**《应用数据结构》课程设计指导书**

**课程编号：**4170251110

**课程名称：**应用数据结构/Applied Data Structure

**周数/学分：**1/1

**先修课程：**应用数据结构

**适应专业：**工科类

**开课学院、系或教研室：**管理学院信息管理与信息系统系

# 一、目的和要求

## 1．目的

课程设计是对学生的一种全面综合训练，是与课堂听讲、自学和练习相辅相成的必不可少的一个教学环节。通常，设计题中的问题比平时的习题复杂得多，也更接近实际。设计着眼于原理与应用的结合，使学生学会如何把书上学到的知识用于解决实际问题，培养软件工作所需要的动手能力；另一方面，能使书上的知识变“活”，起到深化理解和灵活掌握教学内容的目的。平时的练习较偏重于如何编写功能单一的小算法，而设计题是软件设计的综合训练，包括问题分析、总体结构设计、用户界面设计、程序设计基本技能和技巧，多人合作，以致一整套软件工作规范的训练和科学作风的培养。此外，还有很重要的一点是：机器是比任何教师都严厉的检查者。

## 2．要求

为了达到上述目的，本课程安排了若干设计题目，各题的训练重点在于基本的数据结构，而不强调面面俱到。这些题目与教科书的各章只具有粗略的对应关系，一个设计题常常涉及几部分教学内容。根据本课程的授课对象，所有题目的难度相比计算机专业有所降低，但仍有一定的系数保证。

每个设计题采取了统一的格式，由问题描述、基本要求、输入输出、实现提示、选做内容五个部分组成。问题描述旨在为读者建立问题提出的背景环境，指明问题“是什么”。基本要求则对问题进一步求精，划出问题的边界，指出具体的参量或前提条件，并规定该题的最低限度要求。输入输出旨在为检查学生上机作业提供方便，在完成设计题时应自己设计完整和严格的测试方案，当数据输入量较大时，提倡以文件形式向程序提供输入数据。在实现提示部分，对实现中的难点及其解法思路等问题作了简要提示。选做部分向那些尚有余力的读者提出了更严峻的挑战，同时也能开拓其他读者的思路，在完成基本要求时力求避免就事论事的不良思想方法，尽可能寻求具有普遍意义的解法，使得程序结构合理，容易修改扩充。

本设计题目编排的策略是：一方面使其难度和工作量有保证，另一方面给学生提供的辅助和可以模仿的成分也较多。当然，还应指出的是，提示的实现方法未必是最好的，学生不应拘泥于此，而应努力开发更好的方法和结构。

# 二、课程设计内容

课程设计总共有不少于十道题目，学生可以任选其中一道。

**特别说明：所有在课程设计指导书上列出的题目，每个人均须选择一道题独立完成，不可分工合作；每道题每个班选做的人数分配额度由授课老师决定。**

## QQ群重复成员查找

### 【问题描述】

QQ群是聚集相似人群非常有用的工具。然而群容量是有限的，当一个群人数满了以后，群主往往需要开设一个新群来容纳更多的人。当一个群主拥有了至少两个或以上的相同主题的QQ群，并且人数达到一定规模的时候，问题就会显现出来。

人心总是很好奇也很贪婪的，当一个普通群员发现群主还有另外一个相同主题的群，总是会产生进去探索一番的心理，于是就会出现多个QQ群里有很多重复成员的情形。有些情况下，重复成员是必要的，比如管理员、特邀嘉宾等；但更多时候，群主可能并不希望同一个人用相同的QQ号同时占据多个群的名额，因此他需要把这些“不小心”放进去的人找出来，然后……。

找出这些重复人员并不是一件容易的事情，尤其是当两个群的人数都很多时。现在你的任务就是帮助苦恼的群主解决这个难题。腾讯公司并没有提供群成员名单的导出功能，好在我们有另外的途径来获得。具体方法请参见附录一。

必须提醒的是，导出的文件并没有使用自动换行，但是从图 4中可以看出，某些成员的QQ号被强制分成两行显示了。你得首先弄清楚这是怎么回事，并找到解决对策。特别提醒，该文件的内容不允许被更改！亦即你的程序必须要有办法找到第一个群成员开始的地方并正确辨别需要的信息，答辩时将由老师使用自备数据检验。

### 【基本要求】

1. 以自己同时加入的两个QQ群作为数据来源。这些群最好是同一个群主为了同一个主题而开设的，如果没有，可以使用老师提供的数据。
2. 群成员由QQ号唯一确定。
3. 找出的重复成员名单输出到一个文本文件中保存，文件名自定。
4. 如何定位第一个群成员可以参考串的模式匹配算法，要求必须使用KMP算法。
5. 对于任意一个群，都能按照根据群名片和QQ号两种方式从小到大排序输出全体成员，且结果保存在文本文件中便于查看。

### 【输入输出】

输入数据从保存的群名单文件中获得。

输出形式为：（按QQ号升序排列）

重复QQ号 x群名称 x群身份 y群名称 y群身份

1111111 路人甲 无 路人乙 群主

243160620 周锐　@xg1101 x群管理员 1101-周 锐 无

470061490 杜军　@xg0903 无 流氓杜 y群管理员

724824868 燕翔 群主 大學有道 无

…… …… …… …… ……

若没有重复的，则输出“找不到”。

### 【实现提示】

由于无法确知有多少人会重复，所以所有搜索到的QQ号都必须存储起来，以便比对。此时可以考虑使用散列存储方式。尽管群内成员总数可以视为未知，但群成员容量有上限（目前腾讯允许的群成员总人数最大值为2000人，哈希函数所需要的向量空间大小可以参考这些信息）。查找的方法自定。

### 【选做内容】

## QQ群成员发言频率统计

### 【问题描述】

QQ群是用来和多人讨论共同话题的良好平台。然而，当群的人数达到一定程度时，总会有很多人变成潜水党。尽管腾讯提供了所有群员最后一次发言的时间，并且最近还推出了“群成员等级头衔”等发言频度评价体系。但是作为群主，往往还要关心更多的数据，例如，哪些人发言的次数最多，哪些人发言的次数最少。而这需要有尽可能完整的聊天历史记录来提供原始数据。不幸的是，腾讯最近取消了这一功能。因此，只要某个群成员拥有最完整的群聊天历史记录，我们也就能依据它进行相关统计，从而得到关心的结果。

首先，我们需要导出一个群的聊天记录，具体方法参见附录二。

由于是聊天记录，自动换行与否已经不重要。如何搜索聊天记录才是需要你关心的问题。特别提醒，导出的聊天记录文件内容不允许被更改！亦即你的程序必须要有办法找到第一个聊天记录开始的地方并正确辨别需要的信息，答辩时将由老师使用自备数据检验。

### 【基本要求】

1. 以自己加入的某个QQ群作为数据来源。如果没有，可以使用老师提供的数据。由于某些群成员入群后可能从未发言，或者因为某种原因曾经发过言的群成员聊天记录缺失，或者留有发言记录的某个群成员现在已经不在群内等等因素，本题只统计所有出现在聊天记录中的成员，并假设他们就是群的所有成员。群成员由QQ号唯一确定。
2. 统计所有群成员（不含10000号）的发言记录条数和最近发言时间，并能根据发言次数按照升序和降序两种方式排列，并输出到文本文件中保存，以便查看。文件名自定。
3. 按0:00～5:59、6:00～11:59、12:00～17:59、18:00～23:59四个时段进行统计，计算每个发言者在这四个时段发言的次数占其总发言数的百分比。结果既可以完整输出到文本文件（按QQ号大小升序排列显示），也能单独指定某个QQ号进行查看。
4. 按上述四个时段，从第一条聊天记录开始到最后一条记录，按天统计所有发言者的人数（不是次数）。最后求出平均每天每时段的发言者人数

### 【输入输出】

输入数据从保存的群聊记录文件中获得。

输出形式为（样例）：

1. 群成员发言记录条数统计（按发言记录条数升序显示）  
    群成员QQ号 网名 发言次数 最近发言时间  
    123456789 样例 1 2010-01-07  
    …… …… …… ……
2. 按时段统计每个人发言总条数和百分比（按QQ号升序显示）  
    QQ号 网名 0:00～5:59 6:00～11:59 12:00～17:59 18:00～23:59  
   1234567 张三 16,0.61% 23,0.87% 946,35.96% 1646,62.56%  
   …… …… …… …… …… ……
3. 按时段统计发言者人数（不是发言条数）  
   时段 0:00～5:59 6:00～11:59 12:00～17:59 18:00～23:59  
   活跃人数 65 174 375 1238

### 【实现提示】

由于无法确知有多少人会发言，所以所有搜索到的QQ号都必须存储起来（注意，QQ号有长有短，且10000号须排除在外），以便比对。此时可以考虑使用散列存储方式。尽管群内成员总数可以视为未知，但群成员容量有上限（目前腾讯允许的群成员总人数最大值为2000人）。这样，哈希函数所需要的向量空间大小可以参考这些信息。

值得注意的是，本题并不关心聊天的具体内容，而只注重发言者QQ号以及相关时间，因此在统计每个成员的相关信息时，可以参考数据库技术中的实体属性图方式，若将每个QQ号视为实体，事先把涉及到的属性罗列出来，然后精心构造结构体，这样在统计时可以省去很多麻烦。

### 【选做内容】

1. 将所有未发言者以及有最后发言时间（以腾讯显示时间为准）但无发言记录者纳入统计样本空间。
2. 把上述的四个时段细分为24个时段，精确统计所有天数每一个时段群内发言者的人数（不是次数），并输出直方图。

## QQ群聊天记录过滤与甄别

### 【问题描述】

每个资深的QQ群管理员都有一个共同的烦恼，就是当自己的群稍微有点名气，人数略微多了一些，各种网络骗子广告人士蜂拥而至，把个好端端的群弄的乌烟瘴气。尤其是在群管理员不在的时候，很容易失控。于是，监控群聊天记录就成了管理员非常重视的一项日常工作。

为了弥补群管不在线时对聊天动态监管不力的影响，事后对记录进行分析和过滤也是亡羊补牢的手段之一。因此，对群聊记录的分析需求应运而生。然而，当群聊记录的条数海量增加以后，人工查找的工作也变得不太现实。于是，针对敏感词的过滤程序就成了必要的手段之一。而这一切，同样需要有完善的群聊天记录作为保证。

由于这只是课程设计，所以我们不打算按照真实情况将各种真正的敏感词列举出来去进行实际检测，而是假设一些常见词作为验证手段。为此，我们需要设立一份“敏感词清单”。

经常发送带有“敏感词”的人应该受到必要的警告或更严厉的惩罚，因此，对于曾经有前科的发言者，我们需要建立不良言论记录档案，亦即所谓的“黑名单”。这份黑名单应该包括QQ号（不列出网名是因为群名片可能经常发生变化）、犯规次数和最后一次犯规的日期。然后，根据黑名单里的历史记录和最新的检测结果决定谁将接受更为严厉的处罚。例如，假设某成员在2013年6月1日前已经有过两次发送“敏感词”的记录，经过检测，发现其在2013年6月1日以后又有两次新的违禁记录，那么程序就应该能及时做出提示。

首先，我们需要导出一个群的聊天记录，具体方法参见附录二：

特别提醒，导出的聊天记录文件内容不允许被更改！亦即你的程序必须要有办法对各种可能的聊天内容进行甄别辨识，答辩时将由老师使用自备数据检验。

### 【基本要求】

1. 以自己加入的某个QQ群作为数据来源。如果没有，可以使用老师提供的数据。群成员由QQ号唯一确定。
2. 通过搜索整个文件，分别找出发送图片、表情、超链接和超长内容（例如，只对单次发言字符超过100的进行统计，如何处理汉字和西文字符自行斟酌）最多的QQ号
3. 自定义一份虚拟的“敏感词”清单，保存在文本文件filter.txt中，然后以此为依据检测整个聊天记录，并根据指定的日期输出在此日期之前的“违禁”人员名单，同样保存到文本文件blcklst.txt中。
4. 以上一条要求中的日期为界限，检测该日期之后发生的违禁事件，并提示新的违禁者和有前科者的再次违禁记录，并将结果更新到黑名单中。

### 【输入输出】

输入数据从保存的群聊记录文件中获得。

输出形式为：

1. 聊天分类记录：  
   发送图片最多者 发送总次数  
   12345 654  
   发送表情最多者 发送总次数  
   67890 1357  
   发送超链接最多者 发送总次数  
   13579 2254  
   发送超长内容最多者 发送总次数  
   24680 369
2. 敏感词清单（文件filter.txt内容，可自定义）：  
   呵呵  
   老师  
   C语言  
   ……  
     
   违禁人员黑名单（文件blcklist.txt内容）：  
   截至2014.05.31  
   QQ号 犯规次数 最后犯规日期  
   123456 2 2014.03.28  
   …… …… ……
3. 最新检测结果（按QQ号升序排列）：  
   自2013.06.01至今  
   QQ号 前科次数 犯规次数 最后犯规日期  
   111111111 0 2 2013.06.25  
   724824868 2 3 2013.07.18  
   …… …… …… ……

### 【实现提示】

本题难点在于通过模式匹配搜索到敏感词后，需要返回搜索该敏感词的发送者以及发送时间，这当然可以通过逆向搜索来实现，但不如先记录发言者位置，再匹配发言内容更有效。

聊天内容中的图片和表情都有方括号括起，超链接也可以通过http来简单辨识（虽然很容易把通过Web QQ和手机QQ发送的成员错误统计进去），只有超长内容可能会遇到麻烦。如果搜索支持unicode，那么一个汉字可以当成一个字符来处理，但更普遍的情形是一个汉字按照两个字节来处理。具体如何操作，可根据实际情况决定。

表面上看来不涉及大量的QQ号存储，但实际上是有可能的。因为我们的敏感词清单不是正式的真正敏感词，而是常规词汇，有可能导致大量QQ号中招，因此，仍然需要考虑采用Hash存储。当然，你也可以用真正的敏感词（诸如“我勒个去”之类）来检验。

基本要求4的实现需要有要求3的成果，因此要读取blacklist文件内容并更新。然后在屏幕上按照输入输出第3条的形式显示。

### 【选做内容】

1. 找出通过手机发送消息最多的QQ号（使用手机发送的消息末尾会有“(来自手机QQ： http://mobile.qq.com/c )”的提示）。
2. 找出通过Web QQ发送消息最多的QQ号（使用Web QQ发送的消息末尾会有“正在使用WebQQ：”或者“正在使用Q+ Web：”的提示）

## 最便宜航空线路求解

### 【问题描述】

飞机是远途旅行必不可少的工具，但是高昂的价格也令人颇费思量，因此人们往往宁愿用时间来换金钱而选择较慢的交通工具。不过，当航班线路变得丰富之后，人们也发现很多时候选择直达航班往往并非最省钱的路径，比如飞往美国选择从韩国转机就比直飞要便宜多了（虽然这样有坠机的危险）。这样一来，当采用曲线行进能达到省钱也省时间的目的时，坐飞机还是可以接受的。

为了能够更好地解决路径的选择问题，我们模拟一下通过合适的算法来选出最优航空线路。参考<http://flight.qunar.com/>上的数据，自选至少10个城市和至少20条航线（不含中转），要求任意两城市间都可达（这并不意味着两个城市之间就一定有直飞航班，比如大陆到台湾），且至少有2条可选飞行路线。预先定义每条航线的最高定价，航线票价每季度都有折扣机会，但并非必须。每条线路每季度的折扣率由随机函数产生，范围是0.20～1.0（小数点后取两位）。

本程序不考虑机场建设费和燃油附加费，只计算机票价格，将其作为路径上的权值处理，耗费矩阵存储结构自行选择。在真实的情形中，同一线路每天可能有许多航班，也可能每周只有一两趟航班，即便同一天的航班早晚价格也可能有所不同。为简便计，假定每条线路每天只有一趟航班，不考虑航空公司的差别。

### 【基本要求】

1. 所有原始数据一律事先用文本文件保存，读入数据时从文件获取。
2. 航班价格可以用随机函数生成，但建议从网上直接复制真实数据。
3. 可输出所有城市名称以及每一城市所有可直飞到达的城市列表。
4. 可输出任一指定直飞线路（若没有则显示无直飞航班）四个季度的全票和折扣价格
5. 输入月份、起始城市和目的城市名称，程序能显示出直飞（如果有）以及最经济的中转路线选择方案。

### 【输入输出】

从文件中读取输入，具体格式自行决定。

1. 输出所有城市名以及每一城市所有可直飞到达的城市列表（按城市名拼音升序排列）：  
   始发地 可直飞目的地  
   北京 长春 广州 杭州 昆明 上海 首尔 武汉 香港  
   ……  
   首尔 北京 上海 台北 香港  
   ……
2. 指定线路折扣和全价：  
   输入：北京-台北  
   输出：无直飞航班  
   输入：北京-广州  
   输出：季度 全票价格 折扣 折后价格（不计小数，四舍五入）  
    1 1240 6.5 806  
    2 1192 7.0 834  
    3 1475 4.6 679  
    4 1230 8.7 1070
3. 输入月份、起始城市和目的城市名称，输出直飞和最佳方案  
   输入：5 北京-台北  
   输出：无直飞航班，最佳线路：北京-香港，1503，香港-台北，791，合计2294  
   输入：7 广州-首尔  
   输出：直飞，2460，最佳线路：广州-上海，880，上海-首尔，773，合计1653  
   输入：10 武汉-上海  
   输出：直飞，278，最佳线路：武汉-上海，278，合计278  
   ……

### 【实现提示】

考虑到航线不太可能有单向情况，可将图简化成为无向图，但必须保证所有结点连通，所有线路（边）可以采用结构体形式保存其全价和随机产生的折扣，四个季度可以有不同的折扣和全价。可以考虑在折扣全部产生后直接使用Floyd算法。

本题相对简单，学霸学神们敬请避开，为基础不太好的同学留条生路。

### 【选做内容】

若增加不同航空公司的情况，考虑当不同航空公司的最高价格和折扣情况在同一条航线上不同时，程序能给出更合理的选择结果。

## 超长数字串

### 【问题描述】

给一个数字串STR：123456789101112131415161718192021222324……它是由所有自然数从小到大依次排列起来的。任意给一个数字串T，容易知道它一定在STR中出现无穷多次。现要求任意给一个数字n作为串T，并按上述规则将从1～n的所有数字连续串在一起组成一个完整的串S。求出串T在串S中所有出现的位置。例如，若n=81，则从1到81的所有自然数连在一起构成串S，而“81”首次出现的位置为27。你的任务是找出此串中“81”出现的所有位置和出现次数。最后，给出一份列表，列出n从1到100时，所有字符串值为n的出现首位置和次数。

### 【基本要求】

1. 输入n值后首先计算出串S的长度L，计算公式的理论推导自行完成。
2. 串S的存储采用动态内存分配方式，并且要求将串指针和串长变量封装在一起。
3. 程序需要实现两个功能（可通过按键选择实现），其一为能够接受从键盘上输入的n值，并以此为依据生成串S，然后输出串值为n的子串在S中出现的所有位置和次数。其二为通过循环计算出n从1到100时，所有字符串值为n的出现首位置和次数，并将结果存储在一个文本文件中。

### 【输入输出】

对于第一个功能，输入输出形式为：

输入数据：输入数值n：12

输出数据：在从1到12所构成的串中，12出现的位置分别是：1，14。共出现2次。

对于第二个功能，输出形式为：

n 出现首位置 出现次数

1 1 1

2 2 1

…… …… ……

21 15 2

…… …… ……

### 【实现提示】

本题相对简单，学霸学神们敬请避开，为基础不太好的同学留条生路。

### 【选做内容】

将上述构成串S的规则从顺序改为逆序。

## Huffman编／译码器

### 【问题描述】

利用Huffman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，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）。对于双工信道（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Huffman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：

（l）I：初始化。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。利用已建好的Huffman树（如不在内存，则从文件hfmTree中读入），对文件ToBeTran中的正文进行编码，然后将结果存入文件CodeFile中。

（3）D：译码。利用已建好的Huffman树将文件CodeFile中的代码进行译码，结果存入文件TextFile中。

（4）P：印代码文件。将文件CodeFile以紧凑格式显示在终端上，每行50 个代码。

（5）T：印哈夫曼树。将已在内存中的哈夫曼树以直观的方式（树或凹入表形式）显示在终端上，同时将此字符形式的哈夫曼树写入文件TreePrint中。

### 【输入输出】

（l）利用教材中的数据或自行选择一份英文文本材料调试程序。

（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，并实现以下报文的编码和译码："THIS PROGRAM IS MY FAVORITE"。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字符 |  | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 频度 | 186 | 64 | 13 | 22 | 32 | 103 | 21 | 15 | 47 | 57 | 1 | 5 | 32 | 20 | 57 | 63 | 15 | 1 | 48 | 51 | 80 | 23 | 8 | 18 | 1 | 16 | 1 |

### 【实现提示】

（1）文件CodeFile的基类型可以设为子界型bit=0..1。

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

（3）在程序的一次执行过程中，第一次执行I,D或C命令之后，Huffman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，因为文件hfmTree可能早已建好。

### 【选做内容】

（1）上述CodeFile的基类型实际上可能占用了存放一个整数的空间，只起到示意或模拟的作用。现使CodeFile的基类型package=integer，把Huffman码紧缩到一个整形变量中去，最大限度地利用码点存储能力，试改写此系统。

（2）修改系统，实现对系统的原程序的编码和译码（主要是行尾符编/译码问题）。

（3）实现各个转换操作的源/目文件均由用户在选择此操作时指定。

## 钓鱼

### 【问题描述】

在一条水平路边，有n（2 ≤ n ≤ 25）个钓鱼池，从左到右编号为1、2、3、……、n。小明有H（1 ≤ H ≤ 16）个小时的空余时间，他希望用这些时间钓到尽量多的鱼。他从池塘1出发向池塘n走，有选择地在一些池塘边停留一定的时间钓鱼，最后在某一个池边结束钓鱼。小明测出从第i个池塘到第i+1个池塘需要走5T*i*分钟的路。还测出在第i个池边停留，第一个5分钟可以钓到鱼量F*i*，以后再每钓5分钟鱼，鱼量减少D*i*。假设没有其他人钓鱼，也没有其它因素影响小明钓到期望数量的鱼，请编程求出能钓最多鱼的方案。

### 【基本要求】

很显然，在分析本题数据前你需要首先构造出数据，这里我们给出一个示例：

|  |  |  |  |
| --- | --- | --- | --- |
| 池塘编号 | 从上一池塘到本池塘的行程时间T | 首个5分钟可以钓到的鱼量F | 每隔五分钟减少的鱼量D |
| 1 | 2 | 14 | 4 |
| 2 | 1 | 3 | 1 |
| 3 | 3 | 20 | 6 |
| 4 | 1 | 11 | 2 |
| 5 | 4 | 9 | 2 |
| 6 | 2 | 23 | 11 |
| …… | …… | …… | …… |

例如，小明从起点出发到1号池塘，路途需要花费5T1=10分钟，在此停留15分钟，第一个5分钟可以钓到14条鱼，第二个5分钟可以钓到14-4=10条鱼，第三个5分钟可以钓到10-4=6条鱼。于是小明在第一个池塘总计钓到30条鱼，花费25分钟。然后他可能认为第二个池塘不值得停留，于是越过第二个池塘，直接走向第三个池塘，路途花费5T2+5T3=5\*1+5\*3=20分钟，然后在第三个池塘停留10分钟，钓到20+(20-6)=34条鱼。……

究竟在应该每个池塘停留多久？这正是需要你通过算法进行计算的。上述表格的具体数值可以通过随机函数生成，但你得注意不要让F和D的差值太过离谱。

另外，由于输出内容较多，一屏往往显示不完，所以输出结果必须保存到文本文件中以便单独查看。

### 【输入输出】

总过有多少个池塘，以及小明总共有多少时间，可以在程序运行时从键盘输入。上述表格数据应提供两种方式输入：1)通过随机函数生成，且可以二进制形式保存到用于输入的文件中；2)通过输入文件读入。

经过计算以后的输出格式（示例）为：

小明拥有的时间：10小时，合计600分钟

池塘总数：18

各池塘参数：

池塘编号 行程时间 首次鱼量 递减鱼量

1 2 14 4

2 1 3 1

……

具体过程：

抵达1号池塘：花费时间10分钟，停留时间15分钟，获得鱼量14+(14-4)+(14-2\*4)=30

抵达2号池塘：花费时间5分钟，停留时间0分钟，获得鱼量0

抵达3号池塘：花费时间15分钟，停留时间10分钟，获得鱼量20+(20-6)=34

……

最终统计结果，小明路途花费合计xxx分钟，在各池塘停留合计yyy分钟，总计耗时zzz分钟，获得鱼量合计www条。

### 【实现提示】

从题目可以看出，小明每做一件事情所花费的时间都是5分钟为一个基本单位，因此可以直接用5分钟作为一个基本计时单位来简化处理方式。尽管各个池塘的可钓鱼量并不均等，但我们可以假定这些池塘是按照鱼量从大到小排列的（想想看为什么？），然后枚举这些池塘每一个可操作时段的理论收获值，并通过贪心算法来计算最大可钓鱼量。亦即可以每次选一个鱼最多的池塘钓一次鱼，对于每个池塘来说，由于在任何时候鱼的数目只和小明在该池塘钓鱼的时长有关，而和钓鱼的总时长无关，所以这个策略一定是最优的。问题只在于，每个池塘是否应该一直钓到无鱼可钓为止？这就是需要你动脑筋去思考的问题了。如果这个问题你能找到一条清晰的思路来解决，将对你的编程思维模式起到极好的锻炼作用。

### 【选做内容】

## 锁链

### 【问题描述】

一个国王因为听信谗言将一个无辜的数学家关进了监狱。虽然事后发现确属冤枉，但碍于面子，国王不肯认错。为了挽回，于是国王决定用Bytish锁链将其锁在墙上。这种锁链由n（10≤n≤1000）个固定在墙上的铁环和铁棒组成。由于环不是都套在棒上，要想把整副锁链取下是十分困难的。数学家必须自己通过不断取下和套上铁环最终将所有铁环都取下才能获得自由。取下或套上铁环的规则是：

* 铁环从1、2、……、n依次编号。
* 一次只能把一个环取下或套上。
* 编号为1的环无论何时都能取下或套上。
* 如果编号为1、……、k-1（1≤k≤n）的环已经从棒上取下，并且k环套在棒上，则可以取下或套上编号为k+1的环。

写一个程序，读入锁链描述并计算从棒上取下所有环所需的最少步数。

### 【基本要求】

显然，可以运用递归的方法解决此问题。但是你能否找到一个非递归算法呢？

### 【输入输出】

输入：环的总数n。

输出：为尽量体现程序输出结果的层次，可以按照从n、n-1、n-2、……、1的顺序，将移除掉n号环的全部过程作为一个段落输出，然后将移除n-1号环的全部过程也作为一个段落输出，其余依此类推。

### 【实现提示】

显然，通过枚举前i个环的解锁过程可以从中找出解题的思路。

当n=1时，直接移除即得解。

当n=2时，显然不能先移出1环，再移除2环。因为根据约束条件，必须1环在棒上才能移除2环。因此移除的步骤应该是先2环，再1环。

当n=3时，移除3环的条件是2环在，1环卸，于是可以先移除1环，然后移除3环，然后套上1环，然后移除2环和1环。

当n≥4的情形请自行推导。

本题相对简单，学霸学神们敬请避开，为基础不太好的同学留条生路。

### 【选做内容】

## 蚂蚁和瓢虫

### 【问题描述】

蚂蚁和蚜虫是共生的。蚜虫分泌出蜜汁给蚂蚁食用，蚂蚁帮助蚜虫赶走天敌瓢虫。在蚂蚁山附近有一棵树，这里是蚜虫生活的地方。有n只蚂蚁兵，用1到n编号。一只瓢虫威胁着这个小小的生态系统，它经常出现在蚜虫活动的地方。当瓢虫落到树上时，蚂蚁兵会出动把它赶走。它们按照如下规则移动：

1. 树上任意两点之间都只有一条路径，所有的蚂蚁都能同时发现瓢虫的降临，因此都会沿着它所在点到瓢虫的路径前进。每移动一个位置，花费1个时间单位。
2. 若瓢虫降落的地点恰好有一只蚂蚁，则该蚂蚁立刻赶走瓢虫，并记录其赶走瓢虫次数加1。
3. 如果某只蚂蚁的路径上有另外一只蚂蚁，那么距离目标较远的蚂蚁待在原地不动，较近的那只蚂蚁继续前进。
4. 如有多只蚂蚁要进入同一个结点，那么选择编号最小的蚂蚁，其余的蚂蚁留在原位置不动。
5. 当蚂蚁到达了瓢虫的位置后，把它赶走，然后停留在该位置，并记录其赶走瓢虫次数加1。

若整个系统仅有一只瓢虫，且设树的总结点数m≥2n，试模拟一下此生态系统的运行。

### 【基本要求】

1. 用清晰明确的语言表述树的存储方法。
2. 将事件按照事件单位分段模拟。注意：同一个结点任一时刻只能有一只生物，或蚂蚁、或瓢虫。
3. 瓢虫的出现事件通过随机函数实现。
4. 用清晰明确的语言表述算法的实现原理或机制。
5. 将规则3的条件放宽为，当瓢虫降落后，所有蚂蚁都先观察自己前进的路途前方有无别的蚂蚁，如果有则自己原地不动，只有当自己前方没有别的蚂蚁时才会向瓢虫所在结点移动。
6. 将输出结果保存到文本文件中。

### 【输入输出】

输入：1）蚂蚁兵的数量ants；2）树的结点数nodes。

输出：初始状态，时间为0，生成树形结构并显示树结构。

按照时间段从1开始，显示瓢虫出现的结点编号，以及发生移动的蚂蚁编号及其移动路径，直到瓢虫被赶走。例如：

时间点1：瓢虫出现在结点35。

时间点2：蚂蚁9从结点15移向结点18，蚂蚁13从结点34移向结点2，蚂蚁17从结点16移向结点2，因有编号更小的蚂蚁同时移向结点2，故蚂蚁17原位不动。

……

时间点i：蚂蚁9从结点34移向结点35，瓢虫被赶走。蚂蚁9记功一次。

时间点i+1：瓢虫出现在结点27。

……

时间点n：运行结束（确保结束时没有瓢虫在树上）。

统计结果：总共出现瓢虫x次，蚂蚁9赶走5只瓢虫，蚂蚁16赶走3只瓢虫，……

### 【实现提示】

由于这是一棵任意树，因此树的存储方法至关重要。我们可以确定的是这棵树没有回路，因此将瓢虫出现的结点作为根是个不错的主意。可以考虑通过左孩子右兄弟法来转换此树。可以借用图的广度优先遍历思路来解决此问题。

### 【选做内容】

1. 修改规则3，令所有蚂蚁在瓢虫降落后都同时移动，仅当蚂蚁前进的路上相邻结点有另一只蚂蚁时才停止移动。
2. 若允许最多3只瓢虫同时出现，算法应如何改进？

## 网线编号

### 【问题描述】

网络工程师们刚刚为大楼布好了网线，可由于工作疏忽，从机房到办公室的网线有交叉。在机房，所有网线从左到右编号为1，2，3，……，N，而在办公室，网线从左到右的顺序出现了混乱，无法立刻确定其线序。好在他们发现两条线最多只相交一次，而且可以检测出任意两条线是否相交。如下左图，实线表示两条线交叉，虚线表示不交叉。现给出每两条线是否交叉的信息，请计算办公室内从左到右各条线的编号。例如，从下左图的信息可以确定实际布线情况如下右图所示，编号依次为3，2，4，1，5。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 3 | 2 | 4 | 1 | 5 |

1

2

3

4

5

### 【基本要求】

网线相交情况严格符合定义——即两条线最多只相交一次。且最终的结果必须是物理上可以实现的，亦即你必须弄清楚左图中那些实线与实线、实线与虚线之间的交叉关系。请以清晰明确的语言阐述你研究的结论。线路的数量通过键盘输入，交叉的情况通过算法（尽量）采用随机函数生成。交叉的情况应通过矩阵形式输出，最后通过算法输出上下各端口的实际对应情况。

### 【输入输出】

输入：端口数ports

输出：交叉情况矩阵，线路对应表。

### 【实现提示】

两条线最多只相交一次，如果导线x和导线y相交，且已知x<y，则在办公室端导线x排在y的后面。反之，如果不相交，则x排在y的前面。这样，根据导线两两是否相交的信息，可以知道导线两两之间的先后顺序。有了这些顺序，剩下的工作就是拓扑排序了。

### 【选做内容】

## 可怜的奶牛

### 【问题描述】

农夫John有n（n≤10000）头奶牛。可是由于它们产的奶太少，农夫对它们很不满意，决定每天把产奶最少的一头做成牛肉干吃掉。但还是有一点舍不得，John打算,如果有不止一头奶牛产奶最少，当天就大发慈悲，放过所有的牛。

由于John的奶牛产奶是周期性的，John在一开始就能了解所有牛的最终命运。不过他的数学很差，所以请你帮帮忙，算算最后有多少头奶牛可以幸免于难。每头奶牛的产奶周期Ti可能不同，但不会超过10，也不会小于1。在每个周期中，奶牛每天产奶量不超过100。

### 【基本要求】

本题核心是每次取一个最小元素，不过，由于元素有周期性变化，所以不能把它们直接组织成一个堆。由于周期也不相同，只知道周期最大不超过10天，所以模拟的天数可能需要至少2520天（这是1、2、……、10的最小公倍数）。

1. 生成奶牛参数表，包括每头牛的周期数以及它们在这些周期内每天的产奶量。考虑到数量比较大，要求用随机函数生成。
2. 以天为单位，模拟一个合理的时间段（例如2520）天，显示每天被屠宰的牛编号或者因为有相同的牛产奶最少而大赦牛棚。

### 【输入输出】

输入：牛的总数cows，需要模拟的天数days。

输出：将随机生成的牛参数输出到一个文本文件中，要求格式为

周期为1的奶牛：编号 第1天（产奶量，下同）

45 34

97 56

…… ……

周期为2的奶牛：编号 第1天 第2天

243 7 39

651 75 37

…… …… ……

…… …… …… ……

周期为6的奶牛：编号 第1天 第2天 第3天 第4天 第5天 第6天

167 5 7 48 76 99 37

617 18 26 67 87 75 29

…… …… …… …… …… …… …… ……

然后输出从第1天开始的模拟结果（也需要输出到另一个文本文件中）：

第1天，产奶最少的牛编号是：…… x号牛被屠宰或没有牛被屠宰

第2天，产奶最少的牛编号是：…… y号牛被屠宰或没有牛被屠宰

……

统计：总共被宰掉xx头奶牛，到模拟结束，总共有yy天没有发生奶牛被屠宰事件。

### 【实现提示】

周期同为t的奶牛在未被吃掉以前，每天的最小产奶量也是以t为周期的。因此如果把周期相同的奶牛合并起来，每天只需要比较10类奶牛中每类牛的最小产奶量就可以了。例如：

周期为6的奶牛参数表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 牛编号 | 第1天 | 第2天 | 第3天 | 第4天 | 第5天 | 第6天 |
| 167 | 5 | 7 | 48 | 76 | 99 | 37 |
| 617 | 18 | 26 | 67 | 87 | 75 | 29 |
| 284 | 51 | 1 | 33 | 76 | 54 | 47 |
| 486 | 29 | 36 | 33 | 88 | 28 | 41 |
| 合并结果 | 5（牛167） | 1（牛284） | 33（多牛） | 76（多牛） | 28（牛486） | 29（牛617） |

这样，在和其它组比较时，如果是第1天，就把牛167拿去比较即可。如果是第3天，因为多头牛一样小，则此组中必然不会有牛被宰杀，于是此组数据可以免于比较。这样，每次只需要比较k组牛的“代表”即可。

只要周期为6的牛都不被吃掉，这个表就是一直有效的。但是在吃掉一头奶牛后，需要修改此表，使它仍然记录每天的产奶量。可以重新计算此表，但复杂度较高；也可以把一个周期中每天的产奶量组织成堆，每次删除操作后执行堆排序，则可降低复杂度。

### 【选做内容】

## 友好城市

### 【问题描述】

某国有一条横贯东西的大河，河有笔直的两岸，岸上各有位置不同的N（N≤5000）个城市。北岸的每个城市恰好有一个友好城市在南岸，而且不同城市的友好城市不相同。

每对友好城市都向中央政府申请在河上开辟一条直线航道连接两个城市，但由于河上常年有雾，政府决定避免任意两条航道交叉（若航道交叉，船只可能在雾中相撞）。编程帮助政府做出一些批准和申请的决定，使得在保证任意两条航线不相交的情况下被批准的申请最多。

### 【基本要求】

1. 随机生成河道两岸友好城市数据。可以令北岸城市编号为N1、N2、……、Nk，南岸城市编号为S1、S2、……、Sk。两岸城市数量必然相等，然后随机两两配对，将配对结果输出到文本文件中保存。
2. 以北岸城市为基准，按照时间段随机从中每次挑选一个城市发出开通航线申请，由你的算法代替政府进行审核，如果发生航线交叉，则申请被驳回；如果没有发生交叉，则申请通过，并打上标记。
3. 无论申请是否通过或者驳回，申请过的城市不能再发出新的申请。
4. 指定一个时间段，模拟申请的发生和审核情况。最后统计审核结果（包括批准和驳回的数量）。

### 【输入输出】

输入：城市总数cities，模拟天数days。

输出：城市配对情况（输出到文本文件中）。

模拟每天的申请和审核情况：

第1天：N24提出申请，配对城市S68。申请被批准。

第2天：N12提出申请，配对城市S79。申请被驳回。

第3天：N573提出申请，配对城市S286。申请被批准。

……

统计结果：总计有x个申请被提出，其中y个被批准，z个被驳回。

### 【实现提示】

由于此题降低了难度，因此审批的过程非常简单，几乎不涉及到算法的问题（只需要找出简单的判定规则即可），故此题学神学霸请绕路。但如果有人愿意挑战选做内容，则不受此限制。

### 【选做内容】

考虑到顺序审批会导致极大的资源浪费，将上述过程改为动态。即：

1. 每天发出申请的城市允许不止一个，于是每天就会有多条线路需要审批，审批的原则是要求被批准的申请尽可能多，但线路不交叉的原则则必须保证。
2. 当后续有新的申请被提交后，算法必须和已经被批准的申请结合起来考虑，有可能需要废除某些已经开通的线路以保证更多的申请被批准！
3. 被当场驳回申请的可以以后再次获得申请机会（由随机函数决定），但被废除已开通航线的城市是否应该继续获得重新申请的机会呢？请自行推导。

## 管道铺设施工

### 【问题描述】

需要在某个城市的n个居民区之间铺设煤气管道，则在这n个居民区之间只要铺设n-1条管道即可。假设任意两个居民区之间都可以架设管道，但由于地理环境的不同，所需经费不同。选择最优的施工方案能使投资尽可能少，这个问题即为求网络的最小生成树。

### 【基本要求】

1）参考教材，求解的算法为：在可能假设的m条管道中选取n-1条，既能连通n-1个居民区，又使总投资达到最小。

2）当增加一个顶点和若干条边及相关权值后，程序能自动重新计算。

### 【输入输出】

输入：居民区总数settlements，可以通过指定方式或随机方式生成线路图以及图上权值。网络采用邻接矩阵为存储结构。

输出：以顶点对(i,j)的形式输出最小生成树的边。

以下为一个用于检测的用例

A

I

B

H

G

C

E

D

F

32.8

18.2

12.1

8.7

52.5

79.2

44.6

5.9

56.4

41.1

21.3

67.3

10.5

85.6

98.7

### 【实现提示】

可以选用Kruskal算法或者Prim算法来求最小生成树，无论哪一种算法都要选好恰当的辅助数据结构，以存放边或顶点的集合。若采用Kruskal算法，则为选取当前权值最小的边，还要对边按权值进行排序。

注意整个算法的时间复杂度性，采用何种排序算法应依据边的总数来定夺，如果是变数很大的网络，就应选用先进的排序算法；如果按给定输入输出的小模型，选用一般的简单排序算法即可。

本题难度也不算太高，富有挑战精神的同学可以去试试别的题目。

### 【选做内容】

## 源程序相似性判断

### 【问题描述】

对于两个C语言的源程序清单，用哈希表的方法分别统计两程序中使用C语言关键字的情况，并最终按定量的计算结果，得出两份源程序清单的相似性。

### 【基本要求】

C语言关键字的Hash表可以自建，也可以如实现提示中那样构建。此题的主要工作是扫描给定的源程序，累计在每个源程序中C语言关键字出现的频度。在扫描源程序过程中，每遇到关键字就查找Hash表，并累加相应关键字出现的频度。为保证查找效率，建议Hash表的平均查找长度ASL不大于2。

### 【输入输出】

做几个编译和运行都无误的C程序，程序之间有相近的和差别大的，求出相似值S并对比差异程度。

### 【实现提示】

C语言有32个关键字，假设以二次探测再散列处理冲突，为达到ASL≤2，则要求装载因子*α*≤0.795，因关键字个数*n*=32，应取表长*m*>40。对于二次探测再散列，应取4*j*+3型的素数，故设表长*m*=43。最后根据所设表长设计Hash函数。采用除留余数法，并取*p*=41，设

hash(key)=[(key的首字符序号)\*100+(key的尾字符序号)] mod 41

冲突处理方法为

H*i*+1, *i*+2 = ( H1 ± *i*2 ) mod 41 *i* = 1, 2, 3, …

如下所示为C语言中的每个关键字根据Hash函数计算所得函数值以及发生冲突的次数*c*和经二次探测处理冲突之后记录在Hash表中的下标值*j*。

|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **Hash(key)** | ***j*** | ***c*** |
| auto(0014) | 14 |  | 0 |
| break(0110) | 28 |  | 0 |
| case(0204) | 40 |  | 0 |
| char(0217) | 12 |  | 0 |
| const(0219) | 14 | 15 | 1 |
| continue(0204) | 40 | 41 | 1 |
| default(0319) | 32 |  | 0 |
| do(0314) | 27 |  | 0 |
| double(0304) | 17 |  | 0 |
| else(0404) | 35 |  | 0 |
| enum(0412) | 2 |  | 0 |
| extern(0413) | 3 |  | 0 |
| float(0519) | 27 | 26 | 2 |
| for(0517) | 25 |  | 0 |
| goto(0614) | 40 | 39 | 2 |
| if(0805) | 26 | 30 | 3 |
| **Key** | **Hash(key)** | ***j*** | ***c*** |
| int(0819) | 40 | 1 | 3 |
| long(1106) | 40 | 36 | 4 |
| register(1717) | 36 | 37 | 1 |
| return(1713) | 32 | 33 | 1 |
| short(1819) | 15 | 16 | 1 |
| signed(1803) | 40 | 31 | 6 |
| sizeof(1805) | 1 | 0 | 2 |
| static(1802) | 39 | 38 | 2 |
| struct(1819) | 15 | 11 | 4 |
| switch(1807) | 3 | 4 | 1 |
| typedef(1905) | 19 |  | 0 |
| union(2013) | 4 | 5 | 1 |
| unsigned(2003) | 35 | 34 | 2 |
| void(2103) | 12 | 13 | 1 |
| volatile(2104) | 13 | 9 | 4 |
| while(2204) | 31 | 22 | 6 |

上表中所谓字符序号是指将26个英文字母按照从a到z的顺序，从0到25编号，则表中auto的hash值就是0\*100+14=0014，对41求模后结果就是14；而const的hash值也是14，于是取H2=(14+12)%41=15；同样，continue也要向后挪一位，其最终结果为41；float的hash值为27，于是取H2=(27+12)%41=28，但由于在28的位置上已经有了关键字break，因此取H3=(27-12)%41=26；以此类推，goto需要向前移动一位，而if由于冲突，取+12和-12的结果都发生了冲突，因此只能取H4=(26+22)%41=30；……表中其余元素的计算请自行推导。

扫描两个源程序所统计的所有关键字不同频度，可以得到两个向量。如下面简单的例子所示：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 关键字 | void | int |  | for | char |  | if | else |  | while |
| 程序1中关键字频度 | 4 | 3 |  | 4 | 3 |  | 7 | 0 |  | 2 |
| 程序2中关键字频度 | 4 | 2 |  | 5 | 4 |  | 5 | 2 |  | 1 |
| Hash地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

根据程序1和程序2中关键字出现的频度，可提取到两个程序的特征向量**X1**和**X2**。



一般情况下，可以通过计算向量**X*i***和**X*j***的相似值来判断相应的两个程序的相似性，相似值判别函数计算公式为

 （公式1）

其中。的值介于[0, 1]之间，也称广义余弦，即，当时，显见；当和差别很大时，的值近似为0，*θ*就接近于。例如：



可以用下面的二维图示来直观地表示向量的相似程度。

*θ*

*xj*

*xi*

在有些情况下，还需要做进一步的考虑，如下图所示

*θ*

*xk*

*xi*

*xj*

*D*(*xi, xk*)

*D*(*xi, xj*)

从图中看出，尽管和的值是一样的，但直观上**X*i***与**X*j***更相似。因此当S值接近于1时，为避免误判相似性（可能是夹角很小，模值差很大的向量），应当再次计算**X*i***与**X*j***之间的“几何距离”。其计算公式为

 （公式2）

最后的相似性判别计算可分两步完成：

1）用公式1计算S，把接近1的保留，抛弃接近0的情况（即排除不相似者）

2）对保留下来的特征向量，再用公式2计算D，如D值也比较小，说明两者对应的程序确实可能相似。

S和D达到什么门限才能决定取舍，需要积累经验，选择合适的阈值。

本题很大的工作量将是对源程序扫描，区分出C程序的每一关键字。可以为C语言关键字建一棵键树，扫描源程序和在键树中查找同步进行，以取得每一个关键字。

这种判断方法只是提供一种辅助手段，即便S=1也可能不是同一个程序，S的值很小，也可能算法完全是一样的。例如一个程序使用while语句，另一个使用for语句，但功能完全相同。事实上，当发现S的值接近于1且D又很小时，就应该以人工干预来区分。

### 【选做内容】

## 海战游戏模拟

### 【问题描述】

玩家和电脑各自在两块10x10的棋盘中放置战舰，每方 5 艘，且对对方隐形，放好后不能移动。每艘战舰在棋盘上占据一定数量格子，其中护卫舰 2 格，驱逐舰 3 格，潜艇 3 格，战列舰 4 格，航空母舰 5 格。每艘船可横竖任意放置，但任意两艘船不能相互接触，更不能重叠。玩法是双方轮流“轰炸”对方的战舰，每次轰炸的结果是否击中均显示出来。若击中，则可继续攻击，未击中则由对方出招。要击沉一艘战舰，该战舰的所有格子都必须被命中。游戏目标是赶在对手之前将对方所有战舰都击沉。。

### 【基本要求】

（l）在游戏开始之前的初始化工作中，玩家的战舰摆放由手动形式完成，而电脑的战舰摆放必须通过随机函数形式进行（即每次重玩游戏，电脑的战舰摆放应该不同）。

（2）电脑的探测方向必须表现出一定的算法特征。例如，第一次投弹时，若电脑“炸中”玩家战舰，则其探测方向必须是随机的；但若在轮到电脑出招前，此前已有炸中的成果，则探测方向必须能加以利用。

（3）无论玩家还是电脑，每投下一弹，均需在屏幕上显示结果并保持到游戏结束。至于双方轰炸过程中未中的区域是否对对方开放记录，则可自行决定。例如，玩家轰炸格子[1,1]，结果未中，这一结果可以记录下来，玩家此局就不会再去探测此格。但这一成果是否令对手知晓则可以通过游戏选项选择。若允许对方共享成果，则电脑亦不再探测此格，否则电脑仍有可能探测此处。注意，选项对双方平等处理。

（4）双方战舰的形状不做要求，既然是在格子内，用“\*”号表示即可。轰炸结果，炸中可以用“+”号，没炸中可以用“-”号。

### 【输入输出】

输入：战舰数据由随机数产生器生成。自行决定最合适的输入和存储格式。

输出：自行决定最佳输出格式。

### 【实现提示】

既然是蛮力法探测，作为电脑的探测算法，在已有成果上的探测模式，可以借助迷宫的思路，运用栈和队列等结构来帮助实现。

屏幕重绘需要用到gotoxy函数。“轰炸”方式可以用坐标输入，或者用鼠标点击（须自行查阅C手册）。

### 【选做内容】

## 铁路调度模拟

### 【问题描述】

某东西向的铁路上有一小站，该站只有一条铁路支路可供火车停靠，且该支路最多能够容纳M（1<=M<=3）列火车。火车都有一定长度且长度都相等。为了火车行驶的通畅，规定只许火车自东方进站从西方出站，且先进站的火车必须先出站。

该火车站工作任务繁忙，假设每天都有至多N（N<=250）列火车要求在预定时刻进站，并在站内作一定时间P（P<=10）的停靠。为了尽可能满足每列火车进站的要求，小站的调度工作必须井井有条地开展。在小站每天的工作开始前，工作人员必须阅读所有火车的进站申请，并决定究竟接受哪些火车的申请。而对不能满足要求的火车，只能请它们直通过站，不做任何停留。假设火车进站、出站本身不占用额外时间，只考虑停留时间。将时间划分为基本时间片，一个时间片视为一个基本时间单位，每一个单位时间可以完成一件操作，如进站、出站、过站等。该火车站的总工作时间片数T由键盘输入。

请编写一个程序，模拟小站火车进出站的情况。

### 【基本要求】

（l）假设所有经过该站的火车均有停靠需求。假设火车可以在同一个时间片到达（这并非同时到达，而是指在同一个时间片内依次到达），因此任意时间片是否有火车到达、有多少列到达以及到达后的停留时间等数据均必须通过随机方式产生。

（2）所有火车按自然顺序编号表示。

（3）进站的火车只能在支路上停靠，主干用于过站火车直通，以防止拥塞。

（4）当某一时间片内，支路上停靠的火车达到M列时，后来的列车将无法获得停靠许可，必须强制过站。

（5）当支线上已经有火车停靠时，若后进站的火车要求停靠时间小于已经停靠火车的剩余停靠时间，则后进站的火车将被迫延长停靠时间，导致发车晚点。例如，若在时间片1有火车进站，要求停靠8个时间片，则其应在第9个时间片离开。但若在第3个时间片又有火车要求进站停靠2个时间片。由于此时第一列火车还要停留5个时间片，因此后来的火车必须等到第一列火车出站才能出发。考虑到出站不需要时间，因此第二列火车同样在第9个时刻完成出站，但其晚点发车4个时间片。

（7）根据以上描述可知，同一时间片，可能同时发生几列火车进站、另外几列火车出站、以及若干火车过站等事件。

（6）在模拟结束后，统计该火车站在T个时间片内，火车总数、停靠火车数、过站火车数、平均晚点时间片数，此外还需要计算正点列车、晚点列车、过站列车占总列车数的百分比。

### 【输入输出】

由随机数产生器生成。如果打算提前构造一天的火车进站数据，则这些数据必须存储在文件中，程序直接从文件读取数据并处理。

### 【实现提示】

这个题目可以参考课件中的机场事件模拟。但两个问题的具体实现方式有所不同，必须加以仔细甄别。另外，题目中既有天数又有时间片数，可以按照每天若干（例如100）个时间片来处理，键盘输入总天数，然后运行时间片就是T=天数\*每天的时间片数。

### 【选做内容】

（1）将一条支路改成M条支路，且每条支路只能停靠一列火车。看看此时数据会发生怎样的变化。

（2）在选做（1）的基础上，若允许火车双向通过，为了避免撞车事件，程序应怎样修改才更接近真实？

## 10-20-30游戏

### 【问题描述】

游戏使用除去大小鬼之后的52张牌，J、Q、K的面值为10，A的面值为1，其它牌的面值等于其点数。

将牌洗过后叠在一起放在手上，从最上面开始一次拿出7张牌从左到右摆成一条直线放在桌子上，每一张牌代表一个牌堆。然后每次从手上取最上面的一张牌，从左至右依次在各个牌堆的表面放一张牌，但露出前一张牌的头部（我们称此牌为放在前一张牌的后面，后面的摆放方法均同此规则），当往最右边的牌堆放了一张牌以后，重新往最左边的牌堆上放牌。

当牌堆里的牌张数不少于3张时，开始检测过程。检测规则为：

1. 检查牌堆的最前面两张和最后面一张牌的和，如果等于10、20或30，这三张牌就会从牌堆中拿走，然后按顺序放回手中并压在最下面。如果没有出现这种情况，则进行步骤2。
2. 检查牌堆最前面一张和最后面两张牌的和，看是否等于10、20或30。如果是，操作方法同上，否则进行步骤3。
3. 检查牌堆最后三张牌的和，规则同上。例如，如果某堆牌的牌点顺序为5、9、7、3，那么放上6以后结果为5、9、7、3、6，根据步骤1，应该抽走5、9和6，留下7和3在牌堆中；但是如果放上的不是6而是2，则结果为5、9、7、3、2，根据步骤2，应该抽走5、3、2，留下9和7在牌堆中。
4. 如果某次操作后某堆牌中没有剩下一张牌，那么把该牌堆永远清除掉。如果所有牌堆都清除了，则游戏胜利结束；如果手中没有牌了，则游戏以失败告终；如果游戏永远无法结束，则称游戏出现循环。

请模拟此游戏过程。

### 【基本要求】

手中牌的初始状态通过随机函数生成。尽管此游戏不要求记录花色，但为了真实起见，在实际操作时，要求标记和显示花色♠♥♣♦。花色的表示方法可以查阅ASCII码表，这样每一张牌就都是独一无二的。

算法能准确除了能判断胜利和失败意外，还应能准确判断出什么情况下出现循环并能在屏幕上给出提示。

### 【输入输出】

输入完全通过随机函数产生。

输出格式，在屏幕上方显示手中的剩余牌序列，在下方显示发牌情况。例如：

　手上牌：♠K ♦10 ♣3 ♥3 ♠A ♠2 ♥J ……

牌堆情况：牌堆1 牌堆2 牌堆3 牌堆4 牌堆5 牌堆6 牌堆7

♦5 ♦Q ♥5 ♠A ♣8 ♣6 ♥4

……

每当从手上牌取7张牌到牌堆中发放时，上面的手上牌区域应能将被取走的牌消除（每次一张），同时右边的牌顺序左移。而当有牌放回手上牌后面时，在下面的牌堆区同样应能正确显示从上到下的剩余牌顺序，并在手上牌区域中将放回的牌显示在序列后面。

每次发牌完毕进行判定时，如果某堆牌有合适的牌满足10-20-30规则可以被取走，则应先在该牌堆下方（或上方，自行决定最佳位置）给出标志，表明此牌堆有牌可以被取走。如果有多个牌堆同时有牌可以被取走，则按照从左到右的顺序依次处理。

### 【实现提示】

为防止屏幕滚动，可使用光标定位函数

#include <conio.h>

void gotoxy(int x, int y)

来实现在屏幕指定位置输出（默认情况下屏幕范围为25行80列）。亦即牌局无论进行到何时，屏幕上的内容均不能滚出屏幕外，必须固定显示，因此出牌提示区需要反复覆盖提示。

此题操作过程基本不涉及到复杂算法，只要通过随机函数洗牌的事情做好了，剩下的主要工作基本上就是如何显示，因此对格式输出有较高要求。牌堆中的牌可以通过双端队列的形式实现。唯一的难点在于，如何才能知道操作陷入死循环。请自行通过分析来找出判定条件。

### 【选做内容】

## 决斗

### 【问题描述】

n个角斗士被要求进行生死决斗。规则是：所有人围成一圈，按照一定的顺序拉人，被拉出的角斗士就和紧靠其右的人决斗，失败者的尸体将被抬走（即退出圈子）。

研究表明，最终的结果在相当程度上取决于决斗的顺序。可能出现A胜过B，B胜过C，而C又胜过A。因而史学家决定研究哪些角斗士有可能生还。

将这n个角斗士按照从1～n逆时针方向排成一圈，他们要决斗n-1场。其中第i个人与第i+1个人决斗。死者退出圈子，紧靠死者右边的人成为与赢者直接相邻的人。任意两人之间决斗的胜负通过一个矩阵给出——如果A[i][j]=1，表示i能战胜j；如果A[i][j]=-1则表示i打不过j；当i=j时值为0。

### 【基本要求】

生成胜负关系矩阵A。然后将n个角斗士随机放入一个循环链表中，注意，这n个角斗士有自己的代号，也有在链表中的编号，两者意义不同。若规定从链表的1号开始数数，自行决定一个数字用于从链表中选择角斗士，然后被选出的角斗士和右边的角斗士决斗。胜负关系根据矩阵A决定。之后继续数数，找出下一对需要决斗的对手。

模拟此过程，直到最后一个人。

### 【输入输出】

输入：角斗士总数fighters。通过随机函数生成胜负关系矩阵A。

输出：按照决斗场次输出决斗结果。

### 【实现提示】

基本要求降低了题目的难度，因此本题和经典的约瑟夫环基本类似。但如果按照原题的要求，则此题需要求出所有可能赢得整场决斗的人的序号，这就转化成为一个动态规划问题。有志于挑战此动态规划问题的同学可以做选做内容。

### 【选做内容】

分析矩阵A，对每一个角斗士进行数据分析，看看其在任意次序下有没有可能赢得最终的决斗胜利。

## 邮递员问题

### 【问题描述】

邮递员每天都辛辛苦苦地工作，沿着每条路给村庄（总计有n个村子）的人送信。村子之间有道路相连，形成一个网状结构。每个村可能处于两条路的十字路口上（如村庄1）、四条路的米字路口上（如村庄3）或一条路的中央（如村庄5）。

村子里的人都希望自己村子能尽量早地收到信件，因此和邮递员达成了一个协议：每个村庄都有一个期望值Wi，如果该村子是邮递员经过的第K个不同的村子，那么如果K≤Wi，则村子付给邮递员Wi-K元钱，否则邮递员付给村子K-Wi元钱。另外，每经过一条不同的路，邮局会付给邮递员1元钱。而邮局规定每条路（总共m条）都至少经过一次。邮递员该怎样走，才能挣最多的钱呢？

### 【基本要求】

每个村庄的编号和期望值可以通过随机函数生成，但要注意，任何一个村庄的期望值都应该小于村庄的总数n。允许一些村庄的期望值相等，但不允许所有村庄都把期望值定为1这样的小数字（那样邮递员只好辞职了）。为简化问题，约定将1～n-1随机分派给n-1个村庄作为期望值，剩下的第n个村庄则随机从1～n-1中选一个数作为自己的期望值。这样就只有两个村庄的期望值是一样的。

村庄的线路布局可以自行指定，也可以通过随机函数生成。但必须要有明确的存储方法。

假定每次出发的地点为地图外，出发时邮递员可以空降到任意一个村庄作为自己的第一个到达地点。

算法要求能精确计算出怎样走才能实现邮递员的利益最大化，并讨论一下有没有可能在某个特殊图形中会出现亏损。

### 【输入输出】

输入：村庄总数villages

输出：为便于检验，可以考虑采用二维矩阵的形式表示村庄之间的通路，但有一点必须注意，整个图是二维平面图，不可能出现道路交叉，且所有结点的度只能是2、4或8。例如：

1

8

2

10

5

6

7

4

3

5

9

4

3

1

9

8

7

7

6

2

输出邮递员所经过的路线以及他所赚取或损失的金钱数量。当所有村庄都被遍历后，计算其最终的收入。

### 【实现提示】

本题并没有要求路线不能重复，也没有规定到过的村庄不能再经过一次。所以理论上，邮递员从邮局至多可以赚取m元，m为图中的边数。尽管如此，了解一些欧拉回路的知识对解决本题还是有一定帮助的。

### 【选做内容】

## 玄冥二老

### 【问题描述】

玄冥二老在被张无忌调戏后心灰意冷，决定退隐江湖。鹤笔翁处在城市A，鹿杖客处在另一个未知的城市。他们打算选一个城市X（这个由鹤笔翁来决定）碰头，鹿杖客会赶在鹤笔翁之前到达城市X，然后等待鹤笔翁也赶到城市X汇合。他们决定在城市X好好庆祝一下，由鹤笔翁请客。接着一起回到他们的家乡城市B。由于鹿杖客淫性不改，他要求在鹤笔翁回家线路的城市中找最贵的天上人间去消费！也就是说，假设所有的城市都有天上人间，而在鹤笔翁回家的线路A🡺B所经过的所有城市中要选出其中天上人间消费最贵的那个城市作为碰头的城市X。

鹤笔翁正专注地看回家的地图，地图上标有n（n≤100）个城市和某些城市间直达的道路，以及每条道路的过路费。鹤笔翁还知道在每一座城市的天上人间消费的开销。现在给出地图描述和城市A、B的位置，请帮助计算出鹤笔翁回家的最小费用。

### 【基本要求】

通过随机函数生成每个城市的数据，包括该城市到其它城市有无道路可通，如果可通，过路费是多少，以及该城市的天上人间消费价格（假设天上人间只提供一种消费价格）。

通过算法计算鹤笔翁的回家线路，并计算其最终消费多少（包括过路费，只算鹤笔翁一人）。

### 【输入输出】

输入：城市总数cities。通过二维矩阵表示城市间的道路通行情况以及过路费、天上人间消费价。此矩阵可以保存到文本文件中以便查看。

起点A和终点B通过键盘输入。

输出：鹤笔翁选择的经过城市编号以及路途耗费，碰头的城市编号以及在天上人间消费的结果，最终到达B城市后的消费总值。

### 【实现提示】

由于鹿杖客规定必须选择路线上最贵的天上人间消费，因此不能简单地选择一条几何上的最短路径——那样只需要通过Dijkstra算法即可解决，但是如果路上有一个花费特别贵的城市，鹤笔翁可能会破产。因此可以这样考虑：对于每个店，先把它作为聚会地点X，然后删去比它贵的堕落城市以后，再求出AX和XB的最短路径，连在一起就是所求线路。

### 【选做内容】

## 寻找马航失联客机

### 【问题描述】

马航客机失联事件引发全世界的关注，多国政府投入大批人力物力进行搜寻，却始终没有头绪。一天，一个神秘的电话打进了马方情报机构，声称有关于失联客机的重大线索，愿意无偿提供，但对方同时要求，为了确保不暴露身份，只能派一个人去联络，且只能在指定地点获得通行证后步行穿过沙漠前去，不能乘坐任何交通工具。对方只肯透露联系地点位于阿富汗的某个沙漠绿洲中，而你，被委以重任，成为前去获取情报的唯一人选。

没有水和食物，你将无法前行。为了穿越沙漠，需要购买一些食物。在出发地，你可以采购食物和获得无限多的免费饮用水。但由于背包容量有限，在任意时刻拥有的水和食物总量不得超过一个给定值limit。沙漠中有若干绿洲，在绿洲你可以得到无限多的饮用水。但那些绿洲并不会为你提供食物，不过你可以携带一些食物在那里存储。假设这些食物永远不会过期变质，你可以不断往返来增加一个绿洲的食物储量。当然，你也可以选择就在出发地存储食物。

你将被告知出发地、目的地和所有绿洲的平面位置。每走x单位长的路程你就要消耗y单位的水和z单位的食物。请确定在出发地最少购买的食物量。注意：出发地的食物按整数单位出售且总量为106。

### 【基本要求】

背包的容量可以人为指定，但必须合理，你所假设的数字limit上限不得超过100，下限不少于20。沙漠中的绿洲布局显然也是一个图，此图的存储形式可以自定。但是必须记住，起点和终点应分列图的两侧。

必须确保背包中背负的食物总数无法维持一次走到目的地。因此需要在某些绿洲之间以及绿洲和出发地之间来回奔波搬运食物。当然，我们可以假设到达目的地获得情报以后可以派专机前往迎接，从而不愁返回的问题。

你得精确计算出路途中每个绿洲j从其出发时必须在它那里保存的最低食物总量d[j]。考虑从绿洲i出发，第一次到达绿洲j的情况：需要在绿洲i保存多少食物，才能保证在绿洲j保存d[j]单位的食物？

运用dijkstra算法或bellman-ford算法解决此题。

### 【输入输出】

输入：绿洲总数oasis。上面提到的单位路程x、单位水消耗y和单位食物消耗z（这意味着y/x和z/x的值可以是实数）。

运用随机函数生成绿洲之间的网络（即绿洲之间的路线是有长度的）

输出：你所选择的绿洲编号，每个路段的往返次数，每次出发前背包中的水和食物的携带量，每次行程消耗的食物量（向上取整）以及在每个绿洲每次行动后存储的食物余量。

### 【实现提示】

本题不能使用简单的dijkstra算法进行计算，但可以借助最短路的标号法思想。由于水在每个绿洲都是无限的，所以每次携带的水只需要根据路程计算出最低携带量（也要向上取整）即可，其余的可以全部用来携带食物。注意，你返回也是需要消耗食物的，所以在绿洲留下食物时要把这一因素考虑进来。

剩下的问题就比较简单了，从终点进行逆推即可。

### 【选做内容】

## 魔术球问题

### 【问题描述】

有N根柱子，现在有无限多的中心穿孔的球，按照自然数顺序分别编号。请你把尽量多的球放入这N根柱子，要求：

① 串入球的顺序必须是1、2、3、……，不可以跳过某个球不放，且每次只能在某根柱子的最上面放球。

② 同一根柱子中，相邻两个球的编号和为完全平方数。

请问，最多能放多少个球在这N根柱子上？例如，4根柱子可以放11个球。

### 【基本要求】

在给定柱子总数的情况下，顺序演示到球不能再放入为止。

找出柱子数和可以放入球数之间的函数关系。

### 【输入输出】

若输入为球的总数balls，则输出为需要的柱子数poles

若输入为指定柱子数poles，则输出为可以放入的球的最大数量balls。

无论哪种输入，都需要输出每根柱子的编号以及在其上的球的编号。

### 【实现提示】

把每个球看作一个顶点，如果两个球i和j（i<j）的编号和i+j为平方数，那么把两个顶点连成一条有向边i🡪j。首先解决这样一个判定问题：给定球总数n和柱子数m，能否把所有球放到柱子上？这个问题实质上就是所谓的最小路径覆盖问题——用尽量少的不相交简单路径覆盖有向无环图G的所有顶点。

本题相对简单，学霸学神们敬请避开，为基础不太好的同学留条生路。

### 【选做内容】

## 随机走动

### 【问题描述】

一只贪杯的蟑螂醺醺然在室内地面游荡。地面铺满方砖，共计m\*n块，构成大面积矩阵。蟑螂在方砖间随机爬行，可能想撞大运，找片阿司匹林解酒。假定蟑螂从房间中央一块方砖出发，除非撞墙，可以爬向相邻8块方砖的任意一块，且机会均等。问蟑螂多久才可以在所有方砖上留下行迹？

### 【基本要求】

用仿真的方法模拟蟑螂的爬行方向。统计每次蟑螂爬满所有方砖的爬行次数，将程序运行至少1000次，从结果中推导关于在给定m和n的情况下蟑螂爬满方砖的期望值或区间。

### 【输入输出】

输入：房间参数2<m≤20和2<n≤40，起点坐标（x, y）。

输出：蟑螂每达成目标一次，就输出count数组（称为蟑螂的爬行密度）一次，并对count数组求和。将每次的运行结果保存下来，在程序运行至少1000次后，对结果进行统计归纳，指出蟑螂最可能需要多少次才能爬满所有方砖。

### 【实现提示】

用m\*n的数组count记录蟑螂爬到每块方砖的次数，数组每个单元的初值都是零。蟑螂的当前位置用坐标ibug,jbug表示，下一个移动目标是（ibug+imove[k], jbug+jmove[k]），0≤k≤7，且

|  |  |
| --- | --- |
| imove[0]=-1 | jmove[0]=1 |
| imove[1]=0 | jmove[1]=1 |
| imove[2]=1 | jmove[2]=1 |
| imove[3]=1 | jmove[3]=0 |
| imove[4]=1 | jmove[4]=-1 |
| imove[5]=0 | jmove[5]=-1 |
| imove[6]=-1 | jmove[6]=-1 |
| imove[7]=-1 | jmove[7]=0 |

蟑螂爬向的下一个目标是相邻的8块方砖之一，仿真时每次产生一个0到7的随机数，存放在变量k中。因为蟑螂不会爬出房间，如果产生的随机数对应的坐标撞墙，则忽略不计，继续产生下一个新随机数。蟑螂爬行到任意一块方砖，它的计数值都要加1，记录蟑螂爬过的次数，一直到所有方砖都至少经历一次。

本题相对简单，学霸学神们敬请避开，为基础不太好的同学留条生路。

### 【选做内容】

理论上，1000次运行结果应该符合正态分布，试以直方图的形式将正态曲线表示出来。

## 多跑道机场仿真

### 【问题描述】

某机场有三条跑道，编号为0、1、2。其中0号和1号跑道主要用于降落（即降落优先），2号跑道主要用于起飞（即起飞优先）。因此有两个着陆等待队列，到达机场上空的飞机进入任意一个等待着陆队列，两个着陆队列长度应尽量保持接近。飞机进入着陆等待队列后，要赋予一个唯一的整数（例如奇数）标识，以及一个表示飞机可以滞空的最大时间（因燃油有限），也是整数值，且假定飞机每停留一个时间单位就要消耗掉一个单位的燃油。飞机的起飞可以用三条跑道中的任意一条，但只有一个起飞等待队列，飞机进入起飞等待队列后，也赋予一个唯一的整数（例如偶数）标识。

每一时刻，着陆等待队列中的飞机数量不超过5架，起飞等待队列中的飞机数量也不超过5架。每条跑道在一个时间单位只能让一架飞机着陆或起飞。尽管0号1号主要用于着陆，但如果没有飞机滞空，跑道空闲的时候，也可以同时用于起飞，这样同一个时间单位内就可以允许至多3架飞机同时升空；同样，尽管2号跑道主要用于起飞，但是如果着陆等待队列中的某架飞机滞空时间接近0，必须提升其优先级，此时可以调度2号跑道供其降落，但如果多于一架飞机的滞空时间接近0，则应提升其优先级，令其加塞提前使用其它跑道降落。

### 【基本要求】

飞机的到达事件和起飞事件均由随机函数产生，频度可以由键盘输入（可参考课件中的代码）。到达机场的飞机应先进入等待队列，油量也由随机数产生，数值范围在2到10之间。预警油量为2，即当飞机油量还剩2的时候，必须立刻提升优先级，令其在下一个时间段立刻优先降落，这样在下一个时间段该飞机可以利用仅存的1个油量单位执行降落，否则，如果飞机油量仅剩1还无法在当前时间段降落，那么即使下一个时间段能够降落，它也会因为油料耗尽而坠毁！

无论是到达机场或者申请离场，一旦等待队列满，则将被拒绝服务，被拒绝服务的飞机次数需要统计求和。

### 【输入输出】

输入：程序运行的时间长度，飞机到达的发生频度（单位时间内到达机场的架次，可以是实数），飞机起飞的发生频度（单位时间内申请离开机场的架次，可以是实数）。

输出：每个队列的内容，平均起飞等待时间，平均降落等待时间，总降落架次，总起飞架次，尚未降落的飞机数量，尚未起飞的飞机数量，被拒绝服务的飞机数量，因燃油耗尽而坠毁的飞机数量，跑道空闲百分比。

### 【实现提示】

着陆等待队列由于有紧急降落的引入，需要额外操作，这些操作应该在抽象数据类型中明确定义。考察一下什么情况下会发生飞机坠毁（例如有4架着陆队列的飞机油量同时降为2，由于仅有三个跑道允许降落，此时必然会有一架飞机坠毁）。

### 【选做内容】

## 电梯仿真

### 【问题描述】

某建筑高N（5≤N≤30）层。有一部电梯供乘客使用，假设电梯轿厢足够大，只要在承重范围内，可以装下足够多的人，因此不考虑乘客人数，只对乘客的总体重做出限制，规定乘客总重量超过上限W（1000kg≤W≤2000kg）则电梯发出超载警示。规定每个乘客体重T的范围是10kg≤T≤150kg（不计小数，按整数计算），即体重小于10kg的儿童不能单独乘坐电梯，须由大人怀抱，并视为一个人合并计算。不过本题为简单起见，无需对此进行区分，只认为进入电梯的乘客体重都是符合上述规定的。

电梯能根据乘客的要求在每个楼层都停靠，这里不考虑时间因素，只假定在一个时间单位内即可完成进出所有有需求的乘客，并假定电梯每经过一层楼都需要一个时间单位。例如，电梯在程序开始运行时，初始状态是停靠在1楼。在第1个时间单位内，完成开门、所有乘客进入并关门的要求；第2个时间单位中，电梯上行并停靠在2楼，第3个时间单位内完成开门、部分乘客出，部分乘客入并关门的动作；之后又经过2个时间单位，在第5个时间单位时停靠在4楼，第6个时间单位内完成新的开门，上下客和关门动作。其余依此类推。

当排队人数过多（注意，作为乘客不会考虑排队乘客的总体重，而只会计算人数），超过10人时，若其方向为下楼或虽然是上楼，但目标楼层间隔不超过5时，该乘客会选择放弃排队，改为走楼梯。

初始状态下，电梯停靠在1楼，轿厢内没有乘客。每个楼层如果出现需要乘坐电梯的人，都应该安排进入两个队列，上下方向各一个（1层和N层例外，各自只有一个方向。此外，只要任一楼层任一队列中出现了乘客，就视为自动按下了电梯请求按钮）。若同时在两个方向上都有请求发生，则根据最短响应时间和当前运行方向进行选择，算法自拟。当电梯的额定载重上限W与轿厢内的乘客总重量之差＜10kg时，电梯会自动进入“满员”状态，此时电梯将直接驶向当前运行方向最近的指定楼层，并拒绝响应那些途经楼层外部等待队列的请求。

请模拟这部电梯在指定的时间段M（为一个不小于3N的自然数）内的运行过程，并将结果记录输出到屏幕和文件中。

### 【基本要求】

楼层总数N，电梯轿厢承重上限W从键盘输入。

在每个时间段，哪个楼层出现多少人需要乘坐电梯，以及乘坐方向和目的楼层，连同每个乘客的体重都是由随机函数生成的。

### 【输入输出】

输入：楼层总数N，电梯轿厢承重上限W，程序运行的时间长度M，乘客到达的发生频度（单位时间内到达的人数期望值，可以是实数）。

输出：每一个时间段各个楼层发生的乘客请求数、方向以及每个乘客的体重。程序运行结束后，每个乘客的平均等待时间，尚未得到服务的乘客数量，已经服务结束（指离开电梯）的乘客数量，放弃服务的乘客数量，电梯空闲百分比。

将模拟的过程记录同时输出到屏幕和一个文本文件中。

### 【实现提示】

乘客在没有进入电梯之间，虽然有目标楼层，但此时电梯是不知道的，位于等待队列中的乘客只能按向上或向下键。只有在进入电梯后，目的楼层才会被电梯获得，并按照响应的规则进行单向往复折返运动。

### 【选做内容】

1. 考虑有多部电梯的情形。
2. 考虑有多部电梯，且有些电梯限制停靠楼层的情形（例如分奇偶数或高底层）。

## 中国式买票

### 【问题描述】

春节前夕，一年一度的运输高潮开始了。在12306推出网络售票之前，购买火车票的人都需要到火车站窗口去买票，因此买票的队伍往往拖得很长。此时如果你也来买票，看到一眼望不到头的购票队伍，往往第一反应就是找个排在前面的朋友帮忙。如果真有，那么你就可以排在他后面。这样的行为就叫做“插队”，亦称“加塞”。本题的任务就是模拟此种可以插队的情形。

为简化程序，程序按照每个时间单元（以下简称“时元”）来进行模拟，假设只有一个窗口为购票者服务，每个购票者请求服务的时长（由随机函数在该购票者抵达时临时产生）从1～10个时元不等，每个时元内有且仅有一个购票者抵达并发出入队请求，在下一个时元中进入购票队伍。每个抵达购票队伍的人都会首先扫描投票队伍中有没有自己的朋友。如果有，且朋友未接受服务，则插队排在该朋友后面；但如果有朋友，可该朋友是队首元正在接受购票服务，则不能插队，须排在队尾。一旦队首元的服务时间结束，就会出队，下一个时元立即开始新队首元的服务。

请模拟这个售票窗口在指定的测试用例中为所有购票者服务的运行过程，并将结果记录输出到屏幕和文件中。

### 【基本要求】

程序从“input.txt”文件（此文件内容可手工输入，亦可通过程序生成）读入测试用例。为简化起见，一个文件只包含一个测试用例，用例的第一行是朋友组的数目n（2≤n≤100），从第二行开始，每一行都是以该朋友组中的人数开头，后面接此朋友组中的人员编码，以空格隔开。要求每个人的编码均以组别+编号处理。每个朋友组中的朋友数不少于1人，不超过10人，每个人只属于一个朋友组。

模拟过程在屏幕上显示，而测试结果则要求输出到“output.txt”文件中，统计结果则输出到“stat.txt”文件中。

例如，一个用例文件“input.txt”的内容是：

3

3 A0 A1 A2

4 B0 B1 B2 B3

1 C0

该文件第一行的数字3表示只有3个朋友组，第二行表示属于A组的相互都是朋友的有3人，他们的编号从A0到A2；第三行表示属于B组的相互都是朋友的有4人，他们的编号从B0到B3；第四行表示属于C组的只有1人，编号C0。该用例表示全部需要买票的人数为8人，他们全部服务完毕则模拟结束。

### 【输入输出】

输入：从用例文件“input.txt”读取全部人员分组情况后，将他们分别存入对应的数组，然后通过随机生成朋友组的组别和编号来选择加入队伍的购票者，并在入队前随机生成其购票所需的服务时间。之后，每个时元都重复上述过程，直到所有购票者均进入队列为止。

输出：在屏幕上显示每一个时元发生的事件，包括时元编号，正在购票者编号，请求入队者编号，请求入队者服务时长。在模拟结束后，将测试结果写入“output.txt”文件，其格式为：购票者加入队列时元 购票者编号 该购票者的服务开始时元

以前述用例为例，一个可能的模拟输出过程为：

time BuyerNo. RequestNo ServiceTime

1 B2 5

2 B2 A1 1

3 B2 B0 3

4 B2 C0 8

5 B2 B3 3

6 B2 A0 2

7 A1 B1 1

8 A0 A2 4

9 A0

10 B0

11 B0

……

根据上述抵达购票队伍的过程及朋友关系，最终的服务队列次序为：B2, A1, A0, B0, B3, B1, C0, A2

则“output.txt”文件中的内容应为：

ArriveTime BuyerNo. StartTime

1 B2 2

2 A1 7

3 C0 17

4 B0 10

5 B3 13

6 A0 9

7 B1 16

8 A2 25

而统计文件“stat.txt”的内容则应包括每个人之前被插队的人数（如上例中C0的被插队人数为3人），平均被插队人数，每个人的等待时间（开始服务时间-抵达时间）以及平均等待时间（Σ等待时间/人数）。

### 【实现提示】

购票者的分组和编号可以采用动态分配内存的方式，但是要注意，当朋友组的个数超过26个时需要考虑其它标记方式。

由于有插队者，因此不能单纯用一个数组来表示队列，可以考虑用静态链表方式来表示插队者在入队后的逻辑位置。

### 【选做内容】

1. 考虑有多个窗口的情形。
2. 考虑每个购票者的编码改用名字表示，同时要求名字不重复，这样需要采用散列表来存储大量的姓名。

# 三、课程设计进度要求

|  |  |  |
| --- | --- | --- |
| **序号** | **设计内容** | **所用时间** |
| 1 | 问题分析和任务定义 | 0.5天 |
| 2 | 数据类型和系统设计 | 0.5天 |
| 3 | 编码实现和静态检查 | 1天 |
| 4 | 上机准备和上机调试 | 4天 |
| 5 | 总结和整理设计报告 | 1天 |
| 合 计 | | 7天 |

# 四、课程设计步骤与方法

## 1．问题分析和任务定义

通常，设计题目的陈述比较简洁，或者说是有模棱两可的含义。因此，在进行设计之前，首先应该充分地分析和理解问题，明确问题要求做什么？限制条件是什么。注意：本步骤强调的是做什么？而不是怎么做。对问题的描述应避开算法和所涉及的数据类型，只对所需完成的任务做出明确的回答。例如：输入数据的类型、值的范围以及输入的形式；输出数据的类型、值的范围及输出的形式；若是会话式的输入，则结束标志是什么？是否接受非法的输入？对非法输入的回答方式是什么等。这一步还应该为调试程序准备好输入输出，包括合法的输入数据和非法形式的输入数据。

## 2．数据类型和系统设计

在设计这一步骤中需分逻辑设计和详细设计两步实现。逻辑设计指的是，对问题描述中涉及的操作对象定义相应的数据类型，并按照以数据结构为中心的原则划分模块，定义主程序模块和各抽象数据类型；详细设计则为定义相应的存储结构并写出各函数的伪码算法。在这个过程中，要综合考虑系统功能，使得系统结构清晰、合理、简单和易于调试，抽象数据类型的实现尽可能做到数据封装，基本操作的规格说明尽可能明确具体。作为逻辑设计的结果．应写出每个抽象数据类型的定义（包括数据结构的描述和每个基本操作的规格说明），各个主要模块的算法，并画出模块之间的调用关系图。详细设计的结果是对数据结构和基本操作的规格说明作进一步的求精，写出数据存储结构的类型定义，按照算法书写规范用类C语言写出函数形式的算法框架。在求精的过程中，应尽量避免陷入语言细节．不必过早表述辅助数据结构和局部变量。

## 3．编码实现和静态检查

编码是把详细设计的结果进一步求精为程序设计语言程序。程序的每行不要超过60个字符。每个函数体，即不计首部和规格说明部分，一般不要超过40行，最长不得超过60行．否则应该分割成较小的函数。要控制if语句连续嵌套的深度。其他要求参见算法书写规范。如何编写程序才能较快地完成调试是特别要注意的问题。对于编程很熟练的读者，如果基于详细设计的伪码算法就能直接在键盘上输入程序的话，则可以不必用笔在纸上写出编码．而将这一步的工作放在上机准备之后进行，即在上机调试之前直接用键盘输入。

然而，不管是否写出编码的程序，在上机之前，认真的静态检查是必不可少的。多数初学者在编好程序后处于以下两种状态之一：一种是对自己的“精心作品”的正确性确信不疑；另一种是认为上机前的任务已经完成，纠查错误是上机的工作。这两种态度是极为有害的。事实上，非训练有素的程序设计者编写的程序长度超过50行时，极少不含有除语法错误以外的错误。上机动态调试决不能代替静态检查，否则调试效率将是极低的。

静态检查主要有两种方法，一是用一组输入输出手工执行程序（通常应先分模块检查）；二是通过阅读或给别人讲解自己的程序而深入全面地理解程序逻辑，在这个过程中再加入一些注解和断言。如果程序中逻辑概念清楚，那么后者将肯定比前者有效。

## 4．上机准备和上机调试

上机准备包括以下几个方面：

（1）高级语言文本（体现于编译程序用户手册）的扩充和限制。例如，常用的Borland C（C++）和Turbo C（C++）与标准C（C++）的差别，以及相互之间的差别。（2）如果使用C或C++语言，要特别注意与教科书的类C语言之间的细微差别。（3）熟悉机器的操作系统和语言集成环境的用户手册，尤其是最常用的命令操作，以便顺利进行上机的基本活动。（4）掌握调试工具，考虑调试方案，设计输入输出并手工得出正确结果。“磨刀不误砍柴工”，学生应该能够熟练运用高级语言的程序调试器DEBUG调试程序。

上机调试程序时要带一本高级语言教材或手册。调试最好分模块进行，自底向上，即先调试底层函数，必要时可以另写一个调用驱动程序。这种表面上麻烦的工作实际上可以大大降低调试所面临的复杂性，提高调试工作效率。

在调试过程中可以不断借助DEGUB的各种功能，提高调试效率。调试中遇到的各种异常现象往往是预料不到的，此时不应“冥思苦想”，而应动手确定疑点，通过修改程序来证实它或绕过它。调试正确后，认真整理源程序及其注释，印出带有完整注释的且格式良好的源程序清单和结果。

## 5．总结和整理设计报告

这是程序调试完毕后必须做的工作，也是对自己成果的保证过程。

# 五、课程设计报告与算法书写规范

## 1．课程设计报告规范

设计报告的开头应给出题目、班级、姓名、学号和完成日期，并包括以下七个内容：

### A．需求分析

以无歧义的陈述说明程序设计的任务，强调的是程序要做什么？明确规定：

（l）输入的形式和输入值的范围；

（2）输出的形式；

（3）程序所能达到的功能；

（4）输入输出：包括正确的输入及其输出结果和含有错误的输入及其输出结果。

### B．概要设计

说明本程序中用到的所有抽象数据类型的定义、主程序的流程以及各程序模块之间的层次（调用）关系。

### C．详细设计

实现概要设计中定义的所有数据类型，对每个操作只需要写出伪码算法；对主程序和其他模块也都需要写出伪码算法（伪码算法达到的详细程度建议为：按照伪码算法可以在计算机键盘直接输入高级程序设计语言程序）；画出函数的调用关系图。

### D．调试分析

内容包括：

（l）调试过程中遇到的问题是如何解决的以及对设计与实现的回顾讨论和分析；

（2）算法的时空分析（包括基本操作和其他算法的时间复杂度和空间复杂度的分析）和改进设想；

（3）经验和体会等。

### E．用户使用说明

说明如何使用编写的程序，详细列出每一步的操作步骤。

### F．测试结果

列出测试结果，包括输入和输出。这里的输入输出应该完整和严格，最好多于需求分析中所列。

### G．附录

带注释的源程序。如果提交源程序软盘，可以只列出程序文件名的清单。值得注意的是，报告的各种文档资料，如：上述中的前三部分要在程序开发的过程中逐渐充实形成，而不是最后补写（当然也可以最后用实验报告纸誊清或打印）。

## 2．算法书写规范

### （1）算法说明

算法说明，也称为（算法）规格说明，是一个完整算法不可缺少的部分，应该在算法头（即过程或函数首部）之下以注释的形式写明如下内容：指明算法的功能；参数表中各参量的含意和输入输出属性；算法中引用了哪些全局变量或外部定义的变量，它们的作用、入口初值以及应满足哪些限制条件，例如，链表是否带头结点、表中元素是否有序、按递增还是递减方式有序等。必要时，算法说明还可用来陈述算法思想、采用的存储结构等．递归算法的说明特别重要，学生应该力求将它写成算法的严格定义。

算法说明应该在开始设计算法时就写明，可以在算法设计过程中作一些补充和修改，但是切忌最后补写。对于递归算法的情况，这一点尤其重要。这样做也是递归算法设计的正确而有效的途径，在算法设计（即解决一个问题）的过程中，能否利用自身的处理能力来解决所划分出的一个或几个子问题，全凭检查自身的规格说明而定。书写（递归）算法的规格说明时，应该忽略它如何实现或者假定它能够实现。如何实现的问题正是接下去要做的事。

算法说明书写得不好或不完全时，往往失去了评判一个算法正确与否的标准。书写恰当而又简洁的算法说明是一项具有很强技巧性的活动，通常要经过不断的练习才能达到。

### （2）注释与断言

在难懂的语句和关键的语句（段）之后加以注释可以大大提高程序的可读性。注释要恰当，并非越多越好；此外，注释句的抽象程度应略高于语句（段），例如，应避免用“i增加1”来注释语句“i++;”。

断言是注释的一种特殊写法，是一类特别重要的注释。它是一个逻辑谓词，陈述算法执行到这点时应满足的条件。多写断言式的注释，甚至以断言引导算法段的设计，是提高算法的结构良好性、避免错误和增强可读性的有效手段，是特别值得提倡的。其中最重要的是算法的入口断言和else 分支断言。注意，正确的算法也只能在输入参数值合法的前提下得出正确的结果。如果算法不含参数合法性检测代码段，书写入口断言是最低限度的要求。

### （3）输入和输出

算法的输入和输出可以通过三种途径实现。第一种是通过scanf 和printf 语句实现，其特点是实现了算法与计算环境外部的信息交换；第二种是以算法头中参数表里显式列出的参量作为输入／输出的媒介；第三种是通过全局变量或外部变量隐式地传递信息。后两种方法的特点是实现了一个算法与其调用者之间的信息交换。

如果一个算法是定义在某个数据结构之上的几个操作之一，该数据结构可以不列在算法的参数表中。在其他情况下，应尽量避免使用第三种方法。

### （4）错误处理

尽可能使用函数值返回算法的执行状态（正确／错误，或是错误代码等），便于调用者处理异常情况，有利于培养良好的程序设计习惯。

### （5）语句选用和算法结构

赋值语句、if分支语句和while（或for）循环语句是最基本的三种语句，仅用此三种语句就足以对付一切算法的设计了。实际上，不仅是“足够”，而且是“最好”。这样做对于提高结构良好性和可读性、避免逻辑错误是有益和有效的。switch分支语句是广义的if分支语句，在分支条件复杂时选用可以避免if 语句的多重嵌套，有助于提高算法的可读性，也是一个鼓励使用的语句。一般情况下，不准使用goto语句，个别的特殊情况除外。

算法设计过程中应尽量避免下列所示的语句结构：

do

do {

…

} while

…

} while …

或者

if ( … )

if ( … )

对于第二种情形，如果难以改变，应该对第二个if语句加上一对语句括号，以便明确条件成立时的作用范围。此外，语句的开／闭括号应齐全。

### （6）基本运算

如果题目中未明确要求用某种数据结构上的基本运算编写算法，不得直接利用教科书中给出的基本运算。如果非用不可，则要求将所用到的所有基本运算同时实现。

### （7）几点建议

建议以流程图说明算法；

建议在算法书写完毕后，用边界条件的输入参数值验证一下算法能否正确执行。

由于机房的特殊条件限制，建议将编辑好的程序存放在自带的闪存盘或者移动硬盘上，以免数据损失。

# 六、课程改革特色

与传统做法不同，本设计题目非常详细，这并不会限制学生的想象力，而是要使学生在无形中学会模仿，把思路引上正轨，避免坏结构程序和习惯，使某些难点和重点不会被绕过去，且便于教学检查。此外，严格的规范不但能帮助学生有效避免程序设计中的种种错误观念，更重要的是有利于培养软件工作者不可缺少的科学工作方法和作风。

# 七、课程设计评分标准

**管理学院课程设计评分表**

|  |  |
| --- | --- |
| **评 定 项 目** | **评分成绩** |
| 1．设计原创性达到80%（30分） |  |
| 2．程序执行结果实现指导书中的所有基本要求（20分） |  |
| 3．设计结果（软件程序）正确（10分） |  |
| 4．设计报告的规范化（10分） |  |
| 5. 答辩（30分） |  |
| 总分（100分） |  |

备注：成绩等级：优（90分—100分）、良（80分—89分）、中（70分—79分）、及格（60分—69分）、60分以下为不及格。若第一项低于20分，或第一项与第五项之和低于35分，则课程设计成绩不能及格。

# 八、教材及参考书

**教材：**

严蔚敏，吴伟民编著，数据结构（C语言版），清华大学出版社，2007年4月。

**参考书：**

[1]严蔚敏，陈文博编著，数据结构及应用算法教程，清华大学出版社，2001年2月第1版。

[2] 赵逢禹，罗道昆，路玲，杜光耀编著，数据结构与C语言高级程序设计，北京航空航天大学出版社，1998年4月出版。

执笔：燕翔 日期：2017.5

审阅：江长斌 日期：2017.5

审定：王超 日期：2017.5

# 附录一：导出群成员名单

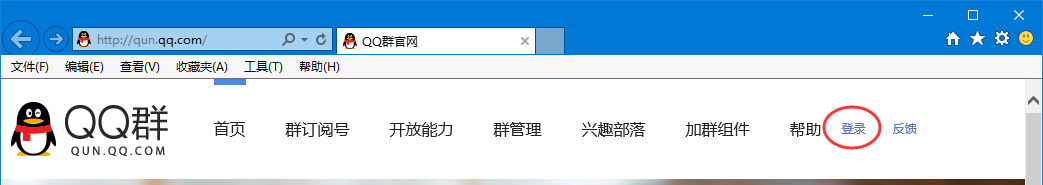
1. 打开浏览器，输入<http://qun.qq.com/>，点击“登录”，进入群空间。  
   

图 1

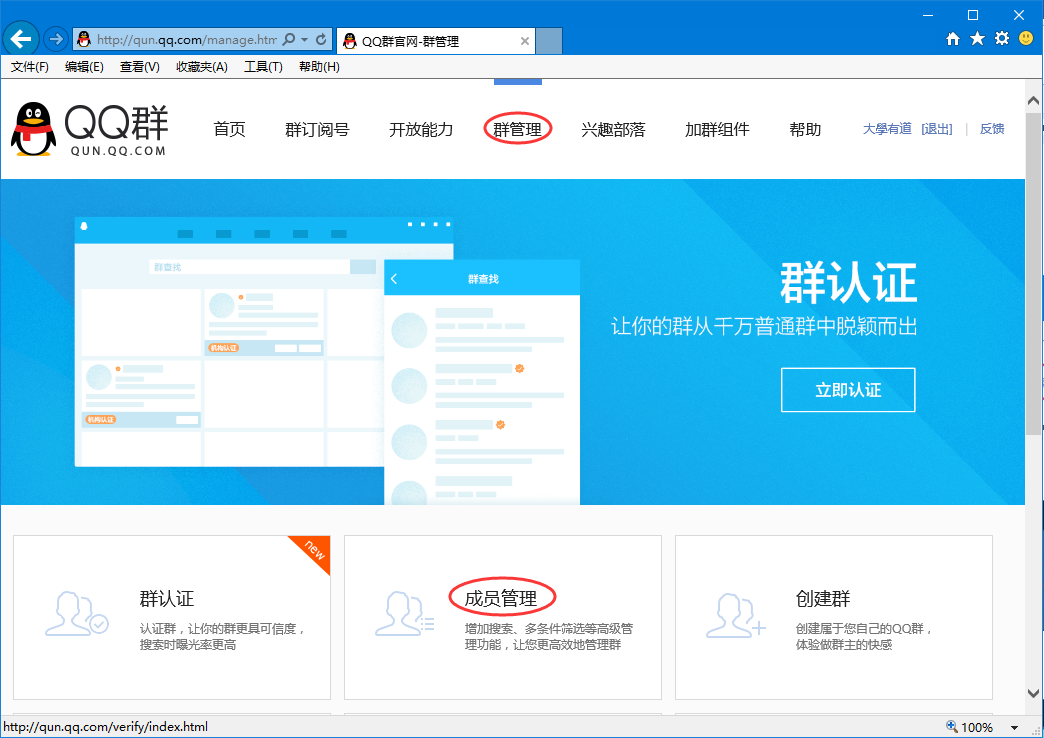
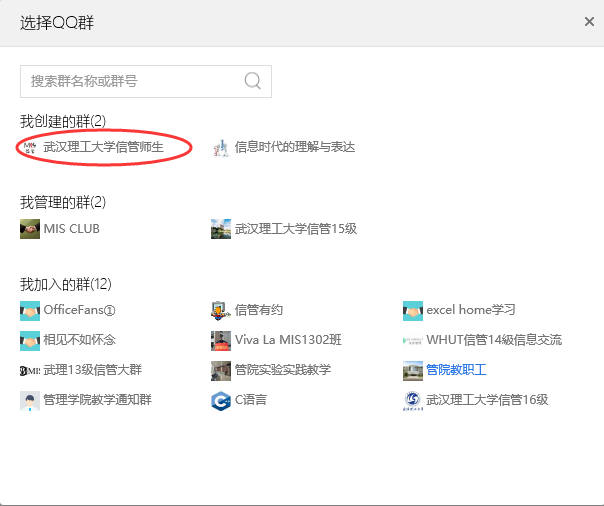
1. 在群空间页面上点击下图所示的“群管理”图标，再点击“成员管理”，选择想查看的群，例如“武汉理工大学信管师生”群，进入群成员清单界面。  
     
   

图 2

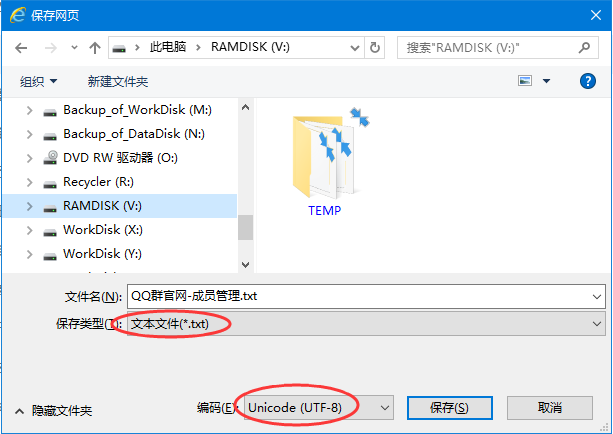
1. 由于网页元素的特殊性，需要按住Ctrl+End键不放，一直到所有成员都显示出来，然后在当前界面下按Ctrl+S键，弹出保存网页窗口，更改保存类型为文本文件，保存之。  
   

图 3

1. 这样，我们就得到了一个群的成员名单文件。里面的内容大体上如下：  
   

图 4

# 附录二：导出群聊记录

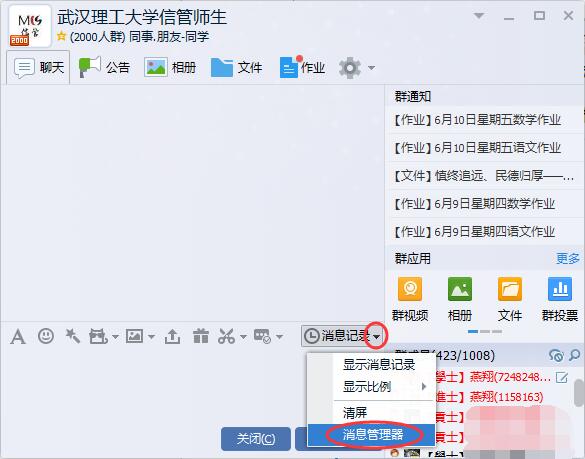
1. 打开群聊界面（此界面因QQ版本不同而有所不同），点击下图所示的图标（要点“消息记录”右边的黑色向下小三角，点错了就看不到下拉菜单，而是直接在右边看到聊天记录了），将会启动消息管理器。  
   

图 5

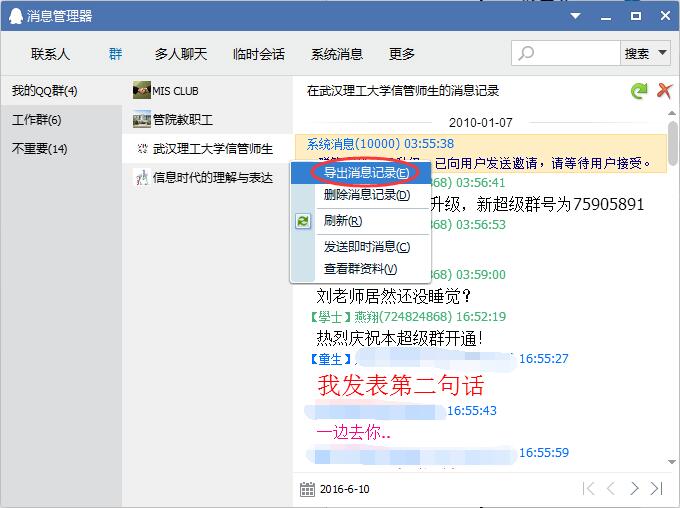
1. 在消息管理器界面中选中要导出记录的群，点右键，选择导出消息记录（注意不要在“工具”选项中点击然后选择“导出全部消息记录”，否则，你的隐私也将暴露在记录里面）：  
   

图 6

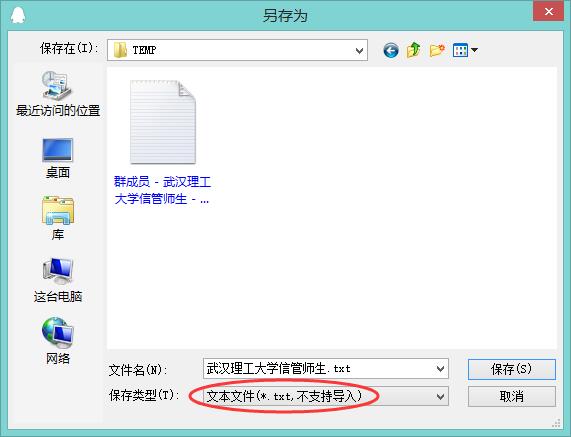
1. 在“另存为”窗口中，将保存类型改为文本文件，保存之。  
   

图 7

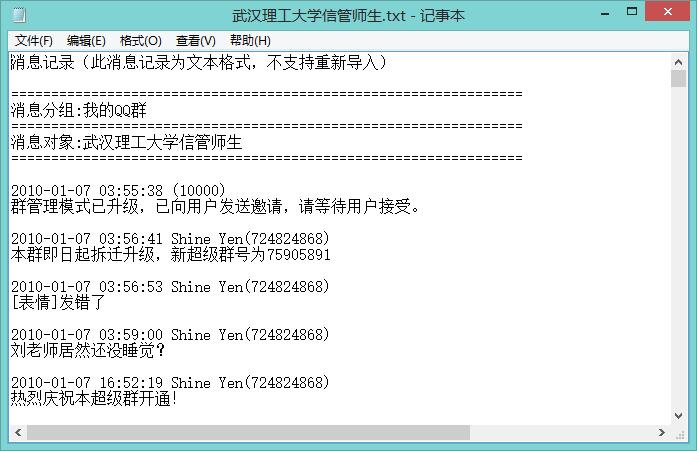
1. 这样，我们就得到了一个群的聊天记录文件。里面的内容大体上如下：  
   

图 8