

淮海工学院计算机工程学院

成绩:

设计名称:	数据结构课程设计
选题名称:	汽车牌照管理系统
姓名:	学号:
专业班级:	计算机科学与技术 G 计 111
系(院):	计算机工程学院
设计时间:	2012.12.24~2013.1.4
设计地点:	软件工程实验室、教室

课程设计报告

签名:

年月

I. 课程设计目的

- 1、训练学生灵活应用所学数据结构知识，独立完成问题分析，结合数据结构理论知识，编写程序求解指定问题。
2. 初步掌握软件开发过程的问题分析、系统设计、程序编码、测试等基本方法和技能；
3. 提高综合运用所学的理论知识和方法独立分析和解决问题的能力；

4. 训练用系统的观点和软件开发一般规范进行软件开发，巩固、深化学生的理论知识，提高编程水平， 并在此过程中培养他们严谨的科学态度和良好的工作作风。

2. 课程设计任务与要求:

任务

根据教材《数据结构 -C 语言描述》（耿国华主编）和参考书《数据结构题集（ C 语言版）》（严蔚敏、吴伟民主编）选择课程设计题目，要求通过设计，在数据结构的逻辑特性和物理表示、数据结构的选用应用、 算法的设计及其实现等方面加深对课程基本内容的理解和综合运用。

设计题目从任务书所列选题表中选取，每班每题不得超过 2 人。

学生自选课题 学生原则上可以结合个人爱好自选课题，要求课题有一定的深度与难度，有一定的算法复杂性，能够 巩固数据结构课程所学的知识。学生自选课题需在 18 周前报课程设计指导教师批准方可生效。

要求:

1、 在处理每个题目时，要求从分析题目的需求入手，按设计抽象数据类型、构思算法、通过设计实现 抽象数据类型、编制上机程序和上机调试等若干步骤完成题目，最终写出完整的分析报告。前期准备工作 完备与否直接影响到后序上机调试工作的效率。在程序设计阶段应尽量利用已有的标准函数，加大代码的 重用率。

2、 设计的题目要求达到一定工作量（ 300 行以上代码） ，并具有一定的深度和难度。

3、 程序设计语言推荐使用 C/C++ ，程序书写规范，源程序需加必要的注释 ；

4、 每位同学需提交可独立运行的程序；

5、 每位同学需独立提交设计报告书（每人一份） ，要求编排格式统一、规范、内容充实，不少于 10 页（代码不算） ；

6、 课程设计实践作为培养学生动手能力的一种手段，单独考核 。

3. 课程设计说明书

一 需求分析

[问题描述]

排序和查找是在数据处理中使用频度极高的操作，为加快查找的速度需对数据记录按关键字排序。在汽车数据的信息模型中，汽车牌照是关键字，而且是具有结构特点的一类关键字，因为汽车牌照号是数字和字母混编的，例如 01B7328，这种记录集合是一个适于利用多关键字进行排序的典型例子。

[基本要求]

(1) 首先利用链式基数排序方法实现排序，然后利用折半查找方法，实现对汽车记录按关键字进行查找。

(2) 汽车记录集合可以人工录入，也可以按自动方式随机生成。本人采用的人工录入。

二 概要设计 1. 有关的数据

```
#define KEY_SIZE 8 #define  
LIST_SIZE 100 typedef struct
```

```
{  
    char key[KEY_SIZE]; char  
    name[10];  
    char carname[20];  
    int next; }RecordType; typedef
```

将车牌号以字符的形式
存储

```
struct  
{  
    RecordType r[LIST_SIZE];  
    int length;  
    int keynum;  
}SLinkList;
```

车主的名字
车的品牌

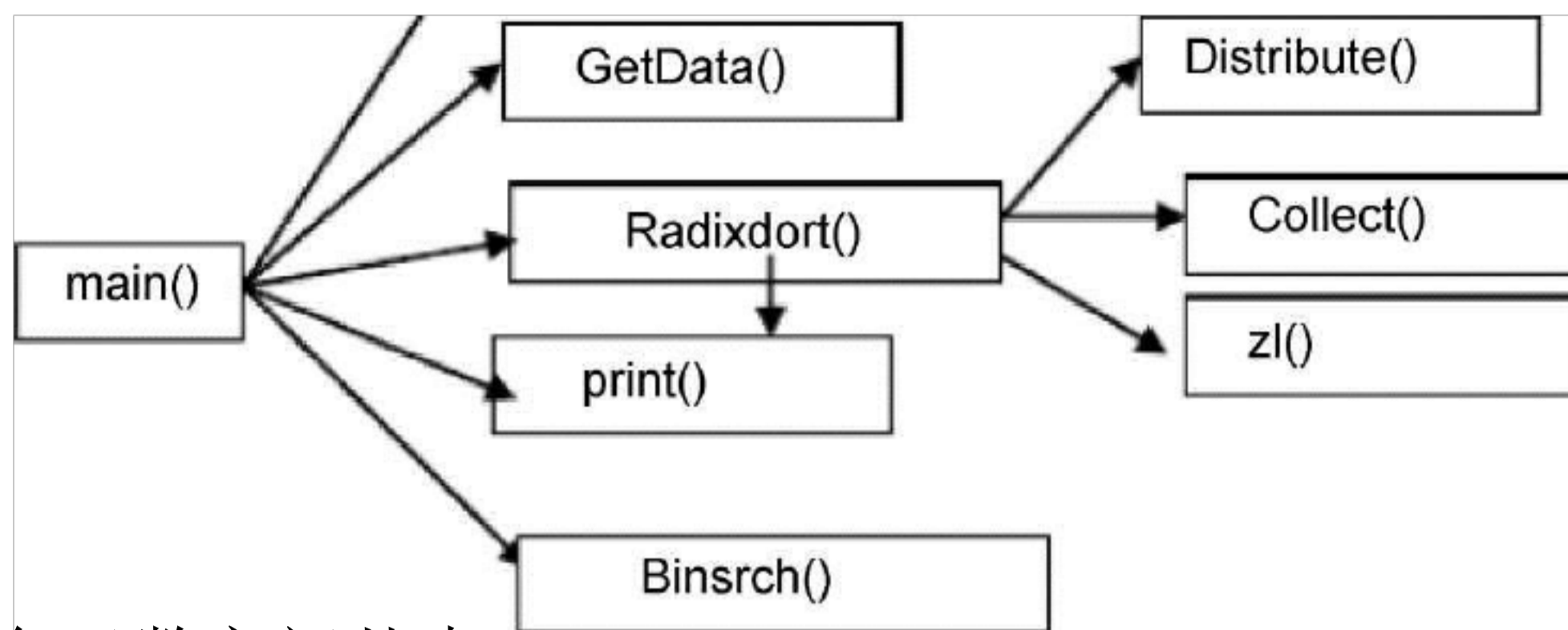
2. 为了实现上述功能，需要使用一下函数：
main()：主函数
noun()：输出提示信息菜单
GetData()：从键盘添加车辆
函数 Distribute() 进行基数排序每
Collect() 进行基数排序每
Binsrch ()：二分查找函数

是一个 RecordType 类型的
数组

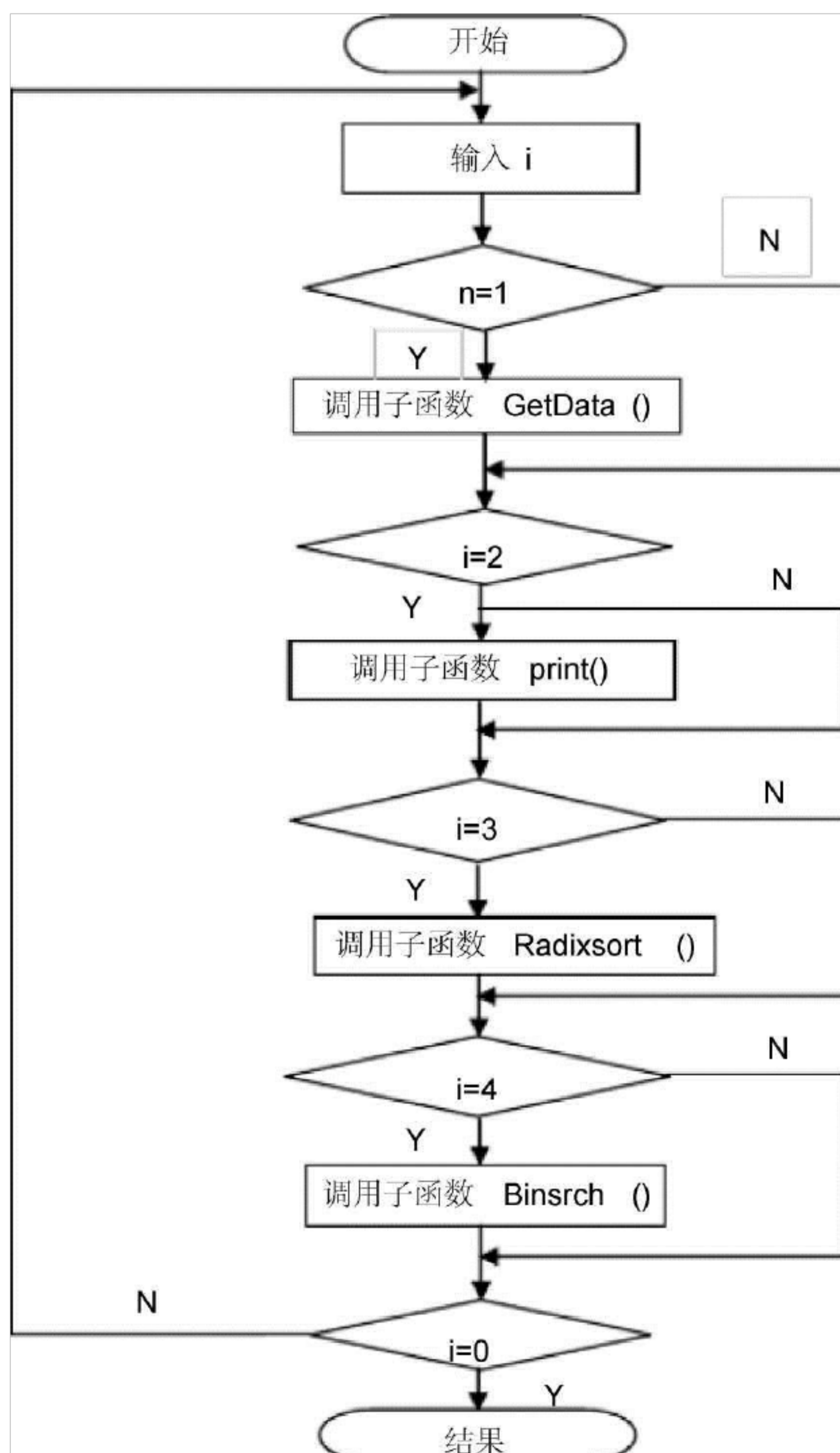
趟的分配函
数
趟的收集函数

print() 输出所有车辆信息函数 Radixsort () 基数排序函数 z1 ()：基数排序后的整理

noun ()



上图为各函数之间基本关系 下图为程序执行的流程图。



三 详细设计

1、基数排序的过程：

二趟收集后的元素序列 (505, 008, 109, 930, 063, 269, 278, 083, 184, 589)

083									
063	184	278	589						
008	109	269	505	930					
k ₀ =0	k ₀ =1	k ₀ =2	k ₀ =3	k ₀ =4	k ₀ =5	k ₀ =6	k ₀ =7	k ₀ =8	k ₀ =9

(c)、按百位数大小将元素分成 10 组

三趟收集后的元素序列(008, 063, 084, 109, 184, 269, 278, 505, 589, 930)

2、二分查找的算法思想:

- (1)、将表中间位置记录的关键字与给定K 值比较, 如果两者相等, 则查找成功。
- (2)、如果两者不等, 利用中间位置记录将表分成前、后两个子表, 如果中间位置记录的关键字大于给定 K 值, 则进一步查找前一子表, 否则进一步查找后一子表。
- (3)、重复以上过程, 直到找到满足条件的记录, 则查找成功, 或者直到分解出的子表不存在为止, 此时 查找不成功。

例如对一有序的数组 a(1, 2, 3, 4, 5, 6, 7, 8, 9)进行查找数 key=6;

首先定义 low=0, high=8, mid=(low+high)/2=4;

第一步: 将 a[mid]与 key 比较, 我们发现 a [mid]<key, 令 low=mid+1=5 ; mid=(low+high)/2=6 第二步:

将 a[mid]与key 比较, 我们发现 a [mid]>key, 此时再令 high=mid-1=5 ; mid=(low+high)/2=5; 第三步:

将 a[mid]与 key 比较, 此时 a[mid]=key, 查找结束, 返回 mid;

第四步: 如果仍未找到, 则继续进行, 直到 low>high, 此时返回 -1, 查找失败;

3、主要函数及功能

1. void Radixsort(SLinkList *l) //基数排序
 //Length个记录存放在数组 r 中, 执行本算法进行基数排序后, 链表中的记录将按关键字 从小到大的顺序链接。 //

```

{
  int n=l->length;
  zimu head, tail;
  shuzi heads, tails;
  for(int i=0;i<=n-1;i++)
    l->r[i].next=i+1;
  l->r[n].next=0;
  for(i=6;i>2;i--) //下标大的为低位,
 从低位开始
  {
    //调用分配函数
    //调用收集函数

    Distribute_s(l->r, i, heads, tails); //
    Collect_s(l->r, heads, tails);
  }

  Distribute_z(l->r, 2, head, tail); // 调用分配函数
  Collect_z(l->r, head, tail); // 调用收集函
  for(i=1;i>=0;i--)
  {
    Distribute_s(l->r, i, heads, tails);
    Collect s(l->r, heads, tails);
  }
}

```

2. Void Distribute_s(RecordType r[], int i, shuzi head, shuzi tail)
 //记录数组 r 中已按低位关键字 key[i+1], ..., key[d] 进行低位优先排序, 本算法按第 i 个关键字 key[i] 建立 10 个队列, 同一个队列中记录的 key[i] 相同。Head[j] 和 tail[j] 分别指向各自队列中第一个和最后一个记录 (j=0, 1, 2, ..., 9)。head[j]=0 表示相应队列为空队列。 //

*_s 表示对数字进行的操作。在程序中还有 _z 表示对字母的操作 *

```

{
    int j, p;
    for(j=0; j<队列的个数 ; j++) 初始化队列
    {
        队列的头指针 =0;全部为 0
        对列的尾指针 =0;全部为 0
    }
    p=第一个数据在数组中的位置
    while(第一个数据在数组中的位置 !=0)
    {
        j=第一个数据的第 i 位在第几个队列

        if(头指针 ==0)
            头指针 =第一个数据载表中的位置 ;
        else
            该队列已有数据的下一个位置 =p 否则将该数在静态链表中的位置放在在
            同一个 队列的数据之后
            尾指针 =p; tial[该数在静态链表中的位置
            p= 下一个数据的位置值 ;
    }
}
void Distribute_z(RecordType r[], int i, zimu head, zimu tail)
{
    int p, j;
    for(j=0; j<=25; j++)
    {
        head[j]=0;
        tail[j]=0;
    }
    p=r[0].next;
}

```



```

while(p!=0)
{
    j=int(int(r[p].key[i])-'A'); if(head[j]==0)head[j]=p; else r[tail[j]].next=p; tail[j]=p;
    p=r[p].next;
}
}

3. Void collect_s(RecordType r[], shuzi head, shuzi tail)
//本算法从 0 到 9 扫描个队列将所有非空队列首尾相接，重新链接成一个链表。

//
int j=0, t;
while(head[j]==0)
    ++j;           //找第一个不为空的队列
r[0].next=head[j]; t=tail[j]; //把 head[j]给第一个数据的位置
while(j<9)
{
    ++j;
    while((j<9)&&(head[j]==0)找到不为 0 的队列
        ++j;
    if(head[j]!=0)
    {
        r[t].next=head[j];
        t=tail[j];
    }
}
r[t].next=0; //使最后一个数的 next=0
}

void Collect_z(RecordType r[], zimu head, zimu tail)字母类型收集重新构成链表
{
    int j=0, t;
    while(head[j]==0)
        ++j;
    r[0].next=head[j]; t=tail[j];
}

```

```

while(j<25)
{
    ++j;
    while((j<25)&&(head[j]==0))
        ++j;
    if(head[j]!=0)
    {
        r[t].next=head[j];
        t=tail[j];
    }
}
r[t].next=0;

```

4. void zl(SLinkList *l) /整理链表顺序

```

{
    int p,q;

    RecordType buf; p=第一个元素在表中的位置 ;
    for(int i=1,表的长度 ;i++)
    {
        while(p<i)
        {
            p= 第 p 个元素的下一个数在表中的位置 ;
            q= 第 p 个元素的下一个数在表中的位置 ;
            if(p!=i)
            {
                buf= 第 p 个元素的地址 ;
                第 p 个元素的地址 = 第 i 个元素的地址 ;交换第 i 个元素的地址与第 p 个
                元素的地址
                第 i 个元素的地址 =buf;
                第 i 个元素的下一个数在表中的位置 =p;
            }
            p=q;
        }
    }
}

```

5. int Binsrch_bin(SLList l, char s[]) // 二分查找 ,s为要找的内容

定义整形三个位置变量 mid ,high, low,并能后两个赋初值; (mid 表示中间, high 表示高位, low 表示低位)

```
While(low<=high)
```

```
{
```

```
    用 mid=(high+low)/2求得 mid 的值;
```

```
    如果 L->r[mid].key==s(要查找的内容); 则返回它在表中的位置 mid
```

```
    如果 L->r[mid].key<s,则将最低位变为 mid+1;
```

```
    如果 L->r[mid].key>s,则将最高位变为 mid-1
```

```
} 执行到些证明在表中没找到要查找的内容, 返回 0;
```

```
}
```

6. void GetData(SLinkList *L) // 从键盘获得数据, 存在表 L 中。

{定义输入的状态变量 x;x不为 0 既认为要输入 定义记录个数的整型变量 j;

输出输入的提示信息 ; scanf("%d",&x)输入 x 的状态; while(x)

```
x=0;
```

```
printf("车牌号: ")输出提示 scanf("%s",&(L->r[j].key)输入节点中对应的量 printf("车主名:"); scanf("%s",&(L->r[j].name));
```

```
    printf("车名: ");
```

```
    scanf("%s",&(L->r[j].carname));
```

```
printf("按任意不为 '0' 的数字继续录入 ***:"); scanf("%d",&x);
```

```
if(x)
```

```
    j++;
```

```
}
```

```
L->length=j;将个数赋给表的长度
```

```
}
```

7. void print(SLinkList *L) 遍历静态表

```
{ int i;
```

```
printf("\t");
```

```
printf("车牌号 车主名 车名 \n"); for(i=1; i<=L->length; i++)  
输出各部分对应的值;
```

```
{
```

```
}
```

8. int Equal(char key1[], char key2[]) 折半查找辅助比较, 判断是否相等, 只比较前 7 位, 第八位是结束符

```
{ for(int i=0; i<7; i++)
```

```
{ if(key1[i]!=key2[i]任意一个不相等就不相等, 返回 0 return 0;
```

```
} return 1; 执行到这说明都相等, 返回不为 0 的值
```

```
}
```

9. int xiao(char key1[], char key2[]) 折半查找辅助比较, 判断较小

```
{ for(int i=0; i<7; i++)
```

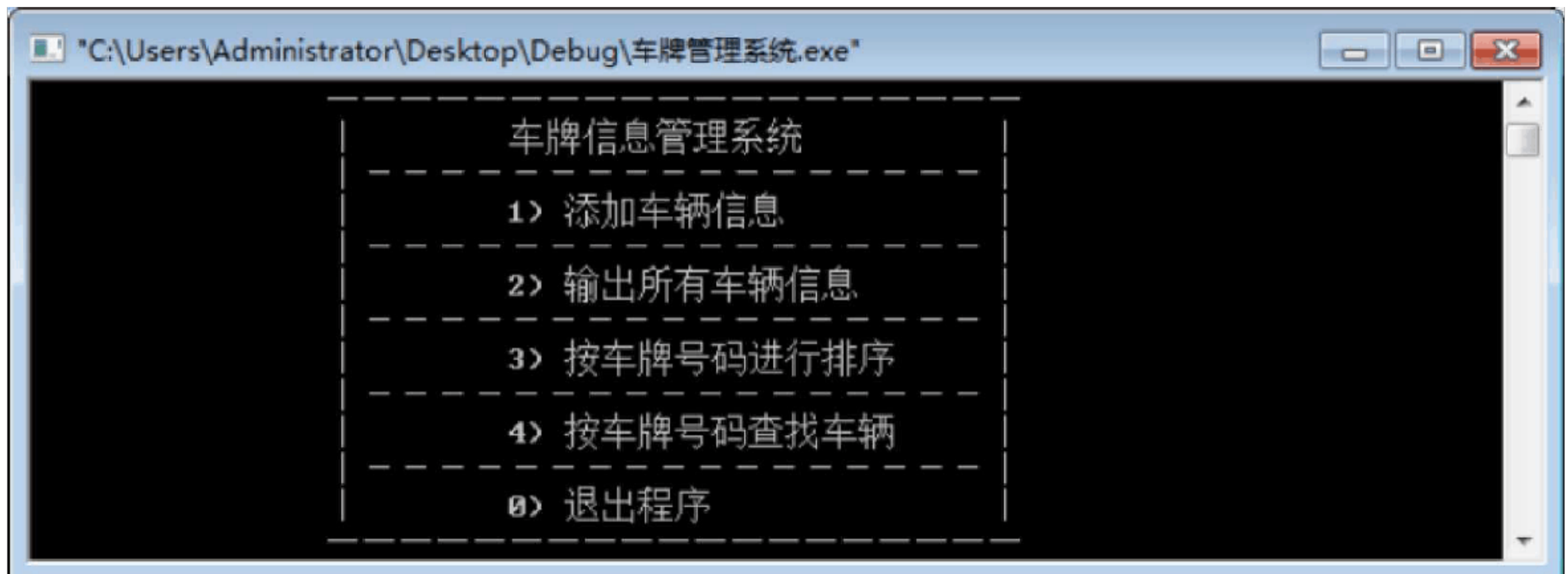
```
{ if(key1[i]<key2[i]) return 1;
```

```
else if(key1[i]>key2[i]) return 0;
```

```
} return 0;
```

四 设计与调试分析

1. 调试输出菜单：



这部分执行成功，为了能够输出对称、格式整齐，所以要不断的调试、修改直到满意。

2. 调试功能 1) 添加车辆信息：

从键盘输入以下几组数据；

车牌号 车主名 车名（按提示输入，以 0 结束轮作输入）

输入成功，退出输入功能也成功，但是要注意在输入时，每个量中间不能输入空格。这样会使 程序默认下个量输入结束。

3. 实现功能 2) 输出所有车辆信息：

输出的车辆信息与输入和一致。此部部执行成功。

4. 实现功能 3) 按车牌号进行排序（从小到大）

按照 2.进行输入：

在这里也套用了功能 2)，从结果可以看出排序成功。

5. 实现功能 4) 按车牌号码查找车辆：

在上面的基础上分别查找 01A1234 和 02A1234

```
*****请选择相应功能对应的数字*****: 4
***请输入要查找的车牌号码***01A1234
查找的车在表中的位置为: 2
车牌号码: 01A1234
车主名: 潘青青
车 名: 奥迪
*****
*****请选择相应功能对应的数字*****: 4
***请输入要查找的车牌号码***02A1234
此车牌号不在列表中
*****
*****请选择相应功能对应的数字*****: 0
半:mp
```

由此可能看出两部分都执行成功

6. 实现功能 5) 退出程序

```
*****请选择相应功能对应的数字*****: 0
Press any key to continue
半:mp
```

退出程序成功。

五 用户手册

- 1、运行程序，根据菜单选择要实现的功能，输入相应的数字。（1：输入数据；2：输出所有元素；3：实现链式基数排序；4：用二分查找在表中按车牌号查找；0：退出程序）
- 2、当选择功能 1 后，根据提示输入相应的信息，在输入时，每个字符串之间不要有空格。按 0 退出输入。（在输入时输入 2 位数字，一个大写字母，然后再输入四位数字）
- 3、当选择功能 2 后，会按格式输出所有节点信息。
- 4、当选择功能 3 后。会输出进行链式排序后的所有节点信息。
- 5、当先择功能 4 后，请输入您要查找车辆的车牌号码。程序会输入相应信息。
- 6、在没有执行第 4 步前，不能执行第五步。
- 7、退出程序请按 0，然后安任意键会关闭运行窗口。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/07800504600007006>