Write recursive contains method

first...

Read over the **_init_** method for class **BTNode**:

```
    '''Binary Tree node.'''

    def __init__(self, data, left=None, right=None):
        ''' (BTNode, object, BTNode, BTNode) -> NoneType

        Create BTNode (self) with data and children left and right.
        '''
        self.data, self.left, self.right = data, left, right
```

next...

Now, read the header and docstring for the function **contains**, and then answer the questions that follow it.

```
def contains(node, value):
    ''' (BTNode, object) -> value

    Return whether tree rooted at node contains value.

    >>> contains(None, 5)
    False
    >>> contains(BTNode(5, BTNode(7), BTNode(9)), 7)
    True
    '''
```

1. One of the examples in **contains** docstring is simple enough not to require recursion (a base case). Write an **if...** expression that checks for this case, and then returns the correct thing. Include an **else...** for when the tree is *less* easy to deal with.

2. Another docstring examples is a typical one which can benefit from recursion. Write code that returns the correct value for this case. **Hint:** Be sure to use the BST property to avoid visiting nodes you don't have to visit.

Now implement the body of **contains**