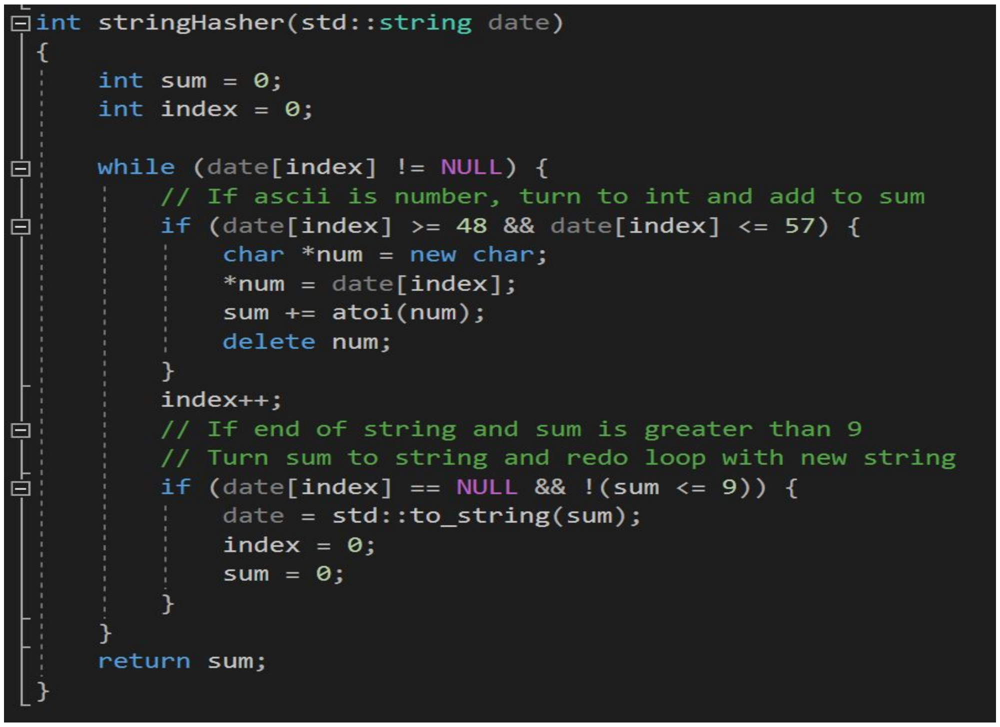
Demonstration Test Plan

Account Object Example:
Name: Nathan Wong
dateCreation: 2018-12-05
sortValue: Name
accountNumber: 123-12-1234
Total: $450
savingsAccount: $300
checkingAccount: $150
Username: animelover03
Password: maggieheartnathan

Modules
(1). Add account
(2). Delete account
    (a.) Undo Feature
(3). Find and display one account
(4). List account in hash sorting
(5). List account using sorted key sequence
(6). Print indented tree
(7). Efficiency
(8). User ATM
(9). Quit

Hash Table

Hash Function

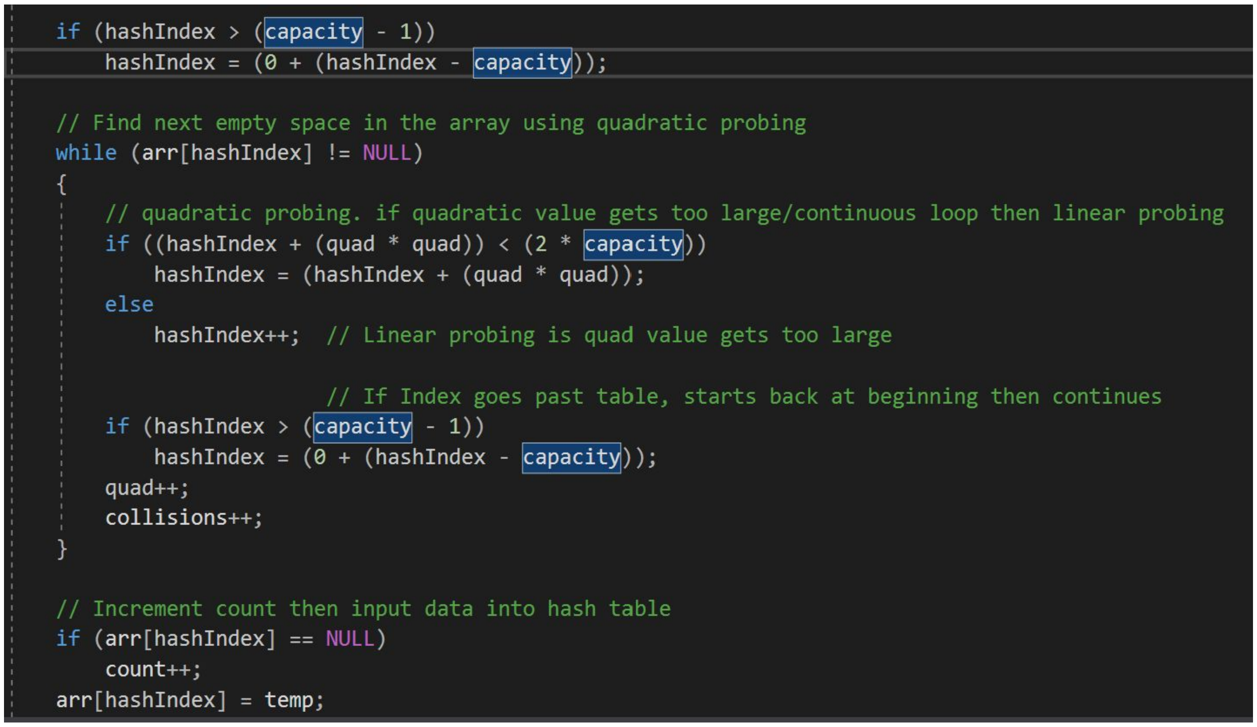


```
int stringHasher(std::string date)
{
    int sum = 0;
    int index = 0;

    while (date[index] != NULL) {
        // If ascii is number, turn to int and add to sum
        if (date[index] >= 48 && date[index] <= 57) {
            char *num = new char;
            *num = date[index];
            sum += atoi(num);
            delete num;
        }
        index++;
        // If end of string and sum is greater than 9
        // Turn sum to string and redo loop with new string
        if (date[index] == NULL && !(sum <= 9)) {
            date = std::to_string(sum);
            index = 0;
            sum = 0;
        }
    }
    return sum;
}
```

Our Hash Function Takes in a string that is the account ID number. It takes all the integer numbers and adding them together. If the number is over a certain number, it takes the new number and adds each integer in the number string together. The sum is returned and that is the hash table index the data will be placed in

Collision Handling



```
if (hashIndex > (capacity - 1))
    hashIndex = (0 + (hashIndex - capacity));

// Find next empty space in the array using quadratic probing
while (arr[hashIndex] != NULL)
{
    // quadratic probing. if quadratic value gets too large/continuous loop then linear probing
    if ((hashIndex + (quad * quad)) < (2 * capacity))
        hashIndex = (hashIndex + (quad * quad));
    else
        hashIndex++;  // Linear probing is quad value gets too large

                      // If Index goes past table, starts back at beginning then continues
    if (hashIndex > (capacity - 1))
        hashIndex = (0 + (hashIndex - capacity));
    quad++;
    collisions++;
}

// Increment count then input data into hash table
if (arr[hashIndex] == NULL)
    count++;
arr[hashIndex] = temp;
```

The Hash Map does quadratic collision handling. If there is a collision, the hash index increments the hash index by n squared for each collision. If the index goes over the capacity of the array, the index overflows to the beginning of the list. If the quadratic value is so big that it skips over the whole array when incrementing, or it increments a size that is twice as big as the capacity, then linear probing takes place.

How to Demonstrate:

Enter Filename

Enter a number that is associated with the desired menu option and follow prompt:

Add Account Module:

Add all the info for the object (example):

John Wick
Fat wick
Wickster
999
999
2000-20-20
333-44-5555

Delete Account:

Delete by inputting the account number. Can undo delete by entering the undo command or you can save changes. Need to confirm save changes by entering yes. Quit command automatically saves changes.

123-12-1234
Delete
142-99-2123
Undo
Save
Yes
Quit

Display One Account:

Display Account by Account ID

420-69-0666

Display Hash Sequence:

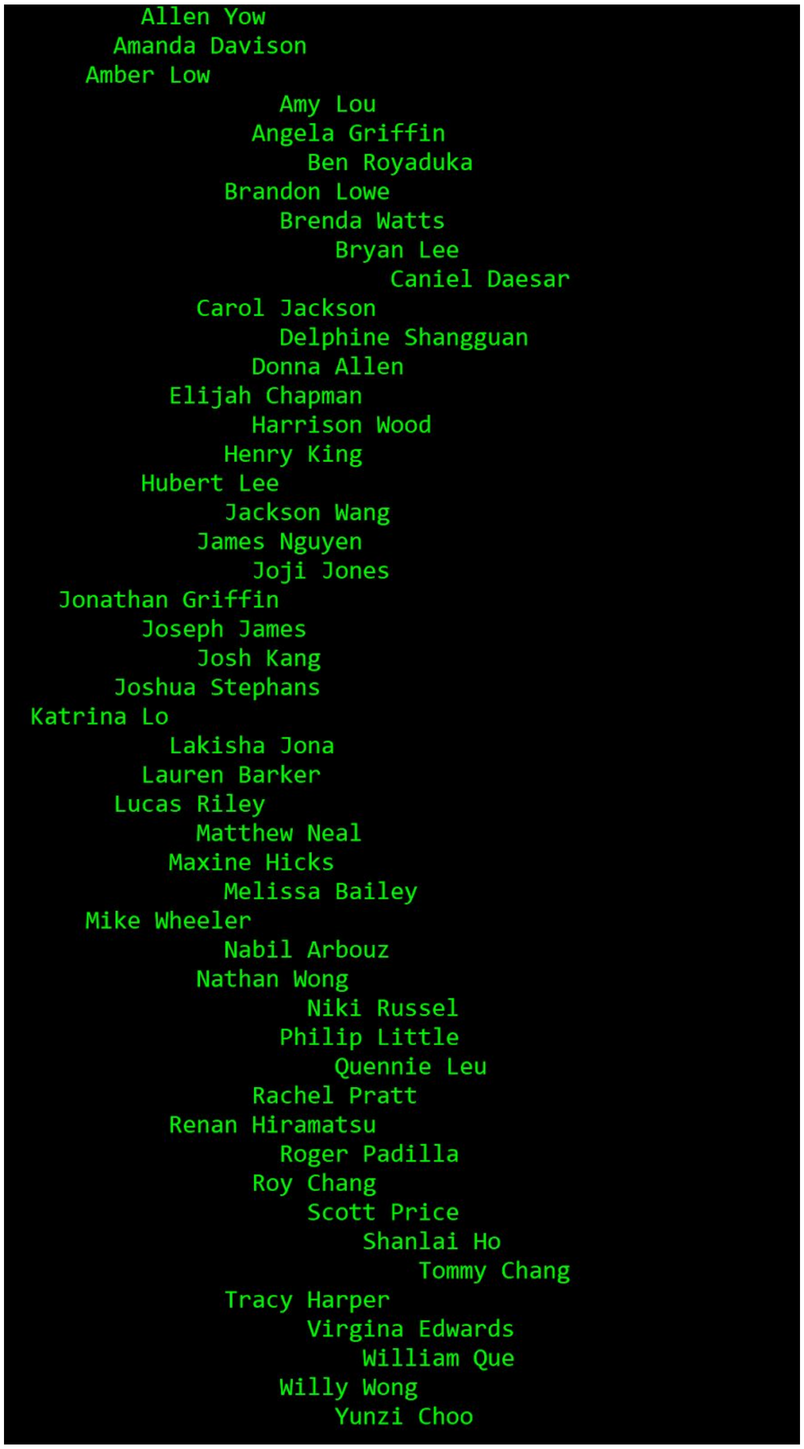
Displays the objects in the order they exist in hash table.

Display Key Sequence:

Display using Inorder Traversal and put in order of account name

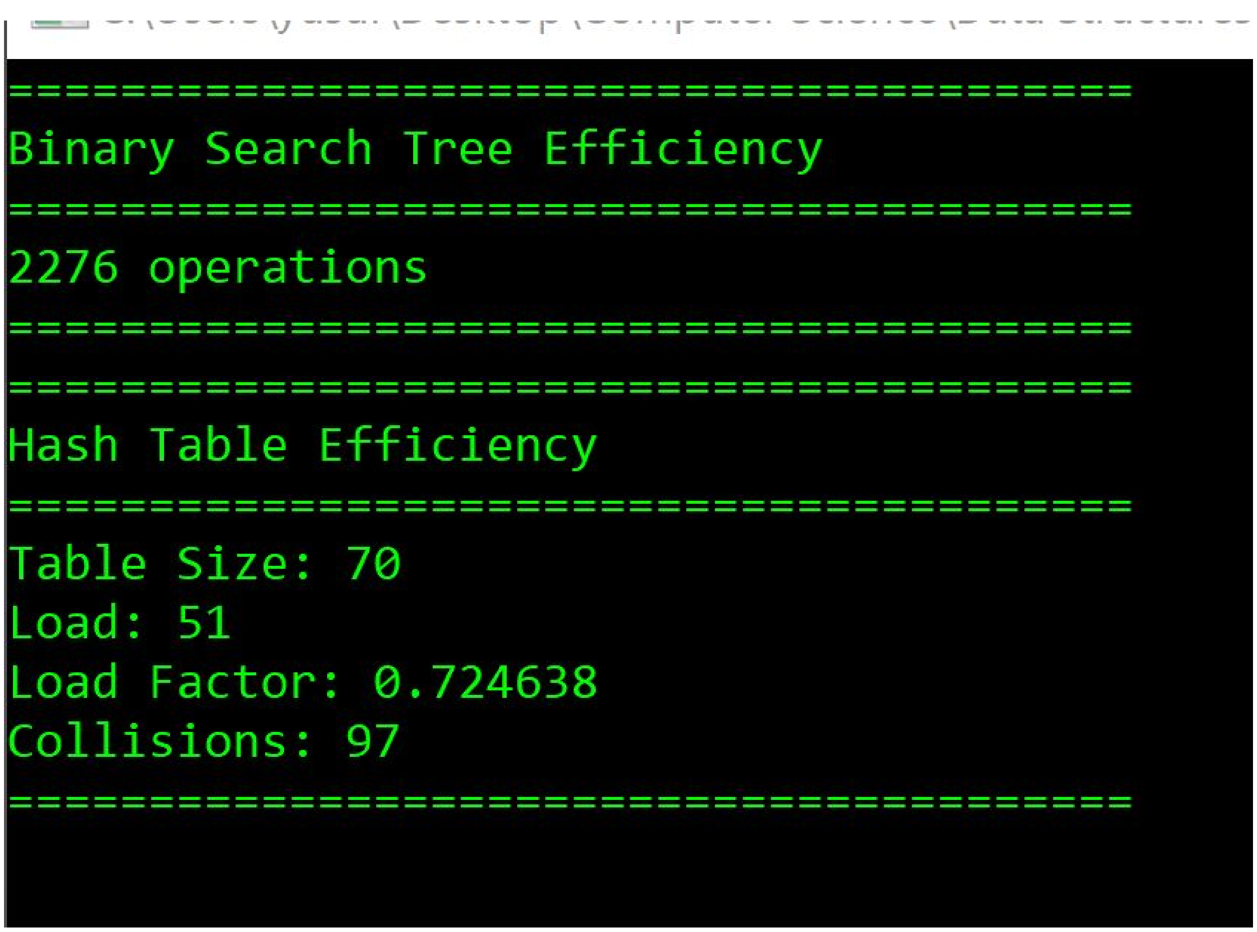
Print Indented Tree

Example:



```
        Allen Yow
      Amanda Davison
    Amber Low
                  Amy Lou
                Angela Griffin
                    Ben Royaduka
              Brandon Lowe
                  Brenda Watts
                      Bryan Lee
                          Caniel Daesar
            Carol Jackson
                  Delphine Shangguan
                Donna Allen
          Elijah Chapman
                Harrison Wood
              Henry King
        Hubert Lee
              Jackson Wang
            James Nguyen
                Joji Jones
  Jonathan Griffin
        Joseph James
            Josh Kang
      Joshua Stephans
Katrina Lo
          Lakisha Jona
        Lauren Barker
      Lucas Riley
            Matthew Neal
          Maxine Hicks
              Melissa Bailey
    Mike Wheeler
              Nabil Arbouz
            Nathan Wong
                    Niki Russel
                  Philip Little
                      Quennie Leu
                Rachel Pratt
          Renan Hiramatsu
                  Roger Padilla
                Roy Chang
                    Scott Price
                        Shanlai Ho
                            Tommy Chang
              Tracy Harper
                    Virgina Edwards
                        William Que
                  Willy Wong
                      Yunzi Choo
```

Efficiency:
Show Efficiency of BST and Hash Table. BST efficiency is based on operations that BST data structures have done
Show the Efficiency of Hash Table by showing Load Factor and Collisions



```
========================================
Binary Search Tree Efficiency
========================================
2276 operations
========================================
========================================
Hash Table Efficiency
========================================
Table Size: 70
Load: 51
Load Factor: 0.724638
Collisions: 97
========================================
```

ATM:

Follow Menu Prompt and do actions of any standard bank account when using an atm. Requires username password and bank id authentication

```
A
Bee
444-44-4444
3) List Account Details
1) Deposit
    Savings
    44
2) Withdraw
    Checking
    44
3) List Account Details
Back345
Quit
```