package com.iimtiaz.day\_15;  
  
import java.util.Iterator;  
import java.util.LinkedHashMap;  
import java.util.Map;  
  
public class LRU\_Cache {  
 public static void main(String[] args) {  
  
 // Create a cache with capacity 2  
 LRUCache lRUCache = new LRUCache(2);  
  
 // Test the put and get methods  
 lRUCache.put(1, 1); // cache is {1=1}  
 lRUCache.put(2, 2); // cache is {1=1, 2=2}  
 System.*out*.println(lRUCache.get(1)); // returns 1  
 lRUCache.put(3, 3); // LRU key was 2, evicts key 2, cache is {1=1, 3=3}  
 System.*out*.println(lRUCache.get(2)); // returns -1 (not found)  
 lRUCache.put(4, 4); // LRU key was 1, evicts key 1, cache is {4=4, 3=3}  
 System.*out*.println(lRUCache.get(1)); // returns -1 (not found)  
 System.*out*.println(lRUCache.get(3)); // returns 3  
 System.*out*.println(lRUCache.get(4)); // returns 4  
 }  
}  
  
class LRUCache {  
  
 int capacity;  
 Map<Integer, Integer> map = new LinkedHashMap<>();  
  
 public LRUCache(int capacity) {  
 this.capacity = capacity;  
 }  
  
 public int get(int key) {  
 if (map.containsKey(key)) {  
 Integer val = map.get(key);  
 map.remove(key);  
  
 map.put(key, val);  
 return map.get(key);  
 } else return -1;  
 }  
  
 public void put(int key, int value) {  
 if (map.containsKey(key)) {  
 map.remove(key);  
 map.put(key, value);  
 } else {  
 if (map.size() < capacity) {  
 map.put(key, value);  
 } else {  
 // removing first element in the map.  
 Iterator<Integer> iterator = map.keySet().iterator();  
 map.remove(iterator.next());  
 map.put(key, value);  
 }  
 }  
 }  
}