谈项目-学习方法-摸索：

做项目：筛选。

语言/框架

总结：学的东西又学习了一遍。总结一次之后，和自己以前看的很像，学了java，看了官方文档，然后看python的文档，真的就很简单。

博客小日记。总结是第二次创作。（老师的十分能力，只能讲7分。你要学习之后，你要写总结，二次学习。）

认真做一个项目，中国人，人很多，但是平凡的人跟多。

（平凡的人太多，稍加努力。你就是top）

你做过一些东西，你才是top的那批人。

解决问题的思路：官方文档》权威指南》stackOverFlow》google》bing》

项目：（面试官关注点。）

1. 能力：

**你了解过什么东西**：用了什么框架。用了哪些模块。一点要自己总结好。框架的架构描述。主要框架部分，然后对sping的

**你深入了解过什么东西**（有没有什么可以了解的。深入的了解：底层数据连接。你要总结，你要脱颖而出。）

**你横向了解哪些**（单机—网络—智能模式。通用的一些总结。对比总结，方法论。）

**你认证找你哪些很水的项目的闪光点，了解共性。**

1. 潜力（差异化）

你怎么解决问题的：（你是怎么解决问题的。你学习的方法论。你的方法论可行，我才要你）

你如何举一反三：（让我想起我做的另一个项目是怎么做的。）

你怎么优化项目：（你的潜力，传输的东西更少，结果一样。带宽降低，可以提高并发量）

你如何快速学习：（还是方法论，基于经验学会的。）

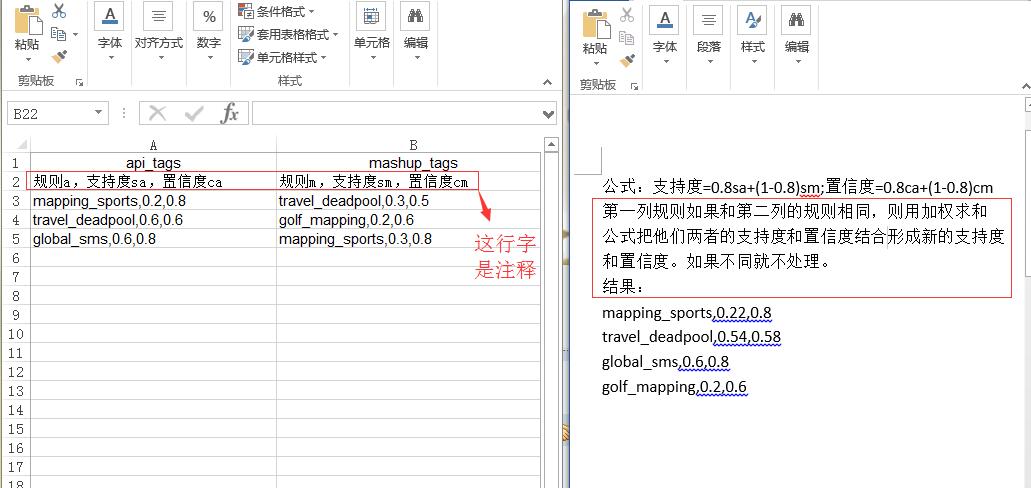
1. 不要吹牛逼，一定要自己先准备好。不要吹，（自己要事先准备好，非常关键。）（数据网站怎么做的。方案对比，提高。）（算法，你不能说你做出来，看别人做出来。这个真的不行，一步一步爬坡，这个非常重要，一个完整的项目，有一两个闪光点。这个非常重要。你不能指望别人糊里糊涂的就把你招进出。（所以你现在学习的时候，真的不要太多。而是要有些深度，要提升。）） （你一个项目就可以体现你的能力，不用想太多。差异化，深入点。做了项目，准备好总结。你做项目的问题）
2. 非常重要的点，你得有。这个非常非常关键。比你学号一点点知识点总要得多。任何事，都有

每天都围绕这六条去学习，记录，总结。你就是王者。

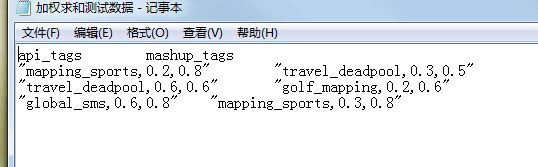
|  |  |  |
| --- | --- | --- |
| 项目准备：（方案）（有一年的时间准备，你比别人优秀的地方就在这里了） | | |
| 一：能力考察。 | | |
| 1. 你了解过什么东西：   【用了什么框架】【做了什么模块】 |  |  |
| 2.你深入了解了什么东西：  【底层是怎么实现的】 |  |  |
| 3.你横向了解了什么东西：  【和你学到的其他东西有什么相关性。】 |  |  |
| 二：潜力考察（体现差异化）【展现你的方法论】 | | |
| 你怎么解决问题的：（你是怎么解决问题的。你学习的方法论。你的方法论可行，我才要你） | 【最致命：跟着视频一步一步学习。】  一定是：  解决问题的思路：官方文档》权威指南》stackOverFlow》google》bing》  Yahoo |  |
| 你如何举一反三：（让我想起我做的另一个项目是怎么做的。） | 比如：【经验1】struts2里面的用到的传值，设置好getset方法，就可以和前台直接用对象.属性名就获取这个属性值。前台设值也一样。深入这里面的原理。就知道：不过是在action中，利用反射机制，来生成对象实例。加上序列化和反序列化的方式做的。  【经验2】另外一个方法是：hibernate中有一个过程是RMO(关系型数据转换为对象型数据的这个映射过程。) 比如后台创建这样一个实体类的时候，就给这个类用上一个方法，让关系型数据映射为我们的对象，为该对象相关属性（用数据库里的字段对应）进行初始化。  【经验3】这个过程你有两个地方用到了，一个是和长理的本科生讲课的时候，用到这个和他们进行讲解，数据库里的一行记录如何变为一个对象的。  【经验4】听马老师的bbs中好像有一个点是javaBean和jsp开发程序的时候，就用到了javabean（可以说起pojo类，和javabean，eneity，model,等概念。）  【经验5】外一个就是你给夏艳敏写她的实验的时候，列表一行一行数据是如何处理的。一个excell表格里的数据是按格式进行存储。现在要对这个数据进行处理。这样的数据有多行，如何进行处理。下面会给这个数据的一个处理方案和做法，感觉自己真是太牛了。哈哈。  【rom-】我把excell表格里的每一个单元格的内容映射为一个对象。然后让对象的字段做运算。这就是一种很经典的方式，建立在自己的摸索之上，就非常有特色。  关于有了这些经验之后，你要知道如何去学习和用已有的学习经验去匹配你以后的学习。  【拓展1】：json的思想。没有仔细了解，必定也是利用了序列化的思想：统一格式，便于解析成对象。那么扩展到xml格式。那还有问题吗？  【拓展2】到RPC(remote procedure call)这样一个概念，那么应该也比较好理解。知道远程的数据格式。然后再我们的程序中开始直接解析数据，生成对象就可以。就是rpc  【扩展3】rpc的一个变现webservice的方法，这里不就简单了么，比如google地图的应该就是提供这样一个服务吧，我们把请求发出，地图服务器端根据我们的请求返回给我们需要的地图数据就可以了，我们这边再解析（这个解析地图都帮我们做好了的，他直接有自己写好了的jar包）  【扩展4】多语言支持机制，这个就是用的类似json这种思想。就是把数据通过标准输入输出来处理，然后用序列化思想和反序列化思想来处理数据。就这样而已。 |  |
| 你怎么优化项目：（你的潜力，传输的东西更少，结果一样。带宽降低，可以提高并发量） |  |  |
| 你如何快速学习：（还是方法论，基于经验学会的。） |  |  |
|  | | |
| 面试官说的每一句话，都是有策略的。那么同样，你所说的每一句话，也要引导面试官向你说的那些话的方向去提问。这样你才有优势（葛亮舌战群儒的被动肯定不希望在面试的时候被你遇上吧。好好把握这样的锻炼机会，你的起薪肯定不是你的技术决定的，而是你说话的能力，你之前面对别人的面试问题回答的水平决定的。你最终的工资，也不是你的能力决定的看，而是你的起薪+你的真实水平。）  例子：面试官问你，spark的问题。你说是函数式编程，有点像java8里的lambda表达式。如果你对lambda很了解，能够说清楚很多原理来。这句话就有得说。  别人可是有面试指南的哦。你如果傻傻的不知道。那么你就彻彻底底输给了别人。所以你要早做准备。重要的真不是技术，而是你把该展现的东西展现出来。 | | |

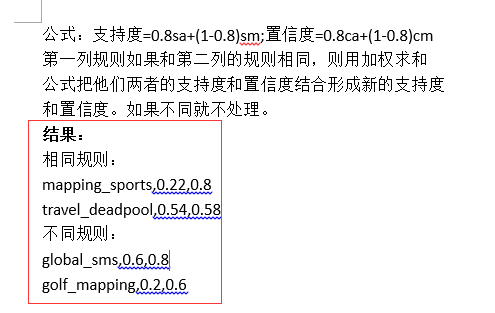
某同学的实验数据处理解决案例：

1.需求：

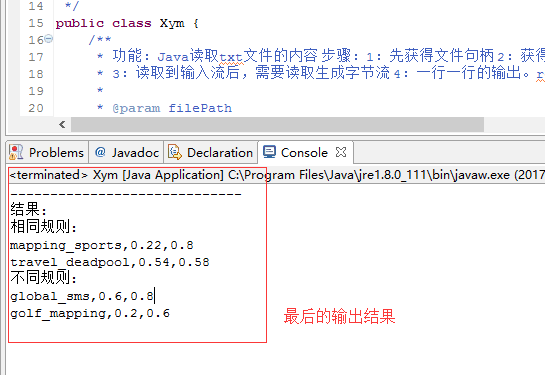


就是如果是两列中相同的规则，就用公式求和一下，不同就直接输出来





最终我的处理结果：大概花了两个小时解决了的方法。也是自己没有多动手。



处理方法代码是这样的。

|  |
| --- |
| 代码： |
| package leet.array;  import java.io.BufferedReader;  import java.io.File;  import java.io.FileInputStream;  import java.io.InputStreamReader;  import java.text.DecimalFormat;  import java.util.ArrayList;  import java.util.Iterator;  import java.util.List;  /\*\*  \* @author jkb  \*/  public class Xym {  /\*\*  \* 功能：Java读取txt文件的内容 步骤：1：先获得文件句柄 2：获得文件句柄当做是输入一个字节码流，需要对这个输入流进行读取  \* 3：读取到输入流后，需要读取生成字节流 4：一行一行的输出。readline()。 备注：需要考虑的是异常情况  \* 4.解析文本数据。  \* @param filePath  \*/  public static void readTxtFile(String filePath) {  try {  String encoding = "GBK";  File file = new File(filePath);  if (file.isFile() && file.exists()) { // 判断文件是否存在  InputStreamReader read = new InputStreamReader(  new FileInputStream(file), encoding);// 考虑到编码格式  BufferedReader bufferedReader = new BufferedReader(read);  String lineTxt = null;  lineTxt = bufferedReader.readLine();  int i = 2;  //list1用于指向源数据的第1列  //list2用于指向源数据的第2列  //list3用于指向更改后数据的相同规则  //list4用于指向更改后数据的不同规则  List<Person> list1 = new ArrayList<Person>();  List<Person> list2 = new ArrayList<Person>();  List<Person> list3 = new ArrayList<Person>();  List<Person> list4 = new ArrayList<Person>();  while ((lineTxt = bufferedReader.readLine()) != null) {  // System.out.println(lineTxt);  String s1 = lineTxt;  String[] strings = s1.split("\t");  for (int j = 0; j < strings.length; j++) {  String s2 = strings[j].substring(1,  strings[j].length() - 1);  String[] para = s2.split(",");  Person person = new Person();  person.setName(para[0]);  person.setS(Double.parseDouble(para[1]));  person.setC(Double.parseDouble(para[2]));  if (i % 2 == 0) {  list1.add(person);  } else {  list2.add(person);  }  i++;  }  }  // System.out.println(list1);  // System.out.println(list2);    //迭代处理list中的内容：  change(list1, list2, list3, list4);  System.out.println("-----------------------------");  System.out.println("结果:");  System.out.println("相同规则:");  for (Iterator iterator = list3.iterator(); iterator.hasNext();) {  Person person = (Person) iterator.next();  System.out.println(person.toString());  }  System.out.println("不同规则:");  for (Iterator iterator = list4.iterator(); iterator.hasNext();) {  Person person = (Person) iterator.next();  System.out.println(person.toString());  }  read.close();  } else {  System.out.println("找不到指定的文件");  }  } catch (Exception e) {  System.out.println("读取文件内容出错");  e.printStackTrace();  }  }  /\*\*  \* 辅助方法(处理解析后的数据。分类存放)  \* @param list1指向源数据的第1列  \* @param list2指向源数据的第2列  \* @param list3用于指向更改后数据的相同规则  \* @param list4用于指向更改后数据的不同规则  \*/  public static void change(List list1,List list2,List list3,List list4){  list1.iterator();  for (Iterator iterator = list1.iterator(); iterator.hasNext();) {  Person p1 = (Person) iterator.next();  for (Iterator iterator2 = list2.iterator(); iterator2.hasNext();) {  Person p2 = (Person) iterator2.next();  if (p1.getName().equals(p2.getName())) {  p1.setS( 0.8\*p1.getS() +0.2\*p2.getS());  p1.setC( 0.8\*p1.getC() + 0.2\*p2.getC());  list3.add(p1);  iterator.remove();  iterator2.remove();  }  }  }  for (Iterator iterator = list1.iterator(); iterator.hasNext();) {  Person p1 = (Person) iterator.next();  list4.add(p1);  }  for (Iterator iterator = list2.iterator(); iterator.hasNext();) {  Person p2 = (Person) iterator.next();  list4.add(p2);  }  }  public static void main(String argv[]) {  String filePath = "D:\\SoftWare\\eclipse\\com.jiang.test\\JavaData\\src\\加权求和测试数据.txt";  // "res/";  readTxtFile(filePath);  }  }  class Person {  private String name;  private double s;  private double c;  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }  public double getS() {  return s;  }  public void setS(double s) {  this.s = s;  }  public double getC() {  return c;  }  public void setC(double c) {  this.c = c;  }  @Override  public String toString() {  DecimalFormat df=new DecimalFormat("########.##");  return name + "," + df.format(s) + "," + df.format(c);  }    }  /\*  \* 删除字符串最后一个字符的几种方法  \*  \* 偶然看到的，记录一下，以免忘记  \*  \* 字符串：string s = "1,2,3,4,5,"  \*  \* 目标：删除最后一个 ","  \*  \* 方法： 1、用的最多的是Substring，这个也是我一直用的  \*  \*  \* s = s.Substring(0,s.Length - 1)  \*  \*  \* 2、用 RTrim，这个我原来只知道用来删除最后的空格，也没有仔细看过其它的用法，才发现可以直接trim掉一些字符  \*  \*  \* s = s.ToString().RTrim(',')  \*  \* 3、用TrimEnd,这个东西和RTrim差不多，区别是这个传递的是一个字符数组，而RTrim可以是任何有效的字符串  \*  \*  \* s=s.TrimEnd(',') //如果要删除"5,"，则需要这么写 char[] MyChar = {'5',','}; s =  \* s.TrimEnd(MyChar); //s = "1,2,3,4"  \*  \* ======================================================== Map<String,String>  \* paraMap=new HashMap<String,String>(); String  \* str="name=zhang,age=30,sex=male"; String[] para=str.split(","); for(int  \* i=0,len=para.length;i<len;i++){ String[] temp=para[i].split("=");  \* paraMap.put(temp[0], temp[1]); } Person p=new Person();  \* p.setName(paraMap.get("name"));  \*/ |

第二篇：

给师姐解答问题：

1.为什么Java里的数组的长度是用length，字符串的长度是用string.length();来做的。能解释一下吗？（更高明的文法，数组取长度和字符串取长度，这两个的区别是什么）开始讲java的面向对象，因为array里封装了一个public属性是length，而且这个属性的值是在初始化对象的时候就赋值了的。而string里是封装了一个public方法，这个方法名为length（）。

【拓展1】System.out.println()这个方法，你经常用吧，能说一下这句话，那是类，那个属性，那个是方法么？答：system是类。Out是system中封装的属性，println（）是out属性里的一个方法。【体现的java里的面向对象，封装，成员和方法】

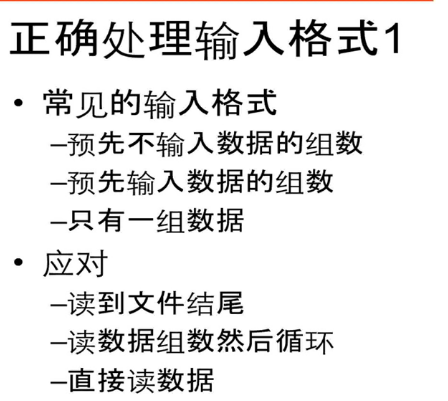
1. 解决编程问题【链家2018届】

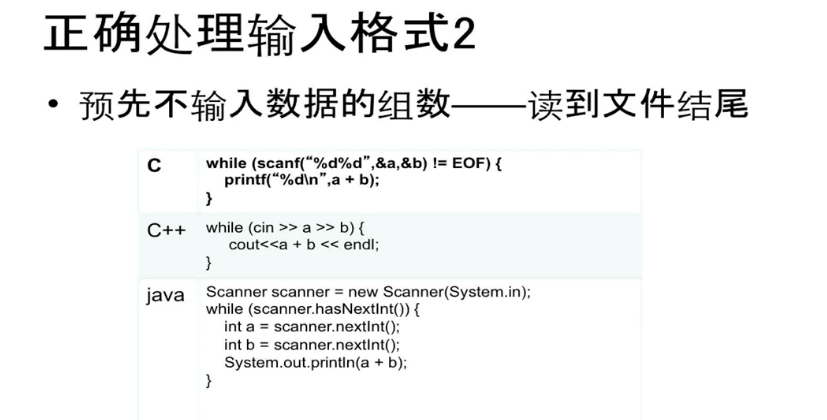
|  |  |
| --- | --- |
| 题目： | D:\Tencent\1344416858\Image\C2C\784648CD4593E20D88A64D8DB046738E.png |
| 思路： | 这个第一步。想法是。如果只有两个区域（2,3区域），那么只用计算本不该在2位置的元素个数就行了，那么就是最终结果。  第二步：如果把2和3看做是一个整体，那么不属于1位置的数据个数就很好计算。  第三步：1位置搞定之后，2位置和3位置，就可以看两个区域。 |
|  | 解决问题的结果是：  Sum = (a12+a13)+Math.max(a23,a32);  真的就是soeasy。 |
| 代码： | package HuaWei;  import java.util.Scanner;  import javax.swing.text.ChangedCharSetException;  public class Main {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  int[] a = new int[scanner.nextInt()];  for (int i = 0; i < a.length; i++) {  a[i] = scanner.nextInt();  }  int sum = change(a);  System.out.println(sum);  }  private static int change(int[] a) {  int count1 = 0;  int count2 = 0;  int count3 = 0;  for (int i = 0; i < a.length; i++) {  if(a[i]==1){  count1++;  }else if (a[i]==2) {  count2++;  }else {  count3++;  }  }  int a12 =0;//代表2占的1位置的个数  int a13 = 0;//代表3占的1位置的个数  int a21 = 0;//代表1占的2位置的个数  int a23 = 0;//代表3占的2位置的个数  int a31 = 0;//代表1占的3位置的个数  int a32 = 0;//代表2占的3位置的个数  for (int i = 0; i < count1; i++) {  if(a[i]==1){  }else if (a[i]==2) {  a12++;  }else {  a13++;  }  }  for (int i = 0; i < count2; i++) {  if(a[i+count1]==1){  a21++;  }else if (a[i+count1]==2) {  }else {  a23++;  }  }  for (int i = 0; i < count3; i++) {  if(a[i+count1+count2]==1){  a31++;  }else if (a[i+count1+count2]==2) {  a32++;  }else {  }  }  int sum = a12+a13;  sum += Math.max(a23, a32);  return sum;  }  } |

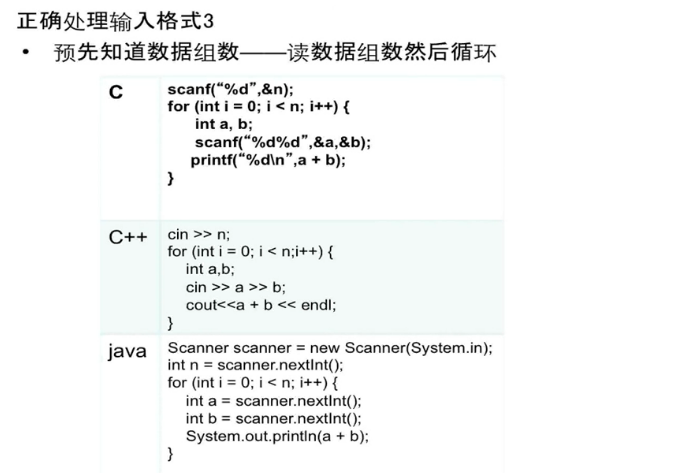
1. 华为的题目：

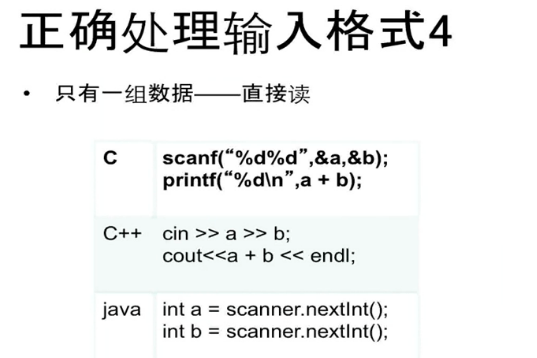
|  |  |
| --- | --- |
| 题目 | D:\Tencent\1344416858\Image\C2C\12E7128DCC7C548495B584C302389019.png |
| 思路： | 核心代码  if(c1[i]==c2[i]){  nowCount++;  maxCount = Math.max(maxCount, nowCount);  }else {  nowCount = 0;  } |
| 解法： | package HuaWei;  /\*\*  \* 陈嘉苗师姐2017.8.16 华为机试  \* @author load  \*  \*/  import java.util.Scanner;  public class FindMatch {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  //System.out.println(findMatch(scanner.nextLine(),scanner.nextLine()));  System.out.println(findMatch(scanner.next(), scanner.next()));  }  public static int findMatch(String str1,String str2){  //1.处理边界情况,长度不等，或者非数值。  if (str1.length()!= str2.length()) {  return -1;  }  try {  int i = Integer.parseInt(str1);  int j = Integer.parseInt(str2);  } catch (NumberFormatException e) {  return -1;  }  //2.主要业务情况  char[] c1= str1.toCharArray();  char[] c2 = str2.toCharArray();  int maxCount = 0;  int nowCount = 0;  for (int i = 0; i < c1.length; i++) {  if(c1[i]==c2[i]){  nowCount++;  maxCount = Math.max(maxCount, nowCount);  }else {  nowCount = 0;  }  }  return maxCount < 2 ? 1:maxCount;  }  } |

第三篇：oj使用篇章。

1. 输入格式处理
2. 

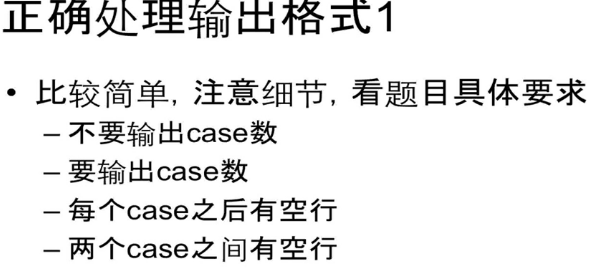


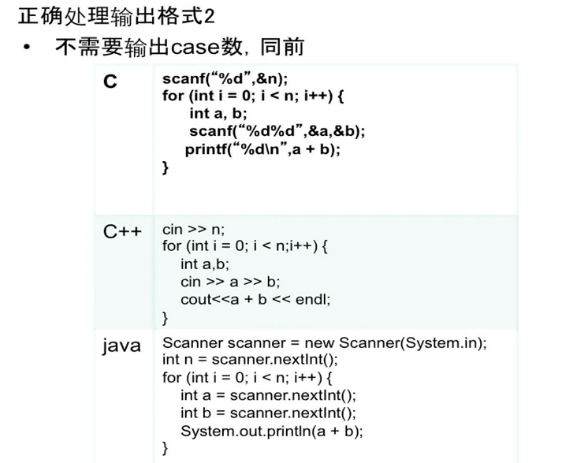




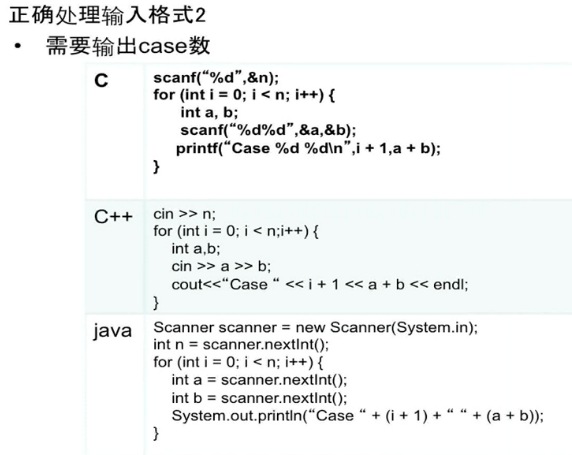
2.输出：

2.1不需要case数

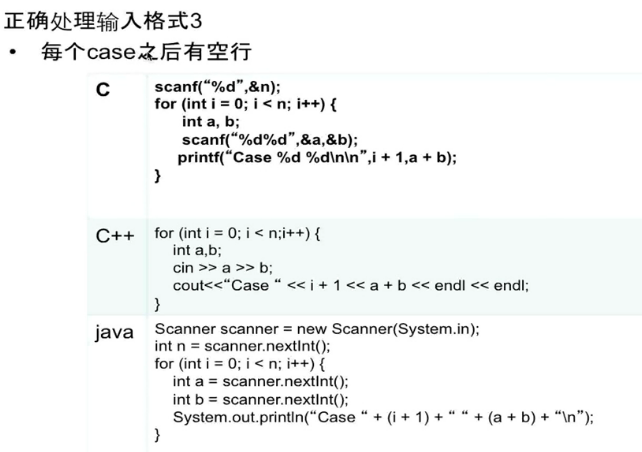


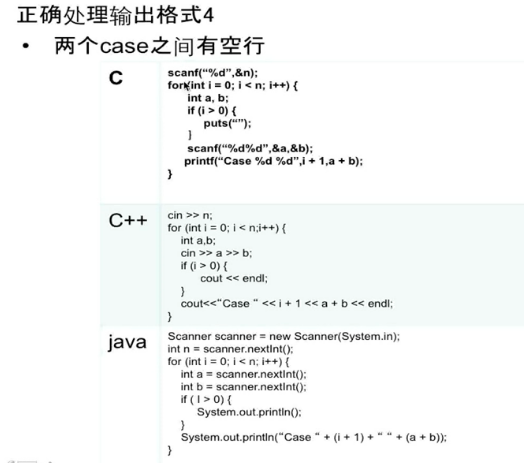


2.2需要case数



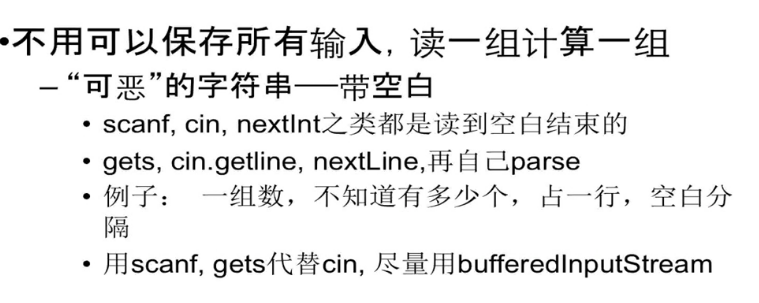
2.3每个case之后有空行

2.4两个case之间有空行



3.输入输出注意事项

3.1输入注意事项



3.2输出注意事项：

