1. Use the following binary tree to write out each of the three traversals indicated below.

```
   1
  /|
 /  \
2    3
 /  \
4    5
   /|
   /  \
 6
```

- Preorder traversal: _1 2 4 3 5 6_
- Inorder traversal: _2 4 1 5 3 4_
- Postorder traversal: _4 2 5 6 3 1_

2. Is this a Binary Search Tree assuming only String keys are shown? (yes or no)? _No_ (must be unique)

3. Is this a Binary Search Tree assuming only String keys are shown? (yes or no)? _No_ (reverse order)
   Actually it could be if we defined the order property to be All nodes to the right are less than, to the left greater than.

4. What potential benefit is there to using a binary search tree rather than an array to store the same collection of key/value mappings of a Map?
   Better runtimes for put, get, and remove. With a BST, they are $O(\log n)$, with an array some are $O(n)$

5. What would be required of the keys for those mappings to be stored in a binary search tree?
   They are Comparable, that is the element implements the Comparable interface so we can see if a key is less than or greater than another

6. Draw a picture ....
   ```java
   aBST.put("Matrix");
aBST.put("Antz");
aBST.put("Bean");
aBST.put("Shaft");
aBST.put("Scream");
aBST.put("Titanic");
   ```

7. ```java
   public int nodesAtLevel(int level) {
       return atDepth(root, level);
   }
   ```
   ```java
   private int atDepth(TreeNode t, int level) {
       if (t == null) {
           return 0;
       } else if (level == 0) {
           return 1;
       } else {
           return atDepth(t.left, level - 1) + atDepth(t.right, level - 1);
       }
   }
   ```
8. public List<E> elementsGreaterThan(E element) {
    ArrayList<E> list = new ArrayList<E>();
    elementsGreaterThan(root, element, list);
    return list;
}

private void elementsGreaterThan(TreeNode t, E element, ArrayList<E> list) {
    if (t != null) {
        elementsGreaterThan(t.left, element, list);
        if (t.data.compareTo(element) > 0)
            list.add(t.data);
        elementsGreaterThan(t.right, element, list);
    }
}

9. public void reverse() {
    mirror(root);
}

private void mirror(TreeNode t) {
    if (t != null) {
        // Swap left and right links
        TreeNode temp = t.right;
        t.right = t.left;
        t.left = temp;
        mirror(t.left);
        mirror(t.right);
    }
}
10. Complete the output that would be generated by this code

```java
-5 -5 12 23
===
true
false
-5
23
3
```

11.
```
null
null
M
null
3
B
C
D
null
```

12. Which city is the capital of Australia? __c. Canberra__ (4pts)

_Do not leave the last question blank, any answer will be worth 4 pts_