

数据结构综合训练习题集

序号	项目名称	任务描述	设计要求	每组学生人数
1.	订票系统	<p>基本要求：</p> <p>(1) 录入：可以录入航班情况（数据可以存储在一个数据文件中，数据结构、具体数据自定）</p> <p>(2) 查询：可以查询某个航线的情况（如，输入航班号，查询起降时间，起飞抵达城市，航班票价，票价折扣，确定航班是否满仓）；</p> <p>(3) 可以输入起飞抵达城市，查询飞机航班情况；</p> <p>(4) 订票：（订票情况可以存在一个数据文件中，结构自己设定），可以订票，如果该航班已经无票，可以提供相关可选择航班；退票：可退票，退票后修改相关数据文件；客户资料有姓名，证件号，订票数量及航班情况，订单要有编号；</p> <p>(5) 修改航班信息：当航班信息改变可以修改航班数据文件。</p>	根据以上功能说明，设计航班信息，订票信息的存储结构，设计程序完成功能；	
2.	用 Huffman 编码压缩文件	准备一个文件，统计该文件中各种字符的频率，对各字符进行 Huffman 编码，将该文件翻译成 Huffman 编码文件，再将 Huffman 编码文件翻译成源文件。		
3.	商品管理系统	<p>以链表结构的有序表表示某商场家电部的库存模型，当有提货或进货时需要对该链表及时地进行维护，每个工作日结束以后，将该链表中的数据以文件形式保存，每日开始营业之前，须将文件形式保存的数据恢复成链表结构的有序表。</p> <p>链表结构的数据域 包括家电名称、品牌、单价和数量，以单价的升序体现链表的有序性。程序功能包括：初始化、创建表、插入、删除、更新数据、查询及链表数据与文件之间的转换等。</p>		
4.	排序算法效率比较	编程实现插入、希尔、快速、堆排序、归并排序算法，并计算每种算法的比较、交换次数。将待排数据从磁盘文件读入，实施排序后将数据写入另一个文件中。		
5.	管道铺设施工的最佳方案选择	<p>$N(N>10)$ 个居民之间需要铺设煤气管道。假设任意两个居民之间都可以铺设煤气管道，但代价不同。事先将任意两个居民之间铺设煤气管道的代价存入磁盘文件中。</p> <p>设计一个最佳方案使得这 N 个居民之间铺设煤气管道所需代价最少，并希望以图形方式在屏幕上输出结果。</p>		

数据结构综合训练习题集

6.	建立 Haffman 树并查询	对文件 file1.txt 中的姓名按姓氏进行统计，计算每个姓氏出现的概率，并生产 Haffman 树，用另一个文件 file2.txt 中的姓氏在 Haffman 树中查询，得出查询完成所用的时间；在 file1.txt 中查询 file2.txt 中姓氏，得出查询完成所用的时间，对两者进行对比，得出结论并写进论文。		
7.	校园导游咨询	<p>1)、功能描述：设计你的学校的校园平面图，所含景点不少于 10 个。以图中顶点表示学校各景点，存放景点名称，代号，简介等信息；以边表示路径，存放路径长度等相关信息。</p> <p>2)、为来访客人提供图中任意景点的问路查询，即查询任意两个景点之间的一条最短的简单路径。</p> <p>3)、为来访客人提供图中任意景点相关信息的查询。</p> <p>测试数据：由读者根据实际情况指定。</p> <p>实现提示：一般情况下，校园的道路是双向通行的，可设校园平面图是一个无向网，顶点和边均含有相关信息。</p>		
8.	网络检查	<p>1). 问题描述</p> <p>从文件中读入一个计算机网络以及机器间的双向连线列表，每一条连线允许两端的计算机进行直接的文件传输，其他计算机间若存在一条连通路径，也可以进行间接的文件传输。请写出程序判断：任意指定两台计算机，它们之间是否可以进行文件传输？</p> <p>2). 基本要求</p> <p>(1) 输入要求：输入若干测试数据组成。对于每一组测试，第 1 行包含一个整数 N (≤ 10000)，即网络中计算机的总台数，因而每台计算机可用 1 到 N 之间的一个正整数表示。接下来的几行输入格式为 I C1 C2 或者 C 或者 C C1C2 或者 S，其中 C1 和 C2 是两台计算机的序号，I 表示在 C1 和 C2 间输入一条连线，C 表示检查 C1 和 C2 间是否可以传输文件，S 表示该组测试结束。</p> <p>当 N 为 0 时，表示全部测试结束，不要对该数据做任何处理。</p> <p>(2) 输出要求：对每一组 C 开头的测试，检查 C1 和 C2 间是否可以传输文件，</p>		

数据结构综合训练习题集

		<p>若可以，则在一行中输出“yes”，否则输出“no”。</p> <p>当读到 S 时，检查整个网络。若网络中任意两机器间都可以传输文件，则在一行中输出“The network is connected.”，否则输出“There are k components.”，其中 k 是网络中连通集的个数。</p> <p>两组测试数据之间请输出一空行分隔。</p>		
9.	散列表的设计与实现	<p>【问题描述】 设计散列表实现电话号码查找系统。</p> <p>【基本要求】 1) 设每个记录有下列数据项：电话号码、用户名、地址； 2) 从键盘输入各记录，分别以电话号码和用户名为关键字建立散列表； 3) 采用一定的方法解决冲突； 4) 查找并显示给定电话号码的记录； 5) 查找并显示给定用户名的记录。</p> <p>【进一步完成内容】 1) 系统功能的完善； 2) 设计不同的散列函数，比较冲突率； 3) 在散列函数确定的前提下，尝试各种不同类型处理冲突的方法，考察平均查找长度的变化。</p>		
10.	二叉平衡排序树	<p>问题描述：从一棵空树开始创建，在创建过程中，保证树的有序性，同时还要针对树的平衡性做些调整。最终要把创建好的二叉排序树转换为二叉平衡排序树。</p> <p>基本要求：1). 创建（插入、调整、改组） 2). 输出</p>		
11.	构造可以使 n 个城市连接的最小生成树	<p>问题描述：给定一个地区的 n 个城市间的距离网，用 Prim 和 Kruskal 算法建立最小生成树，并计算得到的最小生成树的代价。</p> <p>基本要求： 1、城市间的距离网采用邻接矩阵表示，邻接矩阵的存储结构定义采用课本中给出的</p>		

数据结构综合训练习题集

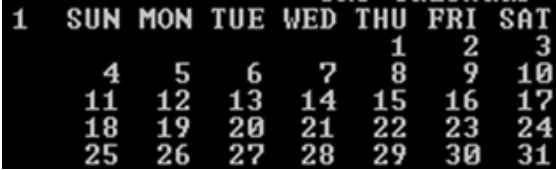
		定义，若两个城市之间不存在道路，则将相应边的权值设为自己定义的无穷大值。 要求在屏幕上显示得到的最小生成树中包括了哪些城市间的道路，并显示得到的最小生成树的代价。 2、表示城市间距离网的邻接矩阵（要求至少 6 个城市，10 条边） 3、最小生成树中包括的边及其权值，并显示得到的最小生成树的代价。		
12.	线索二叉树的应用	要求：实现线索树建立、插入、删除、恢复线索的实现。		
13.	图的遍历的实现	1) 先任意创建一个图； 2) 图的 DFS 的递归和非递归算法的实现 3) 图的 BFS 的算法实现 4) 要求用有向图和无向图分别实现 5) 要求用邻接矩阵、邻接表多种结构存储实现		
14.	哈夫曼编码/译码器	【问题描述】 设计一个利用哈夫曼算法的编码和译码系统，重复地显示并处理以下项目，直到选择退出为止。 【基本要求】 1) 将权值数据存放在数据文件(文件名为 data.txt，位于执行程序的当前目录中) 2) 分别采用动态和静态存储结构 3) 初始化：键盘输入字符集大小 n、n 个字符和 n 个权值，建立哈夫曼树； 4) 编码：利用建好的哈夫曼树生成哈夫曼编码； 5) 输出编码； 6) 设字符集及频度如下表： 字符 空格 A B C D E F G H I J K L M 频度 186 64 13 22 32 103 21 15 47 57 1 5 32 20 字符 N O P Q R S T U V W X Y Z 频度 57 63 15 1 48 51 80 23 8 18 1 16 1		

		<p>【进一步完成内容】</p> <p>1) 译码功能；</p> <p>2) 显示哈夫曼树；</p> <p>3) 界面设计的优化</p>		
15.	停车场管理	<p>设停车场是一个可停放 n 辆汽车的狭长通道，且只有一个大门可供汽车进出。汽车在停车场内按车辆到达时间的先后顺序，依次由北向南排列（大门在最南端，最先到达的第一辆车停放在车场的最北端），若车场内已停满 n 辆汽车，则后来的汽车只能在门外的便道上等候，一旦有车开走，则排在便道上的第一辆车即可开入；当停车场内某辆车要离开时，在它之后进入的车辆必须先退出车场为它让路，待该辆车开出大门外，其他车辆再按原次序进入车场，每辆停放在车场的车在它离开停车场时必须按它停留的时间长短交纳费用。试为停车场编制按上述要求进行管理的模拟程序。</p> <p>以栈模拟停车场，以队列模拟车场外的便道，按照从终端读入的输入数据序列进行模拟管理。每一组输入数据包括三个数据项：汽车“到达”或“离去”信息、汽车牌照号码以及到达或离去的时刻。对每一组输入数据进行操作后的输出信息为：若是车辆到达，则输出汽车在停车场内或便道上的停车位置；若是车辆离去，则输出汽车在停车场内停留的时间和应交纳的费用（在便道上停留的时间不收费）。钱以顺序结构实现，队列以链表结构实现。</p>		
16.	电梯运行仿真程序	<p>问题描述：编写一个程序，模拟办公大楼中全部电梯的工作过程。该仿真程序可以用来监测系统运行情况，改善大楼管理，它也可以看成是一种游戏程序。</p> <p>系统初步描述：</p> <p>（1）办公大楼有若干层（例如，10 层），每层都有电梯可到达，全楼有若干部（例如，不多于 10 部）电梯同时供使用，电梯容量为 24 人，电梯运行每上一层楼需 5 秒，在某一层停下至少需 15 秒。其运行状态可分：向上、向下、停止，当前乘客数，当前所在层数。它设有一个“按钮数组”，例如第五层的按钮按下，意味着有乘客在第 5 层到达目标层，等等。</p>		

		<p>(2) 在楼的每一层，有电梯数，有按钮表示有人等待向上或向下，由若干人在等待，有若干电梯在本层停下，等等。</p> <p>(3) 在大楼中（包括进出）的总人数不超过 500 人，每个人站在电梯前有个目标层，他有一个最大的忍受等待时间，因为他可以选择电梯或是步行走楼梯，等等。</p> <p>(4) 还有下面若干假设：在每个时间段要进大楼的人数在 0~199 之间随机取值；</p> <p>(5) 用电梯的每个人的目标层在 1~10 之间取值；一个人在进电梯或改走楼梯之前的等待时间在 180~360 秒范围内随机发生；一个人到达目标层后第二次再乘电梯中间的工作时间在 400~6600 秒间随机取值。</p>  <p>基本要求：设计出如上图所示的图形化界面，模拟电梯运行。</p>		
17.	文件目录管理系统	<p>问题描述：文件是管理用户信息和应用程序的一种工具。每个文件有唯一的文件名，可以通过文件名访问文件，同时可对文件进行生成、删除及文件名修改等操作。文件系统对若干文件进行管理时将所有的文件目录组合在一起构成一个目录文件。通过对目录文件的管理达到“按名存取”的目的，目录文件常采用的组织结构是树型目录结构。</p>		

数据结构综合训练习题集

		<p>基本要求：</p> <p>函数功能要划分好，程序要有必要的注释。</p> <p>用户通过界面菜单选择以下操作：（注意，以下操作仅需对目录进行操作，不需要实际生成文件）</p> <p>（1） 生成文件，选择路径和文件名，实现对文件的生成。</p> <p>（2） 删除文件，对指定文件进行删除操作。</p> <p>（3） 修改文件，对指定文件进行内容修改或者文件名修改。</p> <p>（4） 输出该目录结构。</p>		
18.	身份证管理程序	<p>该程序应该具有下列功能：</p> <p>（1）通过键盘可以输入身份证信息，大量信息可存放在文件中。身份证包含的信息请参看自己的身份证；</p> <p>（2）给定身份证号码，显示其身份证信息；</p> <p>（3）给定省份的编号，显示该省的人数；</p> <p>（4）给定某区的编号，显示该区的人数；</p> <p>（5）给定身份证号码，可以修改该身份证信息；</p> <p>（6）给定身份证号码，可以删除该身份证信息；</p>	提供一些统计各类信息的功能。例如男女的人数、比例；以及哪年、哪月、哪日出生的人数等。界面要合理。	
19.	集合操作	<p>用单链表模拟有序集合，实现集合的加入一个元素、删除一个元素、集合的交、并、差运算。</p> <p>（1）用单链表存放集合中的元素，链表中的元素按大小存放；</p> <p>（2）实现集合加入一个元素删除一个元素的元素操作；</p> <p>（3）实现集合的交、并、差集合操作；</p>	完成任务描述中的各种功能，自己可以适当增加必要的功能。	
20.	万年历查询程序。	<p>实现万年历程序</p> <p>功能要求：</p> <p>（1）提供菜单方式选择，假定输入的年份在 1940-2040 年之间。</p> <p>（2）输入一个年份，输出是在屏幕上显示该年的日历。</p> <p>（3）输入年月，输出该月的日历。如：</p>		

		 <p>(4) 输入年份、月份、日期，计算得到的是这一天据今天有多少天，星期几； (5) 输入公历的年月日，输出农历年月日。 (6) 输入农历节气，输出当年农历的年月日及公历年月日。可以假定只涉及年份是1940年到2040年。</p>		
21.	数字游戏的设计	<p>实现一个简单的猜数字游戏</p> <p>(1) 一个四位数，各位上的数字不重复，从1到9。 (2) 按以下提示猜出这个四位数。 (3) 每次猜测输入的数据给出类似的提示*A*B。 (4) 其中A前的*代表你本次猜对了多少个数字。 (5) 其中B前的*代表你本次猜对的数字并且位置正确的个数。 (6) 给定猜测次数，如果超过次数未猜中，游戏失败</p>		
22.	个人电话号码查询系统	<p>实现简单的个人电话号码查询系统，根据用户输入的信息（如姓名，身份证号，电话号码、邮件地址等）进行快速查询。</p> <p>基本要求：</p> <p>(1) 插入：实现将用户的信息插入到系统中； (2) 删除：删除某个用户的信息； (3) 修改：修改某个用户的信息； (4) 查询：根据姓名、身份证号等查询用户信息（包括简单条件查询，组合条件查询、模糊查询等）； (5) 排序：对于用户信息进行排序，提高查询速度；</p>		1

数据结构综合训练习题集

		<p>(6) 输出：输出用户信息。</p> <p>提示：</p> <p>(1) 在内存中，设计数据结构存储电话号码的信息；在外存中，利用文件的形式来保存电话号码信息，系统运行时，将电话号码信息从文件调入内存来进行插入、查找等操作。</p> <p>(2) 如果数据的插入删除频繁，可以考虑采取二叉排序树组织电话号码信息（也可采用较复杂的平衡二叉树），可以提高查找和维护的时间性能。</p> <p>(3) 选择不同的排序和查找算法，尽可能提高查找和维护性能。</p>		
23.	单源最短路径求解	<p>给定一个带权有向图 $G=(V,E)$，其中每条边的权是一个非负实数。另外，还给定 V 中的一个顶点，成为源。现在计算从源到其他各顶点的最短路径。路径的长度是指路上各边权值之和。</p>		1
24.	找数字对	<p>输入 N ($2 \leq N \leq 100$) 个数字(在 0 与 9 之间), 然后统计出这组数种相邻两数字组成的链环数字对出现的次数。例如：</p> <p>输入：$N=20$ {表示要输入数的数目}</p> <p>0 1 5 9 8 7 2 2 2 3 2 7 8 7 8 7 9 6 5 9</p> <p>输出 $(7,8)=2$ $(8,7)=3$ {指 $(7,8)$、$(8,7)$ 数字对出现次数分别为 2 次、3 次}</p>		
25.	n 元多项式乘法	<p>(1) 界面友好，函数功能要划分好</p> <p>(2) 总体设计应画一流程图</p> <p>(3) 程序要加必要的注释</p> <p>(4) 要提供程序测试方案</p> <p>(5) 程序一定要经得起测试，宁可功能少一些，也要能运行起来，不能运行的程序是没有价值的。</p>		
26.	图书登记管理程序	<p>该程序应该具有下列功能：</p> <p>(1) 通过键盘输入某本图书的信息；</p> <p>(2) 给定图书编号，显示该本图书的信息；</p> <p>(3) 给定作者姓名，显示所有该作者编写的图书信息；</p>		

数据结构综合训练习题集

		<p>(4) 给定出版社，显示该出版社的所有图书信息；</p> <p>(5) 给定图书编号，删除该本图书的信息；</p> <p>(6) 提供一些统计各类信息的功能。</p>		
27.	超长正整数的乘法	<p>设计一个算法来完成两个超长正整数的乘法。</p> <p>算法提示：</p> <p>首先要设计一种数据结构来表示一个超长的正整数，然后才能够设计算法。</p>		
28.	稀疏矩阵的压缩与还原	<p>一个矩阵含有非零元素比较少，而零元素相对较多，这样的矩阵称为稀疏矩阵，对稀疏矩阵的存储我们不用完全用二维数组来存储，可以用一个三元组，即任意一个稀疏矩阵可以用一个只有三列的二维数组来存放，</p> <p>要求把给定的稀疏矩阵用为三元组表示；同时把三元组转换为稀疏矩阵形式。</p>		
29.	文章编辑	<p>输入一页文字，程序可以统计出文字、数字、空格的个数。静态存储一页文章，每行最多不超过 80 个字符。</p> <p>要求：</p> <p>(1) 分别统计出其中英文字母数和空格数及整篇文章总字数；</p> <p>(2) 统计某一字符串在文章中出现的次数，并输出该次数；</p> <p>(3) 删除某一子串，并将后面的字符前移。</p> <p>存储结构使用线性表，分别用几个子函数实现相应的功能；</p> <p>输入数据的形式和范围：可以输入大写、小写的英文字母、任何数字及标点符号。</p> <p>输出形式：</p> <p>(1) 分行输出用户输入的各行字符；</p> <p>(2) 分 4 行输出“全部字母数”、“数字个数”、“空格个数”、“文章总字数”</p> <p>(3) 输出删除某一字符串后的文章；</p>		
30.	链表的综合算法设计	<p>设有一职工文件，其结构为：职工号(no)、姓名(name)、部门号(depno)、工资数(salary)、职工号指针(pno)、部门号指针(pdepno)、工资数指针(psalary)，设计一程序，从一文件中读取记录到单链表中，并完成如下功能：</p> <p>(1) 输入：添加一个职工记录；(2) 输出：输出全部职工记录；</p> <p>(3) 按 no 排序：通过 pno 指针将职工记录按 no 从小到大链接起来；</p> <p>(4) 按 no 输出：沿 pno 链输出全部职工记录；</p>		

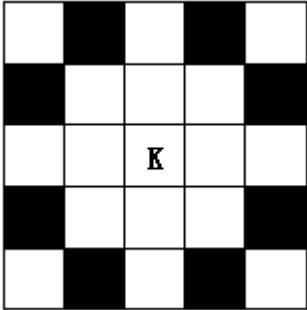
数据结构综合训练习题集

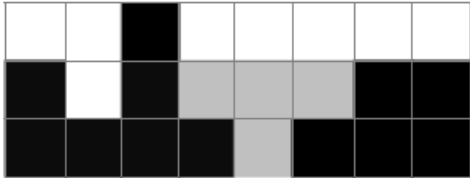



		<p>(5) 按 depno 排序：通过 pdepno 指针将职工记录按 depno 从小到大链接起来；</p> <p>(6) 按 depno 输出：沿 pdepno 链输出全部职工记录；</p> <p>(7) 按 salary 排序：通过 psalary 指针将职工记录按 salary 从小到大链接起来；</p> <p>(8) 按 salary 输出：沿 psalary 链输出全部职工记录；</p> <p>(9) 全清：删除职工文件中的全部记录；</p> <p>(10) 存贮退出：将单链表中的全部结点存贮到职工文件中，然后退出程序运行。</p>		
31.	关键路径问题	<p>基本要求：</p> <p>(1) 对一个描述工程的 AOE 网，建立其存储结构；（注：数据的输入可以是键盘输入或文件输入两种方式）</p> <p>(2) 判断该 AOE 网是否能够顺利进行。</p> <p>(3) 若该工程能顺利进行，输出完成整项工程至少需要多少时间，以及每一个关键活动所依附的两个顶点、最早发生时间、最迟发生时间。（注：结果的输出可以是屏幕输出和文件输出两种方式）</p>		
32.	邮路问题	<p>问题描述：一个邮递员从邮局选好邮件去投递，然后回到邮局。当然他必须经过他所管辖的每条街至少一次。请为他设计一条投递路线，使其所行的路程尽可能地短。</p> <p>基本要求：</p> <p>(1) 设计邮递员的辖区，并将其抽象成图结构进行表示，建立其存储结构。（注：数据输入可以是键盘输入和文件输入两种方式）</p> <p>(2) 按照输入邮局所在位置，为邮递员设计一条最佳投递路线，要能考虑到辖区一般情况。</p> <p>(3) 界面要求：有合理的提示和人机交互。</p>		
33.	机器人布线	<p>布线区域分成 $m \times n$ 的方格阵列。要求确定连接方格 s 到方格 d 的最短布线方案。布线的时候，电路只能沿着直线或者直角布线，有障碍的方格做了封锁标记 (X)，其他线路不允许穿过被封锁的方格。</p>	<p>主要功能：</p> <p>(1) 从文件中读出题目的输入；</p> <p>(2) 向屏幕上打印出题</p>	1

		<div><table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>S</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td></td></tr><tr><td></td><td></td><td>X</td><td>X</td><td>X</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td>X</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>D</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table></div> <p>(1)用文件保存布线区域，用 1、0 分别表示某个格子是否有障碍；S,D 表示起点和终点；</p> <p>(2)给出最短的布线路径长度；</p> <p>(3)用文件保存布线路径，用*表示布线的方格；</p>										S		X	X	X	X								X				X	X	X		X				X	X	X						X	X	X		X	X								D									目的计算结果；	
	S		X	X	X	X																																																														
						X																																																														
		X	X	X		X																																																														
		X	X	X																																																																
		X	X	X		X	X																																																													
							D																																																													
34.	字符串距离	<p>开发计算两个字符串间的编辑距离，LCS 距离和 N-gram 距离的函数。</p> <p>（1）编辑距离</p> <p>字符串 a 和 b 的编辑距离 $ED(i, j)$ 表示把字符串 a 转换成 b 所需要的最少操作次数，这些操作可以是：插入一个字符，删除一个字符，替换一个字符。显然，$ED(i, j)$ 越小，a，b 越相似。$ED(i, j)$ 可按下列公式计算：</p>		1																																																																

		$ED(0,0) = 0$ $ED(0,i) = ED(i,0) = i$ $ED(i,j) = \begin{cases} ED(i-1,j-1) & a_i = b_j \\ 1 + \min(ED(i-1,j), ED(i,j-1), ED(i-1,j-1)) & a_i \neq b_j \end{cases}$ <p>(2) LCS 相似度</p> <p>字符串 a 和 b 的 LCS (Longest Common Subsequence) 相似度是 a 和 b 间的最大相同子串的长度。显然 LCS(i, j) 越大, a, b 越相似。a, b 的 LCS 相似度定义如下:</p> $LCS(i,j) = \begin{cases} 0 & i = 0 \text{ 或 } j = 0 \\ 1 + LCS(i-1,j-1) & a_i = b_j \\ \max(LCS(i-1,j), LCS(i,j-1)) & a_i \neq b_j \end{cases}$ <p>(3) N-gram 相似度</p> <p>设 Ngram(a) 是字符串 a 中长度为 N 的子串的集合。两个字符串 a, b 的 N-gram 相似度 NG(a, b) 定义如下:</p> $NG(a,b) = \frac{ Ngram(a) \cap Ngram(b) }{ Ngram(a) \cup Ngram(b) }$ <p>NG(a, b) 越大, 字符串 a, b 越相似。</p>		
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

35.	推箱子	<p>推箱子是一个很经典的游戏. 今天我们来玩一个简单版本. 在一个 5×5 的房间里有一个箱子和一个搬运工, 搬运工的工作就是把箱子推到指定的位置, 注意, 搬运工只能推箱子而不能拉箱子, 因此如果箱子被推到一个角上那么箱子就不能再被移动了, 如果箱子被推到一面墙上, 那么箱子只能沿着墙移动. 同时, 房间里头还有若干障碍物 (用阴影部分表示)。</p> <p>现在给定房间的结构, 箱子的位置, 搬运工的位置和箱子要被推去的位置, 请你计算出搬运工至少要推动箱子多少格。</p> <p>(1) 从文件中读出房间布局; (2) 计算出推箱子的最小格数;</p>	<p>主要功能:</p> <p>(1) 从文件中读出题目的输入; (2) 向屏幕上打印出题目的计算结果;</p>	1
36.	跳马	<p>在国际象棋中, 马的走法与中车象棋类似, 即俗话说的“马走日”, 下图所示即国际象棋中马 (K) 在一步能到达的格子 (其中黑色的格子是能到达的位置)。</p>	<p>主要功能:</p> <p>(1) 从文件中读出题目的输入; (2) 向屏幕上打印出题目的计算结果;</p>	1

		 <p>现有一 200*200 大小的国际象棋棋盘，棋盘上仅有一个马，给定马的当前位置 (S) 和目标位置 (T)，求出马最少需要多少跳才能从当前位置到达目标位置。</p> <p>(1) 输入：每一行有四个以空格分隔的整数，分别表示马当前位置及目标位置的横、纵坐标 C (x, y) 和 G (x, y)。坐标由 1 开始。</p> <p>(2) 输出：对于每个测例，在单独的一行内输出一个整数，即马从当前位置跳到目标位置最少的跳数。</p> <p>(3) 输入样例： 1 1 2 1 输出样例： 3 输入样例： 1 5 5 1 输出样例： 4</p>		
37.	俄罗斯方块	<p>龙哥小时候最爱的游戏就是俄罗斯方块了,当年他可是个高手,每次游戏他都会选择最快的速度,以至于根本来不及将方块转向而仅仅能够进行左右移动. 为了能够坚持更久, 必须尽可能地使“落下来方块”与“底下已有方块”上表面完全贴合. 在熟悉掌握程序设计后龙哥想要用程序来模拟小时候玩俄罗斯方块的过程, 下面请你来帮龙哥参谋一下吧:-)</p>	<p>主要功能:</p> <p>(1) 从文件中读出题目的输入;</p> <p>(2) 向屏幕上打印出题目的计算结果;</p>	1

		<div data-bbox="591 268 1131 446">  <p>YES</p> </div> <div data-bbox="591 555 1120 734">  <p>NO</p> </div> <div data-bbox="591 786 1064 949"> <p>注:  表示"落下来方块"</p> <p> 表示"底下已有方块"</p> <p>根据输入的矩阵表示出方块的形状</p> </div> <div data-bbox="560 997 1590 1326"> <p>(1) 输入包括两个部分:</p> <ol style="list-style-type: none"> 1、落下来方块的矩阵(第一行两个小于 5 的整数 a、b 由空格隔开,从下一行开始是一个 a 行 b 列的矩阵,1 表示方块,0 表示空) 2、底下已有方块的矩阵(第一行两个小于 10 的整数 c、d 由空格隔开,从下一行开始是一个 c 行 d 列的矩阵,1 表示方块,0 表示空.输入底下已有方块矩阵时需确保不存在朝下的表面) <p>(2) 输出:</p> <p>根据"落下来方块"和"底下已有方块"的形状,若"落下来方块"的下表面与"底下已有"</p> </div>		
--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

数据结构综合练习题集

		<p>方块”的上表面可能完全贴合则输出一行 “YES” 否则输出一行 “NO”</p> <p>Sample Input</p> <p>2 3</p> <p>111</p> <p>010</p> <p>3 8</p> <p>00100000</p> <p>10100011</p> <p>11110111</p> <p>3 2</p> <p>11</p> <p>10</p> <p>10</p> <p>2 8</p> <p>11001110</p> <p>11011111</p> <p>Sample Output</p> <p>YES</p> <p>NO</p>		
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

数据结构综合训练习题集

38.	棋盘问题	<p>(OJ1255)在一个给定形状的棋盘（形状可能是不规则的）上面摆放棋子，棋子没有区别。要求摆放时任意的两个棋子不能放在棋盘中的同一行或者同一列，请编程求解对于给定形状和大小的棋盘，摆放 k 个棋子的所有可行的摆放方案 C。</p> <p>输入：每组数据的第一行是两个正整数，n k，用一个空格隔开，表示了将在一个 n*n 的矩阵内描述棋盘，以及摆放棋子的数目。 $n \leq 8$, $k \leq n$ 随后的 n 行描述了棋盘的形状：每行有 n 个字符，其中 # 表示棋盘区域，. 表示空白区域（数据保证不出现多余的空白行或者空白列）。</p> <p>输出：对于每一组数据，给出一行输出，输出摆放的方案数目 C （数据保证 $C < 2^{31}$）。</p> <p>输入样例：</p> <pre>2 1 #. .#</pre> <p>4 4</p> <pre>...# ..#. .#.. #...</pre> <p>输出样例：</p> <pre>2 1</pre>	<p>主要功能：</p> <p>(1) 从文件中读出题目的输入；</p> <p>(2) 向屏幕上打印出题目的计算结果；</p>	1
-----	------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------	---

数据结构综合训练习题集

39.	算 24	<p>每局4个整数，运用四则运算（可以加括号），但为了降低程序设计的难度，除法的结果只保留整数部分，即$5/2=2$。给出计算出24的方法。</p> <p>输入：本题包含多个测例。数据的第一行有一个整数 N（0从第二行开始的 N 行，各包含4个不大于15的以空格分隔的非零整数。</p> <p>输出：对于每个测例，如果可以计算得到24，则输出 “Yes”，否则输出 “No”。每个输出占一行。</p> <p>输入样例：</p> <pre>2 2 2 3 3 2 4 9 10</pre> <p>输出样例：</p> <pre>Yes Yes</pre>	<p>主要功能：</p> <p>(1) 从文件中读出题目的输入；</p> <p>(2) 向屏幕上打印出题目的计算结果；</p>	1
40.	农场灌溉问题	<p>(0J1144) 一农场由图所示的十一种小方块组成，蓝色线条为灌溉渠。若相邻两块</p> <p>灌溉渠相连则只需一口水井灌溉。</p>	<p>主要功能：</p> <p>(1) 从文件中读出题目的输入；</p> <p>(2) 向屏幕上打印出题目的计算结果；</p>	1

		<div data-bbox="577 252 1590 813"> </div> <p>输入：给出若干由字母表示的最大不超过50×50具体由(m, n)表示，的农场图</p> <p>输出：编程求出最小需要打的井数。每个测例的输出占一行。当$M=N=-1$时结束程序。</p> <p>输入样例： 2 2 DK HF 3 3 ADC FJK IHE -1 -1</p> <p>输出样例： 2 3</p>		
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

数据结构综合训练习题集

41.	城市之间	<p>(OJ1129)Vela 在玩游戏时遇到麻烦了，需要你帮忙：这个游戏有 M 个城市，有的城市有传送点，可以直接到达另外一些城市，如 a[2][3]表示可以从城市2直接到达3。无法直接到达的城市，有的可以通过某些城市中转到达。Vela 在城市 X，她想知道是否可以到达某城市 Z。</p> <p>输入：此题第一行输入 N 表示城镇数目 (N 为小于10的正整数)；从第二行开始输入一个 N*N 的矩阵，若 a[m][k]=1 (0<=m, k<n), 就可以从城市 m 直接到达 k，否则不可以直接到达。第 N+2行输入两个数字 P, Q。</p> <p>输出：如果可以从 P 到达 Q，输出1，否则输出0。</p> <p>输入样例：</p> <pre>3 1 1 0 0 1 1 0 0 1 0 2</pre> <p>输出样例：</p> <pre>1</pre> <p>提示：0不能直接到达2，但可以先到达1，再从1到达2。</p>	<p>主要功能：</p> <p>(1) 从文件中读出题目的输入；</p> <p>(2) 向屏幕上打印出题目的计算结果；</p>	1
42.	本班同学通讯录设计	<p>要求：小巧实用，具有添加，查询和删除功能。姓、名、英文名、QQ 号、电子邮箱、籍贯、电话号码组成，姓名可以由字符和数字混合编码。电话号码可由字符和数字组成。实现功能为：</p> <p>系统以菜单方式工作、信息录入功能、信息浏览功能。</p> <p>输入个人关键字信息（电话/籍贯/QQ 号/邮箱等，）能实现查询功能、信息修改功能、系统退出功能</p>		

数据结构综合训练习题集

43.	最小生成树问题 (<p>(1) 根据普里姆算法思想，画出构造该无向带权图最小生成树的过程。</p> <p>(2) 根据克鲁斯卡尔算法思想，画出构造该无向带权图最小生成树的过程。</p>		
44.	排班系统	<p>学校实验楼有 7 名保安人员：钱、赵、孙、李、周、吴、陈。由于工作需要轮休制度，一星期中每人休息一天。预先让每一个人选择自己认为合适的休息日。请编制程序，打印轮休的所有可能方案。当然使每个人都满意，例如每人选择的休息日如下：</p> <p>钱：星期一、星期六 赵：星期二、星期四 孙：星期三、星期日 李：星期五 周：星期一、星期四、星期六 吴：星期二、星期五 陈：星期三、星期六、星期日</p> <p>运行结果：</p> <p>Solution: 1</p> <p>赵 钱 孙 李 周 吴 陈</p> <p>=====</p> <p>星期四 星期一 星期三 星期五 星期六 星期二 星期日</p> <p>Solution: 2</p> <p>赵 钱 孙 李 周 吴 陈</p> <p>=====</p> <p>星期四 星期一 星期日 星期五 星期六 星期二 星期三</p> <p>Solution: 3</p> <p>赵 钱 孙 李 周 吴 陈</p>		

数据结构综合训练习题集

		<pre> ===== 星期四 星期六 星期三 星期五 星期一 星期二 星期日 Solution: 4 赵 钱 孙 李 周 吴 陈 ===== 星期四 星期六 星期日 星期五 星期一 星期二 星期三 </pre>		
45.	C 语言关键字中英翻译机。	要求输入中文的名词和关键字，可以将其翻译成英语，如输入“基本整形”+回车，得到 int；输入英文的单词 int，则可以翻译成中文“基本整形”。可模拟文曲星来实现。可多次查询，输入 bye 时退出。		
46.	汉诺塔移动	输入盘子数（2 个以上有效），移动速度，开始演示汉诺塔移动的步骤，要求：盘子，A，B，C 柱需要自己绘制，初始时盘子在 A 柱上通过 B 柱最终移动到 C 柱上，显示出盘子在几个柱之间的移动过程。		