**《数据结构》实验计划**

**（1）实验课程安排**

**（1.1）例题讲解：所有例题已经提前布置，包括例题在内的实验题会在实验课当天早8:00在pintia开放。要求同学在当周实验课前提前思考并写好例题代码。老师在课堂上会随机提问3名左右同学分享自己的解决方案，并根据互动情况决定平时实验分数。**

**（1.2）随堂练习：同学自行完成其余题目，包括例题在内的实验题会在实验课当天早8:00在pintia开放。实验课上有助教巡视，不会做的可以问助教。可以在课堂以外时间做题，只要在pintia关闭时间前完成提交就可以。一般一次实验给一周的时间完成。**

**（2）实验计划**

**实验课具体时间地点参照分组表（注意，每个实验组的上课时间不一定相同）**

**实验计划（周次）**

**实验一：线性表及其应用（2）**

**实验二：栈和队列的表示和实现及应用（4）**

**实验三：数组和广义表的应用，包含一题串（6）**

**实验四：二叉树的遍历（8）**

**实验五：树的应用（10）**

**实验六：图的遍历（12）**

**实验七：图的应用（13）**

**实验八：查找（14）**

**实验九：内部排序（15）**

**实验十：安排一次实验考试（16）**

**实验一 线性表及其应用**

**目标：**1、熟练掌握调试代码的方法

2、优化算法，满足时间效率的要求

3、练习通过动态分配内存来创建数组

4、练习顺序表的基本操作

5、练习链表的基本操作

**例题1：约瑟夫环（顺序结构）**

### N个人围成一圈顺序编号，从1号开始按1、2、3......顺序报数，报p者退出圈外，其余的人再从1、2、3开始报数，报p的人再退出圈外，以此类推。 请按退出顺序输出每个退出人的原序号。

### 输入格式:

### 输入只有一行，包括一个整数N(1<=N<=3000)及一个整数p(1<=p<=5000)。

### 输出格式:

### 按退出顺序输出每个退出人的原序号，数据间以一个空格分隔，但行尾无空格。

### 输入样例:

### 在这里给出一组输入。例如：

### 7 3

### 输出样例:

### 3 6 2 7 5 1 4

**例题2：链表去重**

### 给定一个带整数键值的链表 L，你需要把其中绝对值重复的键值结点删掉。即对每个键值 K，只有第一个绝对值等于 K 的结点被保留。同时，所有被删除的结点须被保存在另一个链表上。例如给定 L 为 21→-15→-15→-7→15，你需要输出去重后的链表 21→-15→-7，还有被删除的链表 -15→15。

### 输入格式：

### 输入在第一行给出 L 的第一个结点的地址和一个正整数 N（≤105，为结点总数）。一个结点的地址是非负的 5 位整数，空地址 NULL 用 -1 来表示。

### 随后 N 行，每行按以下格式描述一个结点：

### 地址 键值 下一个结点

### 其中地址是该结点的地址，键值是绝对值不超过104的整数，下一个结点是下个结点的地址。

### 输出格式：

### 首先输出去重后的链表，然后输出被删除的链表。每个结点占一行，按输入的格式输出。

### 输入样例：

### 00100 5

### 99999 -7 87654

### 23854 -15 00000

### 87654 15 -1

### 00000 -15 99999

### 00100 21 23854

### 输出样例：

### 00100 21 23854

### 23854 -15 99999

### 99999 -7 -1

### 00000 -15 87654

### 87654 15 -1

**实验二 栈和队列的表示和实现及应用**

**目标**：1、练习栈的基本操作及应用

2、练习队列的基本操作及应用

3、练习循环队列的基本操作

### 例题：后缀表达式求值

### 后缀表达式，又称逆波兰式，指的是不包含括号，运算符放在两个运算对象的后面，所有的计算按运算符出现的顺序，严格从左向右进行。

### 运用后缀表达式进行计算的具体做法：

### 建立一个操作数栈S。然后从左到右读表达式，如果读到操作数就将它压入栈S中，如果读到n元运算符(即需要参数个数为n的运算符)则取出由栈顶向下的n项操作数进行运算，再将运算的结果代替原栈顶的n项压入栈中。重复上面过程，如果后缀表达式读完且栈中只剩一个操作数，则该数就是运算结果；如果后缀表达式读完但是栈中操作数多于一个，则后缀表达式错误；如果栈中操作数只剩一个，但是后缀表达式还未读完且当前运算符为双元操作符，则后缀表达式同样错误。

### 输入格式:

### 在一行中输入一个以#号结束的非空后缀式，#不属于表达式的一部分，操作数和运算符都以空格分隔，运算数为绝对值不超过100的整数，运算符仅有+、-、\*、/ 四种。

### 输出格式:

### 输出后缀式计算结果，所有的计算都只取结果的整数部分。题目保证计算的中间和最后结果的绝对值都不超过109。

### 如果执行除法时出现分母为零的非法操作，则在一行中输出：Error: X/0，X是当时的分子。

### 如果后缀表达式中运算符多了或者少了，则在一行中输出：Expression Error: X，X是当时栈顶元素。

### 输入样例1:

### 5 -2 + 3 \* #

### 输出样例1:

### 9

### 输入样例2:

### 5 -2 2 + / #

### 输出样例2:

### Error: 5/0

### 输入样例3:

### 5 -1 3 + / - \* #

### 输出样例3:

### Expression Error: 2

**实验三 数组和广义表的应用**

**目标：**1、练习串的基本操作及应用

2、练习稀疏矩阵的压缩存储及运算

**例题：稀疏矩阵加法**

### 给定两个矩阵A和B，求其和矩阵C=A+B。

### 输入格式:

### 第一行包含两个数Row和Col，分别表示矩阵的行数和列数，A和B的维度是一致的。

### 第二行只有一个数N1​，表示接下来要输入的A中的非零元素的个数。

### 接下来是N1​行，每一行都是i j A[i,j] 这样的形式，表示的A中第i行第j列的元素A[i,j]，为了与大多数编程语言保持一致，它们都是从零开始的，也就是说下标的有效范围是[0,Row−1]×[0,Col−1]。

### 在N1​行之后，是一个数N2​，表示矩阵B中非零元素的数量，此后N2​行描述B中的非零元素，它们与此前描述A中非零元素的形式一致。

### 矩阵元素的输入均遵循行主序。这里的所有的输入均可用int类型正确表示，可以假设输入均是合法的。

### 输出格式:

### 第一行输出和矩阵C=A+B中的绝对值大于0.1的元素个数N3​，此后是N3​行，按照行主序输出其中的非零元素，依次是行、列的下标和对应的元素。

### 输入样例:

### 2 2

### 1

### 1 1 1

### 1

### 0 0 1

### 输出样例:

### 2

### 0 0 1

### 1 1 1

**实验四 二叉树的遍历**

**目标：**1、熟悉树的非线性存储结构特点

2、熟练掌握二叉树的各种遍历操作

3、利用遍历操作解决其他操作的应用问题

**例题：根据后序和中序遍历输出先序遍历**

### 本题要求根据给定的一棵二叉树的后序遍历和中序遍历结果，输出该树的先序遍历结果。

### 输入格式:

### 第一行给出正整数N(≤30)，是树中结点的个数。随后两行，每行给出N个整数，分别对应后序遍历和中序遍历结果，数字间以空格分隔。题目保证输入正确对应一棵二叉树。

### 输出格式:

### 在一行中输出Preorder: 以及该树的先序遍历结果。数字间有1个空格，行末不得有多余空格。

### 输入样例:

### 7

### 2 3 1 5 7 6 4

### 1 2 3 4 5 6 7

### 输出样例:

### Preorder: 4 1 3 2 6 5 7

**实验五 树的应用**

**目标：**1、利用树的性质解决更多应用问题

2、熟练掌握完全二叉树的性质

3、掌握哈夫曼树的创建及遍历

### 例题：笛卡尔树

**笛卡尔树是一种特殊的二叉树，其结点包含两个关键字K1和K2。首先笛卡尔树是关于K1的二叉搜索树，即结点左子树的所有K1值都比该结点的K1值小，右子树则大。其次所有结点的K2关键字满足优先队列（不妨设为最小堆）的顺序要求，即该结点的K2值比其子树中所有结点的K2值小。给定一棵二叉树，请判断该树是否笛卡尔树。**

**输入格式:**

**输入首先给出正整数N（≤1000），为树中结点的个数。随后N行，每行给出一个结点的信息，包括：结点的K1值、K2值、左孩子结点编号、右孩子结点编号。设结点从0~(N-1)顺序编号。若某结点不存在孩子结点，则该位置给出−1。**

**输出格式:**

**输出YES如果该树是一棵笛卡尔树；否则输出NO。**

**输入样例1:**

**6**

**8 27 5 1**

**9 40 -1 -1**

**10 20 0 3**

**12 21 -1 4**

**15 22 -1 -1**

**5 35 -1 -1**

**输出样例1:**

**YES**

**输入样例2:**

**6**

**8 27 5 1**

**9 40 -1 -1**

**10 20 0 3**

**12 11 -1 4**

**15 22 -1 -1**

**50 35 -1 -1**

**输出样例2:**

**NO**

**实验六 图的遍历**

**目标：**1、熟练掌握图的深度优先搜索

2、熟练掌握图的广度优先搜索

3、学会最小生成树的求解

**例题：图的遍历**

给定一个有N个顶点和E条边的无向图，请用DFS和BFS分别列出其所有的连通集。假设顶点从0到N−1编号。进行搜索时，假设我们总是从编号最小的顶点出发，按编号递增的顺序访问邻接点。

输入格式:

输入第1行给出2个整数N(0<N≤10)和E，分别是图的顶点数和边数。随后E行，每行给出一条边的两个端点。每行中的数字之间用1空格分隔。

输出格式:

按照"{ v 1 v 2 ... v k }"的格式，每行输出一个连通集。先输出DFS的结果，再输出BFS的结果。

输入样例:

8 6

0 7

0 1

2 0

4 1

2 4

3 5

输出样例:

{ 0 1 4 2 7 }

{ 3 5 }

{ 6 }

{ 0 1 2 7 4 }

{ 3 5 }

{ 6 }

**实验七 图的应用**

**目标：**1、熟练掌握图的最短路径算法

2、能用最短路径算法解决实际应用问题

3、学会利用拓扑排序解决任务调度和工期计算问题

### 例题：旅游规划（25分）

### 有了一张自驾旅游路线图，你会知道城市间的高速公路长度、以及该公路要收取的过路费。现在需要你写一个程序，帮助前来咨询的游客找一条出发地和目的地之间的最短路径。如果有若干条路径都是最短的，那么需要输出最便宜的一条路径。

### 输入格式:

### 输入说明：输入数据的第1行给出4个正整数N、M、S、D，其中N（2≤N≤500）是城市的个数，顺便假设城市的编号为0~(N−1)；M是高速公路的条数；S是出发地的城市编号；D是目的地的城市编号。随后的M行中，每行给出一条高速公路的信息，分别是：城市1、城市2、高速公路长度、收费额，中间用空格分开，数字均为整数且不超过500。输入保证解的存在。

### 输出格式:

### 在一行里输出路径的长度和收费总额，数字间以空格分隔，输出结尾不能有多余空格。

### 输入样例:

### 4 5 0 3

### 0 1 1 20

### 1 3 2 30

### 0 3 4 10

### 0 2 2 20

### 2 3 1 20

### 输出样例:

### 3 40

**实验八 查找**

**目标：**1、在有序序列中可以通过二分查找快速定位所需要的数据

2、熟练掌握二叉排序树的构造和查找方法

3、练习哈希表的构建与查找等基本操作

### 例题：QQ账号的申请与登陆 （哈希表的构建与查找）

### 实现QQ新帐户申请和老帐户登陆的简化版功能。最大挑战是：据说现在的QQ号码已经有10位数了。

### 输入格式:

### 输入首先给出一个正整数N（≤105），随后给出N行指令。每行指令的格式为：“命令符（空格）QQ号码（空格）密码”。其中命令符为“N”（代表New）时表示要新申请一个QQ号，后面是新帐户的号码和密码；命令符为“L”（代表Login）时表示是老帐户登陆，后面是登陆信息。QQ号码为一个不超过10位、但大于1000（据说QQ老总的号码是1001）的整数。密码为不小于6位、不超过16位、且不包含空格的字符串。

### 输出格式:

### 针对每条指令，给出相应的信息：

### 1）若新申请帐户成功，则输出“New: OK”； 2）若新申请的号码已经存在，则输出“ERROR: Exist”； 3）若老帐户登陆成功，则输出“Login: OK”； 4）若老帐户QQ号码不存在，则输出“ERROR: Not Exist”； 5）若老帐户密码错误，则输出“ERROR: Wrong PW”。

### 输入样例:

### 5

### L 1234567890 myQQ@qq.com

### N 1234567890 myQQ@qq.com

### N 1234567890 myQQ@qq.com

### L 1234567890 myQQ@qq

### L 1234567890 myQQ@qq.com

### 输出样例:

### ERROR: Not Exist

### New: OK

### ERROR: Exist

### ERROR: Wrong PW

### Login: OK

**实验九 内部排序**

**目标：**1、练习各种排序算法

2、灵活应用排序方法解决实际应用问题

**例题：**排序

### 给定N个（长整型范围内的）整数，要求输出从小到大排序后的结果。

### 本题旨在测试各种不同的排序算法在各种数据情况下的表现。各组测试数据特点如下：

### · 数据1：只有1个元素；

### · 数据2：11个不相同的整数，测试基本正确性；

### · 数据3：103个随机整数；

### · 数据4：104个随机整数；

### · 数据5：105个随机整数；

### · 数据6：105个顺序整数；

### · 数据7：105个逆序整数；

### · 数据8：105个基本有序的整数；

### · 数据9：105个随机正整数，每个数字不超过1000。

### 输入格式:

### · 输入第一行给出正整数N（≤105），随后一行给出N个（长整型范围内的）整数，其间以空格分隔。

### 输出格式:

### · 在一行中输出从小到大排序后的结果，数字间以1个空格分隔，行末不得有多余空格。

### 输入样例:

### · 11

### 4 981 10 -17 0 -20 29 50 8 43 -5

### 输出样例:

### · -20 -17 -5 0 4 8 10 29 43 50 981