Solution@11-09-23

129. Sum Root to Leaf Numbers

```
class Solution {
    private int result;
    private void findsum(TreeNode root, int val) {
        int curr = val*10 + root.val;
        if(root.left == null && root.right == null){
            result += curr;
            return;
        }
        if(root.left != null) findsum(root.left, curr);
        if(root.right != null) findsum(root.right, curr);
    }

    public int sumNumbers(TreeNode root) {
        if(root == null) return 0;
        result = 0;
        findsum(root, 0);
        return result;
    }
}
```

150. Evaluate Reverse Polish Notation

```
class Solution {
    public int evalRPN(String[] tokens) {
        if (tokens.length == 0 || tokens == null)
            return -1;
        Stack<Integer> stack = new Stack<>();
        for (String token : tokens) {
            if (token.equals("+")) {
                stack.push(stack.pop() + stack.pop());
            }
            else if (token.equals("*")) {
                stack.push(stack.pop() * stack.pop());
            }
            else if (token.equals("-")) {
                int b = stack.pop();
                int a = stack.pop();
                stack.push(a - b);
            }
            else if (token.equals("/")) {
                int b = stack.pop();
                int a = stack.pop();
                stack.push(a / b);
            }
            else {
                stack.push(Integer.valueOf(token));
            }
        }
        return stack.pop();
    }
}
```