**题目描述：**

**数列的定义如下： 数列的第一项为n，以后各项为前一项的平方根，求数列的前m项的和。**

**输入**

**输入数据有多组，每组占一行，由两个整数n（n<10000）和m(m<1000)组成，n和m的含义如前所述。**

**输出**

**对于每组输入数据，输出该数列的和，每个测试实例占一行，要求精度保留2位小数。**

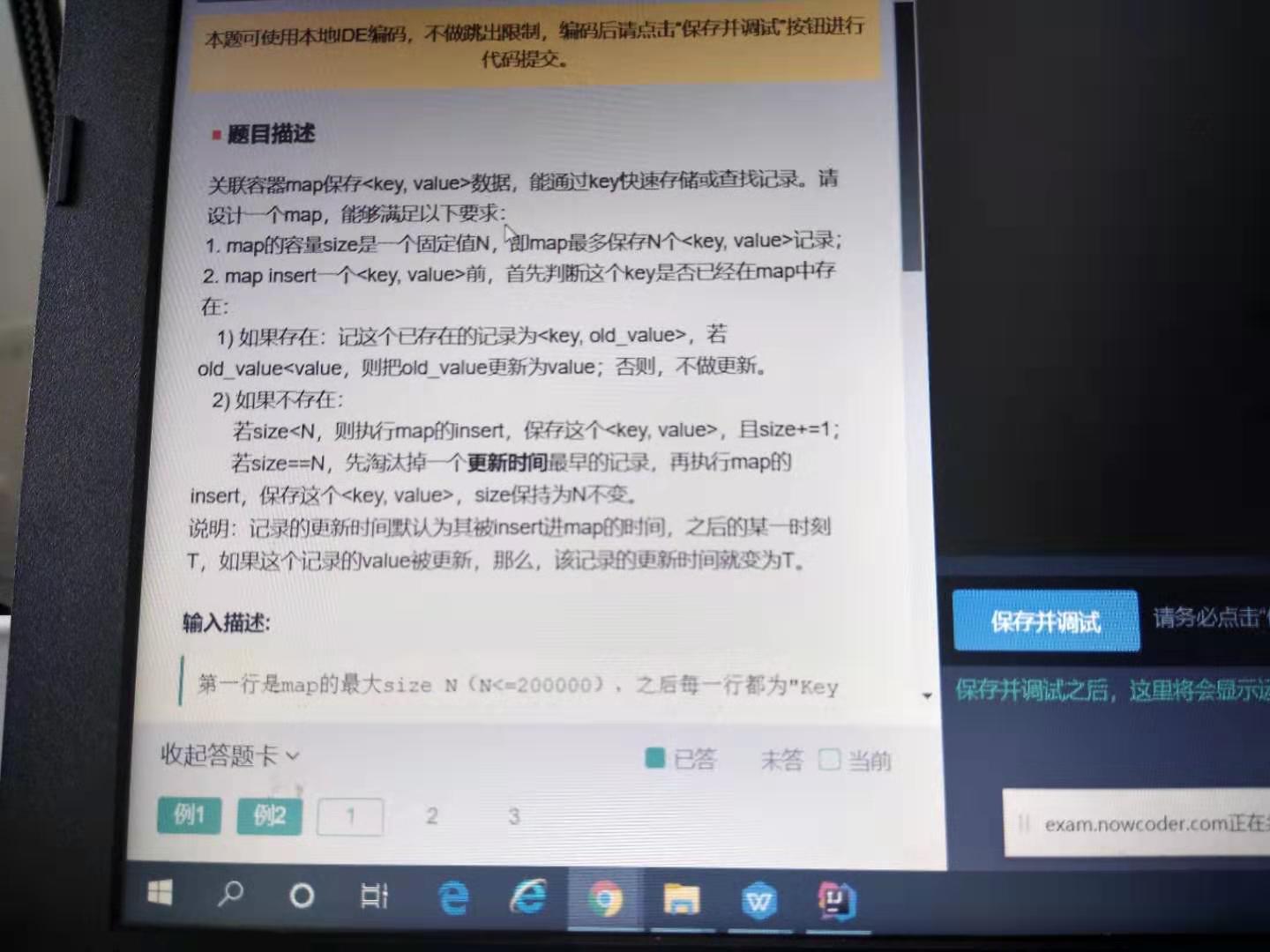
**样例输入**

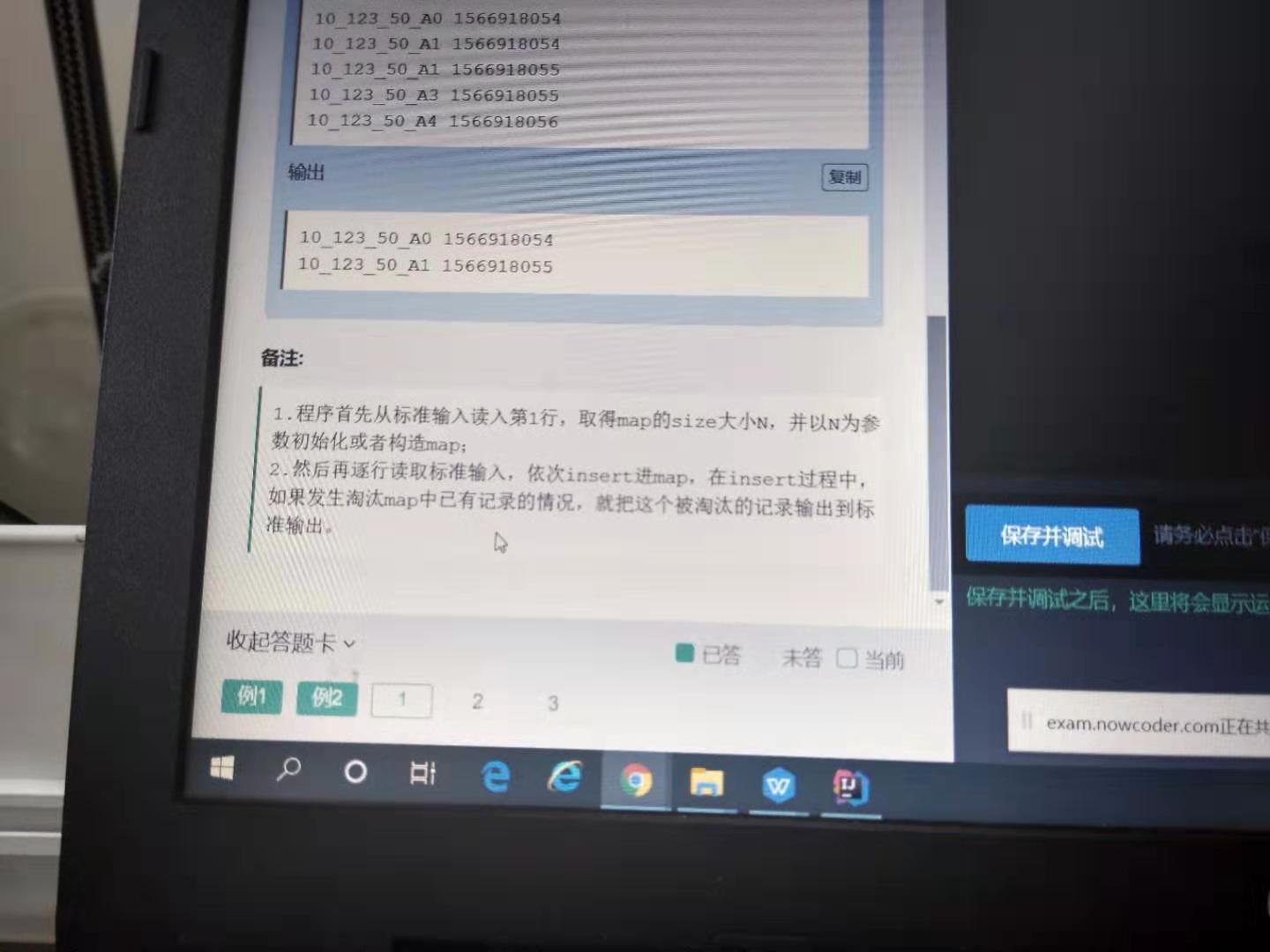
**81 4  
2 2**

**样例输出**

**94.73  
3.41**

**public static void main(String[] args){  
 int m;  
 double sum,n;  
 Scanner scanner = new Scanner(System.*in*);  
 while (scanner.hasNext()){  
 n = scanner.nextInt();  
 m = scanner.nextInt();  
 sum = 0;  
 for (int i = 0; i < m; i++){  
 sum = sum + n;  
 n = Math.*sqrt*(n);  
 }  
 System.*out*.printf("%.2f", sum);  
 System.*out*.println();  
 }**





**package com.qst.dao;  
  
import java.util.LinkedHashMap;  
import java.util.Map;  
import java.util.Scanner;  
*/\*\*  
 \* @Description :  
 \* @Author: Liruilong  
 \* @Date: 2019/9/8 16:47  
 \*/*public class Main {  
 public static LinkedHashMap<String , Long> *linkedHashMap* = new LinkedHashMap<>();  
 public static Integer *size* = 0 ;  
 public static void print(){  
 Scanner scanner = new Scanner(System.*in*);  
 *size* = scanner.nextInt();  
 while (true){  
 *insert*(scanner.next(),scanner.nextLong());  
 }  
 }  
 public synchronized static void insert(String string, long longs){  
 if (*linkedHashMap*.containsKey(string)){  
 *// 存在* if (*linkedHashMap*.get(string) < longs){  
 *linkedHashMap*.put(string, longs);  
 }  
 }else {  
 if ( *linkedHashMap*.size() < *size* ){  
 *linkedHashMap*.put(string, longs);  
 }else if (*linkedHashMap*.size() == *size*){  
 for (Map.Entry<String, Long> entry : *linkedHashMap*.entrySet()){  
 System.*out*.println();  
 System.*out*.println(entry.getKey()+ " " + entry.getValue());  
 *linkedHashMap*.remove(entry.getKey());  
 *linkedHashMap*.put(string, longs);  
 break;  
 }  
 }  
 }  
 }  
 public static void main(String[] args) {  
 *print*();  
 }  
}**