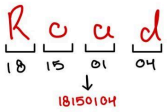
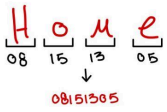
1. See design and remove test below
2. **Tests**
    a. Assigment 7.hpp line 29 function checks if imputed value is a string
    b. Assigment 7.hpp line 84 - 90 checks if word is in the provided array
3. **Fundamentals**
    a. Hashing
        i. Assigment 7.hpp lines 29 - 60 takes in a string and converts it to a hash
        ii. Assigment 7.hpp lines 63 - 93 takes in a hash and converts it to a string
    b. Insert
        i. Assigmnet 7 bt.hpp lines 28 - 69 adds the hashes to a BST
    c. Contains
        i. Assigmnet 7 bt.hpp lines 63 - 66 updates the count if collisions happen
    d. Delete
        i. Assigmnet 7 bt.hpp lines 143 - 274 deletes notes and returns their values
4. **Smarter hash**
    a. Assigmnet 7 bt.hpp lines 63 - 66 Utilized chaining by storing multiple values with the same hash key within the BST which effectively handles collisions
5. Collisions can have a impact on the worst case time complexity, average case for a BST is O(log n) but can be reduced to O(n) when all values are hashed to the same hash basically making it a linked list

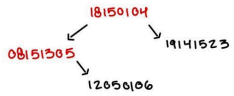
String -> Hash
· use a Map to assign alphabet to numbers
· use a for loop to loop through word and assign to number
· append each number to a string of numbers
· use string for ".append"

Hash -> String
· for loop that steps through the Hash
· take each number use the same map to unhash
· Return word

Storage
· use BST for storage
· My BST Handled collisions








```
Value: 2011811 Count: 1 L: null R: 19200118
Value: 2051212 Count: 1 L: null R: null
Value: 2091804 Count: 1 L: 2051212 R: 6091805
Value: 3151204 Count: 1 L: null R: null
Value: 4011811 Count: 1 L: 3151204 R: 4151518
Value: 4151518 Count: 1 L: null R: null
Value: 6091805 Count: 1 L: 4011811 R: null
Value: 6091908 Count: 1 L: 2091804 R: 12050106
Value: 8151305 Count: 1 L: null R: null
Value: 12050106 Count: 1 L: 8151305 R: null
Value: 13151514 Count: 1 L: 6091908 R: 18010914Value: 18010914 Count: 1 L: null R: 19141523
Value: 18150104 Count: 1 L: null R: null
Value: 18151520 Count: 1 L: 18150104 R: null
Value: 19141523 Count: 1 L: 18151520 R: 19151407
Value: 19151407 Count: 1 L: null R: null
Value: 19200118 Count: 1 L: 13151514 R: 23091404
Value: 23011813 Count: 1 L: null R: null
Value: 23091404 Count: 1 L: 23011813 R: null

Hash being deleted: 19200118 Word: star

Value: 2011811 Count: 1 L: null R: 23011813
Value: 2051212 Count: 1 L: null R: null
Value: 2091804 Count: 1 L: 2051212 R: 6091805
Value: 3151204 Count: 1 L: null R: null
Value: 4011811 Count: 1 L: 3151204 R: 4151518
Value: 4151518 Count: 1 L: null R: null
Value: 6091805 Count: 1 L: 4011811 R: null
Value: 6091908 Count: 1 L: 2091804 R: 12050106
Value: 8151305 Count: 1 L: null R: null
Value: 12050106 Count: 1 L: 8151305 R: null
Value: 13151514 Count: 1 L: 6091908 R: 18010914Value: 18010914 Count: 1 L: null R: 19141523
Value: 18150104 Count: 1 L: null R: null
Value: 18151520 Count: 1 L: 18150104 R: null
Value: 19141523 Count: 1 L: 18151520 R: 19151407
Value: 19151407 Count: 1 L: null R: null
Value: 23011813 Count: 1 L: 13151514 R: 23091404
Value: 23091404 Count: 1 L: null R: null
```