Design Documentation for AVL Tree Implementation

```
Successful Insert!
Successful Insert!
Successful Insert!
Successful Insert!
Successful Insert!
Successful Insert!
Successful Insert!
Successful Insert!
Successful Insert!
Successful Insert!
10 20 30 40 50 60 70 80 90 100
Successful Find!
Finding 60: 60
Successful Find!
Finding 100: 100
Finding 25: Not in tree
Finding 96.3: Not in tree
Successful Remove!
Successful Remove!
10 20 30 40 60 70 80 90

C:\Users\ctlus\source\repos\DS_Projects\x64\Debug\Lab_3-1.exe (process 25312) exited with code 0.
Press any key to close this window . . .
```

Overview:
This implementation of an AVL tree ensures the tree remains balanced during insertions and deletions, providing efficient operations with logarithmic time complexity. The AVL tree maintains a balance factor at each node, ensuring that the height difference between the left and right subtrees never exceeds 1.

Key Features:
- Insert, Delete, and Retrieve Operations: All operations maintain tree balance using rotations, ensuring logarithmic time complexity.
- Balancing Mechanism: After each insertion or deletion, the tree is balanced using left and right rotations.
- In-Order Traversal: Allows tree elements to be printed in sorted order.

Class and Method Descriptions:
1. tNode (Tree Node Structure)
- Represents each node in the AVL tree.
- Each node contains:
  - Pointers to left and right children.
  - A value (double).
  - A height value for balancing.

2. AVL Class (Self-balancing Binary Search Tree)
Private Methods:
- `insertRec`: Recursively inserts nodes and balances the tree.
- `retrieveRec`: Recursively searches for a value in the tree.
- `removeRec`: Recursively removes nodes and balances the tree.
- `rotateRight` and `rotateLeft`* Perform right and left rotations to correct imbalances.
- `balance`: Checks and adjusts the balance factor after insertions or deletions.

- `height` and `getBalance`: Helper methods to compute node height and balance factor.
- `findMin`: Finds the node with the minimum value in a subtree.
- `deleteTree`: Recursively deletes all nodes from the tree.
- `printHelp`: Recursively prints the tree in sorted (in-order) order.

Public Methods:
- `insert`: Inserts a new value into the tree.
- `retrieve`: Retrieves a value if present in the tree.
- `remove`: Removes a value from the tree.
- `printInOrder`: Prints the tree in sorted order using in-order traversal.

3. Rotations and Balancing
The AVL tree maintains balance by applying left and right rotations to nodes when their balance factor exceeds a predefined threshold:
- Left Rotation: Used to balance right-heavy nodes.
- Right Rotation: Used to balance left-heavy nodes.
- Balancing: If a node becomes unbalanced (balance factor >1 or <-1), the appropriate rotation(s) are applied to restore balance.

4. In-Order Traversal
In-order traversal ensures that the nodes are printed in sorted order by visiting the left subtree, the node, and then the right subtree.

Complexity Analysis:
- Time Complexity:
  - Insert, Delete, and Retrieve: O(log n) due to the balanced nature of the tree.
  - Rotations: O(1) for each rotation during rebalancing.
- Space Complexity: O(n), where n is the number of nodes in the tree.

Conclusion:
This AVL tree implementation ensures efficient operations with logarithmic time complexity by balancing the tree after insertions and deletions. The use of rotations keeps the tree height small, allowing for quick search, insertion, and deletion operations. The in-order traversal method ensures that elements are printed in sorted order.

Hash Table with AVL Tree Collision Resolution

```
Noah Brown : 239-596-5873 || Noah Miller : 347-725-2884 || Noah Brown : 684-348-2612 || Noah Miller : 700-132-2600 || Noah Thomas : 240-704-8746 || Noah Thomas : 913-826-6570 || Noah Wilson : 658-233-8778 ||
Noah Wilson : 778-972-8087 || Noah Smith : 102-296-8897 || Noah Smith : 208-760-1713 || Noah Jackson : 702-218-2137 || Noah Jackson : 995-838-9112 || Noah Johnson : 657-376-4989 || Olivia Miller : 373-176-686
1 || Olivia Miller : 405-675-7694 || Olivia Brown : 480-999-4928 || Olivia Thomas : 119-746-4188 || Olivia Smith : 338-972-8496 || Olivia Jackson : 668-541-2101 || Olivia Davis : 216-143-4733 || Olivia Willia
ms : 314-833-4808 || Olivia Williams : 560-838-1988 || Olivia Williams : 637-152-4964 || Olivia Davis : 674-387-7447 || Olivia Anderson : 310-935-4170 || William Brown : 153-381-6018 || Alexander Brown : 182-
153-4596 || William Brown : 684-691-4674 || Alexander Miller : 849-535-9922 || William Miller : 862-804-5970 || Alexander Davis : 792-654-3712 || William Anderson : 122-964-6627 || William Anderson : 487-581-
6518 || Alexander Anderson : 723-384-2018 || Alexander Johnson : 405-205-9273 || William Johnson : 677-260-6876 || Alexander Johnson : 731-362-7031 || Emma Miller : 138-964-2978 || Emma Miller : 414-100-8111
|| Emma Brown : 537-620-3055 || Emma Smith : 491-293-8767 || Emma Smith : 704-310-6785 || Emma Jackson : 398-876-4947 || Emma Davis : 204-214-5082 || Emma Anderson : 714-785-5457 || Emma Johnson : 114-185-205
8 || James Miller : 929-666-3301 || James Davis : 635-564-8463 || James Davis : 682-558-3291 || James Williams : 687-595-5022 || James Davis : 929-428-6555 || James Anderson : 691-187-9881 || James Anderson :
 885-473-2069 || James Johnson : 325-588-5733 || James Johnson : 636-658-2678 || Sophia Brown : 292-244-3966 || Sophia Brown : 868-477-5750 || Liam Miller : 874-880-8597 || Liam Miller : 913-487-4119 || Liam
Thomas : 120-961-3179 || Liam Thomas : 850-830-7418 || Liam Wilson : 213-427-1511 || Liam Wilson : 462-518-1294 || Liam Wilson : 502-572-9385 || Sophia Wilson : 877-863-4748 || Sophia Smith : 144-196-3624 ||
Sophia Smith : 623-688-5240 || Sophia Smith : 695-222-3981 || Liam Smith : 810-780-4530 || Sophia Smith : 906-241-1365 || Liam Jackson : 529-210-4511 || Sophia Jackson : 978-554-5575 || Liam Williams : 170-50
2-5632 || Sophia Davis : 557-301-4599 || Sophia Williams : 817-438-3758 || Sophia Anderson : 998-565-1390 || Liam Johnson : 341-429-9727 || Ava Miller : 579-720-5871 || Ava Thomas : 279-181-3859 || Ava Wilson
 : 727-276-8966 || Ava Smith : 798-801-2215 || Ava Jackson : 205-909-6831 || Ava Jackson : 702-125-1339 || Ava Jackson : 742-857-9865 || Ava Jackson : 803-418-1804 || Ava Williams : 135-370-6753 || Ava Willia
ms : 566-707-6368 || Isabella Thomas : 120-651-8128 || Isabella Thomas : 426-954-4094 || Isabella Wilson : 651-463-6940 || Isabella Smith : 308-477-2736 || Isabella Smith : 908-282-9039 || Isabella Jackson :
440-871-6740 || Isabella Jackson : 457-259-3643 || Isabella Anderson : 161-968-9818 || Isabella Anderson : 328-312-8546 || Isabella Anderson : 335-839-9481 || Isabella Anderson : 488-237-4421 || Isabella Ande
rson : 730-509-3877 || Isabella Anderson : 929-601-5452 ||

PRINTING ALL "Liam"s:
Liam Miller : 874-880-8597 || Liam Miller : 913-487-4119 || Liam Thomas : 120-961-3179 || Liam Thomas : 850-830-7418 || Liam Wilson : 213-427-1511 || Liam Wilson : 462-518-1294 || Liam Wilson : 502-572-9385 |
| Liam Smith : 810-780-4530 || Liam Jackson : 529-210-4511 || Liam Williams : 170-502-5632 || Liam Johnson : 341-429-9727 ||

REMOVING ALL "Isabella Anderson"s:
Noah Brown : 239-596-5873 || Noah Miller : 347-725-2884 || Noah Brown : 684-348-2612 || Noah Miller : 700-132-2600 || Noah Thomas : 240-704-8746 || Noah Thomas : 913-826-6570 || Noah Wilson : 658-233-8778 ||
Noah Wilson : 778-972-8087 || Noah Smith : 102-296-8897 || Noah Smith : 208-760-1713 || Noah Jackson : 702-218-2137 || Noah Jackson : 995-838-9112 || Noah Johnson : 657-376-4989 || Olivia Miller : 373-176-686
1 || Olivia Miller : 405-675-7694 || Olivia Brown : 480-999-4928 || Olivia Thomas : 119-746-4188 || Olivia Smith : 338-972-8496 || Olivia Jackson : 668-541-2101 || Olivia Davis : 216-143-4733 || Olivia Willia
ms : 314-833-4808 || Olivia Williams : 560-838-1988 || Olivia Williams : 637-152-4964 || Olivia Davis : 674-387-7447 || Olivia Anderson : 310-935-4170 || William Brown : 153-381-6018 || Alexander Brown : 182-
153-4596 || William Brown : 684-691-4674 || Alexander Miller : 849-535-9922 || William Miller : 862-804-5970 || Alexander Davis : 792-654-3712 || William Anderson : 122-964-6627 || William Anderson : 487-581-
6518 || Alexander Anderson : 723-384-2018 || Alexander Johnson : 405-205-9273 || William Johnson : 677-260-6876 || Alexander Johnson : 731-362-7031 || Emma Miller : 138-964-2978 || Emma Miller : 414-100-8111
|| Emma Brown : 537-620-3055 || Emma Smith : 491-293-8767 || Emma Smith : 704-310-6785 || Emma Jackson : 398-876-4947 || Emma Davis : 204-214-5082 || Emma Anderson : 714-785-5457 || Emma Johnson : 114-185-205
8 || James Miller : 929-666-3301 || James Davis : 635-564-8463 || James Davis : 682-558-3291 || James Williams : 687-595-5022 || James Davis : 929-428-6555 || James Anderson : 691-187-9881 || James Anderson :
 885-473-2069 || James Johnson : 325-588-5733 || James Johnson : 636-658-2678 || Sophia Brown : 292-244-3966 || Sophia Brown : 868-477-5750 || Liam Miller : 874-880-8597 || Liam Miller : 913-487-4119 || Liam
Thomas : 120-961-3179 || Liam Thomas : 850-830-7418 || Liam Wilson : 213-427-1511 || Liam Wilson : 462-518-1294 || Liam Wilson : 502-572-9385 || Sophia Wilson : 877-863-4748 || Sophia Smith : 144-196-3624 ||
Sophia Smith : 623-688-5240 || Sophia Smith : 695-222-3981 || Liam Smith : 810-780-4530 || Sophia Smith : 906-241-1365 || Liam Jackson : 529-210-4511 || Sophia Jackson : 978-554-5575 || Liam Williams : 170-50
2-5632 || Sophia Davis : 557-301-4599 || Sophia Williams : 817-438-3758 || Sophia Anderson : 998-565-1390 || Liam Johnson : 341-429-9727 || Ava Miller : 579-720-5871 || Ava Thomas : 279-181-3859 || Ava Wilson
 : 727-276-8966 || Ava Smith : 798-801-2215 || Ava Jackson : 205-909-6831 || Ava Jackson : 702-125-1339 || Ava Jackson : 742-857-9865 || Ava Jackson : 803-418-1804 || Ava Williams : 135-370-6753 || Ava Willia
ms : 566-707-6368 || Isabella Thomas : 120-651-8128 || Isabella Thomas : 426-954-4094 || Isabella Wilson : 651-463-6940 || Isabella Smith : 308-477-2736 || Isabella Smith : 908-282-9039 || Isabella Jackson :
440-871-6740 || Isabella Jackson : 457-259-3643 ||

PRINTING ALL "Shaibal Chakrabarty"s:
PRINTING ALL "Lucas Li"s:


INSERTING "Lucas Li" and "Shaibal Chakrabarty":
Noah Brown : 239-596-5873 || Noah Miller : 347-725-2884 || Noah Brown : 684-348-2612 || Noah Miller : 700-132-2600 || Noah Thomas : 240-704-8746 || Noah Thomas : 913-826-6570 || Noah Wilson : 658-233-8778 ||
Noah Wilson : 778-972-8087 || Noah Smith : 102-296-8897 || Noah Smith : 208-760-1713 || Noah Jackson : 702-218-2137 || Noah Jackson : 995-838-9112 || Noah Johnson : 657-376-4989 || Olivia Miller : 373-176-686
1 || Olivia Miller : 405-675-7694 || Olivia Brown : 480-999-4928 || Olivia Thomas : 119-746-4188 || Olivia Smith : 338-972-8496 || Olivia Jackson : 668-541-2101 || Olivia Davis : 216-143-4733 || Olivia Willia
ms : 314-833-4808 || Olivia Williams : 560-838-1988 || Olivia Williams : 637-152-4964 || Olivia Davis : 674-387-7447 || Olivia Anderson : 310-935-4170 || Lucas Li : 469-555-1212 || William Brown : 153-381-601
8 || Alexander Brown : 182-153-4596 || William Brown : 684-691-4674 || Alexander Miller : 849-535-9922 || William Miller : 862-804-5970 || Alexander Davis : 792-654-3712 || William Anderson : 122-964-6627 ||
William Anderson : 487-581-6518 || Alexander Anderson : 723-384-2018 || Alexander Johnson : 405-205-9273 || William Johnson : 677-260-6876 || Alexander Johnson : 731-362-7031 || Emma Miller : 138-964-2978 ||
Shaibal Chakrabarty : 214-768-2000 || Emma Miller : 414-100-8111 || Emma Brown : 537-620-3055 || Emma Smith : 491-293-8767 || Emma Smith : 704-310-6785 || Emma Jackson : 398-876-4947 || Emma Davis : 204-214-5
082 || Emma Anderson : 714-785-5457 || Emma Johnson : 114-185-2058 || James Miller : 929-666-3301 || James Davis : 635-564-8463 || James Davis : 682-558-3291 || James Williams : 687-595-5022 || James Davis :
929-428-6555 || James Anderson : 691-187-9881 || James Anderson : 885-473-2069 || James Johnson : 325-588-5733 || James Johnson : 636-658-2678 || Sophia Brown : 292-244-3966 || Sophia Brown : 868-477-5750 ||
Liam Miller : 874-880-8597 || Liam Miller : 913-487-4119 || Liam Thomas : 120-961-3179 || Liam Thomas : 850-830-7418 || Liam Wilson : 213-427-1511 || Liam Wilson : 462-518-1294 || Liam Wilson : 502-572-9385 |
| Sophia Wilson : 877-863-4748 || Sophia Smith : 144-196-3624 || Sophia Smith : 623-688-5240 || Sophia Smith : 695-222-3981 || Liam Smith : 810-780-4530 || Sophia Smith : 906-241-1365 || Liam Jackson : 529-21
0-4511 || Sophia Jackson : 978-554-5575 || Liam Williams : 170-502-5632 || Sophia Davis : 557-301-4599 || Sophia Williams : 817-438-3758 || Sophia Anderson : 998-565-1390 || Liam Johnson : 341-429-9727 || Ava
 Miller : 579-720-5871 || Ava Thomas : 279-181-3859 || Ava Wilson : 727-276-8966 || Ava Smith : 798-801-2215 || Ava Jackson : 205-909-6831 || Ava Jackson : 702-125-1339 || Ava Jackson : 742-857-9865 || Ava Ja
ckson : 803-418-1804 || Ava Williams : 135-370-6753 || Ava Williams : 566-707-6368 || Isabella Thomas : 120-651-8128 || Isabella Thomas : 426-954-4094 || Isabella Wilson : 651-463-6940 || Isabella Smith : 308
-477-2736 || Isabella Smith : 908-282-9039 || Isabella Jackson : 440-871-6740 || Isabella Jackson : 457-259-3643 ||
C:\Users\ctlus\source\repos\DS_Projects\x64\Debug\Lab_3-2.exe (process 22192) exited with code 0.
Press any key to close this window . . .
```

1. Introduction

This design involves a hash table implementation where collisions are resolved using AVL trees. The hash table stores entries of `person` objects, where each entry includes a first name, last name, and a phone number. The hash table's buckets are implemented using AVL trees to ensure that insertion, deletion, and search operations are efficient, even when multiple items hash to the same bucket.

2. Key Components

- person Structure:
  Stores the first name, last name, and phone number of a person.
- AVL Tree (Self-Balancing Binary Search Tree):
  Each bucket in the hash table is an AVL tree, which ensures that the tree remains balanced. AVL trees provide logarithmic time complexity for insertion, deletion, and search operations.
- Hash Table:
  The hash table uses a hash function to map keys (in this case, the phone number) to an index in the table. The index points to an AVL tree which stores the `person` objects that hash to the same index.

3. Functionality Overview

- Insert Operation:
  The `insert` function adds a new `person` object into the AVL tree corresponding to the hash index computed from the phone number. If the phone number already exists, no new insertion is made.
- Search Operation:
  The `retrieve` function searches for a `person` object by their phone number. The hash of the number is computed to locate the corresponding AVL tree, where the search is then

performed.

- Delete Operation:
  The `remove` function deletes a `person` object by their phone number, or by their first and last name. The relevant AVL tree is identified using the hash function, and the node is deleted from the tree. The AVL tree is rebalanced after deletion to maintain efficiency.
- Collision Handling:
  Collisions (when two phone numbers hash to the same index) are resolved using separate chaining, where each hash table bucket is an AVL tree that can store multiple `person` objects.

4. Time Complexity Analysis
- Insert:
  - Best Case: O(1), when the hash function directs the insertion into an empty bucket.
  - Worst Case: O(log n), where n is the number of elements in the AVL tree in a particular bucket.
- Search:
  - Best Case: O(1), when the searched element is found quickly using the hash function.
  - Worst Case: O(log n), due to the logarithmic search time in the AVL tree.
- Delete:
  - Best Case: O(1), when the AVL tree has very few elements or a simple structure.
  - Worst Case: O(log n), where rebalancing in the AVL tree is required.

5. Space Complexity Analysis
- Space Complexity: O(m + n), where `m` is the number of buckets in the hash table and `n` is the total number of elements. Each bucket (AVL tree) may hold multiple elements, but AVL trees are space-efficient due to their binary search tree nature.

6. Hashing Strategy
The hash table uses the FNV-1a hashing algorithm, which is a fast, non-cryptographic hash function suitable for generating hashes from strings (in this case, phone numbers). The hash value is used to determine the index of the bucket (AVL tree) where a `person` object is stored. If the computed index is already occupied by another object (collision), the AVL tree handles it by storing multiple objects in a balanced structure.

7. Hash Table Size
The size of the hash table is chosen dynamically based on the expected number of elements, ensuring an optimal load factor of 0.75. This minimizes collisions while ensuring efficient memory usage. The size is determined by finding the next prime number greater than or equal to the estimated number of elements.

8. Conclusion
This design ensures efficient storage and retrieval of `person` objects by combining a hash table for quick access with AVL trees for balanced, efficient collision resolution. The approach ensures that operations like insertion, deletion, and search remain performant even under hash collisions.