```
package finalhashmap;

 import.java.util.ArrayList;

 public class Map<K,V> {

     ArrayList <Map<K,V>> buckets;
     int count;
     int numBuckets;

     public Map() {
         buckets = new ArrayList<>();
         numBuckets = 20;
         for(int i=0;i<numBuckets;i++) {
             buckets.add(null);
         }
     }
    private int getBucketIndex(K,Key) {
        int hc = Key.hashCode();
        int index = hc%numBuckets;
        return index;
    }

    public int size() {
        return count;
    }
    public V getValue(K Value) {

    }


   public void insert (K value,V Value) {

       int bucketIndex = getBucketIndex(Key);
       MapNode<K,V> head = buckets.get(bucketIndex);
         MapNode<K,V> prev = null;

       while(head! = null) {
           if(head.key.equals(key)) {
               if(prev! = null) {
                   prev.next = head.next;
               } else {
                   buckets.set(bucketIndex,head.next);
               }
               count--;
               return head.value;
           }
           prev = head;
```

```
        head = head.next;
    }

    return null;
}

private void reHash() {
    ArrayList<MapNode<K,V>> temp = buckets;
    buckets = new ArrayList<>();
    for(int i=0;i<2*numBuckets;i++) {
        buckets.add(null);
    }
    count = 0;
    numBuckets = numBuckets*2;
    for(int i=0;i<temp.size();i++) {
        MapNode<K,V> head = temp.get(i);
        while(head! = null)
        K Key = head.key;
        V value = head.value;
        insert(key,value);
        head = head.next;
    }

}

    // element is not there.insert at 0th position of linked list
    head = buckets.get(bucketIndex);
     MapNode<K,V> head =  new  MapNode<>(Key,Value);
     newNode.next = head;
     buckets.set(bucketIndex ,  newNode);
     count++;
     double loadFactor = (1.0* count)/numBuckets;
     if(loadfactor> 0.7) {
         reHash();
     }

}
```

MapUse.Java

```
 package finalhashmap;

public class MapUse {

    public static void main(String[] args) {
        Map<String, Integer> map = new Map<>();
        for(int i=0;i<20;i++) {
            map.insert("abc" +i, i+1);
            System.out.println(map.loadFactor());
        }
        map.removeKey("abc3");
        map.removeKey("abc7");
        for(int i=0;i<20;i++) {
            System.out.println("abc"+ i+":" + map.getvalue("abc" + i));
        }
    }
}
```