Algorithm Analysis and Data Structures: Assignment 6 – Hashing

Implement Hash table.

Pick 20 random words.  Each word must be of different lengths, maximum length 8 and minimum length 3.

The words will be of letters a-z, A-Z and the space character.

Insert them into a hash table.

You can use a library for only the hash function.

The collision resolution scheme should be open addressing - quadratic.

Initially the table size is 31.  The program should increase the table size and rehash at load factor of .5

At the end print the total number of collisions you get.

Code:

**package** Hashing;

**import** java.util.\*;

**public** **class** HashTable {

**public** **static** String[] *hashArray*;

**static** **double** *loadfactor* = 0.5;

**public** **static** **int** *size*;

**public** **int** tableSize = 31;

**public** **static** **int** *collisions* = 0;

**public** HashTable() {

*hashArray* = **new** String[tableSize];

}

**public** **int** hash(String sWord, **int** j) {

**int** hashval = 0;

**for** (**int** i = 0; i < sWord.length(); i++)

hashval += sWord.charAt(i);

**return** (hashval + (j \* j)) % tableSize;

}

**public** **void** InsertIntoHashTable(String s) {

**boolean** contains = **false**;

**if** (*size* > *loadfactor* \* tableSize)

resize();

**for** (**int** j = 0; j < *hashArray*.length; j++) {

**if** (*hashArray*[j] != **null**) {

**if** (*hashArray*[j].equalsIgnoreCase(s)) {

contains = **true**;

**break**;

}

}

}

**if** (!contains) {

**int** h = 0;

**while** (**true**) {

**int** i = hash(s, h);

**if** (*hashArray*[i] == **null**) {

*hashArray*[i] = s;

*size*++;

**return**;

} **else** {

*collisions* += 1;

h += 1;

}

}

}

}

**public** **void** resize() {

**int** prevSize = tableSize;

tableSize = *nextPrime*(prevSize \* 2);

String[] ref\_Array = *hashArray*;

*hashArray* = **new** String[tableSize];

*size* = 0; // setting size to zero

**for** (**int** i = 0; i < prevSize; i++) {

String s = ref\_Array[i];

**if** (s != **null**) {

InsertIntoHashTable(s);

}

}

}

**public** **static** **int** nextPrime(**int** n) {

**if** (n <= 0)

n = 3;

**if** (n % 2 == 0)

n++;

**for** (; !*isPrime*(n); n += 2)

;

**return** n;

}

**public** **static** **boolean** isPrime(**int** n) {

**if** (n == 2 || n == 3)

**return** **true**;

**if** (n == 1 || n % 2 == 0)

**return** **false**;

**for** (**int** i = 3; i \* i <= n; i += 2)

**if** (n % i == 0)

**return** **false**;

**return** **true**;

}

**public** **static** **void** main(String[] args) {

HashTable ht = **new** HashTable();

String[] input = { "aaa", "source", "data", "amore", "Cricket", "Crack", "volcano", "bxe", "zswe", "zaza", "abcde",

"ab vd", "and", "arm", "qwerty", "cho cho", "chings", "render", "ram", "random", "abcdge" };

**for** (**int** i = 0; i < input.length; i++) {

ht.InsertIntoHashTable(input[i]);

}

**for** (**int** i = 0; i < *hashArray*.length; i++) {

**if** (*hashArray*[i] != **null**)

System.***out***.println(i + "-" + *hashArray*[i] + " ");

}

System.***out***.println();

System.***out***.println("Total no of collisions: " + *collisions*);

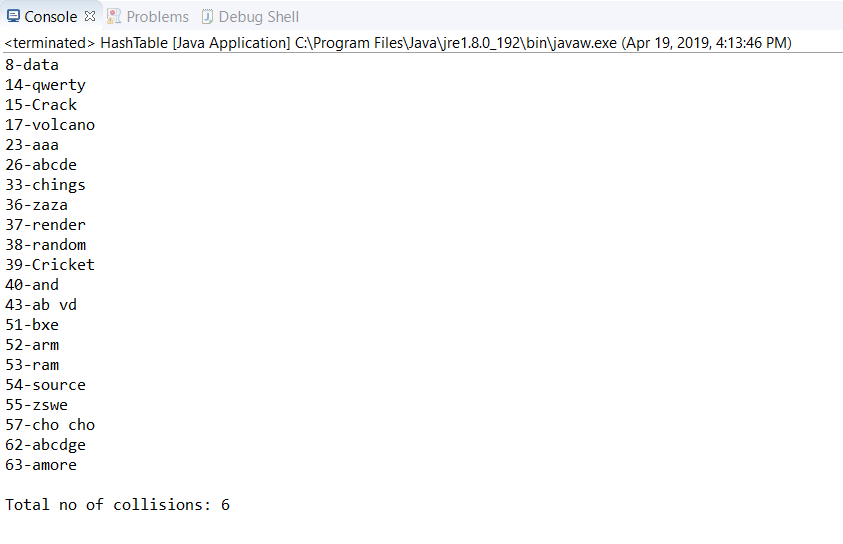
}

}

Instructions to compile:

1. Create a new java project in eclipse named Hashing
2. Create a new package in the source folder named Hashing
3. Create a java class in the same package with the name HashTable
4. Place the code and save
5. Run the code

Screen Shot of the output

****