CSC 212 Programming Assignment # 2
Linear Hashing
Due date: 14/12/2017

Guidelines:	This is an **individual** assignment.
	The assignment must be submitted to **Web-CAT**

Complete the code of the class *LinearHash* below, which implements a hash table where:

- The hash function is division (the table size is passed as parameter to the constructor).

- The collision resolution strategy in linear rehashing (c is passed as parameter to the constructor).

You need to implement the two methods:

- *insert*: Inserts the key and data in the hash table. Assume that the key does not exist (no need to check for that). The method returns the number of probes that were needed to insert the key.

- *find*: Searches for a key k. If k exists *current* is set to point to that element and true is returned. If the key does not exist, *current* does not change and false is returned.

Do not change the other methods.

```
public class LinearHash<T> {

        public enum Status {
                empty, occupied, deleted
        };

        private int maxSize;
        private int size;
        private int c;
        private int current;
        private int[] keys;
```

```java
        private Status[] statusT;
        private T[] data;

        public LinearHash(int maxSize, int c) {

                this.maxSize = maxSize;
                this.c = c;
                size = 0;
                current = -1;
                keys = new int[maxSize];
                statusT = new Status[maxSize];
                data = (T[]) new Object[maxSize];

                // Initialize all cells to empty
                for (int i = 0; i < maxSize; i++) {
                        statusT[i] = Status.empty;
                }
        }

        public int size() {
                return size;
        }

        public boolean full() {
                return size == maxSize;
        }

        public T retrieve() {
                return data[current];
        }

        public void update(T val) {
                data[current] = val;
        }

        public void delete() {
                statusT[current] = Status.deleted;
                size--;
        }

        public int insert(int key, T val) {
        }

        public boolean find(int key) {
        }
}
```