**全国计算机技术与软件专业技术资格（水平）考试**

**2012年下半年 程序员 下午试卷**

（考试时间 14:00～16:30 共 150 分钟）

|  |
| --- |
| **请按下述要求正确填写答题纸** |

1.在答题纸的指定位置填写你所在的省、自治区、直辖市、计划单列市的名称。

2.在答题纸的指定位置填写准考证号、出生年月日和姓名。

3.答题纸上除填写上述内容外只能写解答。

4.本试卷共6道题，试题一至试题四是必答题，试题五至试题六选答 1 道。每

题 15 分，满分 75 分。

5.解答时字迹务必清楚，字迹不清时，将不评分。

6.仿照下面例题，将解答写在答题纸的对应栏内。

**例题**

2012 年下半年全国计算机技术与软件专业技术资格（水平）考试日期是（1）

月（2）日。

因为正确的解答是“11 月 4 日”，故在答题纸的对应栏内写上“11”和“4”

（参看下表）。

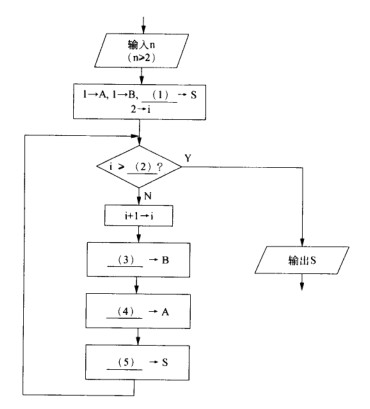
|  |  |
| --- | --- |
| 例题 | 解答栏 |
| （1） | 11 |
| （2） | 4 |

**试题一**

**【说明】**

本流程图用于计算菲波那契数列{a1=1，a2=1，…,an=an-1+an-2!n=3,4,…}的前n项（n>=2) 之和S。例如，菲波那契数列前6项之和为20。计算过程中，当前项之前的两项分别动态地保存在变量A和B中。

**【流程图】**



阅读说明和流程图，填补流程图中的空缺（1）〜（5）

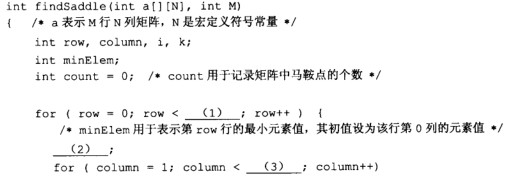
**试题二**

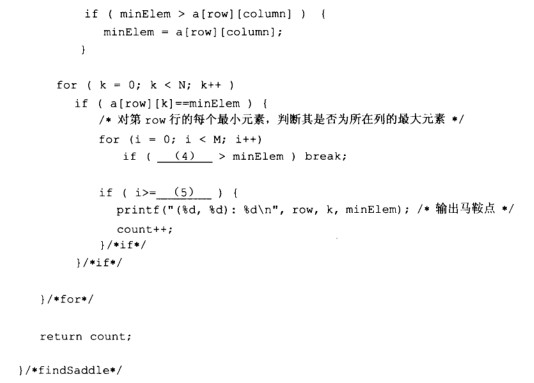
**【说明】**

如果矩阵A中的元素AW]满足条件：A[ij]是第i行中值最小的元素，且又是第j 列中值最大的元素，则称之为该矩阵的一个马鞍点。

一个矩阵可能存在多个马鞍点，也可能不存在马鞍点。下面的函数求解并输出一个 矩阵中的所有马鞍点，最后返回该矩阵中马鞍点的个数。

**【C函数】**





阅读说明和C函数，填充函数中的空缺。

**试题三**

**【说明】**

函数Insert\_key(\*root，key)的功能是将键值key插入到\*root指向根结点的二叉查找

树中（二叉查找树为空时\*root为空指针）。若给定的二叉查找树中已经包含键值为key的结点，则不进行插入操作并返回0;否则申请新结点、存入key的值并将新结点加入树中，返回1。

提示：

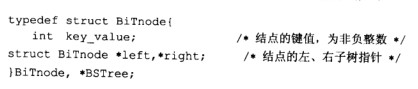
二叉查找树又称为二叉排序树，它或者是一棵空树，或者是具有如下性质的二叉树:

•若它的左子树非空，则其左子树上所有结点的键值均小于根结点的键值；

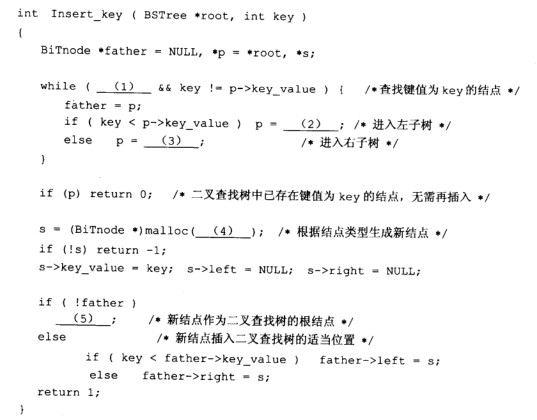
•若它的右子树非空，则其右子树上所有结点的键值均大于根结点的键值；

•左、右子树本身就是二叉查找树。

设二叉查找树采用二叉链表存储结构，链表结点类型定义如下：



**【c函数】**

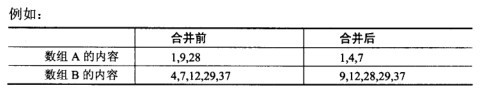


阅读说明和C函数，填充函数中的空缺。

**试题四**

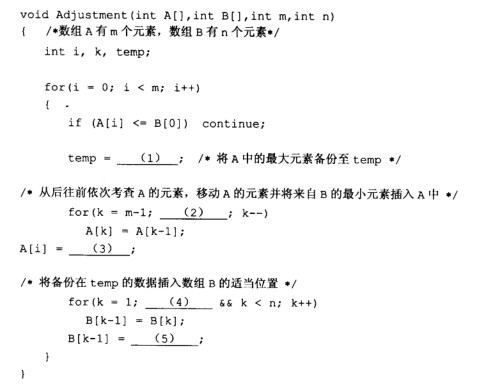
**【说明】**

已知两个整数数组A和B中分别存放了长度为m和n的两个非递减有序序列，函数Adjustment(A，B，m，n)的功能是合并两个非递减序列，并将序列的前m个整数存入A中，其余元素依序存入B中。



合并过程如下：从数组A的第一个元素开始处理。用数组B的最小元素B[0]与数组A的当前元素比较，若A的元素较小，则继续考查A的下一个元素；否则，先将A的最大元素暂存入temp，然后移动A中的元素挪出空闲单元并将B[0]插入数组A，最后将暂存在temp中的数据插入数组B的适当位置（保持B的有序性）。如此重复，直到A中所有元素都不大于B中所有元素为止。

**【C函数】**

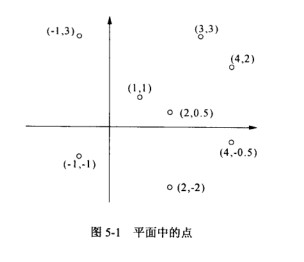


阅读以下说明和C函数，填充函数中的空缺。

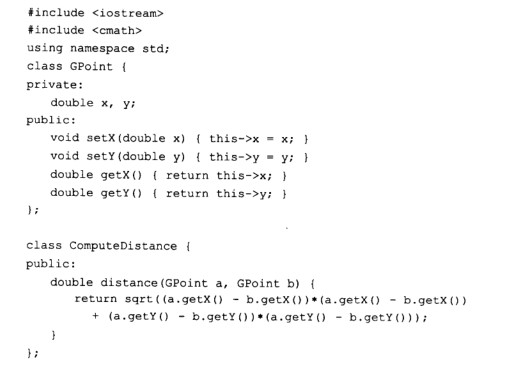
**试题五**

**【说明】**

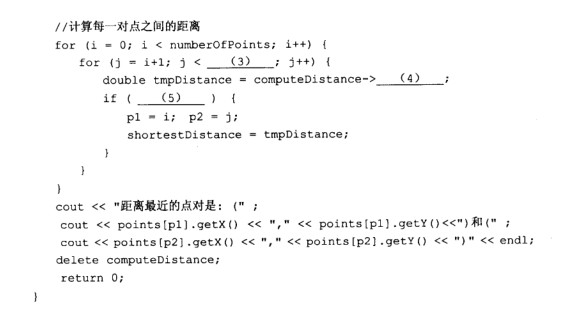
下面的程序用来计算并寻找平面坐标系中给定点中最近的点对（若存在多对，则输 出其中的一对即可）。程序运行时，先输入点的个数和一组互异的点的坐标，通过计算每 对点之间的距离，从而确定出距离最近的点对。例如，在图5-1所示的8个点中，点(1，1) 与(2, 0.5)是间距最近的点对。



**【C++代码】**





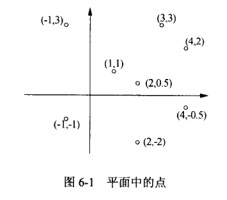


阅读说明和Java程序，填充程序中的空缺。

**试题六**

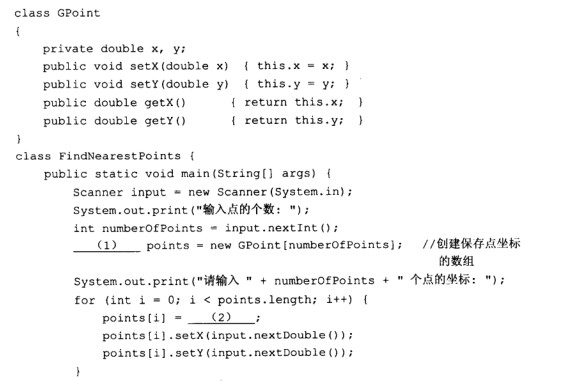
**【说明】**

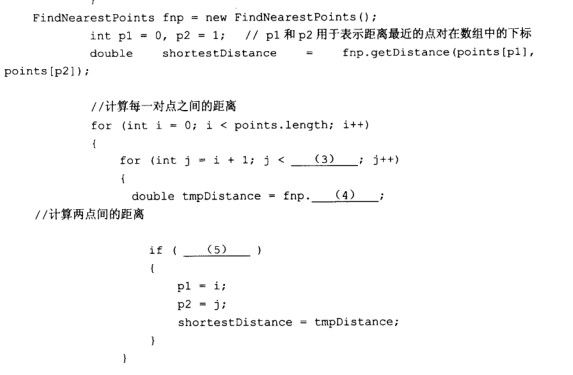
下面的程序用来计算并寻找平面坐标系中给定点中最近的点对（若存在多对，则输出其中的一对即可）。程序运行时，先输入点的个数和一组互异的点的坐标，通过计算每对点之间的距离，从而确定出距离最近的点对。例如，在图6-1所示的8个点中，点(1,1)与(2, 0.5)是间距最近的点对。

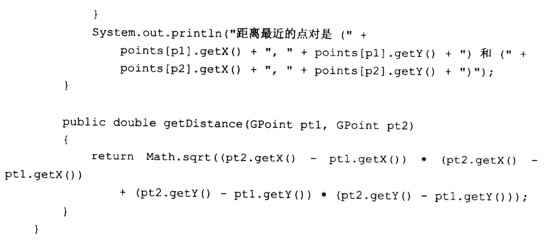


**【Java代码】**

http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/xt-c-12-x-6-1.jpg







阅读说明和Java程序，填充程序中的空缺。