

Problem 3: Find K-th largest element in BST

```
class TreeNode:
```

```
    def __init__(self, val=0, left=None, right=None):
```

```
        self.val = val
```

```
        self.left = left
```

```
        self.right = right
```

```
def kthLargest(root, k):
```

```
    stack = []
```

```
    node = root
```

```
    while node or stack:
```

```
        while node:
```

```
            stack.append(node)
```

```
            node = node.right
```

```
        node = stack.pop()
```

```
        k -= 1
```

```
        if k == 0:
```

```
            return node.val
```

```
        node = node.left
```

```
root = TreeNode(4)
```

```
root.left = TreeNode(2)
```

```
root.right = TreeNode(7)
```

```
root.left.left = TreeNode(1)
```

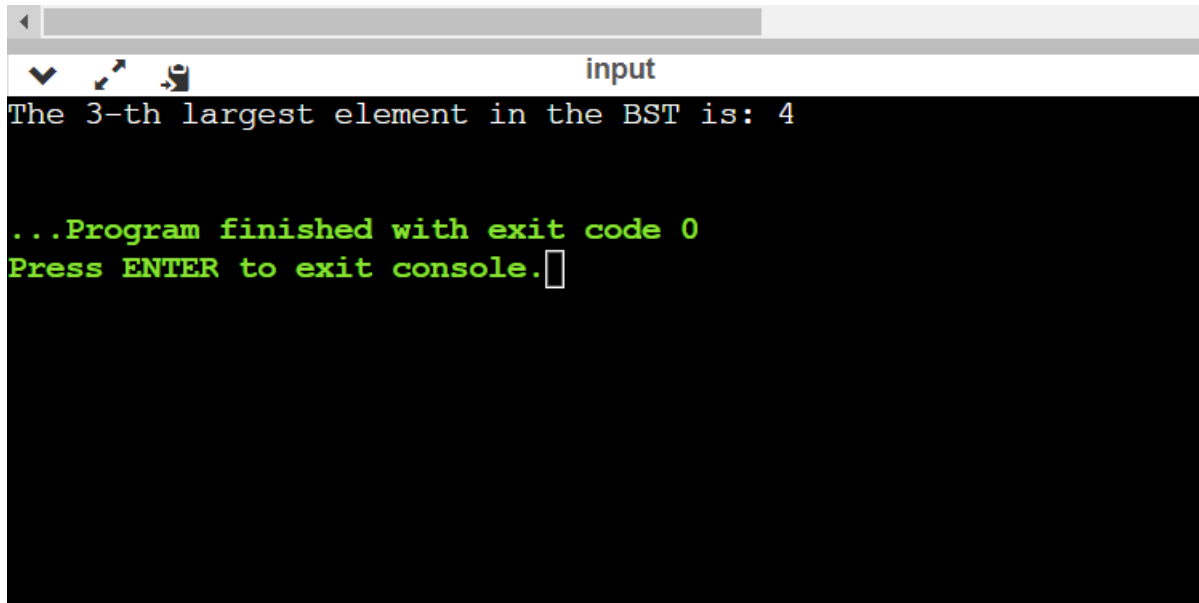
```
root.left.right = TreeNode(3)
```

```
root.right.left = TreeNode(6)
```

k = 3

result = kthLargest(root, k)

print(f"The {k}-th largest element in the BST is: {result}")



The screenshot shows a terminal window with a title bar that includes a back arrow, a maximize button, and a close button, followed by the title "input". The terminal content displays the output of a program: "The 3-th largest element in the BST is: 4". Below this, it shows "...Program finished with exit code 0" and "Press ENTER to exit console." with a cursor at the end of the line.

```
The 3-th largest element in the BST is: 4

...Program finished with exit code 0
Press ENTER to exit console.
```