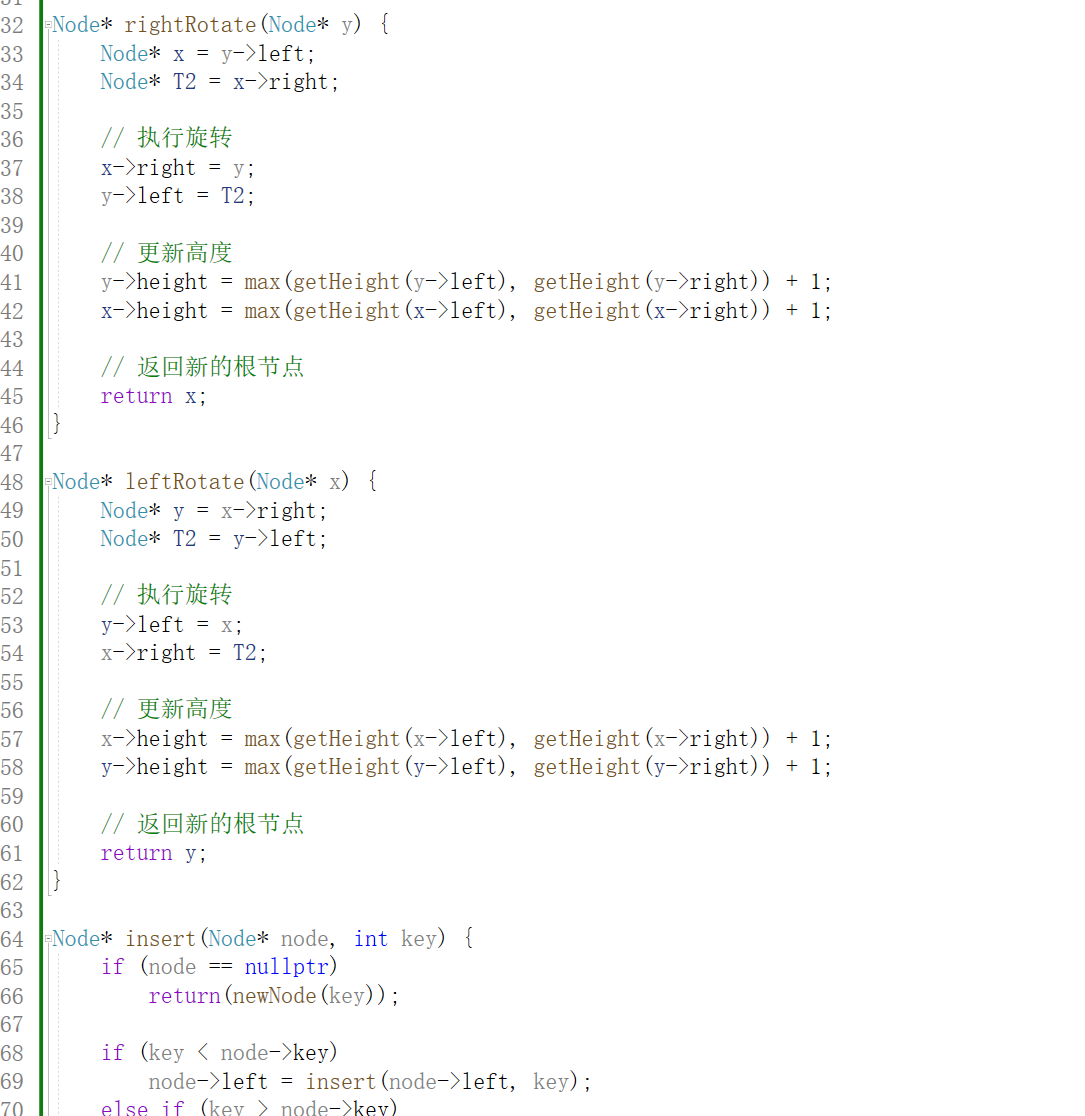
第一题

过程如下：

1. **初始化**：一开始，树是空的。
2. **逐一插入**：逐个将序列中的元素插入到二叉树中。每次插入后，检查树的平衡因子（左子树高度与右子树高度的差），如果某个节点的平衡因子绝对值超过1，则进行相应的旋转操作来恢复平衡。
   * 插入10：树中只有一个节点10。
   * 插入6：6 小于 10，放在10的左子树中。
   * 揑入1：1 小于 10，也小于 6，继绤放在6的左子树中，需要右旋（10）以保持平衡。
   * 揑入4：4 小于 10，在右旋后4将介于6和10之间，放在6的右子树上。
   * 揑入5：5 小于 10，大于6，放在6的右子树的右子树上，现在需要左旋（6）以保持平衡。
   * 揑入11：11 大于 10，放在10的右子树中。
   * 揑入12：12 大于 10，也大于11，放在11的右子树中，需要左旋（10）以保持平衡。
   * 揑入13：13 最大，放在12的右子树上，需要做多次旋转来保持平衡。
   * 揑入8：8 小于 10，大于 6，在6和10之间，放在6的右子树的左子树上。
   * 揑入9：9 小于 10，在6和10之间，放在8的右子树上，需要做旋转来保持平衡。
3. **平衡调整**：在插入每个节点后通过左旋（Left Rotation）或右旋（Right Rotation）来保持树的平衡。更复杂的情况（如双旋转Double Rotation，左-右旋转或右-左旋转）可能需要根据树的实际情况进行具体操作。

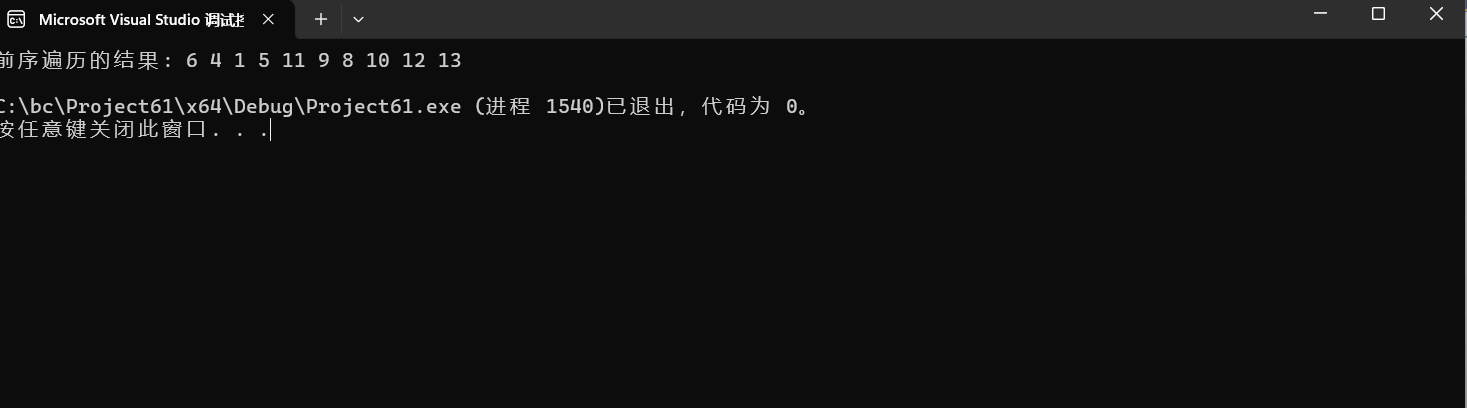
结束后，所有的节点都将被正确插入，并且通过适当的旋转操作，树将被保持为平衡状态。











第二题



