>)(<17,18>)(<42,45>))))', 2)
display(B)
```



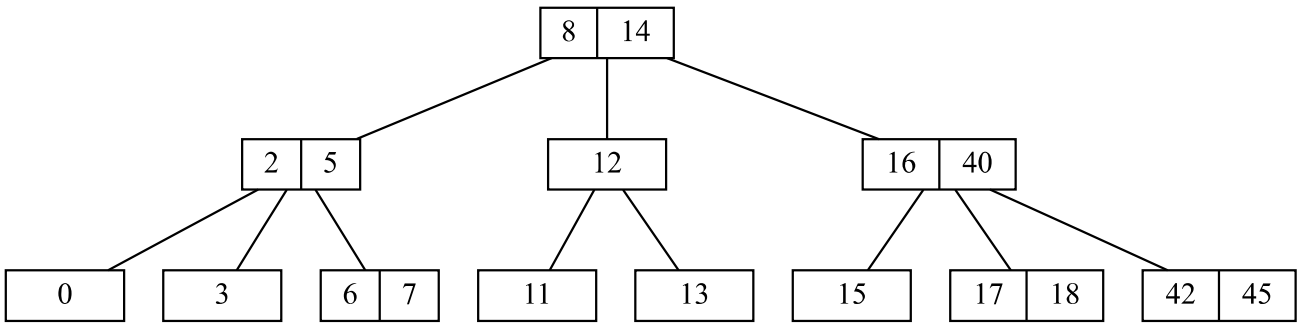
In [3]:
```
B = delete(B, 4)  # simple leaf deletion (precaution principle: 5 and 14 are merged (10 goes down), creating a new root, then 2 and 8 are merged)
display(B)
```



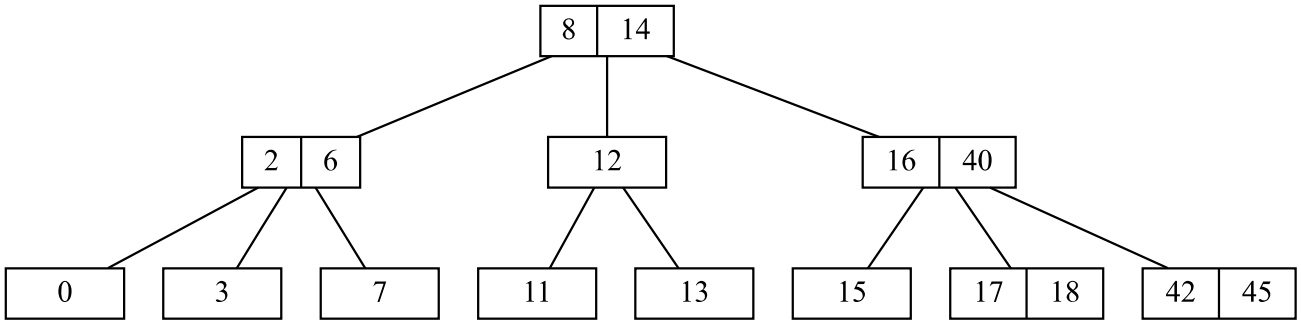
In [4]:
```
B = delete(B, 9)  # leaf deletion that needs a right-rotation
display(B)
```



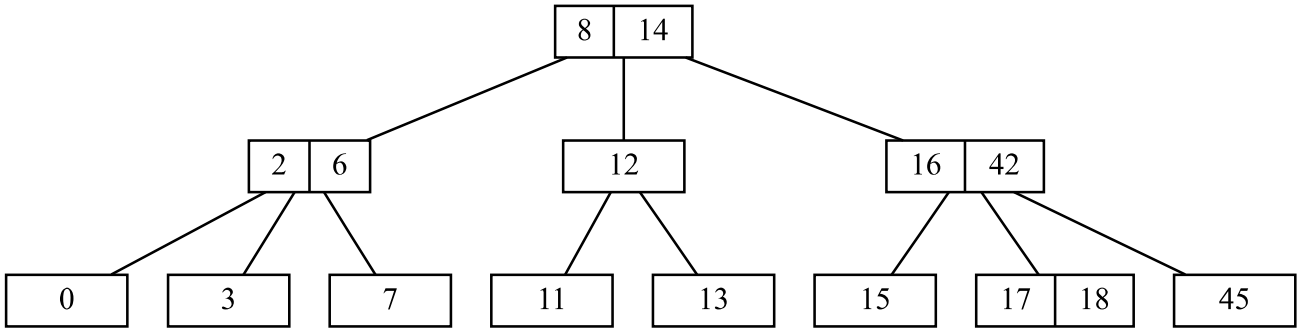
In [5]:
```
B = delete(B, 10)  # internal node deletion, switch with max of left child (then a merge is needed at the leaf level)
display(B)
```



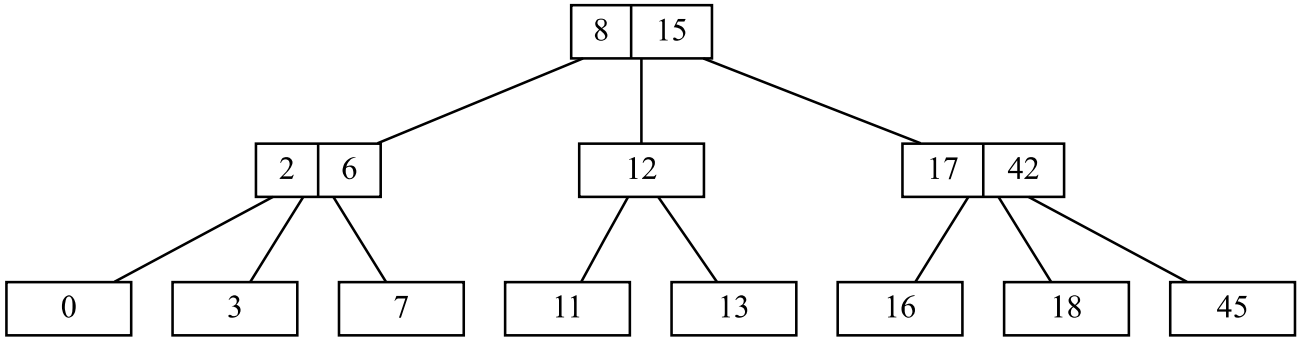
In [6]:
```
B = delete(B, 5)  # internal node deletion, switch with min of right child
display(B)
```



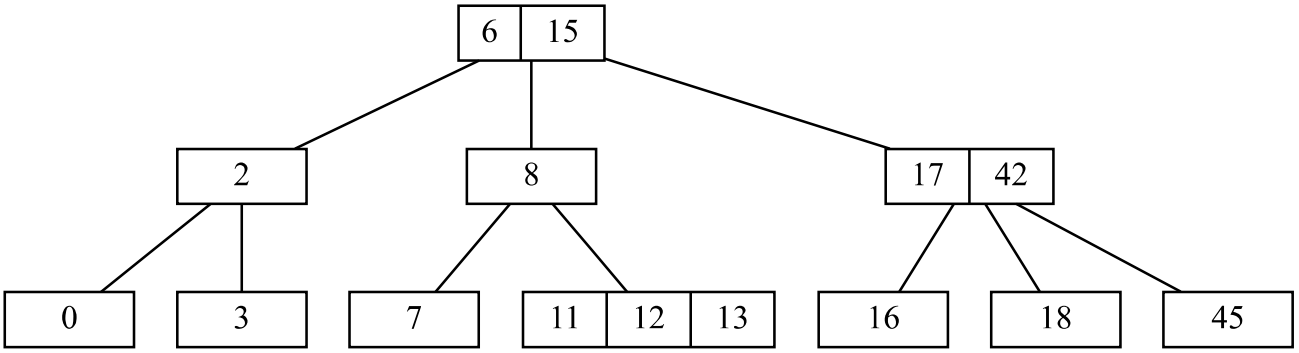
In [7]:
```
B = delete(B, 40)  # simple internal deletion (right child is chosen)
display(B)
```



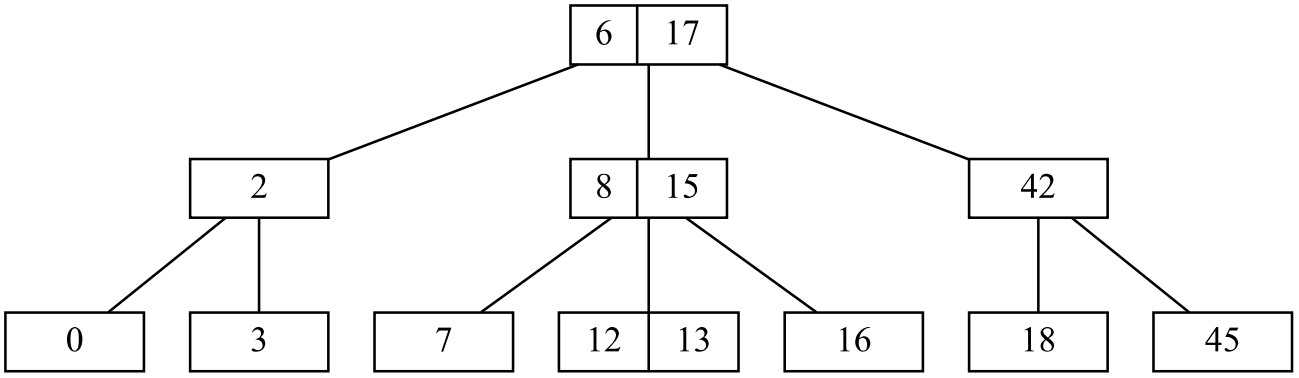
In [8]:
```
B = delete(B, 14)  # internal deletion, right child is chosen, then a left rotation on the leaf is needed
display(B)
```



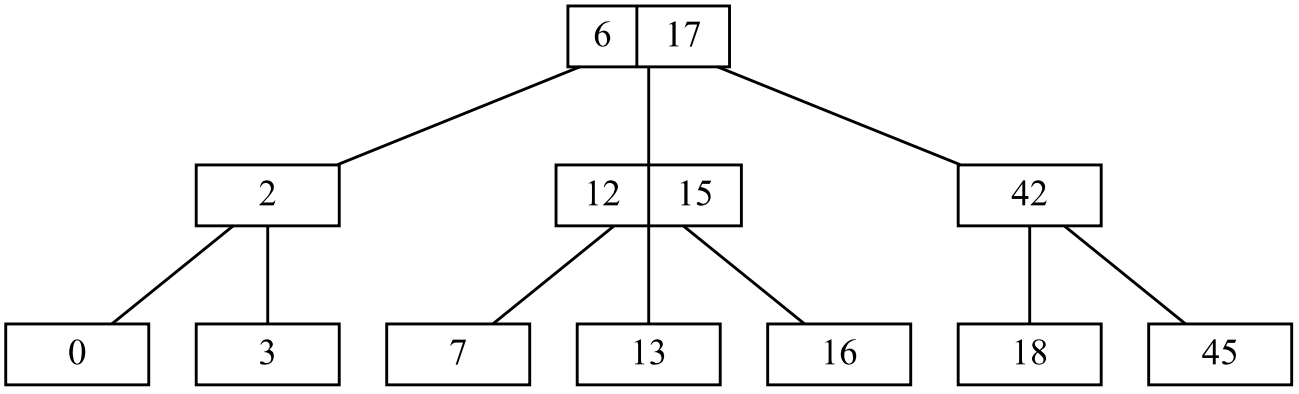
In [9]:
```
B = delete(B, 10)  # non-existant key deletion, a right rotation is done on the secon
d level, moving [7] to the middle node's child list, then [11] and [13] are merged
display(B)
```



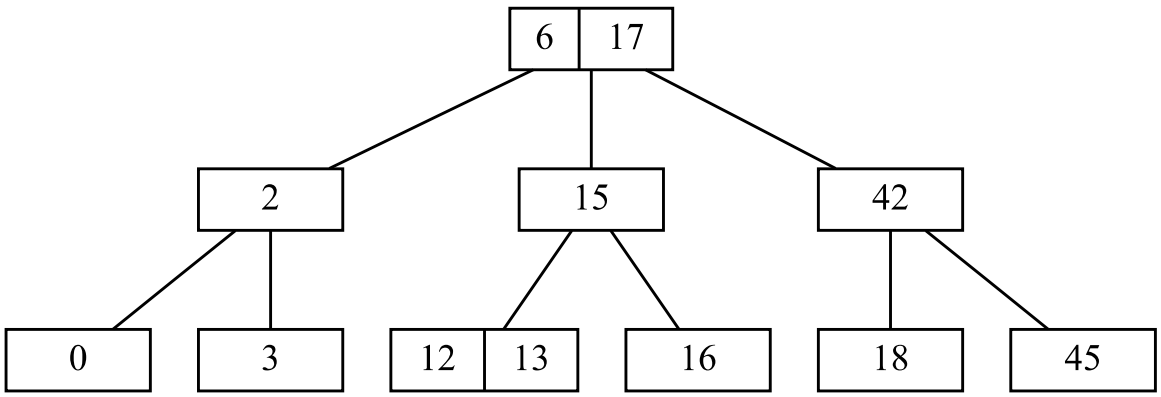
In [10]:
```
B = delete(B, 11)  # leaf deletion, a left-rotation is done on the second level
display(B)
```



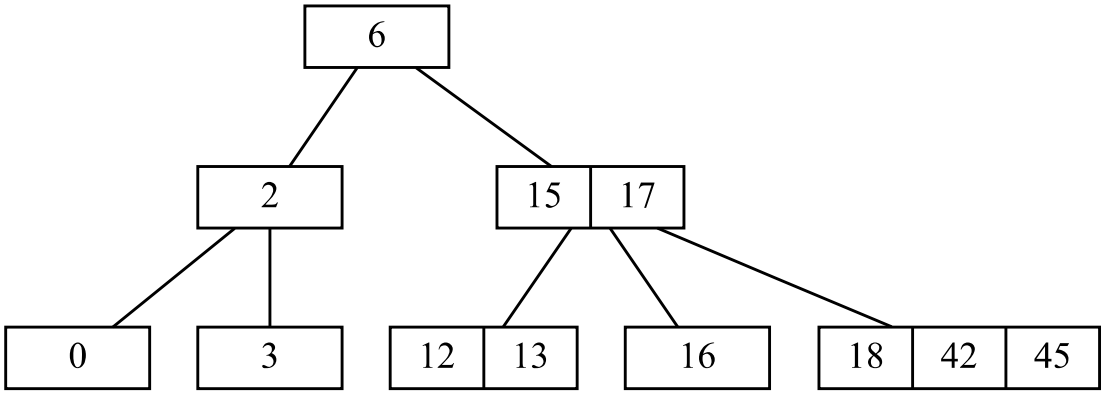
In [11]:
```
B = delete(B, 8)  # simple internal node deletion, no balancing is done
display(B)
```



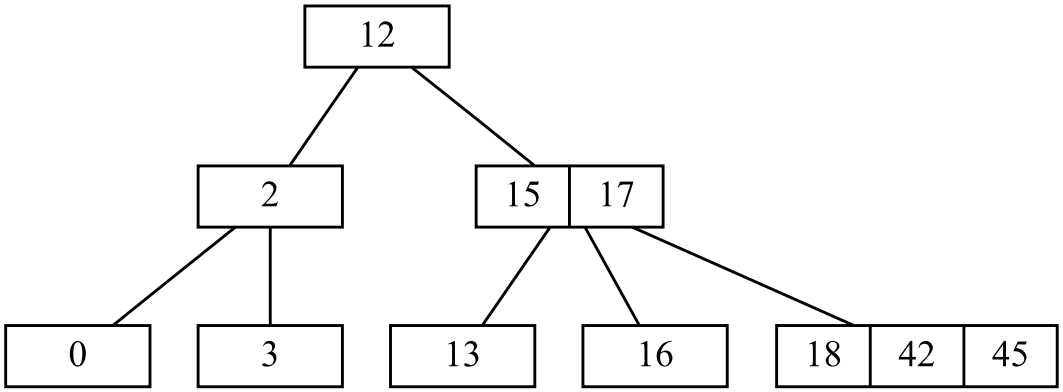
In [12]:
```
B = delete(B, 7)  # leaf deletion, a merge is needed
display(B)
```



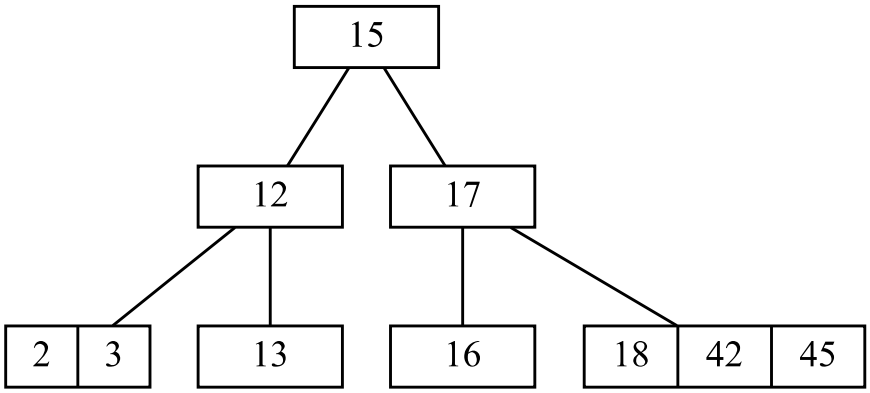
In [13]:
```
B = delete(B, 19)  # non-existant key deletion, two merges are done ([15] and [42] on second level, then [18] and [45] on leaf level)
display(B)
```



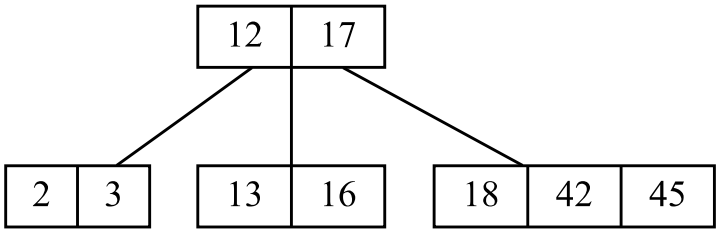
In [14]:
```
B = delete(B, 6)  # simple internal node deletion
display(B)
```



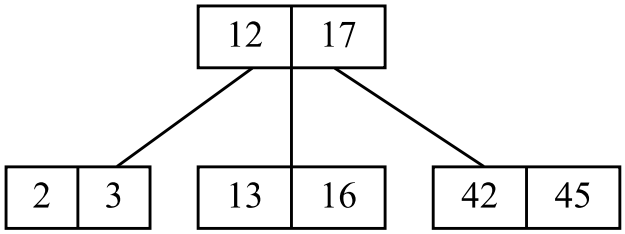
In [15]:
```
B = delete(B, 0)  # one left-rotation on second level, then one merge at the leaf level
display(B)
```



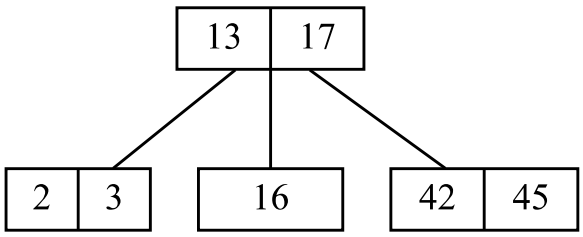
In [16]:
```
B = delete(B, 15)  # one switch, then two merges are made, creating a new root
display(B)
```



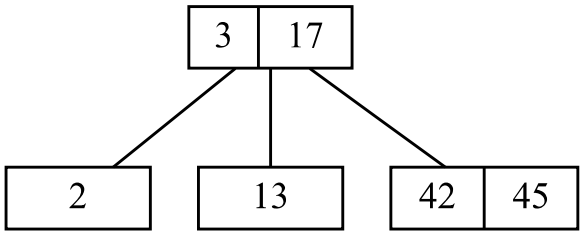
In [17]:
```
B = delete(B, 18)  # simple leaf deletion
display(B)
```



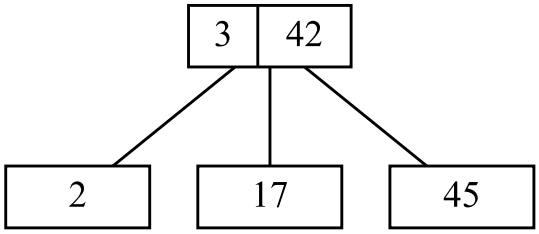
In [18]:
```
B = delete(B, 12)  # simple internal node deletion
display(B)
```



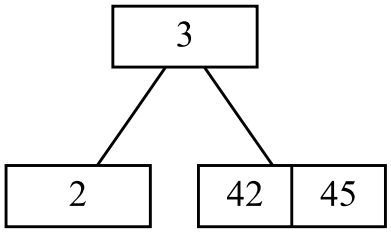
In [19]:
```
B = delete(B, 16)  # one right-rotation is needed
display(B)
```



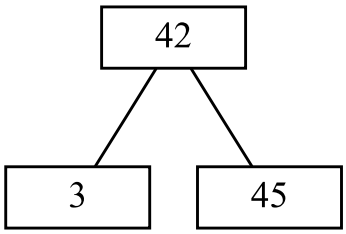
In [20]:
```
B = delete(B, 13)  # one left-rotation is needed
display(B)
```



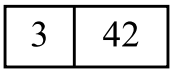
In [21]:
```
B = delete(B, 17)  # one merge
display(B)
```



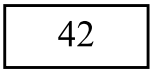
In [22]:
```
B = delete(B, 2)  # last left-rotation
display(B)
```



In [23]:
```
B = delete(B, 45)  # last merge, creating a new root
display(B)
```



In [24]:
```
B = delete(B, 3)  # simple leaf deletion
display(B)
```



In [25]:
```
B = delete(B, 42)  # goodbye
print(B)
```

```
None
```