function HashTable(size) {

this.buckets = Array(size);

this.numBuckets = this.buckets.length;

}

function HashNode(key, value, next) {

this.key = key;

this.value = value;

this.next = next || null;

}

HashTable.prototype.hash = function(key) {

var total = 0;

for (var i = 0; i < key.length; i++) {

total += key.charCodeAt(i);

}

var bucket = total % this.numBuckets;

return bucket;

};

HashTable.prototype.insert = function(key, value) {

var index = this.hash(key);

if (!this.buckets[index]) {

this.buckets[index] = new HashNode(key, value);

}

else if (this.buckets[index].key === key) {

this.buckets[index].value = value;

}

else {

var currentNode = this.buckets[index];

while (currentNode.next) {

if (currentNode.next.key === key) {

currentNode.next.value = value;

return;

}

currentNode = currentNode.next;

}

currentNode.next = new HashNode(key, value);

}

};

HashTable.prototype.get = function(key) {

var index = this.hash(key);

if (!this.buckets[index]) return null;

else {

var currentNode = this.buckets[index];

while (currentNode) {

if (currentNode.key === key) return currentNode.value;

currentNode = currentNode.next;

}

return null;

}

};

HashTable.prototype.retrieveAll = function() {

var allNodes = [];

for (var i = 0; i < this.numBuckets; i++) {

var currentNode = this.buckets[i];

while(currentNode) {

allNodes.push(currentNode);

currentNode = currentNode.next;

}

}

return allNodes;

};

var myHT = new HashTable(30);

myHT.insert('Dean', 'dean@gmail.com');

myHT.insert('Megan', 'megan@gmail.com');

myHT.insert('Dane', 'dane@yahoo.com');

myHT.insert('Dean', 'deanmachine@gmail.com');

myHT.insert('Megan', 'megansmith@gmail.com');

myHT.insert('Dane', 'dane1010@outlook.com');