## 小型文本编辑器

任务：设计一个文本编辑器，使其具有通常编辑器(如Notepad)所应具备的基本功能。

要求：编辑器应具备如字符串查找，字符串剪切，字符串粘贴，字符串替换，统计字数，统计行数等基本功能。

## 用密钥K对密文C解密得到明文P

任务：设明文P=P0P1P2…Pn和密钥K=K0K1K2…Km(n>=m)中的字符Pi(1<=i<=n)或Kj(1<=j<=m)的ASCII为00~7FH，用密钥K对明文P进行加密得到密文C=C0C1C2…Cn, 用密钥K对密文C解密得到明文P。

加密： Ci=Pi+Kj (j=i mod (m+1)) (当Ci<=7FH)

Ci=Pi+Kj-80H (j=i mod (m+1)) (当Ci>7FH)

解密： Pi=Ci-Kj (j=i mod (m+1)) (当Ci>=Kj)

Pi=Ci-Kj+80H (j=i mod (m+1)) (当Ci<Kj)

要求：实现加密解密。实现文件存储。

## 媒体流调度问题

任务：设一个多媒体播放系统用来播放一个 video segment 集合 V，该集合中的每一个元素 segment S=(B; L; F)，其中 B 为开始时间，L 为媒体播放时间，即媒体长度，F 为该媒体播放被延迟所造成的损失函数（此处可以假定 F 为关于延迟播放时间 D 的线性函数，如就定义 F(S) = S.D，也可以自己设定）。媒体播放时要求每个 S 的播放不能被中断（不可抢占），只能一个接一个的播放。本课程设计的基本目的是设计给出尽量少损失的调度算法。对于 V={S1，S2，…，Sn}，V 上的一个调度就是 V 集合中各元素的一个置换，即重新排列。本课程设计题目要求给出三个以上的调度方法：此处给出三种，可以也鼓励自己想出新的、更巧妙的调度方法：

① 最早开始时间最早调度

将 V 中的各 S 按其 B 属性从小到大排序，排序结果就是调度结果。

② 最短媒体最先调度

将 V 中的各 S 按其 L 属性从小到大排序，排序结果就是调度结果。

③ 贪心调度

保证每次调度都是此次调度引起最少损失的 Si。

要求：

① 随机的产生初始化 V 集合，V 中的各个 S 也是随机产生的，即其中的 B 和 L

随机设定。

② 实现三种以上的调度算法实现调度，并计算比较各算法造成的损失。

③ 完成贪心调度策略设计并实现。

④ 思考能完成最优调度的算法。

实现提示：阅读贪心法的相关资料。

## .马踏棋盘的求解及演示（八皇后）

任务：设计一个国际象棋的马踏遍棋盘的演示程序.

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,并按求出的马的行走路线,将路线1,2,…,64依次填入一个8×8的方阵,输出之。

测试数据：可自行指定一个马的初始位置(i,j),0≤i,j≤7.

实现提示:下页图显示了马位于方格(2,3)时,8个可能的移动位置.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),(i+1,j-2),(i-1,j-2),(i-2,j-1)

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 |  |  |  |  |  |  |  |  |

但是,如果(i,j)靠近棋盘的边缘,上述有些位置可能超出棋盘范围,成为不允许的位置.8个可能位置可以用两个一维数组HTry1[0..7]和HTry2[0..7]来表示:

HTry1 0 1 2 3 4 5 6 7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| -2 | -1 | 1 | 2 | 2 | 1 | -1 | -2 |

HTry2 0 1 2 3 4 5 6 7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 2 | 1 | -1 | -2 | -2 | -1 |

位于(i,j)的马可以走到的新位置是棋盘范围内的(i+HTry1[h],j=HTry2[h]),其中h=0,1,…,7.

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

(1) 求出从某一起点出发的多条以至全部行走路线.

(2)探讨每次选择位置的“最佳策略”,以减少回溯的次数.

## 货物进栈、出栈算法

任务：商店货架以栈的形式摆放商品，生产日期越近的越靠近栈底,出栈是从栈顶取货，一天营业结束，如果货架不满,则需上货，如果直接将商品摆放到货架上，则会使生产日期越近的越靠近栈顶，这就需要倒货架，仍使生产日期越近的越靠近栈底。

要求：写出货物进栈、出栈算法。

## 银行业务模拟问题描述

任务：客户业务分为两种。第一种是申请从银行得到一笔资金，即取款或借款。第二种是向银行投入一笔资金，即存款或还款。银行有两个服务窗口，相应的有两个队列。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，且申请额超出银行现存资金总额而得不到满足，则立即排入第二队等候，直至满足时才离开银行，否则业务处理完后立即离开银行。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）第二个队列的客户，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。

要求：写一个上述银行业务的事件驱动模拟系统，通过模拟方法求出客户在银行内逗留的平均时间。

## .简单数字图像处理

任务：一幅图像就是一个从位置集到颜色集的变换。考虑二维图像，位置集实际上就是一个矩阵，此时一幅图像实际上就是一个内容为颜色矩阵。如果颜色为 0-255 间的整数， 表示该位置的灰度等级，0 为黑色，255 为白色，此时的图像称为灰度图。而图像的处理就是在该矩阵进行相关计算。一种常见的计算平滑操作就是通过一点和周围 8个点的信息共同决定该点的新值：如一点的新值为该点和周围 8 点颜色值之和的均值， 这一操作可用下图表示。

|  |  |  |
| --- | --- | --- |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

图像平滑模板

锐化操作:

|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| -1 | 9 | -1 |
| -1 | -1 | -1 |

要求：熟悉 Windows 下 BMP 文件的格式，能够实现其读写(只考虑灰度图像)。实现图像的平滑和锐化操作，其它处理操作选做。

## .英文文章检索

任务：实现对字符串和文件方面知识的运用，对英文文章实现查找、插入、删除、替换、定位等操作。

要求：对英文文章实现字符串模式匹配算法和文件操作算法。

## 哈希查找

任务：哈希查找是一种有效的查找方法，是与内存地址有关的查找。该题目是对一组任意的关键字进行哈希设置，并且利用多种方式解决冲突。

要求：用不同的方式建立哈希函数，并且用不同的方法解决冲突。

## B-树模拟实现

任务：实现 B-树的基本操作。

要求：支持阶数设定。实现过程演示。

## 简单矢量图形的几何变换

任务：常见的几何图形（二维图形）都是由点、直线和圆组成，而平面上的点和直线都可用矢量进行表示，如果假设几何图形中不包含圆，那么一个无论多么复杂的几何图形都可用矢量表示并存储，且图形上的几何变换都可通过矢量上的变换得以实现。

例如下图：



P3

P1

P2

P4

矢量图形的实例

图中的十字形状就可用矢量 P1,P2-P1 和 P3,P3-P4 表示，当然其中 P1,P2,P3,P4 都是向量， 分别表示十字图形的四个顶点。其中 P1 表示直线段的起点，P2-P1 表示该直线的方向和长度。

而该图的平移变换就可表示如下(向右水平平移 5 个单位)：

P3

P1

P2

P4

矢量图形向右平移 5 个单位的结果

[P1+H,P2-P1];[P3+H,P3-P4]其中 H 为矢量 5+0i。

要求：能够应用线性数据结构描述简单的矢量图，并能完成简单的几何变换。描述并实现矢量 ADT，包括矢量的表示和存储，以及其上的基本操作：矢量相加；相减；共扼；求模；相乘等。应用矢量实现简单几何图形（由点和直线组成）的表示和存储。同时该几何图形能够保存在文件中（格式自定），并能从文件中读出。实现矢量几何图形的基本操作，包括平移（水平加垂直）；旋转；依直线镜像； 依点镜像；放大；缩小等（可自行扩展）。

选做内容：提供界面实现用户对图形的操作，包括输入图形，进行变换操作等。

实现提示：用二元组表示一个向量，一个几何图形就是一个数组。

## 外部排序模拟

任务：该题目是考察学生对排序知识的了解，并且对内部排序和外部排序的区别和方法有一定的了解，并能够用文件的知识对其实现和模拟。

要求：用文件的知识模拟外部排序，实现多路平衡归并的实现。考虑内存和外存之间的关系。

## 中国邮路问题

任务：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。

要求：

（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)

（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。

（3）界面要求：有合理的提示和人机交互。

## 八数码

任务：在一个3×3的九宫中有1－8这8个数及一个空格随机的摆放在其中的格子里，用户设定一个初始状态和设定一个最终状态，要求动态给出采用A\*算法找出的从初始状态变化到最终状态的最优路径，并显示出该算法下所有可能的路径节点。各状态之间变化的调整规则是：每次只能将与空格（上、下、左、右）相邻的一个数字平移到空格中。

A\*算法描述：在所有结点对应的f(n) = h(n) + g(n) 中选择f(n)最小值，作为下次选择路径的起点。其中：

n：结点的编号。

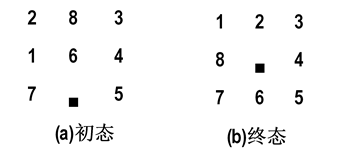
h(n)：与目标状态相比，该状态数码的错位个数。

g(n)：搜索的深度，一般根节点(初始状态)的深度是为0, 其他子节点深度为父结点深度加1。

基本要求：

从终端输入的8数码的初始状态和最终状态。

如：



显示求解中的所有结点，并输出各结点对应的f (n)值；

要求：输出从初态移动到终态的最优移动过程（路径）。界面友好，函数功能要划分好。

## 关键路径问题

任务：当一项工程划分为若干个子任务或活动后，人们不仅需要确定这些活动的先后次序，而且需要进一步计算完成整个工程的时间，确定哪些活动是影响工程进度的关键活动，以便合理地组织人力、物力、财力，加快这些活动的进度，为按时或提前完成整个工程提供保证，这就是关键路径问题。关键路径问题相应的网称为AOE网，其中：顶点表示事件，边表示活动，边上的权表示活动持续的时间。

要求：

（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）

（2）判断该AOE网是否能够顺利进行。

（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）

## 最小生成树问题

任务：已知一个无向连通网表示n个城市以及城市间可能设置的通信线路，其中网的顶点表示城市，边表示两个城市之间的线路，赋于边上的权值表示相应的代价。对于n个点的连通网可以建立许多不同的生成树，每一棵生成树都可以是一个通信网。现在，我们要选择这样一棵生成树，使总的耗费最小。即构造连通网的最小生成树的问题。

要求：

（1）建立城市交通网的存储结构。（注：数据输入可以是键盘输入或文件输入两种方式）

（2）分别用Prim算法和Kruskal算法构造最小生成树，并输出最小生成树的代价及生成树的边。（注：结果的输出可以是屏幕输出和文件输出两种方式）

## 校园导游系统

任务：设计一个校园导游程序，完成校园信息的维护以及为来访的客人提供信息查询等服务功能。

要求：

设计学校的校园平面图，所含景点不少于10个，以图中顶点表示校内各景点，顶点的信息包括：景点名称、代号、简介等，以边表示道路，边上信息包括：两点距离、所需时间等相关信息。（注：数据的输入可以是键盘输入或文件输入两种方式）

提供对校园景点信息的编辑（如：添加、删除、修改等）的功能；

为来访客人提供图中任意景点相关信息的查询（可提供多种查询方式）；

为来访客人提供从校门口到图中任意景点的问路查询（最短路径）；

为来访客人提供图中任意景点间的问路查询。

## 文件目录管理系统

任务：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。

要求：

函数功能要划分好，程序要有必要的注释。

用户通过界面菜单选择以下操作：

生成文件，选择路径和文件名，实现对文件的生成。

删除文件，对指定文件进行删除操作。

修改文件，对指定文件进行内容修改或者文件名修改。

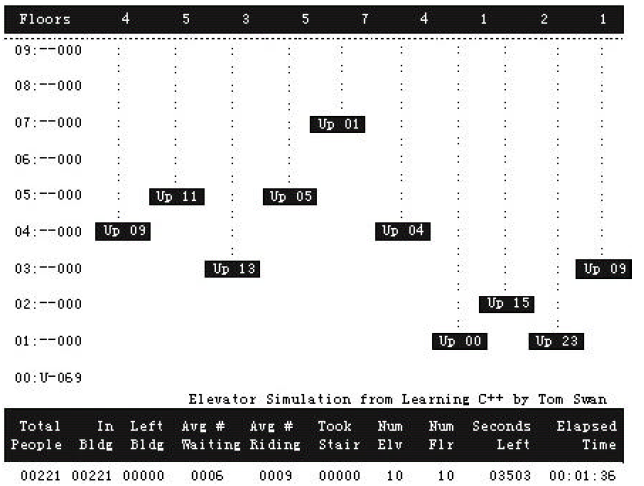
输出该目录结构。

## .置换选择排序

任务：该题目是考察学生对排序知识的了解，并且对内部排序和外部排序的区别和方法有一定的了解，并能够用文件的知识对其实现和模拟。

要求：用文件的知识模拟外部排序，置换选择排序的实现。考虑内存和外存之间的关系。

## 电梯运行仿真程序

任务：该仿真程序可以用来监测系统运行情况，改善大楼管理。系统初步描述：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，电梯运行每上下一层需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。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。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。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；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。

要求：编写一个程序，模拟办公大楼中全部电梯的工作过程。

## 最佳归并树

任务：该题目是考察学生对排序知识的了解，并且对内部排序和外部排序的区别和方法有一定的了解，并能够用文件的知识对其实现和模拟。

要求：用文件的知识模拟外部排序，最佳归并树的实现。考虑内存和外存之间的关系。

## 一个基于 XML 的网站生成器

任务：分析一个 XML 文件，并根据分析结果自动生成一个完整的新闻网站。理解栈结构，并能栈结构解决复杂的实际问题。XML 文件形如：

<newslist>

<article>

<header>

<title>Students Seen on Rice Campus</title>

<author>Donna B. Cerius</author>

<section>Metro</section>

<date>23 Sep 1999</date>

</header>

<body>

……

</body>

</article>

<article>

……

</artile>

……

</newslist>

要求：主页中的链接需依据 XML 中 article 的 section 进行分类；结果网站中的主页中有通向各个 article 页面的链接；结果网站中每个 article 对应一个网页。网站页面的样式自行设计。

实现提示：需参阅关于 XML 和 HTML 的基本介绍。

## 实用的小型通讯录

问题描述：设计一个管理个人通讯录的程序。

要求：完成下列基本功能。

1“文件”管理

1.1导入通讯录：采用手工输入方式，C盘下建立一个通讯录文件，存放与用户相关的人员通讯信息， 通过合并方式，将用户选定通讯录文件与C盘下已有通讯录文件合并，产生新的通讯录。（注意：新通讯录中不能出现重复的人员通讯信息），每条通讯信息由姓名、手机号码、家庭号码、工作单位、家庭住址、与本人关系组成，每个数据项以字符串形式定义。

1.2导出通讯录：将所有通讯信息导出在EXCEL表中存放。将部分满足条件的通讯信息导出EXCEL表中存放

1.3备份通讯录：在默认的C:\BackUp文件夹中，定期备份通讯录中所有信息。备份文件以“BackUp当前日期.txt”命名。

2通讯录管理

2.1通讯录信息浏览：所有通讯信息浏览。某个类别的通讯信息浏览。其中类别有家庭、朋友、同学、同事和其他四类。

2.2通讯录信息添加：按照类别添加通讯信息，如果类别缺省，则属于“其他”类中。添加后，需要及时更新通讯录文件。

2.3通讯录信息删除：删除后，需要及时更新通讯录文件。

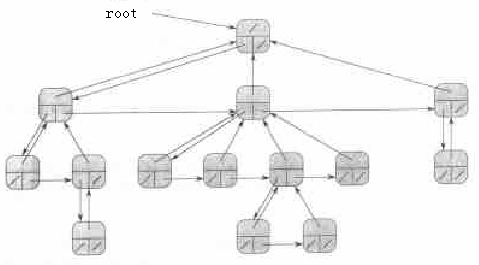
2.4通讯录信息查询：按照姓名、电话号码等字段，进行精确、模糊查询，并在屏幕上输出查询结果。

2.5通讯录信息修改:修改后，需要及时更新通讯录文件。

3通讯录类别管理:类别添加;类别删除;类别修改;类别删除后，下属的所有通讯信息，应该归为“其他”类中。

## 模拟文件目录系统

任务：本设计需完成两部分工作：一个是定义并实现称为 CatalogTree 的 ADT，用它来表达字符串集合组成的有序树；另一个是一个 Shell 的应用程序，用它来模拟文件目录系统，并提供模拟操作界面。

CatalogTree 的组织结构如下图(带父结点指针的儿子—兄弟链树)：

CatalogTree 的结构示意图

针对于目录系统，CatalogTree 的结点存放的数据内容为字符串，每个结点对应一

个目录项，该目录项可以是目录，也可以是文件，如果是目录就可以再存放其它目录或文件，即非叶结点；如果是文件就是叶结点。从根结点到该结点路经所有结点的字符串用“/” 进行组合后就是该目录项的绝对路径， 用来唯一的标识该目录。例如：

/usr/li/email/student/。

目录系统具有如下基本操作：

dir ——列出当前目录下的所有目录项

cd ——打出当前目录的绝对路经

cd ..——当前目录变为当前目录的父目录

cd str——当前目录变为 str 所表示路径的目录

mkdir str ——在(当前目录下)创建一个子目录(名为 str)

mkfile str ——在(当前目录下)创建一个文件(名为 str)

delete str ——删除(当前目录下)名为 str 的目录或文件

要求：① 描述并实现 CatalogTree 的 ADT，包括其上的基本操作：如插入一个结点，寻找一个结点，返回一个结点的最左儿子等（具体情况依据应用自定）。

② 应用 CatalogTree 的 ADT 实现一个完成文件目录系统的 Shell 应用程序。

③ 该 Shell 是一个不断等待用户输入命令的解释程序，根据用户输入的命令完成相关操作，直到退出（quit）。命令名及其含义如上所述。

④ 目录树结构可以保存（save）到文件中，也可从文件中读出（load \*.dat）。

⑤ dir 命令的结果应能够区分是子目录和还是文件。

## 应用堆实现一个优先队列并实现作业的优先调度

任务：优先队列 ***priority queue*** 是一种可以用于很多场合的数据结构，该结构支持如下几本操作：Insert (S, x) – 将元素 x 插入集合 S；Minimum (S) – 返回 S 中最小的关键字；Extract–Min (S) – 删除 S 中的最小关键字；DecreaseKey(S,n,key) – 将 S 中的结点 n 的关键字减小为 key。

要求以堆作为辅助结构设计并实现一个优先队列。要将堆结构嵌入到队列结构中， 以作为其数据组织的一部分。应用该优先队列实现作业的优先调度：一个作业 ti =（si，ei），si 为作业的开始时间（进入时间），ei 为作业的结束时间（离开时间）。作业调度的基本任务是从当前在系统中的作业中选取一个来执行，如果没有作业则执行 nop 操作。本题目要求的作业调度是基于优先级的调度，每次选取优先级最高的作业来调度，优先级用优先数（每个作业一个优先数 pi）表征，优先数越小， 优先级越高。作业 ti 进入系统时，即 si 时刻，系统给该作业指定其初始优先数 pi = ei -si，从而使越短的作业优先级越高。该优先数在作业等待调度执行的过程中会不断减小，调整公式为：pi = pi - wi，其中的 wi 为作业 ti 的等待时间：wi = 当前时间-si。一旦作业被调度，该作业就一直执行，不能被抢占，只有当前执行作业指向完成时，才产生下一轮调度。所以可以在每次调度前动态调整各作业的优先数。编程实现这样一个作业调度系统。

要求：① 给出优先队列的 ADT 描述，包括队列的逻辑结构及其上基本操作。② 以堆结构为辅助结构实现优先队列的存储表示并实现其上的基本操作。③ 作业集合中的各作业随机生成，根据作业的 s 属性和 e 属性动态调整作业队列， 不断加入作业，作业结束删除作业。④ 要对作业调度的结果给出清晰的输出信息，包括：何和作业进入，何时调度哪个作业，何时离开，每个作业等待多长时间，优先数的动态变化情况等。

## 图书管理

任务：图书管理基本业务活动包括：对一本书的采编入库、清除库存、借阅和归还等等。试设计一个图书管理系统，将上述业务活动借助于计算机系统完成。

要求：（1）每种书的登记内容至少包括书号、书名、作者、现存量和总库存量等（2）作为演示系统，不必使用文件，全部数据可以都在内存存放。但是由于上述四项基本业务活动都是通过书号（即关键字）进行的，所以要用B树对书号索引，以获得高效率。（3）系统应实现的操作及功能定义如下：①采编入库：新购入一种书，经分类和确定书号后登记到图书帐目中去。如果这种书在帐目中已有，则只将总库存量增加。②清除库存：某种书已无保留价值，将它从图书帐目中注销。③某种书的现存量大于零，则借出一本，登记借阅者的图书证号和归还期限。④归还：注销对借阅者的登记，改变该书的现存量。⑤显示：以凹入表的形式显示B树。这个操作是为了调试和维护的目的而设置的。下列B树的打印格式如下所示：

60

30

20

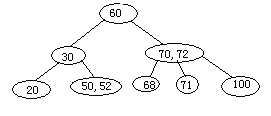
50，52

70，72

68

71

100



## 稀疏矩阵运算器

问题描述：稀疏矩阵是指那些多数元素为0的矩阵。利用“稀疏”特点进行存储和计算可以大大节省存储空间，提高计算效率。实现一个能进行稀疏矩阵基本运算的运算器。

## 算术表达式求值演示

任务：表达式求值是实现程序设计语言的基本问题之一，也是栈的应用的一个典型例子。设计一个程序，演示用算符优先法对算术表达式求值的过程。

要求：以字符序列的形式从终端上输入语法正确的、运算量可以是变量的实数类型表达式。利用教材中给出的算符优先关系，实现对算术四则混合运算表达式的求值，并演示在求值中运算符栈、运算数栈、输入字符和主要操作的变化过程。

提示：（1）设置运算栈和运算数栈辅助分析算符优先关系。（2）在输入表达式的字符序列的同时，完成运算符和运算数的识别处理，以及相应的运算。（3）在识别出运算数的同时，要将其字符序列形式转换成实数形式。

## 文档集合上的查询

任务：设计数据结构完成在一个文档集合的存储，并构造算法实现其内容的查询。该设计包括三个部分：1应用数据结构完成文档集合的内容(基于单词的)存储，并为下一步的查询建立索引。2就单个单词的查询请求，设计算法进行查询。3对多个单词通过 AND 和 OR 构造的复杂查询进行处理(此处可只做两个单词的情况)。具体情形如下面的例子：

*Doc1:* I like the class on data structures and algorithms.

*Doc2:* I hate the class on data structures and algorithms.

*Doc3:* Interesting statistical data may result from this survey.

Here are the answers to some queries:

Query 1: data

Doc1, Doc2, Doc3

Query2: data AND structures

Doc1, Doc2

Query 3: like OR survey

Doc1, Doc3

要求：① 文档集合中的文档数不能少于 20 个。② 数据结构的设计以及查找算法的构造应考虑如何最大程度的提高查询效率。③ 查询效率的提高应是综合多种查询的，而不是只针对一种查询的优化。④ 给出查询效率的模拟实验数据。设计良好的索引结构对于本题是非常重要的

## 实现并对比三种基本的字符串匹配算法

任务：字符串匹配问题是：假定文本时一个长度为 n 的数组 T[1..n]，字符串匹配问题是：假定文本时一个长度为 n 的数组 T[1..n]，模式是一个长度为mn 的数组 P[1..m]，如果 0sn-m，并且 T[s+1..s+m] = P[1..m]，则称模式 P 完成了和T 的匹配，且 P 在 T 中出现的位移为 s，如下图所示：



字符串匹配问题

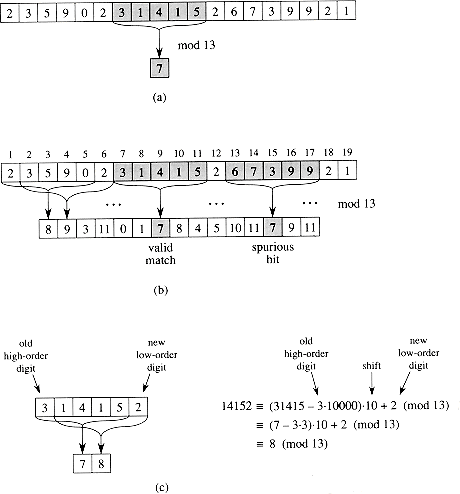
本课程设计要实现三种基本的字符串匹配算法：朴素匹配算法；Rabin-Karp 算法；KMP 算法。朴素匹配算法是最简单的匹配算法，可以形象地看成是用一个包含模式 P 的模板沿文本滑动，同时对每个位移注意模板的上的字符是否与文本中的相应字符相等。Rabin-Karp 算法基本思想是将模板 P 用一个数 p（通常是十进制）表示，如果字符串中的每位都是0-9 的整数：

p = P[m] + 10（P[m-1] + 10（P[m-2]+ … + 10(P[2] + 10P[1])…））

同时将 T 中每个 m 位连续的串上计算出同样的正数值，有 n-m 个：t1，t2，…，tn-m。一种较为巧妙的方法是在 ts 基础上可按如下方式计算 ts+1：

ts+1 = 10（ts – 10m-1 T[s+1]）+ T[s+m+1]

由于这样的值非常大，可能会导致溢出，通常的处理办法是计算 p mod q 和 ts mod q，计算时会用到性质 a+b mod n = a mod n + b mod n；a \* b mod n = a mod n \* b mod n。Rabin-Karp 算法的具体过程如下图所示：



Rabin-Karp 字符串匹配算法示意图

KMP 算法是一种设计得很巧妙但也较复杂的一种算法，是应用自动机思想实现的匹配算法。

要求：① 编程实现三种字符串匹配算法。②分析三种算法的时间复杂性，通过应用三种算法在一个较长英文文本中查找一个子串的实验数据来对比三种算法的运行时间。

## 长整数四则运算

任务：设计数据结构完成长整数的表示和存储，并编写算法来实现两长整数的加、减、乘、除等基本代数运算。

要求：① 长整数长度在一百位以上。② 实现两长整数在取余操作下的加、减、乘、除操作，即实现算法来求解 a+b mod n, a-b mod n, ab mod n, ab mod n 。③输入输出均在文件中。④分析算法的时空复杂性。实现提示：需将长整数的加法转化为多个一般整数加法的组合。

## 索引文件

有若干学生的成绩数据如下，将这些数据保存到st数组中：

学号 姓名 年龄 性别 语文 数学 英语

1 陈华 20 男 78 90 84

5 张明 21 男 78 68 92

8 王英 20 女 86 81 86

3 刘丽 21 女 78 92 88

2 许可 20 男 80 83 78

4 陈军 20 男 78 88 82

7 马胜 21 男 56 67 75

基于这些数据，编程序实现下面的功能：

(1)将st数组中学生记录写入stud.dat文件作为主文件

(2)输出主文件中的学生记录

(3)建立与主文件相对应的索引文件，其中每个记录由两个字段组成：学号和该学生记录在数据文件中的位移量（例：上面列出数据中学号为2的学生许可，其位移量是5。由于本例用定长文件，可以根据位移量计算记录在主文件中的相对地址）。索引文件按学号有序。

(4)输出索引文件中的全部记录

(5)根据用户输入的学号，利用索引文件，用二分查找法找到对应的记录号，再通过主文件输出该记录。

(6) 再次采用二叉平衡树实现索引。

## 首次适应算法FF

任务：实现操作系统中存储分配的空闲分区链算法：首次适应算法FF。在此算法中，空闲区链按起始地址递增顺序排列，在进行内存分配时，从链首开始顺序查找，直到找到一个能满足其大小要求的空闲区为止。然后，再按照作业的大小，从该分区划出一块内存给请求者，余下的空闲分区仍留在空闲链中。本算法的特点是底地址端造留许多碎片，高地址端集中了大空闲区。这样有利于给后到的大作业分配大的内存空间。

要求： 模拟实现分配过程。

## 基于Floyd算法的多重最短路问题的改进算法

任务：路径分析是网络分析最基本的问题,其核心是对最短路径的求解。Floyd算法是一种求取最短路的经典算法。查阅相关文献，例如<http://xueshu.baidu.com/usercenter/paper/show?paperid=2f1103fbf3fe0dd8f508be7f0555c9cf&site=xueshu_se>或<http://www.doc88.com/p-9012008607486.html>，实现算法的改进。

要求： 实现该算法，并与原算法比较。

## 随时找到数据流中的中位数

任务：有一个源源不断地吐出整数的数据流，假设你有足够的空间来保存吐出的数。请设计一个名叫MedianHolder的结构，MedianHolder可以随时取得之前吐出所有数的中位数。

要求： 1）如果MedianHolder已经保存了吐出的N个数，那么任意时刻将一个新数加入到MedianHolder的过程，其时间复杂度是O(logN)。 2）取得已经吐出的N个数整体的中位数的过程，时间复杂度为O(1)。3）为了做到“任何时刻”的要求，那么堆的设计不能采用固定数组的实现方式，因为会有扩容的负担。为了解决这个问题，需要设计一个没有扩容负担的堆结构。参考：<https://blog.csdn.net/zuochao_2013/article/details/80017720>，https://blog.csdn.net/qq\_34342154/article/details/78454581