Task 10: Binary Trees

- Create a Binary Tree Library file `tree.c` with a corresponding `tree.h` that allows one to call `tree_insert` to insert an integer into the tree. The tree should always remain sorted so that the
  left child is smaller or equal to the root node, which is smaller than the right node. Example:

```
    1
   / \
  2   5
 / \ / \
3  4 6  7
```

- If you like, you can start out with the code that we created two weeks ago during the lesson: https://jkrbs.github.io/c_lessons/task_solution_sources/intermediate_4_binary_tree.c
- Add a `tree_print` method that prints the tree as an ordered list.
- Create a small main file that uses `tree.h`, Adds 1 to 10 to a the tree and then prints the tree.
- Bonus: Can you write a method that prints the tree as shown above?

Task 9: Hints

- The new `struct list_node` could look like this:

```
1 typedef struct list_node {
2     int value;
3     struct list_node* next;
4     struct list_node* previous;
5 } list_node;
```

- `previous` of the first node is `NULL`
- `next` of the last node is `NULL`
- If you forgot how the pointers have to point, look at the slides again, maybe they can help.
- Bonus Hint: Recursion :).