I built an interactive AVL tree that reads positive integers, inserting if new and deleting if present, and prints the tree after each operation until a non-positive value ends the program. I kept the provided third-example code and only added a small contains(root, key) helper plus a simple validated input loop. Time complexity is O(log n) for the search, insert, and delete with O(1) left or right rotations and O(n) for printing. I verified this by toggling values by inserting them and then deleting them, and confirmed the tree stays balanced after each change.