第一部分 线性表

第 1.1 题：长整型运算

【问题描述】

编制一个实现任意长的整型进行加法运算的演示程序。

【基本要求】

（1）输入和输出形式：按中国对于长整数的表示习惯，每四位一组，组间用逗号隔开；

（2）相加过程中不能破坏两个操作数链表；不能给长整数规定上限。

【测试数据】

（1）0;0;应输出 0

（2）-2345,6789;-7654,3211; 应输出-1,0000,0000

（3）-9999,9999;1,0000,0000,0000; 应输出 9999,0000,0001

（4）1,0001,0001;-1,0001,0001; 应输出 0

（5）1,0001,0001;-1,0001,0000; 应输出 1

（6）-9999,9999,9999;-9999,9999,9999; 应输出-1,9999,9999,9998

（7）1,0000,9999,9999;-1,0001,0001; 应输出 0

【实现提示】

（1）可以考虑用链表实现，每个结点包含一个 4 位整数，即不超过 9999 的非负整数，整个链表表示为万进制数。

（2）可以利用头结点的数据域代表长整数的符号。

【选作题目】

（1）实现长整数的四则运算。

（2）实现长整数的乘方和阶乘运算。

第 1.2 题：一元稀疏多项式计算器

【问题描述】

设计一个一元稀疏多项式简单计算器。

【基本要求】

（1）输入并建立多项式。输入格式：

n

c_1 e_1

c_2 e_2

…

c_n e_n

其中 n 是多项式的项数，c_i 和 e_i 分别是第 i 项的系数和指数，c_i 和 e_i 之间用空格分开。

（2）多项式输出按照指数降序排列，输出形式为类数学表达式。例如：多项式

$-3x^8+6x^3-18$ 的输出形式为-3x^8+6x^3-18；$x^{15}-8x^7-14$ 的输出形式为 x^15-8x^7-14；注意：系数为 1 的非零次项的输出形式中略去系数 1，如 $1x^8$ 的输出形式为 x^8；$-1x^8$ 的输出形式为-x^8；指数为 1 的项略去指数 1，如 3x。

（3）多项式 a 和 b 相加，建立多项式 a+b。

（4）多项式 a 和 b 相加，建立多项式 a+b。

【测试数据】

（1）$(2x+5x^8-3.1x^{11})+(7-5x^8+11x^9) = (-3.1x^{11}+11x^9+2x+7)$

（2）$(6x^{-3}-x+4.4x^2-1.2x^9)-(-6x^{-3}+5.4x^2-x^2+7.8x^{15})=(-7.8x^{15}-1.2x^9+12x^{-3}-x)$

（3）$(1+x+x^2+x^3+x^4+x^5)+(-x^3-x^4) = (1+x+x^2+x^5)$

（4）$(x+x^3)+(-x-x^3) = 0$

（5）$(x+x^{100})+(x^{100}+x^{200}) = (x+2x^{100}+x^{200})$

（6）$(x+x^2+x^3)+(0) = (x+x^2+x^3)$

（7）互换上述测试数据中的前后两个多项式

【实现提示】

用带头结点的单链表存储多项式。

【选作题目】

（1）求多项式 a 的导函数。

（2）多项式 a 和 b 相乘，建立多项式 ab。

第 1.3 题：停车场管理

【问题描述】

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度交纳费用。试为停车场编制按上述要求进行管理的模拟程序。

【基本要求】

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“离开”信息、汽车牌照号码以及到达或离去的时刻。对每一个数据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或便道上的停车位置；若是车辆离开，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）。栈以顺序结构实现，队列以链表结构实现。

【测试数据】

设 n=2，输入数据为：('A',1,5)，('A',2,10)，('D',1,15)，('A',3,20)，('A',4,25)，('A',5,30)，('D',2,35)，('D',4,40)，('E',0,0)。其中：'A'表示到达（Arrival）；'D'表示离去（Departure）；'E'表示输入结束（End）。

【实现提示】

需要另外设一个栈，临时停放为给要离开的汽车让路而从停车场退出的汽车，也用顺序存储结构实现。输入数据按照到达或离开的时刻有序。栈中每一个元素表示一辆汽车，包括两个数据项：汽车的拍照号码和进入停车场的时刻。

【选作题目】

（1）两个栈共享空间，思考应开辟数组空间是多少？

（2）汽车可以直接从便道上开走，此时排在它前面的汽车要先开走让路，然后再依次排到队尾。

第 1.4 题：马踏棋盘

【问题描述】

设计一个国际象棋的马踏遍棋盘的演示程序。

【基本要求】

将马随机放在国际象棋的8×8棋盘board[8][8]的某个方格中，马按走棋规则进行移动。要求每个方格只进入一次，走遍棋盘上全部 64 个方格。编制非递归程序，求出马的行走路线，并按求出的行走路线，将数组 1，2，……，64 依次填入一个 8×8 的方阵并输出之。

【测试数据】

输入马的初始位置（i,j），0≥i,j≤7。

【实现提示】

下图显示了马位于(2,3)时，8 个可能的移动位置。

	0	1	2	3	4	5	6	7
0			8		1			
1		7				2		
2				H				
3		6				3		
4			5		4			
5								
6								
7								

一般来说，当马位于位置(i,j)时，可以走到下列 8 个位置之一：

(i-2,j+1),(i-1,j+2),(i+1,j+2),(i+2,j+1),(i+2,j-1),(j+1,j-2),(i-1,j-2),(i-2,j-1)

但是，如果(i,j)靠近棋盘的边缘，上述有些位置可能超过棋盘范围，成为不允许的位置。8 个可能位置可以用两个一维数组 Htry1[8]和 Htry2[8]来表示：

	0	1	2	3	4	5	6	7
Htry1	-2	-1	1	2	2	1	-1	-2
	0	1	2	3	4	5	6	7
Htry2	1	2	2	1	-1	-2	-2	-1

位于(i,j)的马可以走到的新位置是在棋盘范围内的(i+Htry1[h],j+Htry2[h])，其中 h=0,1,2,…,7。

每次在多个可走位置中选择其中一个进行试探，其余未曾试探的可走位置必须用适当结构妥善管理，以备试探失败时的“回溯”（晦棋）使用。

【选作题目】

（1）求出从某一个起点出发的全部行走路径。

（2）探讨每次选择位置的“最佳策略”，以减少回溯的次数。

第 1.5 题：银行业务模拟

【问题描述】

客户业务分为两种：第一种是申请从银行得到一笔资金，即借款或取款；第二种是想银行投入一笔资金，即存款或还款。银行有两个服务窗口，相应地有两个队列。客户到达银行后先排第一个队。处理每个客户业务时，如果属于第一种，且申请额超过银行现存资金总额而得不到满足，则立刻排入第二个队等候；否则业务处理完后立刻离开银行。每接待完一个第二种业务的客户，则顺序检查和处理（如果可能）第二队列中的客户，对能满足的申请者予以满足，不能满足者重新排到第二个队列的队尾。注意，在此检查过程中，一旦银行资金总额少于或等于刚才第一个队列中最后一个客户（第二种业务）被接待之前的数额，或者本次已将第二个队列检查或处理了一遍，就停止检查（因为此时已不可能还有能满足者）转而继续接待第一个队列的客户。任何时刻都只开一个窗口。假设检查不需要时间。营业时间结束时所有客户立即离开银行。

写一个上述银行业务的时间驱动模拟系统，通过模拟方法求出客户在银行内停留的平均时间。

【基本要求】

利用动态存储结构实现模拟。

【测试数据】

一天营业开始时银行拥有的存款未 10000（元），营业时间为 600（分钟）。其他模拟参考量自定，注意测定两种极端的情况：一是两个到达事件之间的间隔时间很短，而客户的交易时间很长；另一个恰好相反，设置两个到达事件的间隔时间很长，而客户的交易时间很短。

【实现提示】

事件有两类：到达银行和离开银行。初始时银行现存资金总额为 total。开始营业后的第一个事件是客户到达，营业时间从 0 到 closetime。到达事件发生时随机地设置此客户的交易时间和距下一个到达事件之间的时间间隔。每个客户要办理的款额也是随机确定的，用负值和正值分别表示第一类和第二类业务。变量 total,closetime 以及上述两个随机量的上下界均交互地从终端读入，作为模拟参数。

两个队列和一个时间表均要用动态存储结构实现。注意弄清应该在什么条件下设置离开事件，以及第二个队列用怎样的存储结构实现时可以获得较高的效率。注意：事件表是按时间顺序有序的。

第 1.6 题：程序分析

【问题描述】

读入一个 C++程序，统计程序中代码、注释和空行的行数以及函数的个数和平均行数，并利用统计信息分析评价该程序的风格。

【基本要求】

（1）把 C++程序文件按字符顺序读入。

（2）边读入程序，边识别统计代码行、注释行和空行，同时还要识别函数的开始和结束，以便统计其个数和平均行数。

（3）程序的风格评价分为代码、注释和空行三个方面。每个方面分别为 A,B,C 和 D 四个等级。等级的划分标准为：

	A 级	B 级	C 级	D 级

代码（函数平均长度）	10~15 行	8~9 或 16~20 行	5~7 或 21~24 行	<5 或>24 行
注释（占总行数比率）	15~25%	10~14 或 26~30%	5~9 或 31~35%	<5%或>35%
空行（占总行数比率）	15~25%	10~14 或 26~30%	5~9 或 31~35%	<5%或>35%

以下是对程序文件 example.cpp 分析的输出结果示例：

The results of analyzing program file “example.cpp”:

Lines of code: 180

Lines of comments: 63

Blink lines: 52

Code Comments Space

61% 21% 18%

The program includes 9 functions.

The average length of a section of code is 12.9 lines.

Grade A：Excellent routine size style.

Grade A：Excellent comments style.

Grade A：Excellent white space style.

【测试数据】

先对较小的程序进行分析。当你的程序能正确运行时，对你的程序本身进行分析。

【实现提示】

为了实现的方便，可作以下约定：

（1）头两个字符是’//’的行称为注释行。除了空行和注释行外，其余均为代码行（包括类型定义、变量定义和函数头）

（2）每个函数代码行数（除去空行和注释行）称为该函数的长度。

（3）每行最多只有一个“{”、“}”、“switch”、“struct”（便于识别函数的结束行）。

【选作题目】

（1）报告函数的平均长度。

（2）找出最长函数及其在程序中的位置。

第二部分 树形结构

第 2.1 题：哈夫曼编/译码器

【问题描述】

利用哈夫曼编码进行通信可以大大提高信道利用率，缩短信息传输时间，降低传输成本。但是，这要求在发送端通过一个编码系统对待传数据预先编码，在接收端将传来的数据进行译码（复原）。对于双工通信（即可以双向传输信息的信道），每端都需要一个完整的编/译码系统。试为这样的信息收发站写一个哈夫曼码的编/译码系统。

【基本要求】

一个完整的系统应具有以下功能：

（1）I：初始化（Initialization）。从终端读入字符集大小 n，以及 n 个字符和 n 个权值，建立哈夫曼树，并将它存于文件 hfmTree 中。

（2）E：编码（Encoding）。利用已建好的哈夫曼树，对文件 plainFile 中的正文进行编码，然后将结果存入文件 codeFile 中。

（3）D：译码（Decoding）。利用已建好的哈夫曼树，对文件 codeFile 中的代码进行译码，然后将结果存入文件 textFile 中。

（4）P：打印代码文件（code Printing）。将文件 codeFile 显示在终端上，每行 50 个代码。同时将此字符形式的编码文件写入文件 codePrint 中。

（5）T：打印哈夫曼树（Tree printing）。将哈夫曼树以直观的方式（树或凹入表形式）显示在终端中，同时将此字符形式的哈夫曼树写入文件 treePrint 中。

【测试数据】

（1）已知某系统在通信联络中可能出现 8 种字符，其出现频率分别为 0.05，0.29，0.07，0.08，0.14，0.23，0.03 和 0.11。使用该例子的数据调试程序。

（2）用下表给出的字符集和频度的实际统计数据建立哈夫曼树，并实现以下报文中的编码和译码：”THIS PROGRAM IS MY FAVORITE”。

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	17	47	57	1	5	33	20
字符	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
频度	57	63	15	1	49	51	80	23	8	19	1	16	1	

【实现提示】

（1）用户界面可以设计为“菜单“方式：显示上述功能符合，再加上“Q”表示退出运行 Quit。请用户键入一个选择功能符，此功能符执行完毕后再显示此菜单，直至某次用户选择了“Q”为止。

（2）在程序的一次执行过程中，第一次执行 I,D 或 C 命令后，哈夫曼树已经在内存了，不必再读入。每次执行中不一定执行 I 命令。

【选作题目】

从键盘任意输入一段报文，首先统计字符的频度，然后建立哈夫曼树，并给出报文的编码/译码。

第三部分 集合

第 3.1 题：内部排序算法比较

【问题描述】

在教科书中，各种内部排序算法的时间复杂度分析结果只给出了算法执行时间的阶或大概执行时间。试通过随机数据比较各算法的关键字比较次数和关键字移动次数，以取得直观感受。

【基本要求】

（1）对以下 6 中常用的内部排序算法进行比较：冒泡排序、直接插入排序、简单选择排序、快速排序、希尔排序和堆排序。

（2）待排序表的表长不小于 100；其中的数据要用伪随机数产生程序产生；至少要用 5 组不同的输入数据作比较；比较的指标为有关键字参与的比较次数和关键字的移动次数（关键字一次交换计为 3 次移动）。

（3）最后要对结果做出简单分析，包括对各组数据得出结果波动大小的解释。

【测试数据】

由随机数产生器生成。

【实现提示】

主要工作是设法在已知算法中的适当位置插入对关键字的比较次数和移动次数的计数操作。程序还可以考虑几组数据的典型性，如，正序、逆序和不同程序的乱序。

【选作题目】

（1）增加折半插入排序、归并排序等。

（2）对不同的输入表长作试验，观察检查两个指标相对于表长的变化关系。还可以对稳定性作验证。

第 3.2 题：多关键字排序

【问题描述】

多关键字的排序有其一定的实用范围。例如：在进行高考分数处理时，除了需要对总分进行排序外，不同的专业对单科分数的要求不同，因此尚需在总分相同的情况下，按用户提出的单科分数的次序要求排出考生录取的次序。

【基本要求】

（1）假设某年某地的高考成绩汇总表记录数不超过 10 000，考试科目共 6 门，每科成绩范围为 0 至 100。

（2）从高到低输入单科的优先次序，按照高考成绩总和从高到低进行排列，相同总分的按照单科的优先次序进行从高到低进行排列。

【测试数据】

由随机数产生器生成。

第 3.3 题：哈希表设计

【问题描述】

针对某个集体（比如你所在的班级）中的“人名”设计一个哈希表，完成相应的建表和查表程序。

【基本要求】

假设人名为中国人姓名的汉语拼音形式。带填入哈希表的人名共有 30 个。哈希函数用除留余数法构造，用线性探测法或开散列（链地址法）处理冲突。

【测试数据】

取周围较熟悉的 30 个人名。

【选作题目】

（1）从教材上介绍的几种哈希函数构造方法中选出适用者并设计几个不同的哈希函数，比较它们的地址冲突率。

（2）研究 30 个人名的特点，努力找到一个哈希函数，使得对于不同的拼音名一定不发生地址冲突。

（3）在哈希函数确定的前提下尝试各种不同处理冲突的方法，考察平均查找长度的变化。

第四部分 图

第 4.1 题：教学计划编制问题

【问题描述】

大学的每个专业都有制定教学计划。假设任何专业都有固定的学习年限，每个学年有两个学期，每个学期的时间长度和学分上限值均相等。每个专业开设的课程都是确定的，并且课程在开设时间的安排必须满足先修关系。每门课程有哪些先修课程都是确定的，可以有任意多门，也可以没有。每门课恰好占一个学期。试在这样的前提下设计一个教学计划编制程序。

【基本要求】

（1）输入参数包括：学期总数，一个学期的学分上限，每门课的课程号（固定占 3 位的字母数字串）、学分和直接先修课的课程号。

（2）允许用户指定下列两种编排策略之一：一是使学生在各学期中的学习负担尽量均匀；二是使课程尽可能地集中在前几个学期中。

（3）若根据给定的条件问题无解，则报告适当的信息；否则将教学计划输出到用户指定的文件中。计划的表格格式自行设计。

【测试数据】

学期总数：6；学分上限：10；该专业共开设 12 门课，课程信息如下表所示：

课程编号	课程名称	学分	先修课程编号
C01	程序设计基础	2	无
C02	离散数学	3	C01
C03	数据结构	4	C01,C02
C04	汇编语言	3	C02
C05	语言的设计和分析	2	C03,C04
C06	计算机原理	3	C11
C07	编译原理	4	C05,C03
C08	操作系统	4	C03,C06
C09	高等数学	7	无
C10	线性代数	5	C09
C11	普通物理	2	C09
C12	数值分析	3	C09,C10,C01

【选作题目】

产生多种不同的方案，并使方案之间的差异尽可能地大。

第 4.2 题：最小生成树问题

【问题描述】

若要在 n 个城市之间建设通信网络，只需要架设 n-1 条线路即可。如何以最低的经济代价建设这个通信网，是一个网的最小生成树问题。

【基本要求】

（1）利用 Prim 算法求网的最小生成树，起始顶点可以由用户输入。

（2）以文本形式输出生成树中各条边以及它们的权值。

【测试数据】

已知图的顶点集 V={a,b,c,d,e,f,g,h}，边集 E={(a,b,4),(a,c,3),(b,c,5),(b,d,5),(b,e,9),(c,d,5),(c,h,5),(d,e,7),(d,f,6),(d,g,5),(d,h,4),(e,f,3),(f,g,2),(g,h,6)}。

【实现提示】

构成最小生成树的网一定是无向图。设图的顶点数不超过 30 各，并为简单起见，网中边的权值设成小于 100 的整数，可利用随机数函数生成。

第 4.3 题：校园导游咨询

【问题描述】

设计一个校园导游程序，为来访的客人提供各种信息查询服务。

【基本要求】

（1）设计你所在学校的校园平面图，所含景点不少于 10 个。以图中顶点表示校内各景点，存放景点名称、代号、简介灯信息；以边表示路径，存放路径长度等相关信息。

（2）为来访客人提供图中任何景点相关信息的查询。

（3）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路径。

【测试数据】

由所在学校的实际情况指定。

【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图。顶点和边均含有关信息。

【选作题目】

（1）提供图中任意景点问路查询，即求任意两个景点之间的所有路径。

（2）提供校园图中多个景点的最佳访问路线查询，即求途径这多个景点的最短路径。

（3）提供校园导游图的景点和道路的修改和扩充功能。

（4）扩充每个景点的邻接景点的方向等信息，使得路径查询结果能提供详尽的导向信息。

【选作题目】

产生多种不同的方案，并使方案之间的差异尽可能地大。