TEST CASES

Test Case 1: Insertion and In-order Traversal

- Inserts five names with corresponding phone numbers into the BST.
- After insertion, it performs an in-order traversal, which will print the names in alphabetical order (due to the properties of BST).

Test Case 2: Searching

- Searches for three names in the BST: "Akshay", "Himanshu", and "Venika".
- "Akshay" and "Himanshu" exist in the tree, so their phone numbers will be returned.
- "Venika" does not exist in the tree, so the search will return None.

Test Case 3: Deleting a Leaf Node

- Deletes the node "Snehitha", which is a leaf node (has no children).
- After deletion, performs an in-order traversal to verify that "Snehitha" is no longer in the tree.

Test Case 4: Deleting a Node with One Child

- First, "Nayana" is reinserted, then deleted again.
- "Nayana" is a node with one child (after reinsertion), so its deletion will replace it with its child.
- After deletion, performs an in-order traversal to verify the structure.

Test Case 5: Deleting a Node with Two Children

- Deletes the node "Himanshu", which has two children.
- The in-order successor of "Himanshu" will replace its value.
- After deletion, performs an in-order traversal to verify the structure.

Test Case 6: Deleting the Root

- Deletes the root node "Akshay".
- After deletion, "Jayana" becomes the new root due to the in-order successor replacement.
- Performs an in-order traversal to verify the updated structure.

Test Case 7: Deleting a Non-Existent Node

- Attempts to delete the node "Venika", which does not exist in the tree.
- Since the node is not found, no changes are made to the tree.
- Performs an in-order traversal to confirm that the structure remains unchanged.

Test Case 8: Updating an Existing Entry

- Inserts "Jayana" again with a new phone number, which effectively updates her phone number.
- Since the name "Jayana" already exists, the phone number is updated without changing the structure of the tree.
- Performs an in-order traversal to verify the update.