

```

lemma sameElements<T>(tree:Tree<T>, element:T)
ensures treeContains(tree, element) <==> listContains(flatten(tree), element)
{
  match tree
  case Leaf => {}
  case Node(Tree1, Tree2, T1) => {
    memberOfAppend(element, flatten(Tree1), flatten(Tree2));
    sameElements(Tree1, element);
    sameElements(Tree2, element);

    assert treeContains(tree, element)
      == treeContains(Node(Tree1, Tree2, T1), element) //using definition of tree

      == treeContains(Tree1, element) || treeContains(Tree2, element) || element==T1 //using treeContains func

      == element==T1 || listContains(flatten(Tree1), element) || listContains(flatten(Tree2), element) //using inductive hypothesis

      == listContains(append(flatten(Tree1), flatten(Tree2)), element) //using memberOfAppend method

      == listContains(Cons(T1, append(flatten(Tree1), flatten(Tree2))), element) //using listContains func

      == listContains(Cons(T1, flatten(tree)), element)

      == listContains(flatten(tree), element);
  }
}
/* wrong code

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

No problems have been detected in the workspace.