实验一：Josephus环问题

**1.问题描述**

设编号为：l，2，…，n的n个人按顺时针方同围坐圈，约定编号为k ( 1≤ k≤n)的人按顺时针方向从l开始报数，数到m的那个人出列，它的下位又从1开始报数，数到m的那个人又出列，依次类推，直到所有人出列为止。由此，产生一个个出队编号的序列。试设计算求n个人的出列顺序。

**2.基本要求**

程序运行时，首先要求用户指定人数n、第1个开始报数的人的编号k及报数上限值m。然后出列的顺序打印出相应的编号序列。

**3.提示与分析**

1）由于报告m的人出列的动作对应数据元素的删除操作，而且这种侧除操作比较频繁，因此单向循环链表适于作为存储结构来模拟此过程。而且，为了保征程序指针每一次都指向一个具体的数据元素结点，应使用不带头结循环链表作为存储结构。相应地，需要注意空表和非空表的区别。

2）算法思路：先创建一个含有n个结点的单循环链表，然后由第1个结点起从1开始计算（此时假设K=1），计到m时，对应结点从链表中删除，接下来从被删除结点的下一个结点重新开始从1开始计数，计到m时，从链表中删除对应结点，如此循环，直至最后一个结点从链表中删除，算法结束。

**4.选作内容**

1）在顺序结构上实现本算法，需要考虑如何实现循的顺序结构。

2）m不再固定。假设n个人每人持有一个密码（正整数），从编号为K的人开始从开始顺序报数，报到m的人出列，此时将他的密码作为新的m值，从他顺时针方向上的下一个人开始重新从1报数，报到m的人出列，而后将他的密码作为新的m值，如此循环下去，直到所有人全部出列为止。

3）显示仿真的运算界面。

实验二：模拟停车场管理

**1.问题描述**

设停车场只有一个可停放n辆汽车的狭长通道，且只有一个大门可供汽车进出。汽车在停车场内车辆到达的先后顺序依次排列，若车场内已停满，则后来的汽车只能在门外的便道上等候，一旦停车场内有车开走，则排在便道上的第一辆车即可进入；当停车场内某辆车要离开时，由于停车场是狭长的通道，在它之后开入的车辆必须先退出车场为它让路，待该辆车开出大门后，为它让路的车辆再按原次序进入车场。在这里假设汽车不能从便道上开走。

**2.基本要求**

按照从终端输入数据序列进行模拟管理。

栈用顺序结构实现，队列用链式结构实现。

每一组输入数据包括3个数据项：汽车“到达”或“离去”的信息、汽车牌照号码、汽车“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，则输出车辆在停车场内或便道上的停车位置；若是车辆离去，则输出车辆在停车声内停留的时间和应缴纳的费用（假设在便道上等候的时间不收费）。

**3.提示与分析**

根据问题描述可知，使用栈来模拟停车场，使用队列来模拟车场外的便道；还需另设一个辅助栈，临时停放为给要离去的汽车让路而停车场退出来的汽车；输入数据时必须保证按到达或离去的时间有序。

**4.基本功能分析**

主控功能：介绍程序的基本功能，并给出程序功能所对应的键盘操作的提示，如车到来或离去的表示方法，停车场或者便道的状态的查询方法提示等。

汽车到来：首先要查询当前停车场的状态，当停车场非满时，将其驶入停车场（入栈），开始计费；录停车场满时，让其进入便道等候（入队）。

汽车离开停车场：当某辆车要离开停车场的时候，让它后进停车场的车要为它让路，（即将这些车依次“压入”辅助栈），开走请求离开的车，再将辅助栈中的车依次出栈，“压入”停车场；同时根据离开的车在停车场停留的时间进行收费；最后查询是否有车在便道等候，若有，将便道上的第1辆车驶入停车场（先出队，再入栈），开始交费。

状态查询：用来在屏幕上显示停车位和便道上各位置的状态。

**5.选作内容**

1）汽车有不同种类，则它们的占地面积不同，收费标准不同。例如一辆客车和两辆小汽车的占地面积相同，一辆十轮卡车占地面积相当于四辆小汽车的占地面积。

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。

3）停放在便道上的汽车也收费，收费标准比停放在停车场的车低。、、

4）采用更友好的图形界面显示。

实验三：格式化文本

**1.问题描述**

待排版的一段文本存放在一个文本文件中，要求设计一个程序对文本进行格式化排版，即按指定格式对这段文本排版，实现分段、分页等功能，排版后的文本输出到另一个文本文件中。

**2.基本要求**

1）从输入文件" input.txt"中读取待排版文本

2）排版要求

这段文本由多行文宇组成，每一行包含多个单词，单词之间由一个或多个空格分隔；每行的字符不超过60个。假设一个单词不可被分割存放在两行中。

待排版文本中除了普通的文本字符外，特殊字符“#”表示其后面的正文应另起一行，即表示新的一段开始，新段的段首应向右缩进4个字符，要求排版后的文本中不再出现“#”符号。

格式化的文本参数包含：页长(每页的行数)、页宽(每行的字符数)、页眉（每页正文前的空行数)、页脚〔每页正文后的空行数)、起始页号(从l开始编号)。

3）将待排版义本格式化后在屏幕显示，同时保存到输出文件“output.txt "中。

**3.提示与分析**

1）主要数据类型

Typedef struct

{ int PgLength //页长

Int PgWedth //页宽

Int Hlength //页眉

Int FLength //页脚

Int SpgNum //起始页码

｝Formatpm;

1. 待排版的文本。

#define N 3000

Typedef char source Text[N];

Source Text st;

1. 格式化的文本。

Typedef struct

{FormatPm Parameters;

Char text[page][row][col] //页号、行号、列号

}FormatText;

FormatText ft;

2）基本功能分析

1. 从输入文件“input.txt”中读入待排的文本到数组st中，同时识别普通字符与特殊字符，作出适当的处理。
2. 读入格式化文本的参数结息:页长、页宽、页眉等。
3. 格式化排版。首先，初始化结构体变量ft.parameters。然后，报据页眉Hlength的值设置若干空行。即row的初始位为Hlength;报据SpNum设置页号page的初始值；设置col值为0.

依次扫描st中字符st[k]:

1. 若st[k]=’#’，开始新一行，即row++,若row超出页长Pglength，开始新的一页，即page++，同时相应修改row和col的值；再从ft.text[page][row][0]开始，依次写入4个空格字符。
2. 否则，判断当前字符st[k]是否空格。
3. 若非空格，如果st[k]是一个新单词的开始，统计此单词的字符数n，若n小于当前行剩余列数（PgWedth-col）,则向ft.text中写入此单词，同时k+=n;col+=n;否则开始下一行，即row++;若row超出页长PgLength, 开始新的一页，col=0;写入单词，k+=n.
4. 若st[k]为空格，向ft.text中写入一个空格，col++;接下来，判断st中是否为连续m个空格，不再写入ft.text,令k+=m。
5. 依次类推，处理st中的字符至结束。
6. 从结构体遍历ft.parameters中读取格式化参数信息，按此指定格式将ft.text中内容分页，页内分行，逐行的将字符显示到屏幕上，同时写到输出文件“output.txt”中去。

**4.测试数据**

请自行选择1-3篇没有排版的英文文章进行测试。

**5.选作内容**

1）单词可以被连词符“.”分隔为前后两部分分别存放在某行的末尾及下一行的开头。

2)允许用户在输入文件中指定分页符，排版后实现人为的分页

3）用户在程序运行时可重新指定格式化的文本参数:页长、页宽等。

4）用户在程序运行时可指定待版文件及输出文件的文件名。

实验四：数组与特殊矩阵的应用

**1.问题描述**

1）建立一个n\*n的对称矩阵A；用动态分配的一维数组B对矩阵A进行压缩存储，输出矩阵A和一维数组B；

2）在B中查找对称矩阵A中第row行，第col列（下标从1开始）的元素，输出该元素值；

3）建立一个稀疏矩阵C，输入其行数，列数和非零元个数，用三元组顺序表存储该矩阵，并按矩阵形式输出稀疏矩阵B；

4）对稀疏矩阵C做快速转置运算得到矩阵D，并按矩阵形式输出转置后的矩阵D。

**2.基本要求**

1）矩阵元素相关信息要从终端输入；

2）在三元组顺序表中按行优先顺序存放非零元素

3）具体的输入和输出格式不限；

4）算法要具有较好的健壮性，对错误操作要做适当处理。

**3.提示与分析**

问题1、问题2根据一维数组B中元素下标与矩阵A中行下标、列下标之间的对应关系进行。

问题4可以参考教材算法3-9。

**4.测试数据**

矩阵A中数据采用教材图3-11中数据；矩阵C数据采用教材图3-17中数据。

实验五：二叉树的应用

**1.实验任务**

以二叉链表为存储结构，分别实现二叉树的下列运算:

1. 创建一颗二叉树；
2. 分别用前序、中序和后序遍历方法输出该二叉树的结点序列；
3. 以下为选做题，但这几个典型算法要掌握
   1. 在二叉树中查找值为x的元素
   2. 统计二叉树中结点的个数
   3. 求二叉树的高(深)度
   4. 统计二叉树的叶子结点数

**2.任务要求**

二叉树中结点通过键盘输入，通过界面选择提示输入前序、中序或后序相应的选择值，程序执行相应的遍历，并将遍历结果显示在屏幕上。

结点值编写完整的程序，实现任务要求，上机调试程序直到算法运行正确。

**3.提示与分析**

问题a)可参考教材4-1a、4-1b。

问题b)可参考教材算法4-5、算法4-6、算法4-7.

问题c)中操作算法可参考教材算法4-12、算法4-13。

**4.测试数据**

问题a)中二叉树的结构形态及结点信息自行给定。

问题b)和问题c)中二叉树可以自行确定也可以用问题a中创建的树为例进行。

实验六：哈夫曼编码/译码器

**1.问题描述**

哈夫曼编码是一种应用广泛而有效的数据压缩技术。利用哈夫曼编码进行通信可以大大提高信道利用率，加快信息传输速度，降低传输成本。数据压缩的过程称为编码，解压缩的过程称为译码。进行信息传递时，发送端通过一个编码系统对待传数据（明文）预先编码，而接收端将传来的数据（密文）进行译码。要求设计这样一个简单的哈夫曼编码/译码系统。

**2.基本要求**

系统应具备如下几项功能。

构造哈夫曼树及哈夫曼编码：从终端读入字符集大小n、n个字符以及n个对应的权值，建立哈夫曼树；利用已经建好的哈夫曼树求每个叶结点的哈夫曼编码，并保存。

编码：利用已构造的哈夫曼编码对“明文”文件中的正文进行编码，然后将结果存入“密文”文件中。

译码：将“密文”文件中0、1代码序列进行译码。

打印“密文”文件：将文件以紧凑格式显示在终諯上，每行30个代码；同时，将此字符形式的编码文件保存。

打印哈夫曼树及哈夫曼编码：将已在内存中的哈夫曼树以凹入表形式显示在终端上，同时将每个字符的哈夫曼编码显示出来；并保存到文件。

**3.提示和分析**

1）采用静态链表作为哈夫曼树的存储结构。

2）求哈夫曼编码时使用一维结构数组HCode作为哈夫曼树编码信息的存储。

3）基本功能分析。

初始化：从键盘接收字符集大小n，以及n个字符和n个权值。

建立哈夫曼树：构造哈夫曼树，即将HCode数组中的各个位置的各个域都添上相关的值，并将这个结构体数组存于文件HTree.txt中。

构造哈夫曼编码：为从文件HTree.txt中读入相关的字符信息进行哈夫曼编码，然后将结果存入HNode.txt中，同时将字符与0、1代码串的一一对应关系打印到屏幕上。

编码：利用已构造的哈夫曼编码（HNode.txt）对文件SourceFile.txt（明文）中的正文进行编码，然后将结果存入文件CodeFile.txt（密文）中。

译码：将文件CodeFile.txt（密文）中的代码按照3）中建立的编码规则将其翻译成字符集中字符所组成的字符串形式，进行译码，结果存入文件TextFile.txt(明文)中。（如果正确，TextFile.txt的内容与SourceFile.txt的内容一致）

打印哈夫曼树：从HNode数组中读入相关的结点信息，以凹入表方式将各个结点以及叶子结点的权值和左分支上的0和右分支上的1画出来。

**4.选作内容**

系统能够统计待编码文本中所使用的字符和出现的频率，并据此自动构造哈夫曼编码。

实现各个转换操作的源/目文件，均可由用户在选择该操作时指定和命名。

修改你的系统，对你的系统的原程序的编码和译码（主要是将行尾符编/译码问题）。

以直观的形式在屏幕上显示哈夫曼树。

实验七：校园导游程序

**1.问题描述**

到一个陌生的地方去旅游的时候，人们常常需要一个导游在自己游玩的过程中为人们提供更多的服务，比如介绍景点的历史背景等相关信息，推荐到下一个旅游景点的最佳路径，以及解答旅游者所提出的关于旅游经典的相关问询等。对于新生刚刚来到学校，对校园不熟悉的情况也如此，如果能够提供一个程序让新生或者来访的客人主动通过与机器的“对话”来获得相关的信息的话，就会节省大量的人力和时间，而且所提供的信息能够尽可能的准确，详尽。本实验要求设计一个校园导游程序，为用户提供各种信息查询服务。

**2.基本要求**

本次实验需要开发一个简单的校园导游程序，程序的主体功能为：

1）设计你的学校的校园平面图，所包含的景点不少于10个。以图中顶点表示校园内各景点，存放景点名称、简介等信息；以边表示路径，存放路径长度等信息。

2）显示校园平面图，方便用户直观的看到校园的全景图，并确定自己当前所在位置。

3）为用户提供对平面图中任意场所的相关信息的查询。

4）为用户提供对平面图中任意场所的问路查询，即查询任意两点之间的一条最短路径。

**3.提示与分析**

（1）主要数据结构

由于各个场所通过校园内的路径相连，各个场所和连接它们的道路构成了整个校园的地理环境，所以使用图这种数据结构对他们去进行描述。以图中顶点表示校园内各个场所，应包含场所名称、代号、简介等信息；以边表示连接各个场所的道路，应包含道路的代号，路径长度等相关信息。一般情况下，校园的道路是双向通行的。因此，校园平面图可以看作一个无向图。

图的顶点和边均使用结构体类型，整个图的数据结构可采用教材中介绍的各种表示方法，如戴权的邻接矩阵。

（2）基本功能分析

1. 显示校园平面图：平面图中应该醒目地标识出场所的准确名称以备用户查询。
2. 任意场所的相关信息查询：接受用户所输入的用户名称，并将场所的简介信息反馈给用户。
3. 求单源点到其他各点的最短路径的功能模块：计算并记录从校园门口到各个场所的最短路径。
4. 任意场所的纹路查询的功能模块：接收用户所输入的场所名称，在计算的最短路径集合中找到相关项的信息并反馈给用户。

4.测试数据

由读者根据实际情况指定。

5.选作内容

1. 提供图中任意场所的问路查询，即求任意两点之间的最短路径。
2. 提供校园图中多个场所的最佳访问路线查询，即求途径多个场所的最短路径。
3. 校园导游图的场所和道路的修改扩充功能。
4. 实现校园导游图的图形界面。

实验八：职工信息检索系统

**1.问题描述**

若某单位职工有n名，试以职工的姓名为关键码，设计散列表，使得平均查找长度不超过L。请完成相应的建表和查询功能。

**2.基本要求**

假设每名职工的姓名以汉语拼音形式表示。待填入散列表的人名共有n个，取平均查找长度的上限为n=3.请构造合适的散列函数，选择恰当的处理冲突的方法来构造散列表。

输入职工的姓名，实现在散列表中的检索职工信息，显示检索过程。

**3.提示分析**

1）散列函数的构造

散列函数的选取原则是形式简单而分布均匀。一般情况下，姓名包含2~4个汉字，因此拼音的长度不超过24个字符。一个字的拼音一般由声母和韵母构成，汉语拼音中的声母和韵母是有限的，因此可以考虑对声母和韵母进行编号，那么一个姓名的编号就是一串数字，以此数字为关键码，根据职工人数n确定散列表的长度m，采取除留余数法

H（key） = key%p （P选择不超过m的最大素数）

来计算散列地址（当姓名较长时，key位数较多，可先作折叠处理，再计算散列地址）。

2）处理冲突的方法

好的散列函数使得散列地址在散列表中的分布较均匀，发生冲突的机会较小，但冲突是不可避免的，发生冲突时，采用开发定址法处理冲突，可采用二次探测法形成探测序列。

3）查找过程

输入要查找的职工的姓名，根据其姓名的编号key计算散列地址H（key），判断此地址单元存放的元素的关键码是否等于key，若等于，查找成功；否则，根据探测序列，计算下一散列地址，判断此地址单元存放的元素的关键码是否等于key，若等于查找成功；否则继续计算下一个地址，直至查找成功或失败。

**4.选作内容**

1)设计其他几个不同的散列函数，比较它们的地址发生冲突的概率。

2）在确定散列函数的前提下，尝试其他不同处理冲突的方法。

3）研究若干个人名拼音的特点，寻找更好的散列函数，使得对于不同的拼音名映射到不同的散列地址。

课后实验一 个人图书管理系统

**1.问题描述**

人们在自己的工作和生活中会拥有很多的图书，对所购买的图书进行分类和统计是一种良好的习惯，同时也便于对这些图书资料进行整理和查询。图书的各种信息，包括分类、作者、出版日期、价格简介等、辅之以程序来对图书的信息进行统计和查询的工作将使得这种图书管理工作变得轻松简单。为解决这个实际问题开发个人图书管理系统。

**2.基本要求**

这些图书管理系统具备如下的功能。

保存图书的各种相关信息：出版日期、书号、书名、作者、图书类别编号、价格等。

查找功能：按照多种关键码查找需要的图书，查找成功后可以修改记录的相关项。

排序功能：按照多种关键码对所有的图书进行排序，例如按照购买日期进行排序、按图书类别排序等。

显示图书的信息。

其他辅助的维护工作。

**3.提示与分析**

（1）主要数据结构

由于图书的册数较多，而且要在程序不运行的时候仍然要保存着数据，所以采用文件的形式放到外存储器中。需要操作时，从文件中调入内存来进行查找和排序工作，为了接受文件中的内容，需要有一个数据结构与之对应，可以采用这样的结构体数组来接受数据。

（2）基本功能分析

初始化：清空books数组，进入输入状态，接受键盘输入的全部数据保存在books数组中，按某种顺序输入记录。

插入：接受从键盘输入的一条新的记录，按某种顺序（如作者、出版日期等）插入到books数组中。

删除:接收从键盘输入的一本书的信息（如书名、书号、作者等），在books数组中查找，如找到,则从books数组中删除该记录，否则显示“未找到”。

更新：接收从键盘输入的一本书的信息，在 books 数组中查找，如找到，则显示该记录的原数据并提示键盘输入新数据用以替换原有数据，如未找到，则显示“未找到”。

统计：统计每类书籍的数量，显示统计结果。

排序：对books数组中所有记录按“类别 ”排序，类名相同的按“书名”排序（字典序），显示排序结果。

**4.选作内容**

实现模糊查询

列出某个作者全部名著的查询

实现图形化的操作界面。

课后实验二 各种内部排序的性能比较

**1.问题描述**

通过排序过程中关键字比较次数和关键字移动次数，分析比较常用的排序算法的性能。

**2.基本要求**

对直接插入排序，气泡排序，简单选择排序，快速排序，希尔排序，堆排序等常用的内部排序算法进行比较。

排列表的表长不小于200；其中的数据要用伪随机数产生；至少要用6组不同的输入数据（包含正序，逆序和随机次序的情况）作比较；以关键码的比较次数和移动次数（关键码的交换计为3次移动）作为衡量的指标。

最后要对结果做出简单的分析。

**3.提示与分析**

1）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。

2)考虑输入数据的典型性：如正序，逆序和不同程序的乱序。

**4.选作内容**

1）增加对折半插入排序，二路归并排序，基数排序等排序方法的分析。

2）对不同的输入表长作试验，观测两个指标对于表长的变化关系。

3）验证排序方法的稳定性。

课后实验三 模拟银行排队办理业务

**1.问题描述**

到银行办理业务往往需要排队。以前银行采用的是多队列多窗口式的服务，后来大多数的银行安装了电子叫号系统，将服务顺序改为单队列多窗口式的服务，现如今一些银行还增设了VIP客户，服务顺序设为双队列多窗口式服务。请根据下面要求，编写程序模拟这三种不同的银行服务模式，从而比较，分析不同方法产生的效果。

**2.要求**

1）输入下列数据时：银行的窗口数（包括普顾客业务办理窗口数和VIP顾客业务办理窗口数）、顾客总数、每位顾客的类型以及其到达的时间和处理事务的时间，采用先来先服务的排队策略，分别计算在问题描述中提到的3种银行服务模式下，每个顾客的等待时间以及所有顾客的平均等待时间。

2）采用短时间服务业务优先的排队策略，分别计算在问题描述中提到的3种银行服务模式下，每个顾客的等待时间以及所有顾客的平均等待时间。

3）采用长时间服务业务优先的排队策略，分别计算在问题描述中提到的3种银行服务模式下，每个顾客的等待时间以及所有顾客的平均等待时间。

4）通过分析上述排队策略的优劣，提出一种与以上不同的排队策略，分析其特点并分别计算在问题描述中提到的3种银行服务模式下，每个顾客的等待时间以及所有顾客的平均等待时间。