

# 课程设计题目（2021 版）

## 必做题 1-6:

### 1、区块链（必做）（链表）

#### [问题描述]

使用链表设计一个保存信息的系统,该系统拥有类似区块链的设计以防止信息被轻易篡改。

该题目使用一个链表。信息保存在链表的每一个节点中,每个节点需要包含该节点的编号、信息和校验码。其中:

- + 每个节点的编号按照顺序递增,从 0 开始。
- + 节点中包含的信息是字符串,且每个字符的 ASCII 码范围为 0-127,以\0 结束。
- + 每个节点的校验码等于上一个节点的校验码+本节点的节点编号+本节点信息中字符串 ASCII 码之和 mod 113。
- + 首个节点的校验码则是本节点信息中字符串 ASCII 码之和 mod 113。
- + 有效的链表要求所有节点的校验码都能够成功按照上述算法得出。

#### [基本要求]

- (1) 要求从文本文件中输入;
  - (2) 给定链表,检查链表是否有效。若无效,输出首个无效节点的节点编号;
  - (3) 允许向链表中添加信息,要求保证链表始终有效;
  - (4) 篡改一个有效的链表中特定编号的节点信息内容,保持篡改后的链表仍然有效。
- 注意,可能需要篡改多个节点以达到此要求。

### 2、迷宫问题（必做）（栈与队列）

#### [问题描述]

利用栈操作实现迷宫问题求解。

#### [基本要求]

- (1) 从文件中读取数据,生成模拟迷宫地图,不少于 20 行 20 列。
- (2) 给出任意入口和出口,显示输出迷宫路线。

### 3、JSON 查找（必做）（树）

#### [问题描述]

JSON (JavaScript Object Notation) 是一种轻量级的数据交换格式,可以用来描述半结构化的数据。JSON 格式中的基本单元是值 (value),出于简化的目的本题只涉及 2 种类型的值:

\* 字符串 (string): 字符串是由双引号 " 括起来的一组字符 (可以为空)。如果字符串的内容中出现双引号 ", 在双引号前面加反斜杠,也就是用 \" 表示;如果出现反斜杠 \, 则用两个反斜杠 \\ 表示。反斜杠后面不能出现 " 和 \ 以外的字符。例如: ""、"hello"、"\""

\* 对象 (object): 对象是一组键值对的无序集合 (可以为空)。键值对表示对象的

属性，键是属性名，值是属性的内容。对象以左花括号 { 开始，右花括号 } 结束，键值对之间以逗号 , 分隔。一个键值对的键和值之间以冒号 : 分隔。键必须是字符串，同一个对象所有键值对的键必须两两都不相同；值可以是字符串，也可以是另一个对象。例如：{}、{"foo": "bar"}、{"Mon": "weekday", "Tue": "weekday", "Sun": "weekend"}。

除了字符串内部的位置，其他位置都可以插入一个或多个空格使得 JSON 的呈现更加美观，也可以在一些地方换行，不会影响所表示的数据内容。例如，上面举例的最后一个 JSON 数据也可以写成如下形式。

```
{
  "Mon": "weekday",
  "Tue": "weekday",
  "Sun": "weekend"
}
```

给出一个 JSON 格式描述的数据，以及若干查询，编程返回这些查询的结果。

输入格式

第一行是两个正整数 n 和 m，分别表示 JSON 数据的行数和查询的个数。

接下来 n 行，描述一个 JSON 数据，保证输入是一个合法的 JSON 对象。

接下来 m 行，每行描述一个查询。给出要查询的属性名，要求返回对应属性的内容。需要支持多层查询，各层的属性名之间用小数点 . 连接。保证查询的格式都是合法的。

[基本要求]

输出格式

对于输入的每一个查询，按顺序输出查询结果，每个结果占一行。

如果查询结果是一个字符串，则输出 STRING <string>，其中 <string> 是字符串的值，中间用一个空格分隔。

如果查询结果是一个对象，则输出 OBJECT，不需要输出对象的内容。

如果查询结果不存在，则输出 NOTEXIST。

样例输入

```
10 5
{
  "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "2ndStreet",
    "city": "NewYork",
    "state": "NY"
  },
  "esc\\aped": "\\\"hello\\\""
}
```

firstName

address

address.city

address.postal

esc\aped

样例输出

STRING John  
OBJECT  
STRING NewYork  
NOTEXIST  
STRING "hello"

[基本要求]

- (1) 要求从文本文件中输入；
- (2) 本题目其实就是一棵普通的树（即每个结点的孩子数不固定，不能单纯采用  $n$  叉树来解决），可以考虑使用孩子兄弟表示法等进行表示和存储；
- (3) 严格按照要求的输入输出格式进行数据的输入、输出（训练 CSP 考试中的格式化输入输出的正确性）；

#### 4、【4】公交线路提示（必做）（图）

[问题描述]

上网下载真实南京公交线路图，建立南京主要公交线路图的存储结构。

[基本要求]

- (1) 输入任意两站点，给出转车次数最少的乘车路线。
- (2) 输入任意两站点，给出经过站点最少的乘车路线。

#### 5、Hash 表应用（必做）（查找）

[问题描述]

设计散列表实现 VIP 客户发掘。对身份证号进行 Hash，通过对乘客某时间段内的乘机频率、里程数统计，发掘 VIP 客户。

[基本要求]

- (1) 设每个记录有下列数据项：身份证号码（虚构，位数和编码规则与真实一致即可）、姓名、航班号、航班日期、里程。
- (2) 从文件输入各记录，以身份证号码为关键字建立散列表。
- (3) 分别采用开放定址（自行选择和设计定址方案）和链地址两种方案解决冲突；显示发生冲突的次数、每次中解决冲突进行重定位的次数。
- (4) 记录条数至少在 100 条以上。
- (5) 从记录中实现乘客乘机频率、里程数统计，从而发掘 VIP 客户。

#### 6、排序算法比较（必做）（排序）

[问题描述]

利用随机函数产生 10 个样本，每个样本有 50000 个随机整数（并使第一个样本是正序，第二个样本是逆序），利用直接插入排序、希尔排序，冒泡排序、快速排序、选择排序、堆排序，归并排序、基数排序 8 种排序方法进行排序（结果为由小到大的顺序），并统计每一种排序算法对不同样本所耗费的时间。

[基本要求]

- (1) 原始数据存在文件中，用相同样本对不同算法进行测试；
- (2) 屏幕显示每种排序算法对不同样本所花的时间；

## 选做题 7-20:

类型一：CSP 类型选做题（最多计 4 分）

### 7、【2】地铁修建（选做）（图）

[问题描述]

A 市有  $n$  个交通枢纽，其中 1 号和  $n$  号非常重要，为了加强运输能力，A 市决定在 1 号到  $n$  号枢纽间修建一条地铁。

地铁由很多段隧道组成，每段隧道连接两个交通枢纽。经过勘探，有  $m$  段隧道作为候选，两个交通枢纽之间最多只有一条候选的隧道，没有隧道两端连接着同一个交通枢纽。

现在有  $n$  家隧道施工的公司，每段候选的隧道只能由一个公司施工，每家公司施工需要的天数一致。而每家公司最多只能修建一条候选隧道。所有公司同时开始施工。

作为项目负责人，你获得了候选隧道的信息，现在你可以按自己的想法选择一部分隧道进行施工，请问修建整条地铁最少需要多少天。

输入格式

输入的第一行包含两个整数  $n, m$ ，用一个空格分隔，分别表示交通枢纽的数量和候选隧道的数量。

第 2 行到第  $m+1$  行，每行包含三个整数  $a, b, c$ ，表示枢纽  $a$  和枢纽  $b$  之间可以修建一条隧道，需要的时间为  $c$  天。

[基本要求]

输出格式

输出一个整数，修建整条地铁线路最少需要的天数。

样例输入

```
6 6
1 2 4
2 3 4
3 6 7
1 4 2
4 5 5
5 6 6
```

样例输出

```
6
```

样例说明

可以修建的线路有两种。

第一种经过的枢纽依次为 1, 2, 3, 6，所需要的时间分别是 4, 4, 7，则整条地铁线需要 7 天修完；

第二种经过的枢纽依次为 1, 4, 5, 6，所需要的时间分别是 2, 5, 6，则整条地铁

线需要 6 天修完。

第二种方案所用的天数更少。

## 8、【2】URL 映射（选做）（字符串）

### 问题描述

URL 映射是诸如 Django、Ruby on Rails 等网页框架（web frameworks）的一个重要组件。对于从浏览器发来的 HTTP 请求，URL 映射模块会解析请求中的 URL 地址，并将其分派给相应的处理代码。现在，请你来实现一个简单的 URL 映射功能。

本题中 URL 映射功能的配置由若干条 URL 映射规则组成。当一个请求到达时，URL 映射功能会将请求中的 URL 地址按照配置的先后顺序逐一与这些规则进行匹配。当遇到第一条完全匹配的规则时，匹配成功，得到匹配的规则以及匹配的参数。若不能匹配任何一条规则，则匹配失败。

本题输入的 URL 地址是以斜杠 / 作为分隔符的路径，保证以斜杠开头。其他合法字符还包括大小写英文字母、阿拉伯数字、减号 -、下划线 \_ 和小数点 .。例如，/person/123/ 是一个合法的 URL 地址，而 /person/123? 则不合法（存在不合法的字符问号 ?）。另外，英文字母区分大小写，因此 /case/ 和 /Case/ 是不同的 URL 地址。

对于 URL 映射规则，同样是以斜杠开始。除了可以是正常的 URL 地址外，还可以包含参数，有以下 3 种：

字符串：用于匹配一段字符串，注意字符串里不能包含斜杠。例如，abcde0123。

整数：用于匹配一个不带符号的整数，全部由阿拉伯数字组成。例如，01234。

路径：用于匹配一段字符串，字符串可以包含斜杠。例如，abcd/0123/。

以上 3 种参数都必须匹配非空的字符串。简便起见，题目规定规则中 和 前面一定是斜杠，后面要么是斜杠，要么是规则的结束（也就是该参数是规则的最后一部分）。而 的前面一定是斜杠，后面一定是规则的结束。无论是 URL 地址还是规则，都不会出现连续的斜杠。

### 输入格式

输入第一行是两个正整数  $n$  和  $m$ ，分别表示 URL 映射的规则条数和待处理的 URL 地址个数，中间用一个空格字符分隔。

第 2 行至第  $n+1$  行按匹配的先后顺序描述 URL 映射规则的配置信息。第  $i+1$  行包含两个字符串  $p_i$  和  $r_i$ ，其中  $p_i$  表示 URL 匹配的规则， $r_i$  表示这条 URL 匹配的名字。两个字符串都非空，且不包含空格字符，两者中间用一个空格字符分隔。

第  $n+2$  行至第  $n+m+1$  行描述待处理的 URL 地址。第  $n+1+i$  行包含一个字符串  $q_i$ ，表示待处理的 URL 地址，字符串中不包含空格字符。

### 样例输入

```
5 4
/articles/2003/ special_case_2003
/articles/<int>/ year_archive
/articles/<int>/<int>/ month_archive
/articles/<int>/<int>/<str>/ article_detail
/static/<path> static_serve
/articles/2004/
/articles/1985/09/aloha/
```

/articles/hello/  
/static/js/jquery.js

样例输出

year\_archive 2004

article\_detail 1985 9 aloha  
404

static\_serve js/jquery.js

样例说明

对于第 1 个地址 /articles/2004/，无法匹配第 1 条规则，可以匹配第 2 条规则，参数为 2004。

对于第 2 个地址 /articles/1985/09/aloha/，只能匹配第 4 条规则，参数依次为 1985、9（已经去掉前导零）和 aloha。

对于第 3 个地址 /articles/hello/，无法匹配任何一条规则。

对于第 4 个地址 /static/js/jquery.js，可以匹配最后一条规则，参数为 js/jquery.js。

数据规模和约定

$1 \leq n \leq 100, 1 \leq m \leq 100$ 。

所有输入行的长度不超过 100 个字符（不包含换行符）。

保证输入的规则都是合法的。

## 9、【3】行车路线（选做）（图）

[问题描述]

小明和小芳出去乡村玩，小明负责开车，小芳来导航。

小芳将可能的道路分为大道和小道。大道比较好走，每走 1 公里小明会增加 1 的疲劳度。小道不好走，如果连续走小道，小明的疲劳值会快速增加，连续走  $s$  公里小明会增加  $s^2$  的疲劳度。

例如：有 5 个路口，1 号路口到 2 号路口为小道，2 号路口到 3 号路口为小道，3 号路口到 4 号路口为大道，4 号路口到 5 号路口为小道，相邻路口之间的距离都是 2 公里。如果小明从 1 号路口到 5 号路口，则总疲劳值为  $(2+2)^2+2+2^2=16+2+4=22$ 。

现在小芳拿到了地图，请帮助她规划一个开车的路线，使得按这个路线开车小明的疲劳度最小。

[基本要求]

输入格式：

输入的第一行包含两个整数  $n, m$ ，分别表示路口的数量和道路的数量。路口由 1 至  $n$  编号，小明需要开车从 1 号路口到  $n$  号路口。

接下来  $m$  行描述道路，每行包含四个整数  $t, a, b, c$ ，表示一条类型为  $t$ ，连接  $a$  与  $b$  两个路口，长度为  $c$  公里的双向道路。其中  $t$  为 0 表示大道， $t$  为 1 表示小道。保证 1 号路口和  $n$  号路口是连通的。

输出格式

输出一个整数，表示最优路线下小明的疲劳度。

样例输入

6 7

1 1 2 3

1 2 3 2  
0 1 3 30  
0 3 4 20  
0 4 5 30  
1 3 5 6  
1 5 6 1

样例输出

76

样例说明

从 1 走小道到 2，再走小道到 3，疲劳度为  $5^2=25$ ；然后从 3 走大道经过 4 到达 5，疲劳度为  $20+30=50$ ；最后从 5 走小道到 6，疲劳度为 1。总共为 76。

课程设计要求：

- (1) 要求从文本文件中输入；
- (2) 采用适当的数据结构存储由输入数据中的道路所形成的图结构；
- (3) 编写尽可能优的算法，处理好连续走小道造成的疲劳值的指数增长（提示：基于迪杰斯特拉算法进行改进即可完成本题）；
- (4) 除严格按题目要求进行输出以外，还要求输出最优路线的路径，以及从出发点到各个点的最小疲劳值。

## 10、【1】 社交网络图中结点的“重要性”计算（选做）（图）

[问题描述]

在社交网络中，个人或单位（结点）之间通过某些关系（边）联系起来。他们受到这些关系的影响，这种影响可以理解为网络中相互连接的结点之间蔓延的一种相互作用，可以增强也可以减弱。而结点根据其所处的位置不同，在网络中体现的重要性也不尽相同。

“紧密度中心性”是用来衡量一个结点达到其他结点的“快慢”的指标，即一个有较高中心性的结点比有较低中心性的结点能够更快地（平均意义下）到达网络中的其他结点，因而在该网络的传播过程中有更重要的价值。在有  $N$  个结点的网络中，结点  $v_i$  的“紧密度中心性”  $Cc(v_i)$  数学上定义为  $v_i$  到其余所有结点  $v_j (j \neq i)$  的最短距离

$d(v_i, v_j)$  的平均值的倒数：

$$Cc(v_i) = \left[ \frac{1}{N-1} \sum_{i \neq j}^N d(v_i, v_j) \right]^{-1} = \frac{N-1}{\sum_{i \neq j}^N d(v_i, v_j)}$$

对于非连通图，所有结点的紧密度中心性都是 0。

本实验给定一个无权的无向图以及其中的一组结点，要求计算这组结点中每个结点的紧密度中心性。

[基本要求]

(1) 输入说明：输入的第一行给出两个正整数  $N$  ( $N \leq 1000$ ) 和  $M$ ，其中  $N$  是图中结点数，结点编号从 1 到  $N$ ； $M$  ( $M \leq 10000$ ) 是边的数目。随后的  $M$  行中，每行给出一

条边的信息，即该边连接的两个结点编号，中间用空格分隔。最后一行给出需要计算紧密度中心性的这组结点的个数  $K$  ( $K \leq 100$ )，以及  $K$  个结点的编号，用空格分隔。

(2) 输出说明：按照 “ $Cc(i)=x.xx$ ” 的格式输出  $K$  个给定结点的紧密度中心性，每个输出占一行，结果精确到小数点后 2 位。

(3) 测试用例：

```
输入  5 8
      1 2
      1 3
      1 4
      2 3
      3 4
      4 5
      2 5
      3 5
      2 4 3
输出  Cc(4) = 0.80
      Cc(3) = 1.00
```

## 类型二：系统设计型（无上限）

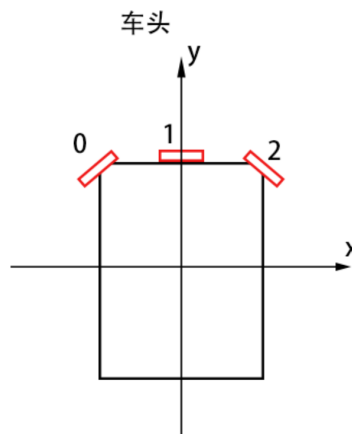
### 11、【3】障碍物检测（选做）（综合）

[问题描述]

毫米波雷达具有多目标探测能力，可以应用于无人驾驶技术上。现有一辆无人车，车头装有三个毫米波雷达，编号  $id$  分别为：0. 左；1. 中；2. 右。保证三个雷达的探测范围互不重叠。

该组雷达每隔一定时间（一个时间片）上报一次数据，我们可以从中判断当前无人车前方是否存在障碍物

车体坐标示意如下：



[输入格式]

从标准输入读入数据。

第 1 行包含一个正整数  $N$ ，表示上报的数据次数。

接下来将有  $N$  行，每行只包含一个正整数，表示当前三个雷达所检测到的有效障碍



物数量  $m$ 。若  $m > 0$ ，则在该行后紧接着有  $m$  行障碍物信息数据。每一行信息数据中包含用空格分隔的 6 个整数  $i$ 、 $t$ 、 $dx$ 、 $dy$ 、 $sx$ 、 $sy$ 。 $i$  代表雷达编号； $t$  代表障碍物 ID（范围为 0-65535）； $dx$  代表障碍物在车体坐标系中的  $x$  坐标，单位为  $cm$ ； $dy$  代表障碍物在车体坐标系中的  $y$  坐标，单位为  $cm$ ； $sx$  代表障碍物相对于车体的相对速度的  $x$  分量； $sy$  代表障碍物相对于车体的相对速度的  $y$  分量。

[输出格式]

输出到标准输出。

输出一个包含  $N$  个元素的序列字符串，元素与元素之间用空格隔开。其中 0 代表无障碍物，1 代表有障碍物。

障碍物存在的判断依据如下：

1. 同一障碍物在 10 个时间片内出现 3 次以；上
2. 只要有一个或多个障碍物满足条件 1，都可判断为障碍物存在；
3. 障碍物与车体坐标中心点的距离需小于 50m 方可判断为有效障碍物；
4. 障碍物相对于车体的相对速度的  $y$  分量需大于  $3m/s$ ；
5. 三个雷达中只要有一个雷达出现有效障碍物即可判断障碍物存在。

[样例 1 输入]

```
10
0
0
0
0
1
0 64 940 430 230 60
1
0 64 870 428 232 58
1
0 64 842 411 214 49
1
0 64 808 422 220 53
1
0 64 771 409 231 57
1
0 64 744 413 224 58
0
```

[样例 1 输出]

```
0 0 0 0 0 1 1 1 1 0
```

[样例 2 输入]

```
5
1
0 64 940 430 230 60
2
0 64 870 402 232 58
1 39 340 -173 567 -4
2
0 64 828 379 170 40
```

1 39 412 -180 559 -1

1

1 39 473 -191 570 -6

0

[样例 2 输出]

0 0 0 1 0

[样例 2 解释]

编号为 64 的障碍物虽然出现在前三个时间片，但是前后分量的速度小于 3m/s。

## 12、连连看（必做）（图）

[问题描述]

建立一个 10\*20 的矩形方格图，其中有 10 种不同的图案，每种图案个数为偶数，填满矩形方格图。

[基本要求]

(1) 随机产生原始数据；

(2) 输入两个位置，如果两者图案相同，并且可以用小于等于 3 条直线相连，即可消除该两个图案。

## 13、【3】基于角色的访问控制（选做）（综合）

[问题描述]

基于角色的访问控制 (Role-based access control, RBAC)，是一种较新且广为使用的资源访问控制机制，本题是对该控制机制的简化。简化版的 RBAC 有用户和用户组两个核心概念，通过将用户添加到用户组中，并将所需的访问权限赋予用户组，从而实现快速灵活地对大量用户进行权限管理。

[基本要求]

(1) 要求从文本文件中输入；

(2) 用户信息包含以下内容：用户编号、姓名。用户组信息包含以下内容：用户组编号、用户组称呼、用户组职责。特别注意，用户组可以拥有不止一种权限；

(3) 按照姓名查询，输出用户信息（包括其本人所属的所有用户组）；

(4) 显示某特定用户组中的所有用户；

(5) 对某用户组添加用户；

(6) 对某用户组删除用户；

(7) 修改某用户或用户组的信息；

(8) 两个用户组之间允许通过继承的方式共享用户和职责；

[继承举例]

系统中有“组长”和“组员”两个用户组。其中“组长”继承了“组员”的所有职责。将某用户加入到“组长”用户组中，在输出该用户所拥有的职责时，需要输出“组长”和“组员”的所有职责。同时输出“组员”用户组的成员时，也需要输出该用户。

## 14、【4】B-树应用（选做）（查找）

[问题描述]

设计并实现 B-树的动态查找。

[基本要求]

- (1) 从文件读取数据
- (2) 实现 B-树的三种基本功能：查找、插入和删除。
- (3) 以可验证的方式输出结果

## 15、【1】算术表达式求值（选做）（栈）

[问题描述]

一个算术表达式是由操作数(operand)、运算符(operator)和界限符(delimiter)组成的。假设操作数是正实数，运算符只含加减乘除等四种运算符，界限符有左右括号和表达式起始、结束符“#”，如：#6+15\*(21-8/4)#。引入表达式起始、结束符是为了方便。编程利用“运算符优先法”求算术表达式的值。

[基本要求]

- (1) 从键盘或文件读入一个合法的算术表达式，输出正确的结果。
- (2) 显示输入序列和栈的变化过程。
- (3) 考虑算法的健壮性，当表达式错误时，要给出错误原因的提示。
- (4) 实现非整数的处理(\*)。

## 16、【3】词梯（选做）（综合）

[问题描述]

词梯：给定两个长度相等的单词，一个作为起点，一个作为终点。每次只改变一个字母，最终，作为起点的单词可以转变为终点的单词。

例如：code→cade→cate→date→data

词梯不是唯一的，如 work→play：

work fork form foam flam flay play  
work pork perk peak pean plan play  
work pork perk peak peat plat play  
work pork perk pert peat plat play  
work pork porn pirn pian plan play  
work pork port pert peat plat play  
work word wood pood plod ploy play  
work worm form foam flam flay play  
work worn porn pirn pian plan play  
work wort bort boat blat plat play  
work wort port pert peat plat play  
work wort wert pert peat plat play

[基本要求]

- (1) 从 <https://github.com/dwyl/english-words> 中下载 words.txt 作为词汇表；
- (2) 给定两个长度相等的单词，一个作为起点，一个作为终点，寻找长度最短的词梯。

## 17、【2】购物网站信息管理（选做）（线性表）

[问题描述]

设计一个程序,对商铺信息管理,商铺信息包括:商铺编号,商铺名,信誉度(0-5), (商品名称 1, 价格 1, 销量 1), (商品名称 2, 价格 2, 销量 2), (商品名称 3, 价格 3, 销量 3) ...。

商品名称包括(毛巾, 牙刷, 牙膏, 肥皂, 洗发水, 沐浴露等 6 种以上商品), 每个商铺具有其中事先确定若干商品及价格, 由文件输入, 销量初始为 0。

[基本要求]

(1) 建立一个单向链表存储所有商铺信息(至少 30 个), 以编号为序, 编号从 1 开始递增, 从文件中读取数据, 并能将数据存储到文件。商铺信息结点的数据结构自行设计。

(2) 可以增、删商铺。增加商铺, 编号自动加一, 插入链表尾部; 删除商铺, 以编号为准, 并修改后续结点的编号, 保持编号连续性。可增、删商品。

(3) 查询某一种商品名称, 建立一个双向循环链表, 结点信息是包含该商品的所有商铺编号、商铺名、信誉度、商品名称、价格、销量, 以信誉度从高至低, 并按销量排序, 并逐一显示。

(4) 购买某一商铺的商品, 修改单向链表中商品的信息的销量。

(5) 建立一个顺序结构, 按商品名 Hash 分配地址, 存储当前每种商品总销量并输出。

(6) 任何的商铺信息变化, 实现文件存储。

## 18、【4】平衡二叉树操作的演示(选做)(查找)

[问题描述]

利用平衡二叉树实现一个动态查找表。

[基本要求]

(1) 从文件读取数据

(2) 实现动态查找表的三种基本功能: 查找、插入和删除。

(3) 以可验证的方式输出结果

## 19、【2】Huffman 编码与解码(选做)(树)

[问题描述]

对一篇不少于 5000 字符的英文文章(source.txt), 统计各字符出现的次数, 实现 Huffman 编码(code.dat), 以及对编码结果的解码(recode.txt)。

[基本要求]

(1) 输出每个字符出现的次数和编码, 并存储文件(Huffman.txt)。

(2) 在 Huffman 编码后, 英文文章编码结果保存到文件中(code.dat), 编码结果必须是二进制形式, 即 0 1 的信息用比特位表示, 不能用字符'0'和'1'表示。

(3) 实现解码功能。

## 20、【3】家谱管理系统(选做)(树)

[问题描述]

实现具有下列功能的家谱管理系统。

[基本要求]

(1) 输入文件以存放最初家谱中各成员的信息, 成员的信息中均应包含以下内容: 姓

名、出生日期、婚否、地址、健在否、死亡日期（若其已死亡），也可附加其它信息，但不是必需的。

(2) 实现数据的文件存储和读取。

(3) 以图形方式显示家谱。

(4) 显示第 n 代所有人的信息。

(5) 按照姓名查询，输出成员信息（包括其本人、父亲、孩子的信息）。

(6) 按照出生日期查询成员名单。

(7) 输入两人姓名，确定其关系。

(8) 某成员添加孩子。

(9) 删除某成员（若其还有后代，则一并删除）。

(10) 修改某成员信息。

(11) 要求建立至少 40 个成员的数据，以较为直观的方式显示结果，并提供文稿形式以便检查。

(12) 界面要求：有合理的提示，每个功能可以设立菜单，根据提示，可以完成相关的功能要求。

(13) 存储结构：根据系统功能要求自行设计，但是要求相关数据要存储在数据文件中。  
测试数据：要求使用 1、全部合法数据；2、局部非法数据。进行程序测试，以保证程序的稳定。

**注：选做题中标注，为该题分值。如【3】，表示分值为 3。**

#### **成绩评定细则：**

1. 正确性、功能的完备性和总程序量：程序是否可以运行，结果是否正确，是否实现要求的所有子功能，是否达到足够的程序代码（50 分）  
    必做题 && 选做题分值 12 以上      (40 分-50 分) (优秀)  
    必做题 && 选做题分值 6 以上      (30 分-39 分) (良好)  
    必做题 && 选做题分值 3 以上      (20 分-29 分) (中等)  
    必做题                               (10 分-19 分) (及格)
  2. 课程设计报告中的算法说明的清晰程度，课程设计报告中总结的深刻程度（20 分）
  3. 独立完成情况（30 分）
- 总计：100 分 以五分制（优、良、中、及格、不及格）对应。
4. 根据完成情况，符合要求，提出申请，才可参评优秀。

#### **加分项目：**

1. 健壮性：异常处理的情况
2. 可读性：代码编写是否规范，是否便于阅读。如函数、变量命名，‘{ }’的缩进，关键位置适量注释等
3. 功能的完善：除要求实现的功能外，完成了其它的功能，实现了功能的完善
4. 界面的设计：良好交互的界面

编程语言：C、C++ 等

检查方式：

1. 总体上检查程序的代码量，正确性，可读性，健壮性，功能的完备性，程序的结构是否合理；局部检查三个以上函数块
2. 检查程序的运行情况

时间安排包括：

- 1 上机时间安排：待定
- 2 课程设计报告电子文档上交时间：待定
- 3 课程设计检查时间：待定

课程设计报告要求：

1. 所有的课程设计报告，均要有封面，包括：数据结构课程设计、班级、学号、姓名、和指导教师；
2. 目录；
3. 对每一题，给出自己采用的数据结构；
4. 给出算法设计思想；
5. 给出实现的源程序，并在必要的代码处给出注释；
6. 给出测试数据和结果；
7. 给出算法的时间复杂度、另外可以提出算法的改进方法；
8. 结束语：说明总体完成情况，每一题的程序代码行，总代码行，心得体会；