**Part 2: Hash table with separate chaining**

In this part, you will store the competition results (<userid>, <score>) from Project 2 into a hash table. Implement all your functions in HashTableChaining.java

function specifications:

* A constructor that accepts the size of the hash table as a parameter and initializes the table with empty linked lists at each index
* A function *insert(key, value)* that adds the key to the hash table. A <key, value> pair associates value/record with a key.
* A function *remove(key)* that removes the key from the table, if it is in the hash table.
* A function *contains(key)* that returns a Boolean indicating if the key is in the table.
* A function *size()* that returns the number of keys in the table.
* A function *hash(key)* that returns the hash value of the hashcode of a key. Use Java built-in hashcode function. Since the hashcode can be negative, use 0x7fffffff as a mask to make it positive. ie. ("userid".hashcode() & 0x7fffffff) % capacity

You may assume that all the input, i.e. keys, are strings and all values are integers. All strings are delimited by a newline character. The initial size of the hash table will be provided, and it will be a prime number.

You may implement your own main method for your own testing but is it NOT required to pass test cases. To run all test cases, click the green button on the left of public class HashTableChainingTest and select Run 'HashTableChainingTest'. You may also run each individual test by clicking the green button on the left of each test and select run.

Test cases details:

hash(): check whether your hash() function is assigning the correct index to the userid

insert\_contains(): check whether insert is successful and whether contains() function is detecting the inserted element

size() : check whether elements are inserted properly and if size() function is returning the correct size

remove(): check whether remove an element is successful and if contains() is detecting the removal

Note: make sure your contains() is working correctly because both insert() and remove() are tested using contains()

Note: in the testcase, change the directory of the file to where you put the file. For example, if you are writing your HashTableChaining.java is in /src and you put test.txt in /src, it will be Scanner sc = **new** Scanner(**new** File(**"src/test.txt"**));