第一章　绪论

一.填空题

1. 数据结构包括数据的逻辑结构、数据的存储结构和 **数据的运算** 。

2. 数据的逻辑结构可以分为 **线性** 和 **非线性** 两大类型。

3. 在算法正确的前提下，评价一个算法好坏的两个主要标准是 **时间复杂度** 和 **空间复杂度** 。

4. 对于给定的 n 个元素,可以构造出的逻辑结构有线性、树形 、 **图形** 和 **集合** 四种。

5. 数据的存储结构不仅有顺序存储结构、链式存储结构，还有 **索引存储结构** 和 **散列存储结构** 。

6. 组成数据的基本单位是 **数据元素** 。

7. 数据结构的两个要素是 **数据元素** 和 **数据元素之间的关系** 。

8. 语句频度是 **语句重复执行的次数** 。

9. 算法是 **对特定问题求解步骤的一种描述，是指令的有限序列** 。

10. 数值计算问题是 **操作对象之间的关系可以用数学方程加以描述的问题** ；非数值计算问题是 **操作对象之间的关系不能用数学方程加以描述的问题** 。

11. 程序是 **算法在计算机上的特定的实现** 。

12. 抽象数据类型是 **指一个数学模型以及定义在该模型上的一组操作** 。

13. 学习数据结构课程的目的是 **非数值计算问题的程序设计** 。

14. 算法可用 **自然语言、流程图、N-S 图、计算机语言、伪码语言等** 描述。

15. 存储密度是 **一个结点中数据元素所占的存储单元和整个结点所占的存储单元之比** 。

二.简答题

1. 试说明算法与程序有哪些区别？

答：至少有四点区别：

（1）程序不一定满足有穷性(如操作系统)；

（2）程序中的指令必须是机器可执行的，算法中的指令则无此限制；

（3）算法代表了对问题的解，程序则是算法在计算机上的特定的实现(一个算法若用程序设计语言来描述，它才是一个程序)；

（4）数据结构+算法=程序。

2. 举一个数据结构的例子，叙述其逻辑结构、存储结构和运算（操作）三方面的内容。

答：例如有一张学生成绩表，记录了一个班的学生各门课的成绩。按学生的姓名为一行记成的表，这个表就是一个数据结构。每个记录（有姓名、学号、成绩等项）就是一个结点，对于整个表来说，只有一个开始结点（它的前面无记录）和一个终端结点（它的后面无记录），其它的结点则各有一个也只有一个直接前趋和直接后继（它的前面和后面均有且只有一个记录）。这几个关系就确定了这个表的逻辑结构——线性结构。

那么我们怎样把这个表中的数据存储到计算机里呢？用高级语言如何表示各结点之间的关系呢？是用一片连续的内存单元来存放这些记录（如用数组存储，亦即顺序存储）还是随机存放各结点数据再用指针进行链接（链式存储）呢？这就是存储结构的问题，我们都是从高级语言的层次来讨论这个问题的。

最后，我们有了这个表（数据结构），肯定要用它，那么就是要对这张表中的记录进行查询、修改、删除等操作，对这个表可以进行哪些操作以及如何实现这些操作就是数据的运算（操作）问题了。

3. 什么叫算法效率？如何度量算法效率？

答：算法效率包括时间效率和空间效率。时间效率指算法运行得有多快；空间效率关心算

法需要的额外空间。算法效率通常用时间复杂度和空间复杂度来度量。

4. 数据的逻辑结构与存储结构的区别和联系是什么？

答：区别：数据的逻辑结构是一个*数学模型*；数据的存储结构是数据的逻辑结构在计算机内部的*存储方式*。

联系：一种逻辑结构可以用多种存储结构来存储；一种存储结构可以用来存多种逻辑结构。

5. 算法有什么特性？评价一个算法有几个标准？

答：一个算法应该具有以下五个特性：

(1)有穷性。 一个算法必须总是在执行有穷步之后结束,且每一步都在有穷时间内完成(有限步完成)。

(2)确定性。算法中每一条指令必须有确切的含义。不存在二义性。且算法只有一个入口和一个出口（无二义性）。

(3)可行性。一个算法是可行的。即算法描述的操作都是可以通过已经实现的基本运算执行有限次来实现的（每一步都可通过执行有限次基本操作实现）。

(4)输入性。一个算法有零个或多个输入,这些输入取自于某个特定的对象集合(有零个或多个输入)。

(5)输出性。一个算法有一个或多个输出，这些输出是同输入有着某些特定关系的量（有一个或多个输出）。

评价一个算法有以下几个标准:

(1) 正确性。算法应满足具体问题的需求。

(2) 可读性。 算法应该好读。以有利于阅读者对程序的理解。

(3) 健壮性。算法应具有容错处理。当输入非法数据时，算法应对其作出反应，而不是产生莫名其妙的输出结果。

(4) 高效性。指算法执行的时间和执行过程中所需要的最大存储空间要少。一般，这两者与问题的规模有关。

第二章　顺序表

一.填空题

1. 当线性表的元素总数基本稳定，且很少进行插入和删除操作，但要求以最快的速度存取线性表中的元素时，应采用 **顺序** 存储结构。

2. 线性表 L=（a1,a2,…,an）用数组表示，假定删除表中任一元素的概率相同，则删除一个元素平均需要移动元素的个数是 **（n-1）/2** 。

3. 顺序表中查找某个元素时，从前到后查找与从后到前查找的时间复杂度 **相** 同。

4. 在具有 n 个元素的顺序表中插入一个元素，合法的插入位置有 **n+1** 个。

5. 在一个长度为 n 的顺序表中第 i 个元素（1<=i<=n）之前插入一个元素时，需向后移动 **n-i+1** 个元素。

6. 顺序存储结构的线性表中所有元素的地址 **一定** 连续。

7. 顺序存储结构的线性表其物理结构与逻辑结构是 **一致** 的。

8. 在具有 n 个元素的顺序存储结构的线性表任意一个位置中插入一个元素，在等概率条件下，平均需要移动 **n/2** 个元素。

9. 在具有 n 个元素的顺序存储结构的线性表任意一个位置中删除一个元素，在等概率条件下，平均需要移动 **（n-1）/2** 个元素。

10. 在具有 n 个元素的顺序存储结构的线性表中查找某个元素,平均需要比较（n+1）/2 次。
11. 顺序存储结构的线性表中，插入或删除某个元素时，元素移动的次数与其位置 有 关（填有或无）。
12. 顺序存储结构的线性表中，访问第 i 个元素与其位置 无 关。（填有或无）。
13. 在具有 n 个元素的顺序存储结构的线性表中要访问第 i 个元素的时间复杂度是 O(1) 。
14. 在顺序表 L 中的 i 个位置插入某个元素 x，正常插入时，i 位置以及 i 位置以后的元素需要后移，首先后移的是 最后一 个元素。
15. 要删除顺序表 L 中的 i 位置的元素 x，正常删除时，i 位置以后的元素需要前移，首先前移的是 第 i+1 个元素(或 i+1 位置上的元素)。
16. 在具有 n 个元素的顺序存储结构的线性表中插入某个元素的时间复杂度是 O(n) 。
17. 若顺序表中的元素是从 1 位置开始存放的，要删除具有 n 个元素的顺序表中某个元素，合法的删除位置是 1 到 n 。
18. 在具有 n 个元素的顺序存储结构的线性表中删除某个元素的时间复杂度是 O(n) 。

二.简答题

1. 下列算法完成在顺序表 SeqL 的第 i 个位置插入元素 x，正常插入返回 1，否则返回 0 或-1，请在空的下划线上填写合适的内容完成该算法。

```
int seq_ins(SeqList *SeqL,int i, DataType x)
{      int j;
       if   ( SeqL->len ==maxsize )                    /*表满*/
     {   printf("the list is full\n");
         return 0;        }
     else if (i<1||i> SeqL->len+1)                  /*位置不对*/
     {        printf("the position is invalid\n ");
          return -1;        }
     else                                          /*正常插入*/
    {     for (j=SeqL->len;j>=i;j--)
           SeqL->data[j+1]= SeqL->data[j];  /*元素后移*/
      SeqL->data[i]=x;    /*插入元素*/
       (SeqL->len)++;                                        /*表长加 1*/
      return 1;
     }
}
```

2. 下列算法完成删除顺序表 SeqL 的第 i 个元素，元素类型为 DataType，其值通过参数 px 返回，请在空的下划线上填写合适的内容完成该算法。

```
int    seq_del (SeqList *SeqL,int i，  DataType *px )
{ int j ;
  if  （SeqL->len==0)                          /*表空*/
       {   printf("the list is empty\n"); return 0;}
  else if( i<1||i> SeqL->len )                    /*位置不对*/
       { printf("\n the position is invalid"); return -1;}
  else                                          /*正常删除*/
      {   *px=SeqL->data[i];                   /*删除元素通过参数 px  返回*/
          for (j=i+1;j<=SeqL->len;j++)
```

```
            SeqL->data[j-1]= SeqL->data[j] ;          /*元素前移*/
        SeqL->len= SeqL->len-1;                  /*表长减 1*/
          return 1;          }
    }
```

3. 什么是线性表？线性表有什么特点？

答：线性表是具有相同数据类型的 n（ ≥0）个数据元素的有限序列。线性表除第一个元素外，其它每一个元素有一个且仅有一个直接前驱；除最后一个元素外，其它每一个元素有一个且仅有一个直接后继。

4. 设有一整型顺序表 L，元素从位置 1 开始存放，下列算法实现将以第一个元素为基准，将其放置在表中合适的位置，使得其前面的元素都比它小，后面的元素均大于等于该元素。请在空的下划线上填写合适的内容完成该算法。

```
void part(SeqList *L)
{     int i, j;                          /*循环变量声明*/
      int   x;
       x= L->data[1] ;                   /*将第一个元素置入 x 中*/
          for(i=2;i<=L->len;i++)
              if( L->data[i]<x  )          /*当前元素小于基准元素*/
                {   L->data[0]=L->data[i];      /*当前元素暂存在 0 位置*/
                    for(j=i-1;j>=1;j--)        /*当前元素前面所有元素后移*/
                        L->data[j+1]=L->data[j];
                    L->data[1]=L->data[0];   /*当前元素从 0 位置移到最前面*/
                 }
        }
```

5. 什么是顺序存储结构？顺序存储结构的优点和缺点各是什么？

答：顺序存储结构是把逻辑上相邻的元素存储在物理位置相邻的存储单元中。通常用数组实现。

顺序存储有三个优点:

（1）方法简单，用数组，容易实现。

（2）不用为表示结点的逻辑关系而增加额外开销。

（3）具有按元素序号随机访问的特点。

顺序存储有两个缺点：

（1）进行插入、删除时，平均移动表中一半的元素，效率较低。

（2）需预先分配足够大的存储空间。空间估计过大，会导致空间闲置（造成浪费）；预先分配过小，又会造成溢出。

三.程序设计题

1. 在顺序存储结构的职工工资表中，职工工资信息包括：职工号（no）、姓名（name）、职称（pro）、工资（sal）等四项信息，请编写一完整的程序，实现以下功能：

（1）创建信息表：从键盘读入所有职工的信息。（3 分）

（2）删除：给定职工号，删除该职工的信息。（6 分）

（3）修改：对职称为“教授”的职工工资加 100。（4 分）

（4）在显示器（屏幕）上显示所有职工的各项信息。（3 分）

（5）主程序以菜单的方式调用以上功能。（4 分）

元素类型及顺序表类型定义如下：

```
typedef struct
```

```
{    char no[8],name[10],pro[6];
float    sal；
} DataType;
typedef struct
{     DataType data[MAXLEN+1]；
       int len;
}SeqList;
```

2. 图书管每本图书包含：书号（no）、书名（name）、现存量（newnum）、总库存量（sum num）四项信息，编写完整程序通过顺序表实现：

（1） 初始化：录入现有的所有图书的四项信息。（3 分）

（2） 借书：每本书每次只能借一本，如果库中有该书，则允许借阅并使该书的现存量减 1，否则给出相应提示信息。（4 分）

（3） 价值估算：统计库中所有书的价钱。价钱为所有书的单价乘以库存量的累加和。（4 分）

（4） 显示：显示图书管所有藏书信息。（3 分）

（5） 主程序以菜单的方式调用以上功能。（4 分）

元素类型及顺序表类型定义 2 分。

3. 设有两个整型顺序表 L1，L2，其元素值递增有序存放，请定义该顺序表的元素类型及表类型（2 分）；设计以下自定义函数：

（1）录入顺序表中所有元素的值。（3 分）

（2）将顺序表 L1，L2 合并为到另外一个顺序表 L3 中，L3 中的元素非递减有序排列。（8 分）

（3）输出顺序表中元素的值。（3 分）

主函数通过调用以上函数实现两个表的合并并显示合并结果。（4 分）

4. 有一个职工基本信息管理，职工信息包含：职工号、姓名、性别；编写完整程序，实现如下功能：

(1)录入函数 input:从键盘读入职工信息。(3 分）

(2)删除函数 delete:给定职工号,删除该职工的信息。(5 分)

(3)插入函数 insert：假定表中职工信息按职工号升序排列，任意给定一职工信息，使得插入后依然有序。(6 分)

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。

5. 有一个学生信息包含：学号 no〈主关键字〉；姓名 name；英语成绩 score。定义学生信息类型 DataType 及顺序表类型 SeqList;

(1)录入函数 input:从键盘读入学生信息。（3 分）

(2)查找函数 search：任意给定一个学号，查找其英语成绩，将其成绩通过函数返回，若找不到，返回-1。（5 分）

(3)插入函数 insert：假定表中学生信息按学号升序排列，任意给定一学生信息，使得插入后依然有序。(6 分)

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。

6. 设有一个超市的库存情况如下表 1 所示：

表 1　超市商品信息

商品编号	商品名称	价格	库存量
10110001	作业本	1.2	20
10110002	铅笔	1.0	10
10110003	钢笔	0.5	30
10110004	铅笔刀	10	5

编写完整程序实现：

(1) 从键盘输入货物信息并将其放在顺序表中。（4 分）。

(2) 假定商品信息按货号升序存放，任意插入一商品信息，要求按货号有序插入到表中。（6 分）

(3) 任意给定一个商品编号，查找其商品名称、价格和库存量，如果存在该商品输出并返回 1，否则返回 0。（4 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。

7. 有一个房产信息管理系统，房产信息包含：门牌号、户主、电话号码、面积。编程实现如下功能（要求用顺序表存储）：

(1) 编写一个初始化函数 input:从键盘读入房产基本信息。(3 分)

(2) 编写一个取暖费用计算函数 cost:任意给定一门牌号，根据门牌号进行查询，找到时，返回应缴纳取暖费，否则返回 0，并且给出提示信息。计算公式为：每户应缴纳费用=面积*4.5 元/m^2。（4 分）

(3) 编写一排序函数 sort：按门牌号升序排列。（4 分）

(4) 编写一个函数 output:输出所有面积低于 90 平方米住户的名称。（3 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。

8. 有一个学生信息包含：学号 no〈主关键字〉；姓名 name；英语成绩 english，计算机成绩 comp,数学成绩 math。定义学生信息类型 DataType 及顺序表类型 SeqList;

(1)录入基本信息函数 input:从键盘读入学生姓名、学号。（3 分）

(2)录入成绩 inp_score：给定课程名称，录入所有人本门课的成绩。（3 分）

(3) 删除函数 del：假定表中学生信息学号升序排列，任意给定一学生学号，删除该学生信息，正常删除返回 1，否则返回 0。(6 分)

(4)输出函数 output:输出所有人的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。

9. 设有一个商品信息表(包括商品编号 no、商品名称 name、商品库存量 count 和商品单价 price)，编程实现以下功能：

（1） 货物信息录入：按货号有序输入学生信息。（3 分）

（2） 进货管理：任意输入一个货物信息，在表中查找该货物，若存在此货物，则将该货物的数量加到表中货物数量中；若不存在该货物，则将该货物信息按照货物号有序插入到表中。（10 分）

（3）货物信息输出:输出所有货物的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

10. 设有一个商品信息表(包括商品编号 no、商品名称 name、商品库存量 count 和商品单价 price)，编程实现以下功能：

（1）货物信息录入：按货号有序输入货物信息。（3 分）

（2）出货管理：函数返回值为购买该货物的金额。任意输入一个货物信息 x，在表中查找该货物，若存在此货物且库存量大于等于 x 的数量，则将表中货物的库存量减去 x 的数量，计算出需支付的金额并返回；若库存量不足，则给出相应的提示并返回 0。若不存在该货物，返回-1。（10 分）

（3）货物信息输出:输出所有货物的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

11. 有一自来水公司水费缴费系统中，数据信息包括：用户名称、编号、用水量、水费、缴费情况（缴清，未缴），请定义用户信息数据类型及顺序表类型并设计如下函数：

函数 1：输入所有用户的名称，编号，用水量，用户名为”???”作为结束符。每个用户的水费通过公式：水费=用水量*0.59 计算得出。用户的缴费情况都设为未缴。（4 分）

函数 2：输入用户编号，查找到该用户信息并将该用户的缴费情况都设为缴清。（4 分）

函数 3：设计一个排序函数，将元素信息按编号有序排列。（4 分）

函数 4：输出所有的未缴费的用户名称。（3 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

12. 设有一个超市商品信息表(包括商品编号 no、商品名称 name、商品库存量 amount 和商品单价 price)，编程实现以下功能：

（1）货物信息录入：输入一批货物信息，货号为“000”时结束。（3 分）

（2）购物管理：输入若干货物货号、数量（即客户所购买的货物信息），货号为“000”时结束，输出所购买的每个货物货号、名称、数量、单价、金额（单价通过查找得到，金额=单价×数量）；最后输出总的价格。（10 分）

（3）货物信息输出:输出所有货物的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

13. 有一学生成绩信息包括：姓名、学号、成绩、名次；编程实现以下功能：

（1）输入所有人姓名、学号、成绩，名次初始化为 0。（3 分）

（2）给定一学生学号，输出其成绩，若不存在该学号，给出相应的错误提示。（4 分）

（3）将学生信息按成绩由高到低排序，并将其名次存入该学生信息的“名次”中。（6 分）

（4）输出所有学生的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

14. 学生考勤管理。学生信息包括：姓名、学号、考勤情况（迟到、旷课、按时上课），成绩及本次已经是第几次考勤。每个人有 20 次考勤，编程实现以下功能：

（1）学生基本信息录入及初始化：输入所有学生的姓名、学号，将每个人的考勤成绩置 0，将考勤次数也置 0。（3 分）

（2）输入考勤情况：每次上课，输入本次考勤，从第一个人开始，依次输入每个人的考勤。（4 分）

（3）计算考勤成绩:迟到得 0.5 分，旷课得 0 分，正常上课得 1 分，计算每个人的考勤成绩。（5 分）

（4）输出所有人的学号、姓名、考勤情况、考勤成绩。（3 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

元素类型定义如下：

```
typedef struct
{    char name[10];    /*name 表示学生姓名*/
char no[12];        /*no 表示学号*/
char kaoqin[21];    /*kaoqin 表示 20 次的考勤，q---缺课，c---迟到，z---按时上课*/
int k;    /*k 表示已经考勤到第几次，初始化时将其置为-1*/
int score;   /*score 表示考勤成绩*/
} DataType;
```

若有 DataType x，则 x. kaoqin[2]表示 x 的第三次考勤。

15. 有一个职工基本信息管理，职工信息包含：职工号、姓名、工资；编写完整程序，实现如下功能：

(1) 录入函数 input:从键盘读入职工信息。(3 分）

(2) 修改函数 modify:给定职工号,查找该职工，若找到修改其信息，若找不到，给出错误提示。(4 分)

(3) 插入函数 insert：假定表中职工信息按职工号升序排列，任意给定一职工信息，使得插入后依然有序。(6 分)

(4) 输出函数 output:输出所有人的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

16. 有一个学生成绩管理，学生信息包含：学号、姓名、成绩；编写完整程序，实现如下功能：

(1) 录入函数 input:从键盘读入学生信息。(3 分）

(2) 修改函数 modify:给定学号,查找该学生，若找到修改其成绩，若找不到，给出错误提示。(4 分)

(3) 排序函数 sort：将学生信息按成绩由高到低排序。 (6 分)

(4) 输出函数 output:输出所有人的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

17. 图书管每本图书包含：书号（no）、书名（name）、现存量（newnum）、总库存量（sum num）四项信息，编写完整程序通过顺序表实现：

(1) 初始化：录入现有的所有图书的四项信息。（3 分）

(2) 清库：给定某书 x 的书号及数量，若找到该书，将该书的现存量及库存量减去 x 的数量，若减后的库存量为 0，则删除该书信息；若找不到该书，则给出错误提示。（9 分）

(3) 显示：显示图书管所有藏书信息。（2 分）

(4) 主程序以菜单的方式调用以上功能。（4 分）

元素类型及顺序表类型定义（2 分）

18. 图书管每本图书包含：书号（no）、书名（name）、现存量（newnum）、总库存量（sum num）四项信息，编写完整程序通过顺序表实现：

（1） 初始化：录入现有的所有图书的四项信息。（3 分）

（2） 检索：给定书名，输出所有与给定书名相同的书的信息。（3 分）

（3） 价值估算：统计库中所有书的价钱。价钱为所有书的单价乘以库存量的累加和。（6 分）

（4） 显示：显示图书管所有藏书信息。（2 分）

（5） 主程序以菜单的方式调用以上功能。（4 分）

（6） 完成元素类型定义及顺序表类型定义（2 分）。

19. 图书管每本图书包含：书号（no）、书名（name）、现存量（newnum）、总库存量（sumnum）四项信息，编写完整程序通过顺序表实现：

（1） 初始化：录入现有的所有图书的四项信息。（3 分）

（2） 查找：给定书号，在表中查找该书，若存在返回其位置，否则返回 0。(4 分)

（3） 进书：给定某个图书的书名、书号、数量，若此次购进的是图书馆已有的图书，则修改其现存量和总库存量；若此次购进的图书是新书，按书号有序插入到表中。（假定表中元素按书号升序排列）（7 分）

（4） 主程序以菜单的方式调用以上功能。（4 分）

（5） 完成元素类型定义及顺序表类型定义（2 分）。

20. 学生学费管理。学生信息包括：姓名、学号、学费、已缴学费、欠费。编写完整程序通过顺序表实现：

（1） 初始化：录入所有人的姓名、学号、学费。（3 分）

（2） 缴费：输入学号及已缴学费，欠费=学费-已缴学费。（5 分）

（3） 催缴学费：对欠费为正数的人，输出其姓名、学号、欠费并统计所有人的欠费总数，总的欠费数目以函数值的形式返回（6 分）

（4） 主程序以菜单的方式调用以上功能。（4 分）

（5） 完成元素类型定义及顺序表类型定义（2 分）。

解答:

1. /*1. 在顺序存储结构的职工工资表中，职工工资信息包括：职工号（no）、姓名（name）、职称（pro）、工资（sal）等四项信息，请编写一完整的程序，实现以下功能：

（1）创建信息表：从键盘读入所有职工的信息。（3 分）

（2）删除：给定职工号，删除该职工的信息。（6 分）

（3）修改：对职称为"教授"的职工工资加 100。（4 分）

（4）在显示器（屏幕）上显示所有职工的各项信息。（3 分）

（5）主程序以菜单的方式调用以上功能。（4 分）

*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{   char no[8],name[10],pro[6];
    float   sal;
} DataType;
typedef struct
{     DataType data[MAXLEN+1];
```

```
        int len;
    }SeqList;

    /*从键盘读入所有职工的信息*/
    void input(SeqList*L)
    { int i;
      printf("input the length\n");
      scanf("%d",&L->len);
      printf("input no,name,pro,sal\n");
      for(i=1;i<=L->len;i++)
       scanf("%s%s%s%f",L->data[i].no,L->data[i].name,L->data[i].pro,&L->data[i].sal);
    }
        /*给定职工号，删除该职工的信息*/
        void del(SeqList*L,DataType x)
        { int j,i=1;          while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在
位置*/
            i++;
          if(i>L->len )
             printf("not find\n");
          else
          {
            for (j=i+1;j<=L->len;j++)                    /*删除元素 X*/
                 L->data[j-1]=L->data[j];
              (L->len)--;
          }
        }
    /*对职称为"教授"的职工工资加 100*/
    void modify(SeqList*L)
    {
    int i;
    for(i=1;i<=L->len;i++)
      if(strcmp(L->data[i].pro,"教授")==0)      L->data[i].sal+=100; }
        /*菜单*/
        void menu()
        {
        printf("1-------------录入信息\n");
        printf("2-------------删除\n");
        printf("3-------------修改\n");
        printf("4-------------输出\n");
        printf("0-------------退出\n");
    }
        void main()
        {
        SeqList* L;
```

```
        DataType x;
        int sel,i;
        L=(SeqList*)malloc(sizeof(SeqList));
        L->len=0;       do
        { menu();
            printf("input your select\n");
            scanf("%d",&sel);
            switch(sel)
            {
            case 1:input(L);break;
            case 2:printf("input no\n");
                   scanf("%s",x.no);
              del(L,x);
              break;
            case 3:modify(L);
                   break;
            case 4:for(i=1;i<=L->len;i++)
                printf("%10s%12s%8s%7.1f\n",L->data[i].no,L->data[i].name,L->data[i].p
ro,L->data[i].sal);
            }
        }while(sel!=0);
        }
```

2. /*2．图书管每本图书包含：书号（no）、书名（name）、现存量（newnum）、总库存量（sumnum）四项信息，编写完整程序通过顺序表实现：

（1） 初始化：录入现有的所有图书的四项信息。（3 分）

（2） 借书：每本书每次只能借一本，如果库中有该书，则允许借阅并使该书的现存量减 1，否则给出相应提示信息。（4 分）

（3） 价值估算：统计库中所有书的价钱。价钱为所有书的单价乘以库存量的累加和。（4 分）

（4） 显示：显示图书管所有藏书信息。（3 分）

（5） 主程序以菜单的方式调用以上功能。（4 分）

元素类型及顺序表类型定义 2 分。

*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{   char no[8],name[10];
    int newnum,sumnum;
    float price;
} DataType;
typedef struct
```

```
{     DataType data[MAXLEN+1];
       int len;
}SeqList;
/*从键盘读入所有图书的信息*/
void input(SeqList*L)
{ int i;
   printf("input the length\n");
   scanf("%d",&L->len);
   printf("input no,name,newnum,sumnum,price\n");
   for(i=1;i<=L->len;i++)
     scanf("%s%s%d%d%f",L->data[i].no,L->data[i].name,&L->data[i].newnum,&L->da
ta[i].sumnum,&L->data[i].price);
}
      /*借书*/
      void brow(SeqList*L,DataType x)
      { int i=1;      while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)    /*查找元素 x 所在位置
*/
          i++;
        if(i>L->len )
          printf("not find\n");
        else if( L->data[i].newnum>0)          L->data[i].newnum--;
        else
          printf("the book is brown");
      }
/*价值估算*/
float volue(SeqList*L)
{ float v=0;      int i;
   for(i=1;i<=L->len;i++)
     v+=L->data[i].price*L->data[i].sumnum;
   return v;
}
      /*显示*/
      void output(SeqList*L)
      {int i;
        printf("no,name,newnum,sumnum,price\n");
        for(i=1;i<=L->len;i++)
          printf("%10s%12s%5d%5d%7.2f\n",L->data[i].no,L->data[i].name,L->data[i].
newnum,L->data[i].sumnum,L->data[i].price);
      }
      /*菜单*/
void menu()
{
printf("1----------录入信息\n");
printf("2----------借书\n");
```

```
printf("3----------估价\n");
printf("4----------输出\n");
printf("0----------退出\n");
      }
void main()
{
SeqList* L;
DataType x;
int sel;
float v;
L=(SeqList*)malloc(sizeof(SeqList));
L->len=0; do
{ menu();
    printf("input your select\n");
    scanf("%d",&sel);
    switch(sel)
    {
    case 1:input(L);break;
    case 2:printf("input no\n");
          scanf("%s",x.no);
      brow(L,x);
      break;
    case 3:v=volue(L);
          printf("the sum vulue is%10.2f\n",v);
          break;
    case 4:output(L);
          break;
    }
}while(sel!=0);
}
```

3. /*设有两个整型顺序表 L1，L2，其元素值递增有序存放，请定义该顺序表的元素类型及表类型（2 分）；设计以下自定义函数：

（1）录入顺序表中所有元素的值。（3 分）

（2）将顺序表 L1，L2 合并为到另外一个顺序表 L3 中，L3 中的元素非递减有序排列。（8 分）

（3）输出顺序表中元素的值。（3 分）

主函数通过调用以上函数实现两个表的合并并显示合并结果。（4 分）

*/

```
#include <stdio.h>
#include <stdlib.h>
#define MAXLEN 100
typedef struct
      {     int data[MAXLEN+1];
```

```
        int len;
    }SeqList;
    /*录入*/
    void input(SeqList*L)
    { int i;
      printf("input the length\n");
      scanf("%d",&L->len);
      printf("input element\n");
      for(i=1;i<=L->len;i++)
        scanf("%d",&L->data[i]);
    }
/*合并*/
void merge(SeqList* A, SeqList* B, SeqList *C)
{   int i,j,k;
        i=1;j=1;k=1;                while(i<=A->len && j<=B->len)
            if(A->data[i]<B->data[j])
                C->data[k++]=A->data[i++];
            else C->data[k++]=B->data[j++];
        while(i<=A->len)
                C->data[k++]=A->data[i++];
        while(j<=B->len)
                C->data[k++]=B->data[j++];
         C->len=k-1; }
    /*显示*/
    void output(SeqList*L)
    {int i;
      for(i=1;i<=L->len;i++)
        printf("%5d\n",L->data[i]);
    }
/*菜单*/
void menu()
{
printf("1----------表 1 录入\n");
printf("2----------表 2 录入\n");
printf("3----------合并\n");
printf("4----------输出合并后的表\n");
printf("0----------退出\n");
    }
void main()
{
SeqList* L1,*L2,*L3;
int sel;
L1=(SeqList*)malloc(sizeof(SeqList));
L1->len=0; L2=(SeqList*)malloc(sizeof(SeqList));
```

```
L2->len=0; L3=(SeqList*)malloc(sizeof(SeqList));
L3->len=0; do
{ menu();
    printf("input your select\n");
    scanf("%d",&sel);
    switch(sel)
    {
    case 1:    printf("input list1\n");
               input(L1);
         break;
    case 2: printf("input list2\n");
             input(L2);
             break;
    case 3:merge(L1,L2,L3);
           break;
    case 4:output(L3);
    }
}while(sel!=0);
}
```

4. /*有一个职工基本信息管理，职工信息包含：职工号、姓名、性别；编写完整程序，实现如下功能：

(1)录入函数 input:从键盘读入职工信息。(3 分）

(2)删除函数 delete:给定职工号,删除该职工的信息。(5 分)

(3)插入函数 insert：假定表中职工信息按职工号升序排列，任意给定一职工信息，使得插入后依然有序。(6 分)

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{   char no[8],name[10],xb[3]; /*注意字符型数组存放字符串时需多申请一个*/
} DataType;
typedef struct
{     DataType data[MAXLEN+1];
      int len;
}SeqList;
      /*从键盘读入所有职工的信息*/
void input(SeqList*L)
{ int i;
   printf("input the length\n");
   scanf("%d",&L->len);
```

```
   printf("input no,name,xb\n");
   for(i=1;i<=L->len;i++)
     scanf("%s%s%s",L->data[i].no,L->data[i].name,L->data[i].xb);
}
/*查找*/
int search(SeqList*L,DataType x)
{
int i=1; while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
     i++;
      if(i>L->len ) return 0; return i;
}
/*给定职工号，删除该职工的信息*/
void del(SeqList*L,DataType x)
{ int j,i;
   i=search(L,x);
   if(i==0 )
      printf("not find\n");
   else
   {
    for (j=i+1;j<=L->len;j++)                          /*删除元素 X*/
            L->data[j-1]=L->data[j];
        (L->len)--;
   }
}
            /*插入职工信息*/
      void ins(SeqList*L,DataType x)
      {
      int i,j;
      i=search(L,x);
            if(i==0 )
      {
         i=1;   /*查找插入位置*/
         while(i<=L->len&&strcmp(x.no,L->data[i].no)>0)
           i++;
         for(j=L->len;j>=i;j--)
           L->data[j+1]=L->data[j];
         L->data[i]=x;
         L->len++;
      }
      else printf("the element already exist\n");
      }
/*菜单*/
void menu()
{
```

```
printf("1----------录入信息\n");
printf("2----------查找\n");
printf("3----------删除\n");
printf("4----------插入\n");
printf("5----------输出\n");
printf("0----------退出\n");                }
void main()
{
SeqList* L;
DataType x;
int sel,i;
L=(SeqList*)malloc(sizeof(SeqList));
L->len=0; do
{ menu();
    printf("input your select\n");
    scanf("%d",&sel);
    switch(sel)
    {
    case 1:input(L);break;
    case 2:printf("input no\n");
          scanf("%s",x.no);
     i=search(L,x);
     if(i==0)          printf("not exsit\n" );
     else
       printf("%10s%12s%4s\n",L->data[i].no,L->data[i].name,L->data[i].xb);
     break;
    case 3:printf("input no\n");
          scanf("%s",x.no);
     del(L,x);break;
    case 4:printf("input no,name,xb\n");
                  scanf("%s%s%s",x.no,x.name,x.xb);
     ins(L,x);
          break;
    case 5:for(i=1;i<=L->len;i++)
       printf("%10s%12s%4s\n",L->data[i].no,L->data[i].name,L->data[i].xb);
    }
}while(sel!=0);
}
```

5. /*有一个学生信息包含：学号 no〈主关键字〉；姓名 name；英语成绩 score。定义学生信息类型 DataType 及顺序表类型 SeqList;

(1)录入函数 input:从键盘读入学生信息。（3 分）

（2）查找函数 search：任意给定一个学号，查找其英语成绩，将其成绩通过函数返回，若找不到，返回-1。（5 分）

(3)插入函数 insert：假定表中学生信息按学号升序排列，任意给定一学生信息，使得插入后依然有序。(6 分)

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{    char no[8],name[10];
     int score;
} DataType;
typedef struct
{     DataType data[MAXLEN+1];
       int len;
}SeqList;
/*从键盘读入所有学生的信息*/
void input(SeqList*L)
{ int i;
   printf("input the length\n");
   scanf("%d",&L->len);
   printf("input no,name,score\n");
   for(i=1;i<=L->len;i++)
     scanf("%s%s%d",L->data[i].no,L->data[i].name,&L->data[i].score);
}
/*查找*/
int search(SeqList*L,DataType x)
{
int i=1; while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
     i++;
      if(i>L->len ) return -1; return L->data[i].score;
}
/*插入学生信息*/
       void ins(SeqList*L,DataType x)
       {
       int i,j;
       i=search(L,x);
             if(i==-1 )
       {
          i=1;   /*查找插入位置*/
          while(i<=L->len&&strcmp(x.no,L->data[i].no)>0)
            i++;
          for(j=L->len;j>=i;j--)
            L->data[j+1]=L->data[j];
```

```
            L->data[i]=x;
            L->len++;
        }
        else printf("the element already exist\n");
        }
/*菜单*/
void menu()
{
printf("1-------------录入信息\n");
printf("2-------------查找\n");
printf("3-------------插入\n");
printf("4-------------输出\n");
printf("0-------------退出\n");
        }
void main()
{
SeqList* L;
DataType x;
int sel,s,i;
L=(SeqList*)malloc(sizeof(SeqList));
L->len=0; do
{ menu();
      printf("input your select\n");
      scanf("%d",&sel);
      switch(sel)
      {
      case 1:input(L);break;
      case 2:printf("input no\n");
             scanf("%s",x.no);
        s=search(L,x);
        if(s==-1)           printf("not exsit\n" );
        else
          printf("the score is%5d\n",s);
        break;
      case 3:printf("input no,name,score\n");
                       scanf("%s%s%d",x.no,x.name,&x.score);
        ins(L,x);
             break;
      case 4:for(i=1;i<=L->len;i++)
          printf("%10s%12s%5d\n",L->data[i].no,L->data[i].name,L->data[i].score);
      }
}while(sel!=0);
}
```

6. /*设有一个超市的库存情况如下表 1 所示：

表 1　超市商品信息

商品编号	商品名称	价格	库存量
10110001	作业本	1.2	20
10110002	铅笔	1.0	10
10110003	钢笔	0.5	30
10110004	铅笔刀	10	5

编写完整程序实现：

(1) 从键盘输入货物信息并将其放在顺序表中。（4 分）。

(2）假定商品信息按货号升序存放，任意插入一商品信息，要求按货号有序插入到表中。（6 分）

(3）任意给定一个商品编号，查找其商品名称、价格和库存量，如果存在该商品输出并返回 1，否则返回 0。（4 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{    char no[8],name[10];
     float price;
     int amount;
} DataType;
typedef struct
{     DataType data[MAXLEN+1];
      int len;
}SeqList;
/*从键盘读入所有货物的信息*/
void input(SeqList*L)
{ int i;
   printf("input the length\n");
   scanf("%d",&L->len);
   printf("input no,name,price,amount\n");
   for(i=1;i<=L->len;i++)
     scanf("%s%s%f%d",L->data[i].no,L->data[i].name,&L->data[i].price,&L->data[
i].amount);
}
/*查找*/
int search(SeqList*L,DataType x)
{
int i=1; while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
     i++;
      if(i>L->len ) return 0; return i;
```

```
}

      /*插入货物信息*/
      void ins(SeqList*L,DataType x)
      {
      int i,j;
      i=search(L,x);
            if(i==0 )
      {
         i=1;   /*查找插入位置*/
         while(i<=L->len&&strcmp(x.no,L->data[i].no)>0)
           i++;
         for(j=L->len;j>=i;j--)
           L->data[j+1]=L->data[j];
         L->data[i]=x;
         L->len++;
      }
      else printf("the element already exist\n");
      }
/*菜单*/
void menu()
{
printf("1----------录入信息\n");
printf("2----------查找\n");
printf("3----------插入\n");
printf("4----------输出\n");
printf("0----------退出\n");
              }
void main()
{
SeqList* L;
DataType x;
int sel,i;
L=(SeqList*)malloc(sizeof(SeqList));
L->len=0; do
{ menu();
     printf("input your select\n");
     scanf("%d",&sel);
     switch(sel)
     {
     case 1:input(L);break;
     case 2:printf("input no\n");
            scanf("%s",x.no);
      i=search(L,x);
```

```
            if(i==0)          printf("not exsit\n" );
            else
              printf("%10s%12s%7.2f%5d\n",L->data[i].no,L->data[i].name,L->data[i].pri
ce,L->data[i].amount);
            break;
          case 3:printf("input no,name,price,amount\n");
                        scanf("%s%s%f%d",x.no,x.name,&x.price,&x.amount);
            ins(L,x);
                break;
          case 4:for(i=1;i<=L->len;i++)
                printf("%10s%12s%7.2f%5d\n",L->data[i].no,L->data[i].name,L->data[i].p
rice,L->data[i].amount);
          }
      }while(sel!=0);
      }
```

7. /*有一个房产信息管理系统，房产信息包含：门牌号、户主、电话号码、面积。编程实现如下功能（要求用顺序表存储）：

(1) 编写一个初始化函数 input:从键盘读入房产基本信息。(3 分)

(2) 编写一个取暖费用计算函数 cost:任意给定一门牌号，根据门牌号进行查询，找到时，返回应缴纳取暖费，否则返回 0，并且给出提示信息。计算公式为：每户应缴纳费用=面积*4.5 元/m^2。。（4 分）

（3）编写一排序函数 sort：按门牌号升序排列。（4 分）

（4）编写一个函数 output:输出所有面积低于 90 平方米住户的名称。（3 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。

*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{    char no[6],name[10];
     float area;
     char tel[15];
} DataType;
typedef struct
{     DataType data[MAXLEN+1];
      int len;
}SeqList;

/*从键盘读入所有住房的信息*/
void input(SeqList*L)
{ int i;
   printf("input the length\n");
```

```
    scanf("%d",&L->len);
    printf("input no,name,area,tel\n");
    for(i=1;i<=L->len;i++)
      scanf("%s%s%f%s",L->data[i].no,L->data[i].name,&L->data[i].area,&L->data[i
].tel);
  }
  /*查找*/
  int search(SeqList*L,DataType x)
  {
  int i=1; while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
      i++;
       if(i>L->len ) return 0; return i;
  }
            /*计算取暖费*/
       float cost(SeqList*L,DataType x)
       {
       int i;
       i=search(L,x);
            if(i==0 )
         return 0;     else return (L->data[i].area*4.5);
       }
  /*排序*/
  void sort(SeqList*L)
  {
    int   i,j,n;
    n=L->len;
    for(i=2;i<=n;i++)          /*外循环控制排序的总趟数*/
     {
     L->data[0]=L->data[i];
     j=i-1;       while (strcmp(L->data[0].no,L->data[j].no)<0)       { L->data[j+1]= L->data[j];
              j--;
         }     //while
         L->data[j+1]= L->data[0];
  }
  }
            /*输出*/
       void output(SeqList*L)
       {
       int i;
       for(i=1;i<=L->len;i++)
         printf("%10s%12s%7.2f%15s\n",L->data[i].no,L->data[i].name,L->data[i].are
a,L->data[i].tel);
       }
  /*菜单*/
```

```
void menu()
{
printf("1----------录入信息\n");
printf("2----------计算取暖费\n");
printf("3----------排序\n");
printf("4----------输出\n");
printf("0----------退出\n");                }
void main()
{
SeqList* L;
DataType x;
int sel;
float x_cost;
L=(SeqList*)malloc(sizeof(SeqList));
L->len=0; do
{ menu();
    printf("input your select\n");
    scanf("%d",&sel);
    switch(sel)
    {
    case 1:input(L);break;
    case 2:printf("input no\n");
          scanf("%s",x.no);
     x_cost=cost(L,x);
     if(x_cost==0)          printf("not exsit\n" );
     else
       printf("%10s%7.2f%\n",x.no,x_cost);
     break;
    case 3:sort(L);
          break;
    case 4:output(L);
          break;
    }
}while(sel!=0);
}
```

8. /* 有一个学生信息包含：学号 no〈主关键字〉；姓名 name；英语成绩 english，计算机成绩 comp,数学成绩 math。定义学生信息类型 DataType 及顺序表类型 SeqList;

(1)录入基本信息函数 input:从键盘读入学生姓名、学号。（3 分）

（2）录入成绩 inp_score：给定课程名称，录入所有人本门课的成绩。（3 分）

(3) 删除函数 del：假定表中学生信息学号升序排列，任意给定一学生学号，删除该学生
信息，正常删除返回 1，否则返回 0。(6 分)

（4）输出函数 output:输出所有人的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 4 分。
*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct student          /*定义元素数据类型，DataType */
{
   char name[10],no[10];
   int math,eng,comp;
}DataType;
typedef struct                  /*定义顺序表类型 SeqList */
{ DataType data[MAXLEN];
   int len;
}SeqList;
      void menu();
      void input(SeqList* L);
      void inp_score(SeqList* L);
      void seq_del(SeqList *SeqL, DataType x);
      void output(SeqList *L);
void main()
{ SeqList *SeqLStu;           /*主函数中，学生信息顺序表用 SeqLStu 表示*/
   DataType x;
   int sel;
   SeqLStu=(SeqList*)malloc(sizeof(SeqList));
    if (SeqLStu==NULL)    exit(0);
    SeqLStu->len=0;                                  /*将表初始化为一个空表*/
    do{ menu();
        printf("\n input your choice:");      /*输入所选择的功能号*/
        scanf("%d",&sel);
        switch (sel)
         {case 1: input(SeqLStu);            break;
           case 2: inp_score(SeqLStu);          break;
       case 3: printf("\n please input element no\n");
              scanf("%s",x.no);
          seq_del(SeqLStu,x);
          break;
       case 4:    output(SeqLStu);
         break;
       case 0: exit(0);
    }
   }while (sel!=0);
}
     /*菜单函数*/
```

```
        void menu()
        {    printf("\n1.............录入姓名学号\n");
             printf("2.............录入成绩\n");
             printf("3.............删除\n");
             printf("4.............显示\n");
             printf("0.............退出\n");
        }
        /*输入学号、姓名*/
        void input(SeqList* L)
        {     int i;
        printf("\n initalizing.............\n");
                  printf("\n input the list's length :");
                       scanf("%d",&L->len);
                       getchar();
                       for (i=1; i<=L->len;i++)
              {    printf("enter no,name\n");
                            scanf("%s%s",L->data[i].no,L->data[i].name);
                  L->data[i].math=0;              L->data[i].comp=0;              L->data[i].eng=0;
       }
        }
/*输入成绩*/
void inp_score(SeqList* L)
{
int i;
        char kch[20];
printf("input the course name\n ");
scanf("%s",kch);
if    (strcmp(kch,"math" )==0)       /*字符串比较*/
{
   printf("\n input the math score \n");
        for (i=1; i<=L->len;i++)
        scanf("%d",&L->data[i].math);
}
else if (strcmp(kch,"eng" )==0) {
   printf("\n input the english score\n ");
   for (i=1; i<=L->len;i++)
     scanf("%d",&L->data[i].eng);
}
else if(strcmp(kch,"comp" )==0) {
   printf("\n input the computer score\n ");
   for (i=1; i<=L->len;i++)
     scanf("%d",&L->data[i].comp);
}
else printf("\n the name is error \n");
```

```
}

    /*删除给定元素*/
    void seq_del(SeqList *L, DataType x)
    {
        int i=1,j;     /*查找元素 x*/
        while(i<=L->len && strcmp(x.no,L->data[i].no)>0)    i++;
        if(i<=L->len&&strcmp(x.no,L->data[i].no)==0) /*找到 x，则删除 x*/
        {
          for (j=i+1;j<=L->len;j++)
            L->data[j-1]=L->data[j];/*元素前移*/
          (L->len)--;                    /*表长减 1*/
        }
        else
          printf("x not exist\n");
    }
/*输出函数*/
void output(SeqList *L)
{
    int i=1;       for(i=1;i<=L->len;i++)
                printf("%12s;%10s;%8d;%8d;%8d\n",L->data[i].no,L->data[i].name,L->
data[i].math,L->data[i].eng,L->data[i].comp);
}
```

9. /*设有一个商品信息表(包括商品编号 no、商品名称 name、商品库存量 count 和商品单价 price) ，编程实现以下功能：

（1） 货物信息录入：按货号有序输入学生信息。（3 分）

（2） 进货管理：任意输入一个货物信息，在表中查找该货物，若存在此货物，则将该货物的数量加到表中货物数量中；若不存在该货物，则将该货物信息按照货物号有序插入到表中。（10 分）

（3）货物信息输出:输出所有货物的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct student          /*定义元素数据类型，DataType */
{
   char name[10],no[10];
   int amount;
}DataType;
typedef struct                  /*定义顺序表类型 SeqList */
{ DataType data[MAXLEN];
```

```
   int len;
}SeqList;
      void menu();
      void input(SeqList* L);
      void ruku(SeqList* L,DataType x);
      void output(SeqList *L);
      int search(SeqList*L,DataType x);
      void main()
      { SeqList *L;            /*主函数中，货物顺序表用 L 表示*/
         DataType x;
         int sel;
         L=(SeqList*)malloc(sizeof(SeqList));
          if (L==NULL)    exit(0);
          L->len=0;                                    /*将表初始化为一个空表*/
          do{ menu();
              printf("\n input your choice:");       /*输入所选择的功能号*/
              scanf("%d",&sel);
              switch (sel)
                {case 1: input(L);             break;
                  case 2:    printf("\n input element no,name,amountiang\n");
                     scanf("%s%s%d",x.no,x.name,&x.amount);
                  ruku(L,x);
                  break;
               case 3:    output(L);
                  break;
               case 0: exit(0);
          }
         }while (sel!=0);
      }
/*菜单函数*/
void menu()
{    printf("\n1.............录入货物信息\n");
     printf("2.............入库\n");
     printf("3.............显示\n");
     printf("0.............退出\n");
}
/*录入货物信息*/
void input(SeqList* L)
{     int i;
printf("\n initalizing.............\n");
          printf("\n input the list's length :");
              scanf("%d",&L->len);
              getchar();
              for (i=1; i<=L->len;i++)
```

```
        {   printf("enter no,name,amount\n");
                    scanf("%s%s%d",L->data[i].no,L->data[i].name,&L->data[i].amount);
        }
    }
            /*查找*/
        int search(SeqList*L,DataType x)
        {
        int i=1;    while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
            i++;
             if(i>L->len ) return 0;      return i;
        }
    /*入库*/
    void ruku(SeqList* L,DataType x)
    {
    int i,j;
        i=search(L,x);
        if(i==0) {     i=1;     while(i<=L->len&&strcmp(L->data[i].no,x.no)<0)   /*查找插入位置
*/
            i++;
       for(j=L->len;j>=i;j--)
        L->data[j+1]=L->data[j];
       L->data[i]=x;
       L->len++;
    }
    else
       L->data[i].amount+=x.amount;
    }
            /*输出函数*/
        void output(SeqList *L)
        {
            int i=1;          for(i=1;i<=L->len;i++)
                        printf("%12s;%12s;%8d\n",L->data[i].no,L->data[i].name,L->data[i
].amount);
        }
```

10. /*设有一个商品信息表(包括商品编号 no、商品名称 name、商品库存量 count 和商品单价 price)，编程实现以下功能：

（1）货物信息录入：按货号有序输入货物信息。（3 分）

（2）出货管理：函数返回值为购买该货物的金额。任意输入一个货物信息 x，在表中查找该货物，若存在此货物且库存量大于等于 x 的数量，则将表中货物的库存量减去 x 的数量，计算出需支付的金额并返回；若库存量不足，则给出相应的提示并返回 0。若不存在该货物，返回-1。（10 分）

（3）货物信息输出:输出所有货物的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct student          /*定义元素数据类型，DataType */
{
   char name[10],no[10];
   float price;
   int amount;
}DataType;
typedef struct                  /*定义顺序表类型 SeqList */
{ DataType data[MAXLEN];
   int len;
}SeqList;
      void menu();
      void input(SeqList* L);
      float chuku(SeqList* L,DataType x);
      void output(SeqList *L);
      int search(SeqList*L,DataType x);
      void main()
      { SeqList *L;           /*主函数中，货物顺序表用 L 表示*/
         DataType x;
         int sel;
         float cost;
          L=(SeqList*)malloc(sizeof(SeqList));
          if (L==NULL)    exit(0);
          L->len=0;                                  /*将表初始化为一个空表*/
          do{ menu();
               printf("\n input your choice:");      /*输入所选择的功能号*/
               scanf("%d",&sel);
               switch (sel)
                {case 1: input(L);            break;
                  case 2:    printf("\n input element no,name,amount\n");
                      scanf("%s%s%d",x.no,x.name,&x.amount);
                  cost=chuku(L,x);
                  if(cost>0) printf("cost:%8.2f\n",cost);
                  break;
              case 3:    output(L);
                break;
              case 0: exit(0);
           }
         }while (sel!=0);
      }
```

```
/*菜单函数*/
void menu()
{    printf("\n1.....录入货物信息\n");
     printf("2.....出库\n");
     printf("3.....显示\n");
     printf("0.....退出\n");
}
/*录入货物信息*/
void input(SeqList* L)
{     int i;
printf("\n initalizing.....\n");
          printf("\n input the list's length :");
               scanf("%d",&L->len);
               getchar();
               for (i=1; i<=L->len;i++)
     {    printf("enter no,name,amount,price\n");
                    scanf("%s%s%d%f",L->data[i].no,L->data[i].name,&L->data[i].amount
,&L->data[i].price);
          }
}
               /*查找*/
        int search(SeqList*L,DataType x)
        {
        int i=1;    while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
             i++;
              if(i>L->len ) return 0;      return i;
        }
/*出库*/
float chuku(SeqList* L,DataType x)
{
int i;
        i=search(L,x);
        if(i==0) {      printf("this element not exist\n");
         return -1; }
else if(L->data[i].amount>=x.amount)
{     L->data[i].amount-=x.amount;
   return x.amount*L->data[i].price;
}
else
{ printf("the amount is not enough\n");
         return 0; }

}
               /*输出函数*/
```

```
void output(SeqList *L)
{
    int i=1;        printf("no              name            price    amount\n");
    for(i=1;i<=L->len;i++)
                printf("%-12s%-12s%-7.2f%-8d\n",L->data[i].no,L->data[i].name,L-
>data[i].price,L->data[i].amount);
}
```

11. /*有一自来水公司水费缴费系统中，数据信息包括：用户名称、编号、用水量、水费、缴费情况（缴清，未缴），请定义用户信息数据类型及顺序表类型并设计如下函数：

函数 1：输入所有用户的名称，编号，用水量，用户名为"???"作为结束符。每个用户的水费通过公式：水费=用水量*0.59 计算得出。用户的缴费情况都设为未缴。（4 分）

函数 2：输入用户编号，查找到该用户信息并将该用户的缴费情况都设为缴清。（4 分）

函数 3：设计一个排序函数，将元素信息按编号有序排列。（4 分）

函数 4：输出所有的未缴费的用户名称。（3 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分 */

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct student          /*定义元素数据类型，DataType */
{
   char name[10],no[10];
   float amount,money;
   int pay;
}DataType;
typedef struct                  /*定义顺序表类型 SeqList */
{ DataType data[MAXLEN];
   int len;
}SeqList;
      void menu();
      void input(SeqList* L);
      void output1(SeqList* L);
      void output2(SeqList *L);
      int search(SeqList*L,DataType x);
      void sort(SeqList*L);
      void main()
      { SeqList *L;          /*主函数中，货物顺序表用 L 表示*/
        int sel,i;
        DataType x;
        L=(SeqList*)malloc(sizeof(SeqList));
         if (L==NULL)    exit(0);
         L->len=0;                          /*将表初始化为一个空表*/
         do{ menu();
```

```
            printf("\n input your choice:");       /*输入所选择的功能号*/
            scanf("%d",&sel);
            switch (sel)
              {case 1: input(L);              break;
                case 2: printf("\n input the no\n");
                              scanf("%s",x.no);
                i=search(L,x);
                if(i!=0) L->data[i].pay=1;              else printf("tne no is error\n");
                break;
              case 3:    sort(L);
                        break;
                 case 4:    output2(L);
                    break;
              case 5: output1(L);
                    break;
              case 0: exit(0);
             }
           }while (sel!=0);
       }
/*菜单函数*/
void menu()
{    printf("\n1.....录入用户信息\n");
     printf("2.....缴费\n");
     printf("3.....排序\n");
     printf("4.....输出未缴费人姓名\n");
     printf("5.....显示\n");
     printf("0.....退出\n");
}
/*录入用户信息*/
void input(SeqList* L)
{     int i=0;
      DataType x;
printf("\n initalizing.....\n");
      printf("\n input the no\n");
scanf("%s",x.no);
while(strcmp(x.no,"???")!=0) {
   printf("input name,amount\n");
       scanf("%s%f",x.name,&x.amount);
   x.money=x.amount*0.59;     x.pay=0;     i++;
   L->data[i]=x;
   printf("input the no\n");
       scanf("%s",x.no);
}
       L->len=i;
```

```
}
    /*查找*/
    int search(SeqList*L,DataType x)
    {
int i=1;    while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
        i++;
          if(i>L->len ) return 0;      return i;
    }
    /*排序*/
void sort(SeqList* L)
{
    int i,j;
    for( i=2;i<=L->len;i++)
       {L->data[0]=L->data[i];
        j=i-1;
        while(strcmp(L->data[0].no,L->data[j].no)<0 )
    {   L->data[j+1]=L->data[j] ;
              j--;
    }
       L->data[j+1]=L->data[0] ;
        }
  }
/*输出函数*/
void output1(SeqList *L)
{
    int i;
    printf("no              name            amount     money     pay\n");
    for(i=1;i<=L->len;i++)
              printf("%-12s%-12s%-7.2f%-7.2f%2d\n",L->data[i].no,L->data[i].name
,L->data[i].amount,L->data[i].money,L->data[i].pay);
}
    /*输出未缴费的用户名称*/
    void output2(SeqList *L)
    {
        int i;
        for(i=1;i<=L->len;i++)
              if(L->data[i].pay==0)     printf("%-12s%\n",L->data[i].name);
    }
```

12. /*设有一个超市商品信息表(包括商品编号 no、商品名称 name、商品库存量 amount 和商品
单价 price)

编程实现以下功能：

（1）货物信息录入：输入一批货物信息，货号为"000"时结束。（3 分）

（2）购物管理：输入若干货物货号、数量（即客户所购买的货物信息），货号为"000"时结束，输出所购买的每个货物货号、名称、数量、单价、金额（单价通过查找得到，金额=单价×数量）；最后输出总的价格。（10 分）

（3）货物信息输出:输出所有货物的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct student          /*定义元素数据类型，DataType */
{
   char name[10],no[10];
   float price;
   int amount;
}DataType;
typedef struct                  /*定义顺序表类型 SeqList */
{ DataType data[MAXLEN];
   int len;
}SeqList;
      void menu();
      void input(SeqList* L);
      void cost_menu(SeqList* L);
      void output(SeqList *L);
      int search(SeqList*L,DataType x);
      void main()
      { SeqList *L;           /*主函数中，货物顺序表用 L 表示*/
         int sel;
         L=(SeqList*)malloc(sizeof(SeqList));
          if (L==NULL)    exit(0);
          L->len=0;                                  /*将表初始化为一个空表*/
          do{ menu();
              printf("\n input your choice:");      /*输入所选择的功能号*/
              scanf("%d",&sel);
              switch (sel)
                {case 1: input(L);            break;
                  case 2: cost_menu(L);
                break;
             case 3:    output(L);
              break;
             case 0: exit(0);
          }
         }while (sel!=0);
      }
```

```
/*菜单函数*/
void menu()
{    printf("\n1.....录入货物信息\n");
     printf("2.....出库\n");
     printf("3.....显示\n");
     printf("0.....退出\n");
}
/*录入货物信息*/
void input(SeqList* L)
{     int i=0;
       DataType x;
printf("\n initalizing.....\n");
       printf("\n input the no\n");
scanf("%s",x.no);
while(strcmp(x.no,"000")!=0) {
   printf("input name,amount,price\n");
        scanf("%s%d%f",x.name,&x.amount,&x.price);
   i++;
   L->data[i]=x;
   printf("input the no\n");
        scanf("%s",x.no);
}
        L->len=i;
}
       /*查找*/
       int search(SeqList*L,DataType x)
       {
       int i=1;    while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
            i++;
              if(i>L->len ) return 0;      return i;
       }
/*账单打印*/
void cost_menu(SeqList* L)
{
int i,k=0,j;
       DataType x,a[MAXLEN ];
       float cost=0; printf(" input the no\n");
scanf("%s",x.no);
while(strcmp(x.no,"000")!=0) {
         i=search(L,x);
   if(i==0)              printf("this no    is error\n");
   else
   {
    printf("input amount\n");
```

```
                scanf("%d",&x.amount);
          if(x.amount>L->data[i].amount)
            printf("the amount error\n");
          else
          { L->data[i].amount-=x.amount; /*修改库存数量*/
             x.price=L->data[i].price;
             strcpy(x.name,L->data[i].name);
             k++;
             a[k]=x;              /*将所购买的货物信息放在数组 a 中*/
          }
        }
             printf(" input the no\n");
             scanf("%s",x.no);
    }
    printf("货号          名称          单价   数量   金额\n");
    for(j=1;j<=k;j++)
         {
      printf("%-12s%-12s%-6.1f%-6d%-7.1f\n",a[j].no,a[j].name,a[j].price,a[j].amo
unt,a[j].price*a[j].amount);
      cost+=a[j].price*a[j].amount;
    }
    printf("应付金额：%7.1f\n",cost);
         }
    /*输出函数*/
    void output(SeqList *L)
    {
        int i;
        printf("no              name          price    amount\n");
        for(i=1;i<=L->len;i++)
                printf("%-12s%-12s%-7.2f%-8d\n",L->data[i].no,L->data[i].name,L->d
ata[i].price,L->data[i].amount);
    }
```

13. /*有一学生成绩信息包括：姓名、学号、成绩、名次；编程实现以下功能：

（1）输入所有人姓名、学号、成绩，名次初始化为 0。（3 分）

（2）给定一学生学号，输出其成绩，若不存在该学号，给出相应的错误提示。（4 分）

（3）将学生信息按成绩由高到低排序，并将其名次存入该学生信息的"名次"中。（6 分）

（4）输出所有学生的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#define MAXLEN 100
typedef struct student          /*定义元素数据类型，DataType */
{
   char name[10],no[10];
   int score,rank;
}DataType;
typedef struct                  /*定义顺序表类型 SeqList */
{ DataType data[MAXLEN];
   int len;
}SeqList;
       void menu();
       void input(SeqList* L);
       int search(SeqList* L,DataType x);
       void sort(SeqList *L);
       void output(SeqList *L);
void main()
{ SeqList *SeqLStu;           /*主函数中，学生信息顺序表用 SeqLStu 表示*/
   DataType x;
   int sel,i;
     SeqLStu=(SeqList*)malloc(sizeof(SeqList));
     if (SeqLStu==NULL)    exit(0);
     SeqLStu->len=0;                                  /*将表初始化为一个空表*/
     do{ menu();
          printf("\n input your choice:");       /*输入所选择的功能号*/
          scanf("%d",&sel);
          switch (sel)
     {
     case 1: input(SeqLStu);             break;
     case 2: printf("input no\n");
               scanf("%s",x.no);
        i=search(SeqLStu,x);
                        if(i==0) printf("the no is error\n");
        else printf("score=%d",SeqLStu->data[i].score);
        break;
     case 3: sort(SeqLStu);           break;
     case 4: output(SeqLStu);         break;
     case 0: exit(0);
     }
   }while (sel!=0);
}
       /*菜单函数*/
       void menu()
       {    printf("\n1.............录入学生信息\n");
            printf("2.............查找\n");
```

```
            printf("3.............排序\n");
            printf("4.............显示\n");
            printf("0.............退出\n");
        }
             /*输入学号、姓名、成绩*/
        void input(SeqList* L)
        {     int i;
        printf("\n initalizing.............\n");
                printf("\n input the list's length :");
                    scanf("%d",&L->len);
                    getchar();
                    for (i=1; i<=L->len;i++)
            {    printf("enter no,name,score\n");
                        scanf("%s%s%d",L->data[i].no,L->data[i].name,&L->data[i].score)
;
                L->data[i].rank=0;            }
        }
    /*查找*/
    int search(SeqList*L,DataType x)
    {
    int i=1; while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
        i++;
         if(i>L->len ) return 0; return i;
    }
            /*排序*/
        void sort(SeqList* L)
        {
            int i,j;
            for( i=2;i<=L->len;i++)
               {L->data[0]=L->data[i];
                 j=i-1;
                 while(L->data[0].score>L->data[j].score )
            {   L->data[j+1]=L->data[j] ;
                        j--;
            }
               L->data[j+1]=L->data[0] ;
                 }
            for( i=1;i<=L->len;i++)
               L->data[i].rank=i;
          }
    /*输出函数*/
    void output(SeqList *L)
    {
        int i=1;       for(i=1;i<=L->len;i++)
```

```
            printf("%12s;%10s;%5d%3d\n",L->data[i].no,L->data[i].name,L->data[i].sc
ore,L->data[i].rank);
    }
```

14. /*学生考勤管理。学生信息包括：姓名、学号、考勤情况（迟到、旷课、按时上课），成绩及本次已经是第几次考勤。每个人有 20 次考勤；

编程实现以下功能：

（1）学生基本信息录入及初始化：输入所有学生的姓名、学号，将每个人的考勤成绩置 0，将考勤次数也置 0。（3 分）

（2）输入考勤情况：每次上课，输入本次考勤，从第一个人开始，依次输入每个人的考勤。（4 分）

（3）计算考勤成绩:迟到得 0.5 分，旷课得 0 分，正常上课得 1 分，计算每个人的考勤成绩。（5 分）

（4）输出所有人的学号、姓名、考勤情况、考勤成绩。（3 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{    char name[10];    /*name 表示学生姓名*/
     char no[12];      /*no 表示学号*/
     char kaoqin[21];    /*kaoqin 表示 20 次的考勤，q---缺课，c---迟到，k---旷课*/
     int k;    /*k 表示已经考勤到第几次，初始化时将其置为 0*/
     float score;   /*score 表示考勤成绩*/
} DataType;
typedef struct                  /*定义顺序表类型 SeqList */
     { DataType data[MAXLEN];
       int len;
     }SeqList;
void menu();
void input(SeqList* L);
void kaoqin(SeqList* L);
void score(SeqList *L);
void output(SeqList*L);
void main()
{ SeqList *L;          /*主函数中，货物顺序表用 L 表示*/
  int sel;
  L=(SeqList*)malloc(sizeof(SeqList));
   if (L==NULL)   exit(0);
   L->len=0;                              /*将表初始化为一个空表*/
   do{ menu();
       printf("\n input your choice:");     /*输入所选择的功能号*/
```

```
        scanf("%d",&sel);
        switch (sel)
         {case 1: input(L);            break;
           case 2: kaoqin(L);
          break;
       case 3:    score(L);
                 break;
           case 4:    output(L);
              break;
       case 0: exit(0);
    }
  }while (sel!=0);
}
      /*菜单函数*/
      void menu()
      {    printf("\n1.............录入学生信息\n");
           printf("2.............考勤\n");
           printf("3.............计算考勤成绩\n");
           printf("4.............显示\n");
           printf("0.............退出\n");
      }
      /*录入学生姓名、学号*/
      void input(SeqList* L)
      {     int i=0;
      printf("输入元素个数\n");
      scanf("%d",&L->len);
            printf("\n  输入学号、姓名\n");
      for(i=1;i<=L->len;i++)
      {
             scanf("%s%s",L->data[i].no,L->data[i].name);
         L->data[i].k=-1; /*未开始考勤*/
         L->data[i].score=0;     }

      }
/*输入考勤*/
void kaoqin(SeqList*L)
{
      int i,j;
      printf("输入考勤，q---缺课，c---迟到，z---按时上课\n");
      getchar();
      for(i=1;i<=L->len;i++)
      {     L->data[i].k++;     /*考勤到第 k 次*/
             j=L->data[i].k;
         L->data[i].kaoqin[j]=getchar(); /*输入第 i 个人的第 j 次考勤*/
```

```
        }
        }
    /*考勤成绩计算*/
    void score(SeqList* L)
    {     char q;
          int i,j;
          for( i=1;i<=L->len;i++)
         for (j=0;j<=L->data[i].k;j++)
         {q=L->data[i].kaoqin[j];
          if(q=='c')
         L->data[i].score+=0.5;      else if(q=='z')
         L->data[i].score+=1;       }
    }
          /*输出函数*/
          void output(SeqList *L)
          {
              int i,j;
              printf("no                name              kaoqin                    score\n");
              for(i=1;i<=L->len;i++)
              {    j=L->data[i].k;
                   j++;
                   L->data[i].kaoqin[j]='\0'; /*在最后一次考勤之后填入 0，所有考勤可以按
字符串处理*/
                     printf("%-12s%-12s%-21s%5.1f\n",L->data[i].no,L->data[i].name,L->dat
a[i].kaoqin,L->data[i].score);
              }
          }
```

15. /*有一个职工基本信息管理，职工信息包含：职工号、姓名、工资；编写完整程序，实现如下功能：

(1)录入函数 input:从键盘读入职工信息。(3 分）

(2)修改函数 modify:给定职工号,查找该职工，若找到修改其信息，若找不到，给出错误提示。(4 分)

(3)插入函数 insert：假定表中职工信息按职工号升序排列，任意给定一职工信息，使得插入后依然有序。(6 分)

（4）输出函数 output:输出所有人的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分。

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{    char no[8],name[10];
```

```
    float    sal;
} DataType;
typedef struct
{     DataType data[MAXLEN+1];
      int len;
}SeqList;
/*从键盘读入所有职工的信息*/
void input(SeqList*L)
{ int i;
   printf("input the length\n");
   scanf("%d",&L->len);
   printf("input no,name,sal\n");
   for(i=1;i<=L->len;i++)
    scanf("%s%s%f",L->data[i].no,L->data[i].name,&L->data[i].sal);
}
      /*查找*/
      int search(SeqList*L,DataType x)
      { int i=1;      while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)    /*查找元素 x 所在位置
*/
          i++;
         if(i>L->len )
            return 0;            return i;
      }
/*修改*/
void modify(SeqList*L,DataType x)
{
int i;
i=search(L,x);
if(i==0)      printf("not exist\n");
else
{printf("input name,sal\n");
   scanf("%s%f",L->data[i].name,&L->data[i].sal);
}
}
      /*插入*/
      void ins(SeqList*L,DataType x)
      {
      int i,j;
      i=search(L,x);
            if(i==0 )
      {
         i=1;   /*查找插入位置*/
         while(i<=L->len&&strcmp(x.no,L->data[i].no)>0)
          i++;
```

```
            for(j=L->len;j>=i;j--)
              L->data[j+1]=L->data[j];
            L->data[i]=x;
            L->len++;
          }
          else printf("the element already exist\n");
          }
    /*菜单*/
    void menu()
    {
    printf("1-------------录入信息\n");
    printf("2-------------修改\n");
    printf("3-------------插入\n");
    printf("4-------------输出\n");
    printf("0-------------退出\n");
    }
    void main()
    {
    SeqList* L;
    DataType x;
    int sel,i;
    L=(SeqList*)malloc(sizeof(SeqList));
    L->len=0; do
    { menu();
        printf("input your select\n");
        scanf("%d",&sel);
        switch(sel)
        {
        case 1:input(L);break;
        case 2:printf("input no\n");
               scanf("%s",x.no);
          modify(L,x);
          break;
        case 3:printf("input no,name,sal\n");
               scanf("%s%s%f",x.no,x.name,&x.sal);
          ins(L,x);
               break;
        case 4:for(i=1;i<=L->len;i++)
           printf("%10s%12s%7.1f\n",L->data[i].no,L->data[i].name,L->data[i].sal);
        }
    }while(sel!=0);
    }
```

16. /*有一个学生成绩管理，学生信息包含：学号、姓名、成绩；编写完整程序，实现如下功能：

(1)录入函数 input:从键盘读入学生信息。(3 分）

(2)修改函数 modify:给定学号,查找该学生，若找到修改其成绩，若找不到，给出错误提示。(4 分)

(3)排序函数 sort：将学生信息按成绩由高到低排序。 (6 分)

（4）输出函数 output:输出所有人的信息。（2 分）

主函数以菜单形式调用以上功能，类型定义 2 分，主函数 3 分 */

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{    char no[8],name[10];
     float    score;
} DataType;
typedef struct
{     DataType data[MAXLEN+1];
       int len;
}SeqList;

/*从键盘读入所有学生的信息*/
void input(SeqList*L)
{ int i;
   printf("input the length\n");
   scanf("%d",&L->len);
   printf("input no,name,score\n");
   for(i=1;i<=L->len;i++)
     scanf("%s%s%f",L->data[i].no,L->data[i].name,&L->data[i].score);
}
       /*查找*/
       int search(SeqList*L,DataType x)
       { int i=1;      while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)    /*查找元素 x 所在位置
*/
              i++;
           if(i>L->len )
              return 0;            return i;
       }
/*修改*/
void modify(SeqList*L,DataType x)
{
int i;
i=search(L,x);
if(i==0)      printf("not exist\n");
else
{printf("input new score\n");
```

```
        scanf("%f",&L->data[i].score);
    }
    }
        /*菜单*/
        void menu()
        {
        printf("1-------------录入信息\n");
        printf("2-------------修改\n");
        printf("3-------------排序\n");
        printf("4-------------输出\n");
        printf("0-------------退出\n");
    }
        /*排序*/
        void sort(SeqList* L)
        {
            int i,j;
            for( i=2;i<=L->len;i++)
               {L->data[0]=L->data[i];
                j=i-1;
                while(L->data[0].score<L->data[j].score )
            {   L->data[j+1]=L->data[j] ;
                    j--;
            }
               L->data[j+1]=L->data[0] ;
                }
          }
    /*输出函数*/
    void output(SeqList *L)
    {
        int i=1;      for(i=1;i<=L->len;i++)
            printf("%12s;%10s;%5.1f\n",L->data[i].no,L->data[i].name,L->data[i].sco
re);
    }
    void main()
        {
        SeqList* L;
        DataType x;
        int sel;
        L=(SeqList*)malloc(sizeof(SeqList));
        L->len=0;       do
        { menu();
            printf("input your select\n");
            scanf("%d",&sel);
            switch(sel)
```

```
        {
        case 1:input(L);break;
        case 2:printf("input no\n");
               scanf("%s",x.no);
          modify(L,x);
          break;
        case 3:sort(L);
               break;
        case 4:output(L);
               break;
        }
    }while(sel!=0);
    }
```

17. /*图书管每本图书包含：书号（no）、书名（name）、现存量（newnum）、总库存量（sumnum）四项信息，编写完整程序通过顺序表实现：

(1) 初始化：录入现有的所有图书的四项信息。（3 分）

(2) 清库：给定某书 x 的书号及数量，若找到该书，将该书的现存量及库存量减去 x 的数
量，若减后的库存量为 0，则删除该书信息；若找不到该书，则给出错误提示。（9 分）

(3) 显示：显示图书管所有藏书信息。（2 分）

(4) 主程序以菜单的方式调用以上功能。（4 分）

元素类型及顺序表类型定义（2 分）

*/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{   char no[8],name[10];
    int newnum,sumnum;
    float price;
} DataType;
typedef struct
{    DataType data[MAXLEN+1];
     int len;
}SeqList;
/*从键盘读入所有图书的信息*/
void input(SeqList*L)
{ int i;
  printf("input the length\n");
  scanf("%d",&L->len);
  printf("input no,name,newnum,sumnum,price\n");
  for(i=1;i<=L->len;i++)
```

```
        scanf("%s%s%d%d%f",L->data[i].no,L->data[i].name,&L->data[i].newnum,&L->da
ta[i].sumnum,&L->data[i].price);
    }
        /*查找*/
        int search(SeqList*L,DataType x)
        { int i=1;      while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)    /*查找元素 x 所在位置
*/
            i++;
          if(i>L->len )
            return 0;            return i;
        }
    /*清库*/
    void del(SeqList*L,DataType x)
    {
      int i,j;
      i=search(L,x);
      if(i==0)     /*找不到*/
       printf("the no id error\n");
      else
      {
       if(x.sumnum<=L->data[i].newnum)
       {
        L->data[i].newnum-=x.sumnum;
        L->data[i].sumnum-=x.sumnum;
        if(L->data[i].sumnum==0)     /*清库后库存为 0，则删除改书的信息*/
        {
         for (j=i+1;j<=L->len;j++)
           L->data[j-1]=L->data[j];
         (L->len)--;
        }
       }
       else
        printf("the amount is error\n");
      }
    }
        /*显示*/
        void output(SeqList*L)
        {int i;
          printf("no,name,newnum,sumnum,price\n");
          for(i=1;i<=L->len;i++)
            printf("%10s%12s%5d%5d%7.2f\n",L->data[i].no,L->data[i].name,L->data[i].
newnum,L->data[i].sumnum,L->data[i].price);
        }
        /*菜单*/
```

```
void menu()
{
printf("1----------录入信息\n");
printf("2----------清库\n");
printf("3----------显示\n");
printf("0----------退出\n");
      }
void main()
{
SeqList* L;
DataType x;
int sel;
L=(SeqList*)malloc(sizeof(SeqList));
L->len=0; do
{ menu();
    printf("input your select\n");
    scanf("%d",&sel);
    switch(sel)
    {
    case 1:input(L);break;
    case 2:printf("input no,amount\n");
          scanf("%s%d",x.no,&x.sumnum);
     del(L,x);
     break;
    case 3:output(L);
     break;
    }
}while(sel!=0);
}
```

18. /*图书管每本图书包含：书号（no）、书名（name）、现存量（newnum）、总库存量（sumnum）四项信息，编写完整程序通过顺序表实现：

（1） 初始化：录入现有的所有图书的四项信息。（3 分）

（2） 检索：给定书名，输出所有与给定书名相同的书的信息。（3 分）

（3） 价值估算：统计库中所有书的价钱。价钱为所有书的单价乘以库存量的累加和。（6 分）

（4） 显示：显示图书管所有藏书信息。（2 分）

（5） 主程序以菜单的方式调用以上功能。（4 分）

（6） 完成元素类型定义及顺序表类型定义（2 分）。

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
```

```
typedef struct
{    char no[8],name[21];
     int newnum,sumnum;
     float price;
} DataType;
typedef struct
{     DataType data[MAXLEN+1];
      int len;
}SeqList;
/*从键盘读入所有图书的信息*/
void input(SeqList*L)
{ int i;
   printf("input the length\n");
   scanf("%d",&L->len);
   printf("input no,name,newnum,sumnum,price\n");
   for(i=1;i<=L->len;i++)
    scanf("%s%s%d%d%f",L->data[i].no,L->data[i].name,&L->data[i].newnum,&L->da
ta[i].sumnum,&L->data[i].price);
}
      /*检索*/
      void search(SeqList*L,DataType x)
      { int i;
         printf("no     name      newnum          sumnum          price\n");
         for (i=1;i<=L->len;i++)
           if(strcmp(L->data[i].name,x.name)==0)
printf("%8s%20s%5d%5d%5.1f\n",L->data[i].no,L->data[i].name,&L->data[i].
newnum,&L->data[i].sumnum,&L->data[i].price);
      }
/*价值估算*/
float volue(SeqList*L)
{ float v=0;     int i;
   for(i=1;i<=L->len;i++)
    v+=L->data[i].price*L->data[i].sumnum;
   return v;
}
      /*显示*/
      void output(SeqList*L)
      {int i;
         printf("no,name,newnum,sumnum,price\n");
         for(i=1;i<=L->len;i++)
           printf("%10s%12s%5d%5d%7.2f\n",L->data[i].no,L->data[i].name,L->data[i].
newnum,L->data[i].sumnum,L->data[i].price);
      }
/*菜单*/
```

```
void menu()
{
printf("1----------录入信息\n");
printf("2----------检索\n");
printf("3----------估价\n");
printf("4----------输出\n");
printf("0----------退出\n");
      }
void main()
{
SeqList* L;
DataType x;
int sel;
float v;
L=(SeqList*)malloc(sizeof(SeqList));
L->len=0; do
{ menu();
    printf("input your select\n");
    scanf("%d",&sel);
    switch(sel)
    {
    case 1:input(L);break;
    case 2:printf("input name\n");
          scanf("%s",x.name);
     search(L,x);
     break;
    case 3:v=volue(L);
          printf("the sum vulue is%10.2f\n",v);
          break;
    case 4:output(L);
          break;
    }
}while(sel!=0);
}
```

19. /*图书管每本图书包含：书号（no）、书名（name）、现存量（newnum）、总库存量（sumnum）四项信息，编写完整程序通过顺序表实现：

（1） 初始化：录入现有的所有图书的四项信息。（3 分）

（2） 查找：给定书号，在表中查找该书，若存在返回其位置，否则返回 0。(4 分)

（3） 进书：给定某个图书的书名、书号、数量，若此次购进的是图书馆已有的图书，则修改其现存量和总库存量；若此次购进的图书是新书，按书号有序插入到表中。（假定表中元素按书号升序排列）（7 分）

（4） 主程序以菜单的方式调用以上功能。（4 分）

（5） 完成元素类型定义及顺序表类型定义（2 分）

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct
{    char no[8],name[21];
     int newnum,sumnum;
     float price;
} DataType;
typedef struct
{     DataType data[MAXLEN+1];
      int len;
}SeqList;
/*从键盘读入所有图书的信息*/
void input(SeqList*L)
{ int i;
   printf("input the length\n");
   scanf("%d",&L->len);
   printf("input no,name,newnum,sumnum,price\n");
   for(i=1;i<=L->len;i++)
     scanf("%s%s%d%d%f",L->data[i].no,L->data[i].name,&L->data[i].newnum,&L->da
ta[i].sumnum,&L->data[i].price);
}
      /*查找*/
      int search(SeqList*L,DataType x)
      { int i=1;      while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)
           i++;
         if(i>L->len )
            return 0;            return i;
      }
/*进书*/
void ins(SeqList*L,DataType x)
{
   int i,j;
   i=search(L,x);
   if(i==0)     /*找不到*/
   {
      i=1;   /*查找插入位置*/
   while(i<=L->len&&strcmp(x.no,L->data[i].no)>0)
     i++;
   for(j=L->len;j>=i;j--)
     L->data[j+1]=L->data[j];
   L->data[i]=x;
```

```
        L->len++;
        }
        else     /*找到*/
        {
              L->data[i].newnum+=x.sumnum;
           L->data[i].sumnum+=x.sumnum;
        }
     }
          /*显示*/
          void output(SeqList*L)
          {int i;
             printf("no,name,newnum,sumnum,price\n");
             for(i=1;i<=L->len;i++)
               printf("%10s%12s%5d%5d%7.2f\n",L->data[i].no,L->data[i].name,L->data[i].
newnum,L->data[i].sumnum,L->data[i].price);
          }
     /*菜单*/
     void menu()
     {
     printf("1----------录入信息\n");
     printf("2----------查找\n");
          printf("3----------进书\n");
     printf("4----------显示\n");
     printf("0----------退出\n");
          }
     void main()
     {
     SeqList* L;
     DataType x;
     int sel,i;
     L=(SeqList*)malloc(sizeof(SeqList));
     L->len=0; do
     { menu();
          printf("input your select\n");
          scanf("%d",&sel);
          switch(sel)
          {
          case 1:input(L);break;
          case 2:printf("input no\n");
                scanf("%s",x.no);
           i=search(L,x);
           if(i==0)          printf("this no error\n");
           else
             printf("%20s%5d%5d%7.2f\n",L->data[i].name,L->data[i].newnum,L->data[i].
```

```
sumnum,L->data[i].price);
            break;
               case 3:printf("input no,name,newnum,sumnum,price\n");
                        scanf("%s%s%d%d%f",x.no,x.name,&x.newnum,&x.sumnum,&x.price);
            ins(L,x);
            break;
          case 4:output(L);
            break;
          }
      }while(sel!=0);
      }
```

20. /*学生学费管理。学生信息包括：姓名、学号、学费、已缴学费、欠费。编写完整程序通过顺序表实现：

（1） 初始化：录入所有人的姓名、学号、学费。（3 分）

（2） 缴费：输入学号及已缴学费，欠费=学费-已缴学费。（5 分）

（3） 催缴学费：对欠费为正数的人，输出其姓名、学号、欠费并统计所有人的欠费总数，总的欠费数目以函数值的形式返回（6 分）

（4） 主程序以菜单的方式调用以上功能。（4 分）

（5） 完成元素类型定义及顺序表类型定义（2 分）。

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 100
typedef struct student          /*定义元素数据类型，DataType */
{
   char no[10],name[10];
   float xf,yjxf,qf;   /*xf---学费  yjxf---已缴学费   qf---欠费*/
}DataType;
typedef struct                  /*定义顺序表类型 SeqList */
{ DataType data[MAXLEN];
   int len;
}SeqList;
      void menu();
      void input(SeqList* L);
      int search(SeqList* L,DataType x);
      void jf(SeqList * L,DataType x);
      float cjxf(SeqList *L);
      void output(SeqList* L);
void main()
{ SeqList *SeqLStu;          /*主函数中，学生信息顺序表用 SeqLStu 表示*/
   DataType x;
   int sel;
```

```
   float zqf;
   SeqLStu=(SeqList*)malloc(sizeof(SeqList));
    if (SeqLStu==NULL)    exit(0);
    SeqLStu->len=0;                                    /*将表初始化为一个空表*/
    do{ menu();
        printf("\n input your choice:");       /*输入所选择的功能号*/
        scanf("%d",&sel);
        switch (sel)
    {
    case 1: input(SeqLStu);           break;
    case 2: printf("input no,yjxf\n");
            scanf("%s%f",x.no,&x.yjxf);
       jf(SeqLStu,x);
       break;
    case 3: zqf=cjxf(SeqLStu);
            printf("总欠费：%8.1f",zqf);
            break;
    case 4: output(SeqLStu);          break;
    case 0: exit(0);
    }
   }while (sel!=0);
}
      /*菜单函数*/
      void menu()
      {    printf("\n1.............录入学生信息\n");
           printf("2.............缴费\n");
           printf("3.............催缴学费\n");
           printf("4.............显示\n");
           printf("0.............退出\n");
      }
      /*输入学号、姓名、学费*/
      void input(SeqList* L)
      {     int i;
      printf("\n initalizing.............\n");
                printf("\n input the list's length :");
                     scanf("%d",&L->len);
                     getchar();
                     for (i=1; i<=L->len;i++)
           {    printf("enter no,name,xf\n");
                          scanf("%s%s%f",L->data[i].no,L->data[i].name,&L->data[i].xf);
                L->data[i].yjxf=0;                         L->data[i].qf=L->data[i].xf;
           }
      }
/*查找*/
```

```
int search(SeqList*L,DataType x)
{
int i=1; while(i<=L->len&&strcmp(L->data[i].no,x.no)!=0)   /*查找元素 x 所在位置*/
    i++;
     if(i>L->len ) return 0; return i;
}
          /*缴费*/
     void jf(SeqList * L,DataType x)
     {
     int i;
     i=search(L,x);
     if(i==0)          printf("the no is error\n");
     else
     {     L->data[i].yjxf=x.yjxf;
        L->data[i].qf=L->data[i].xf-L->data[i].yjxf;
     }
     }
/*催缴学费*/
float cjxf(SeqList *L)
{
int i;
float amount=0;       printf("学号        姓名        欠费\n");
    for(i=1;i<=L->len;i++)
        if(L->data[i].qf>0)       {
     printf("%12s%10s%8.1f\n",L->data[i].no,L->data[i].name,L->data[i].qf);
     amount+=L->data[i].qf;
    }
    return amount;
}
     /*输出函数*/
     void output(SeqList *L)
     {
         int i=1;          printf("学号            姓名          学费       已缴学费  欠费\n");
         for(i=1;i<=L->len;i++)
             printf("%-12s%-10s%-8.1f%-8.1f%-8.1f\n",L->data[i].no,L->data[i].name
,L->data[i].xf,L->data[i].yjxf,L->data[i].qf);
     }
```

第三章　链表

一.填空题

1. 单链表中增加头结点的目的是为了 **简化操作** 。
2. 用单链表方式存储线性表，每个结点需要两个域，一个是数据域，另一个是 **指针域** 。
3. 在一个单链表的 p 所指的结点之后插入一个 s 所指的结点，应执行的操作是：

s->next=p->next 和 **p->next =s** 。

4. 某带头结点的单链表的头指针为 head，判定该链表为空的条件是 **head->next==NULL**。

5. 在带有头结点的单链表 HL 中，要在首元素之前插入一个由指针 p 指向的结点，则应执行 p->next=HL->next 及 **HL->next=p** 操作。

6. 设指针变量 p 指向单链表中某结点 A，则删除结点 A 的后继结点需要的操作为 **p->next=p->next->next**（不考虑存储空间的释放）。

7. 链式存储结构的线性表其元素之间的逻辑关系是通过结点的 **指针** 域来表示的。

8. 单循环链表 L 中指针 P 所指结点为尾结点的条件是 **P->next==L** 。

9. 访问具有 n 个结点的单链表中任意一个结点的时间复杂度是 **O(n)** 。

10. 在单链表 L 中，指针 P 所指的结点有后继结点的条件是 **P->next!=NULL** 。

11. 链式存储结构的线性表中所有元素的地址 **不一定** 连续。

12. 链式存储结构的线性表中，插入或删除某个元素所需的时间与其位置 **无** 关（填有或无）

13. 在单链表 L 中，指针 P 所指的结点为尾结点的条件是 **P->next==NULL** 。

14. 头插法建立单链表时，元素的输入顺序与在链表中的逻辑顺序是 **相反** 的。

15. 尾接法建立单链表时，元素的输入顺序与在链表中的逻辑顺序是 **相同** 的。

16. 若要将一个单链表中的元素倒置，可以借助 **头插法** 建立单链表的思想将链表中的结点重新放置。

17. 线性表用链式存储结构存储比用顺序存储结构存储所占的存储空间 **不一定** 多（填一定或不一定）。

18. 线性表中逻辑上相邻的两个元素其存放位置 **不一定** 相邻。

二.简答题

1. 下列算法完成用头插法为数组 a 中的 n 个元素建立一个带头结点的单链表，并返回其头指针，请在空的下划线上填写合适的内容完成该算法。

```
NodeType *creatl2 (DataType a[ ],int n)
{   NodeType *head, *s;
    int i;
    head=(NodeType*)malloc(sizeof(NodeType));  /*为头结点申请空间*/
    if(head==NULL)  return  NULL;                /*申请空间未成功，返回空*/
    head->next=NULL  ;                          /*链表初始化为空*/
    for (i=1; i<=n; i++)
   {   s=(NodeType*)malloc(sizeof(NodeType));
       s->data=a[n-i];    /*给新结点的数据域赋值*/
       s->next=head->next;              /*新结点的指针域指向头的下一个元素*/
       head->next=s;                    /*头结点指向新结点*/
   }
  return head;      /*返回头指针*/
}
```

2. 下列算法完成用尾接法为数组 a 中的 n 个元素建立一个带头结点的单链表，并返回其头指针，请在空的下划线上填写合适的内容完成该算法。

```
NodeType* creatl1(DataType a[ ],int n)
{   NodeType *head,*tail,*s;          /* head 为头指针，  tail 为尾指针*/
    int i;
    head=initl( );                                  /*链表初始化*/
```

```
    if (head==NULL) return NULL;
     tail=head ;                          /*尾指针初始化为头指针*/
     for (i=0;i<n;i++)
   {   s=(NodeType*)malloc(sizeof(NodeType)) ;     /*为新结点申请空间*/
       if (s==NULL) return NULL;                  /*申请不到空间，则返回空*/
       s->data=a[i] ;                             /*给新结点的数据域赋值*/
       s->next=NULL;                              / *给新结点的指针域赋值*/
       tail->next=s;                                  /*将新结点接到表尾*/
       tail=s   ;                                 /*新结点为当前的表尾*/
     }
   return head;                                       /*返回头指针*/
}
```

3. 什么是链式存储结构？链式存储结构的优点和缺点各是什么？

答：**链式存储结构是对逻辑上相邻的元素不要求其物理位置相邻，元素间的逻辑关系通过附加的指针字段来表示。通常用指针来实现。**

链式存储有两个优点：

（1）进行插入、删除时，不需移动表中的元素，只需修改指针，效率较高。

（2）不需预先分配存储空间，当需要存储空间时，临时开辟空间，不会造成空间浪费。

链式存储有三个缺点:

（1）方法比顺序存储复杂，不易实现。

（2）为表示结点的逻辑关系需要增加额外空间。

（3）不具有按元素序号随机访问的特点。

4. 单链表中头指针、头结点和第一个结点（开始结点）三者的区别是什么？

答：头指针是头结点（或第一个结点）的地址。单链表如果不带头结点的话，头指针是第一个结点的地址；单链表如果带头结点的话，头指针是头结点的地址。

头结点是第一个结点前面的一个结点，它的数据域无定义，指针域中存放的是第一个数据结点的地址，空表时为空。

第一个结点就是第一个存放数据的结点。

5. 已知整型带头结点的单链表 H，下列算法实现将该链表中的元素倒置，若原链表中的元素为 1，2，3，4，5； 倒置后在则变成 5，4，3，2，1. 请在空的下划线上填写合适的内容完成该算法。

```
void reverse ( NodeType *H )
      {   NodeType *p;
          p=H->next;          /*p 指向首元结点*/
          H->next=NULL;     /*头结点的指针域为空*/
          while( p!=NULL   )        /*当 p 结点不为空时*/
              {     q=p;
                    p=p->next;   /*p 指针后移*/
                    /*将 q 所指向的结点插入到头结点之后*/
                     q->next= H->next;
                       H->next=q;
    }
}
```

三.算法设计题

1. 设单链表的结点类型定义如下：

```
typedef struct node
{
    int    data；
    struct node *next；
}NodeType；
```

设计一算法利用单链表中原来的结点将该单链表中元素倒置。即：原来的元素如果为：1，2，3，4，5；倒置后变成 5，4，3，2，1。

2. 设单链表的结点类型定义如下：

```
typedef struct node
{
    int    data；
    struct node *next；
}NodeType；
```

设计一算法尽量利用原表中的结点，将该链表划分成两个链表，一个链表中放置正数和零，另外一个链表中放置负数。

3. 设单链表的结点类型定义如下：

```
typedef struct node
{
    int data；
    struct node *next；
}NodeType；
```

设计一算法删除单链表中重复的结点。

4. 设单链表的结点类型定义如下：

```
typedef struct node
{
    int data；
    struct node *next；
}NodeType；
```

设计一算法删除单链表中所有的元素 x。

5. 设单链表的结点类型定义如下：

```
typedef struct node
{
    int    data；
    struct node *next；
}NodeType；
```

设计一算法尽量用原表中的结点将两个升序排列的单链表合并为一个升序排列的单链表。

6. 设单链表的结点类型定义如下：

```
typedef struct node
{
    int data；
    struct node *next；
}NodeType；
```

设计一算法尽量用原表中的结点将两个升序排列的单链表合并为一个降序排列的单链表。

7. 设单链表的结点类型定义如下：

```
typedef struct node
{
    int    data；
    struct node *next；
}NodeType；
```

设计一算法在带头结点的单链表 L 中，将新元素 x 插入到带头结点的单链表 L 中的元素 elm 之后，若不存在元素 elm，则插入到最后。

8. 设单链表的结点类型定义如下：

```
typedef struct node
{
    int    data；
    struct node *next；
}NodeType；
```

设计一算法在带头结点的单链表 L 中，将新元素 x 插入到带头结点的单链表 L 中的元素 elm 之前，若不存在元素 elm，则插入到最后。

9. 设单链表的结点类型定义如下：

```
typedef struct node
{
    int    data；
    struct node *next；
}NodeType；
```

设计一算法：将任意给定的一个元素 x 插入到升序排列的单链表中，使得该链表中的数据依然有序。

10. 已知一带头结点的单链表 L，结点类型为：

```
typedef struct node
{
int    data;
struct node *next;
}nodetype;
```

设计一算法，求该链表中值最大的结点并返回该结点的指针，若链表为空，则返回一个空指针。

解答:

1. 设计一算法利用单链表中原来的结点将该单链表中元素倒置。即：原来的元素如果为：1，2，3，4，5；倒置后变成 5，4，3，2，1。

```
void reverse (NodeType * H)
        {   NodeType *p;
            p=H-> next;
            H->next=NULL;
            while(p)
             {    q=p; p=p->next;
                  q->next=H->next;
                  H->next=q;      }
          }
```

2. 设计一算法尽量利用原表中的结点，将该链表划分成两个链表，一个链表中放置正数和零，另外一个链表中放置负数。

```
/*拆分单链表函数。正数和零仍放在 head 中,负数放入 head1 中，通过函数返回 head1*/
NodeType *   chaifen(NodeType *head)
{
  NodeType *p,*q,*r,*head1;    head1=( NodeType *)malloc(sizeof(NodeType ));  //为新表建立头结点
head1->next=NULL;
r=head1; p=head;
while(p->next)              // 从原表第一个结点开始
if(p->next->data<0)
    {   q=p->next;          //q 指向负数结点
        p->next=q->next;   //负数结点从原表取出
        r->next=q;          // 负数结点以尾接法插入到新表
        r=q;
}
else p=p->next;
      r->next=NULL;      //新表尾结点的指针域为空
  return head1; }
```

3. 设计一算法删除单链表中重复的结点。

```
void pur_LinkList(NodeType * H)
{ NodeType *p,*q,*r;
   p=H->next;
     if (p==NULL) return；
     while(p->next)
     {   q=p; //找与 p 重复的结点并删除
         while(q->next)
           if (q->next->data==p->data)
                    { r=q->next;   q->next=r->next; free( r ); }
           else q= q ->next;
```

```
            p=p->next;
          }
      }
```

4. 设计一算法删除单链表中所有的元素 x。

```
void pur_LinkList(NodeType* H)
{ NodeType *p,*q,*r;
    q=H;
     while(q->next!=NULL)
       {   //找 x 并删除
           if (q->next->data==x)
           {   p=q->next;   q->next=p->next; free(p ); }
           else q= q ->next;
       }
}
```

5. 设计一算法尽量用原表中的结点将两个升序排列的单链表合并为一个升序排列的单链表。

```
LinkList merge(NodeType *A, NodeType * B)
{    NodeType *p,*q,*tail;
     p=A->next;   q=B->next; //从两个表第一个结点开始
     C=A;
     C->next=NULL;tail=C;
     free( B );
     while(p&&q)      //当两个表都有结点时
      {   if   (p->data<q->data)     //取出两个表当前小的元素
             { s=p; p=p->next; }
         else{s=q;q=q->next;}
         tail->next=s;     //  用尾接法插入到新表中
         tail=s;
}
            if   (p==NULL)   p=q; //将剩余元素放入新表
            tail->next=p;
         }
```

6. 设计一算法尽量用原表中的结点将两个升序排列的单链表合并为一个降序排列的单链表。

```
LinkList merge(NodeType * A, NodeType *B)
{    NodeType* C,*p,*q;
     p=A->next;   q=B->next; //从两个表第一个结点开始
     C=A;
     C->next=NULL;
     free( B );
     while(p&&q)      //当两个表都有结点时
```

```
        {   if   (p->data<q->data)     //取出两个表当前小的元素
              { s=p; p=p->next; }
            else{s=q;q=q->next;}
            s->next=C->next;     // 用头插法插入到新表中
            C->next=s;
    }
            if   (p==NULL)   p=q; //将剩余元素放入新表
            while( p )
                  { s=p;   p=p->next;   s->next=C->next;      C->next=s;}
            }
```

7. 设计一算法在带头结点的单链表 L 中，将新元素 x 插入到带头结点的单链表 L 中的元素 elm 之后，若不存在元素 elm，则插入到最后。

```
void   link_ins(NodeType *head,datatype x,datatype elm)
{
  NodeType *p,*s;
  p=head;
  while(p->next!=NULL&&p->next->data!=elm)
      p=p->next;
s=( NodeType *)malloc(sizeof(NodeType ));
    s->data=x;
    s->next=NULL;
    if(p->next==NULL)   //没找到时，p 指向最后一个结点
        p->next=s;
    else                          //找到时，p->next 指向 elm 元素
     {  s->next=p->next->next;
        p->next->next=s;
     }
    }
```

8. 设计一算法在带头结点的单链表 L 中，将新元素 x 插入到带头结点的单链表 L 中的元素 elm 之前，若不存在元素 elm，则插入到最后。

```
void   link_ins(NodeType *head,datatype x,datatype elm)
{
  NodeType *p,*s;
  p=head;
  while(p->next!=NULL&&p->next->data!=elm)
      p=p->next;
s=( NodeType *)malloc(sizeof(NodeType ));

    s->data=x;
    s->next=NULL;
    if(p->next==NULL)   //没找到时，p 指向最后一个结点
        p->next=s;
```

```
else                    //找到时，p->next 指向 elm 元素.p 指向 elm 之前的结点
 {  s->next=p->next;
    p->next=s;
}
}
```

9. 设计一算法：将任意给定的一个元素 x 插入到升序排列的单链表中，使得该链表中的数据依然有序。

```
void ins(NodeType* L, int x)
{ NodeType* p=L,*s，*q;   //找第一个比 x 大的元素的前驱结点

        while(p->next!=NULL&&p->next->data<x)
            p=p->next;
        //若 x 比所有元素都大，p 指向最后一个结点；否则指向第一个比 x 大的元素的前驱结点，两种情况下，都是将 x 插入到 p 之后
    s=(NodeType*)malloc(sizeof(NodeType));//为新结点申请空间
    s->data=x;
    s->next=p->next;   //新结点插到 p 之后
    p->next=s;
    }
```

10. 设计一算法，求该链表中值最大的结点并返回该结点的指针，若链表为空，则返回一个空指针。

```
NodeType *maxpNodeType* L)
{   NodeType *p=L->next,*s;
    if(p!=NULL)
{    s=p;   //第一个结点初始化为最大的结点
        p=p->next;   //从第二个开始比较
        while(p!=NULL)
        {    if(p->data>s->data)   s=p;   //s 指向当前最大的结点
             p=p->next;
  }
return s；
  }
else
    return   NULL；
}
```

第四章　栈和队列

一.填空题

1. 为了解决普通顺序存储结构队列的“假溢出”现象，节约内存单元，通过在队列操作中加入数学中的 **求余** 运算，可以将其构造成循环队列。

2. 解决计算机与打印机之间速度不匹配问题，须设置一个数据缓冲区，应是一个 **队列** 结

构。

3. 循环队列是为了解决 **假溢出** 问题而将一个顺序表想像成一个首尾相接的顺序表。

4. 在循环队列中，如果其头指针为 front，队列中元素个数为 len，则该队列为空队的条件是 **len==0** 。

5. 队列的插入操作在 **队尾** 进行。

6. 队列的删除操作在 **队头** 进行。

7. 一个栈的输入序列是：1，2，3 则不可能的栈输出序列是 3，1，2 。

8. 队列是限制插入只能在表的一端，而删除在表的另一端进行的线性表，其特点是 **先进先出（或后进后出）** 。

9. 队列的特点是 **先进先出（或后进后出）** 。

10. **队列** 称作先进先出表。

11. 栈和队列是两种 **操作受限制的** 线性表。

12. 在作进栈运算时要判别栈是否 **已满** 。

13. 在作退栈运算时要先判别栈是否 **已空** 。

14. 在循环队列中，为了能够区分队满和队空，往往少用一个元素空间。在这种情况下，队满的条件是 **(*rear* + 1) % MAXSIZE == *front*** (假定循环队列的最大容量为 MAXSIZE，队首是 front，队尾是 rear)。

15. 栈的特点是 **先进后出（或后进先出）** 。

二.简答题

1. 循环队列的优点是什么？如何判别循环队列的"空"或"满"？

答：循环队列的优点是：它可以克服顺序队列的"假上溢"现象，能够使存储队列的向量空间得到充分的利用。　　判别循环队列的"空"或"满"通常有两种方法：

（1）另设一个变量 num 记录当前队列中的元素个数,当 num==0 时队空， num==maxsize 时队满。

（2）少用一个存储单元(即少存一个元素), 队空条件为 front == rear，队满条件为(rear + 1) % maxsize == front 。

2. 栈的特点是什么？队列的特点是什么？

答：栈的特点是先进后出，队列的特点是先进先出。

3. 什么是递归？递归有什么优点？

答：一个直接调用自己或通过一系列调用间接调用自己的过程称为递归。

递归程序结构清晰，可读性强，而且容易用数学归纳法来证明程序的正确性。

4. 什么是栈顶？什么是栈底？

答：允许插入和删除的一端是栈顶。

不允许插入和删除的一端是栈底。

5. 出栈和读栈顶元素有无区别？若有，其区别是什么？

答：出栈和读栈顶元素有区别。在栈 s 存在且非空的情况下，出栈是将栈 s 的顶部元素从栈中删除，栈中少了一个元素，栈发生变化；读栈顶元素是读栈顶元素，栈不变化。

6. 链栈和单链表有什么相同点和不同点？

答：链栈实际就是不带头结点的单链表。其结构完全一样，不同的是链栈只允许头插、头删，而单链表可在其它位置插入和删除。

7. 链队列和单链表有什么相同点和不同点？

答：相同点：结点类型相同。

不同点：链队列只允许头删、尾插，而单链表可在其它位置插入和删除。

8. 什么是顺序表？什么是顺序栈？两者之间有什么联系和区别？

答：线性表的顺序存储是指在内存中用地址连续的一块存储空间对线性表中的各元素按其逻辑顺序依次存放，用这种存储形式存储的线性表称为顺序表。利用顺序存储方式实现的栈称为顺序栈。顺序栈是顺序表的特殊情况，顺序栈只允许在栈顶插入和删除，而顺序表可在其它位置插入和删除。

9. 设栈 S 和队列 Q 的初始状态均为空，元素 e1,e2,e3,e4,e5,e6 依次通过栈 S,一个元素出栈后即进入队列 Q，若 6 个元素出队序列为 e2,e4,e3,e6,e5,e1, 则栈 S 的容量至少为多少？写出其分析过程。

答：至少为 3（分析过程略）。

第五章　串、数组和广义表

一.填空题

1. 将 10 阶的下三角矩阵 A 按列优先顺序压缩存储在一维数组 C 中，则 C 数组的大小应为 **55** 。

2. 已知二维数组 A[10][20]采用行优先方式存储，每个元素占 2 个存储单元,并且 A[0][0]的存储地址是 1000, 则 A[2][8]的存储地址是 **1096** 。

3. 对 5×5 的下三角矩阵进行压缩存储时，如果每个元素要占 3 个字节，共需的存储单元数为 **45** 个。

4. 设数组 B[0..3，1..5]，数组中的任一元素 B[i，j]均占两个单元，从首地址 SA 开始，把数组 B 按行序为主序进行存放，则元素 B[3，4]的地址为 SA+36 。

5. 已知具有 n 个元素的一维数组采用顺序存储结构，每个元素占 k 个存储单元，第一个元素的地址为 LOC(a1)，那么，LOC(ai)= LOC(a1)+（i-1）*k 。

6. 字符串（简称串）是一种特殊的线性表，它的 **数据元素** 仅由一个字符组成。

7. 字符串和线性表一样，它通常采用的存储方式也是 **顺序存储** 和 **链式存储** 。

8. 二维数组的顺序存储可以采用以行为主序存储，也可采用 **以列为主序** 存储。

9. 值相同元素或者零元素分布有一定规律的矩阵称为 **特殊矩阵** 。

10. 对称矩阵是满足 $a_{ij}= a_{ji}(0\leq i,j\leq n-1)$ 条件的矩阵。

11. 有较多值相同元素或较多零元素，且值相同元素或者零元素分布没有一定规律的矩阵称为 **稀疏矩阵** 。

12. 稀疏矩阵的压缩存储除了要保存非零元素的值外，还要保存非零元素在矩阵中的 **行和列** 。

13. 矩阵压缩存储是指为多个值相同的元素分配一个存储空间，对 **值为 0 的元素** 不分配存储空间。

14. 稀疏矩阵的压缩存储通常采用 **三元组表** 和 **十字链表** 存储。

15. n 阶对称矩阵元素可以只存储下三角部分，共需 n(n+1)/2 个单元的空间。

二.简答题

1. 简述下列术语：空串与空白串，主串和子串，串值和串名。

答：空串与空白串：串中所包含的字符个数称为该串的长度。长度为零的串称为空串，它不包含任何字符。仅由一个或多个空格组成的串称为空白串。空串和空白串不同，例如“ ”和“”分别表示长度为 1 的空白串和长度为 0 的空串。

主串与子串：串中任意连续的字符组成的子序列称为该串的子串。包含子串的串相应的称为主串。

串值和串名：串是零个或多个字符组成的有限序列。一般记作 S=“$a_1a_2a_3…a_n$”，其中 a_i(1

≦i≦n) 是一个任意字符，可以是字母、数字或其它字符，S 是串名，引号引起来的字符序列为串值。

2. 按以行序为主序的存储顺序，列出四维数组 a[2][3][2][4]中所有元素在内存中的存储顺序。

答：a[0][0][0][0]、a[0][0][0][1]、a[0][0][0][2]、a[0][0][0][3]、
a[0][0][1][0]、a[0][0][1][1]、a[0][0][1][2]、a[0][0][1][3]、
a[0][1][0][0]、a[0][1][0][1]、a[0][1][0][2]、a[0][1][0][3]、
a[0][1][1][0]、a[0][1][1][1]、a[0][1][1][2]、a[0][1][1][3]、
a[0][2][0][0]、a[0][2][0][1]、a[0][2][0][2]、a[0][2][0][3]、
a[0][2][1][0]、a[0][2][1][1]、a[0][2][1][2]、a[0][2][1][3]、
a[1][0][0][0]、a[1][0][0][1]、a[1][0][0][2]、a[1][0][0][3]、
a[1][0][1][0]、a[1][0][1][1]、a[1][0][1][2]、a[1][0][1][3]、
a[1][1][0][0]、a[1][1][0][1]、a[1][1][0][2]、a[1][1][0][3]、
a[1][1][1][0]、a[1][1][1][1]、a[1][1][1][2]、a[1][1][1][3]、
a[1][2][0][0]、a[1][2][0][1]、a[1][2][0][2]、a[1][2][0][3]、
a[1][2][1][0]、a[1][2][1][1]、a[1][2][1][2]、a[1][2][1][3]、

3. 给定整型数组 b[3][5]，已知每个元素占 2 个字节，b[0][0]的存储地址为 1200，试求在行序为主序的存储方式下：

（1） b[2][4]的存储地址。

（2） 该数组占用的字节个数。

答：（1）loc(b[2][4])= loc(b[0][0])+(2*5+2)*2=1200+24=1224

应该是：loc(b[2][4])= loc(b[0][0])+(2*5+4)*2=1200+28=1228

（2）3*5*2=30

4. 已知数组 A[8][10]，问按列存储数组元素时 A[4][6]的起始地址与数组 A 按行存储时的哪一个元素的起始地址相同？稀疏矩阵的压缩存储有哪两种方法？

答：（1）假定每个元素占 L 个存储单元，按列存储时 loc(A[4][6])= loc(A[0][0])+(8*6+4)*L，设其与数组 A 按行存储时的 A[i][j]的起始地址相同，则

loc(A[0][0])+(8*6+4)*L = loc(A[0][0])+(10*i+j)*L

即 8*6+4=10*i+j

所以 i=5，j=2

也就说，数组 A[8][10]按列存储数组元素时 A[4][6]的起始地址与数组 A 按行存储时 A[5][2]的起始地址相同。

（2）稀疏矩阵的压缩存储有三元组表和十字链表两种存储方法。

5. 什么是稀疏矩阵的三元组表存储？试写出以下矩阵 A 的三元组顺序表。

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 2 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \end{pmatrix}$$

答：存储稀疏矩阵非零元素的值、所在行和列，以及稀疏矩阵非零元素的个数、行数、列数的顺序存储叫稀疏矩阵的三元组表存储。

矩阵 A 的三元组顺序表为：

A=((4,6,5), (1,3,1), (1,5,2), (2,1,3), (3,4,4), (4,2,5))

（用 A[0]存储稀疏矩阵行数、列数、非零元个数）

6. 设 A 是一个具有 m 行 n 列的元素的二维数组，每个元素占用 s 个存储单元， Loc(aij) 为元素 aij 的存储地址，Loc(a00) 是 a00 存储位置, 也是二维数组 A 的基址。若以行序为主序的方式存储二维数组，则元素 aij 的存储位置是什么？若以列序为主序的方式存储二维数组，则元素 aij 的存储位置又是什么？

答：若以行序为主序的方式存储二维数组，则元素 aij 的存储位置为：

$$Loc(aij) = Loc(a00) + (n \times i + j) \times s$$

若以列序为主序的方式存储二维数组，则元素 aij 的存储位置为：

$$Loc(aij) = Loc(a00) + (m \times j + i) \times s$$

7. 什么是字符串？它和线性表有什么联系和区别？

答：字符串（简称串）是一种特殊的线性表，它的数据元素仅由一个字符组成。

字符串的逻辑结构、存储结构和线性表一样，除第一个元素外，其它每一个元素有一个且仅有一个直接前驱，除最后一个元素外，其它每一个元素有一个且仅有一个直接后继，它通常采用的存储方式也是顺序存储和链式存储。字符串和线性表的区别仅在于，线性表中的数据元素可以是任意数据，而字符串中的数据元素只能是一个字符。

8. 什么是三角矩阵？如何将三角矩阵存储到一个一维数组中？

答：以主对角线划分，三角矩阵有上三角和下三角两种。上三角矩阵如图所示，它的下三角（不包括主对角线）中的元素均为常数。下三角矩阵正好相反，它的主对角线上方均为常数，如图所示。在大多数情况下，三角矩阵常数为零。

$$\begin{pmatrix} a_{00} & a_{01} & \cdots & a_{0n-1} \\ c & a_{11} & \cdots & a_{1n-1} \\ & \cdots\cdots\cdots & & \\ c & \cdots & c & a_{n-1n-1} \end{pmatrix} \qquad \begin{pmatrix} a_{00} & c & \cdots & c \\ a_{10} & a_{11} & \cdots & c \\ & \cdots\cdots\cdots & & \\ a_{n-10} & a_{n-11} & \cdots & a_{n-1n-1} \end{pmatrix}$$

(a)上三角矩阵　　(b)下三角矩阵

三角矩阵中的重复元素 c 可共享一个存储空间，其余的元素正好有 n(n+1)/2 个，因此，三角矩阵可压缩存储到一维数组 sa[0..n(n+1)/2]中，其中 c 存放在数组的最后一个分量中。

上三角矩阵中，主对角线之上的第 p 行(0≦p<n)恰有 n-p 个元素，按行优先顺序存放上三角矩阵中的元素 a_{ij} 时，a_{ij} 之前的 i 行一共有

$$\sum_{p=0}^{i-1}(n-p) = i(2n-i+1)/2$$

个元素，在第 i 行上，a_{ij} 前恰好有 j-i 个元素：$a_{ii}, a_{ii+1}, \ldots, a_{ij-1}$。因此，sa[k]和 a_{ij} 的对应关系是：

当 i≦j　　k= i(2n-i+1)/2+j-i

当 i>j　　k=n(n+1)/2

下三角矩阵的存储和对称矩阵类似，sa[k]和 a_{ij} 对应关系是：

当 i≧j　　k=i(i+1)/2+j

当 i<j　　k=n(n+1)/2

第六章　二叉树

一.填空题

1. 具有 256 个结点的完全二叉树的深度为 **8** 。 **应为 9**

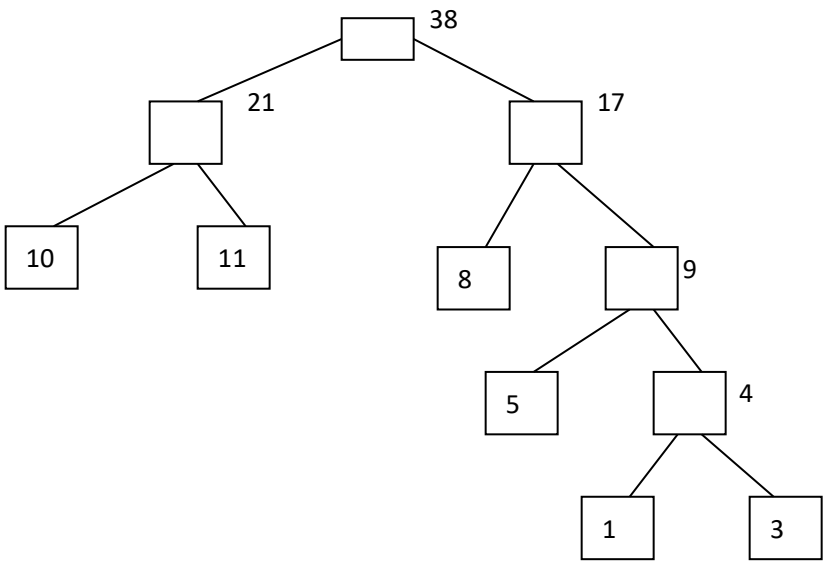
2. 一棵深度为 6 的满二叉树中叶子结点数为 **32** 个。

3. 已知完全二叉树的第 8 层有 8 个结点，则其叶子结点数是 **68** 。
4. 设根结点的层数为 1，则深度为 k 的二叉树最多具有 2^k-1 个结点。
5. 有 3 个结点的二叉树共有 **5** 种形态。
6. 如果要采用顺序存储结构存储二叉树，该二叉树必须是 **完全** 二叉树。
7. 深度为 k 的完全二叉树结点总数最多为 2^k-1 。
8. 深度为 k 的完全二叉树所含叶子结点的个数最多为 2^{k-1} 。
9. n 个权构成一棵 Huffman 树，其结点总数为 **2n-1** 。
10. 具有 n 个结点的二叉树用二叉链表存放，共有 **n+1** 个空指针域。
11. 在一棵二叉树中，度为零的结点的个数为 N0,度为 2 的结点的个数为 N2,则有 **N0 = N2+1**。
12. 高度为 8 的完全二叉树至少有 **64** 个叶子结点。
13. 已知二叉树有 50 个叶子结点，则该二叉树的总结点数至少是 **99** 。
14. 在二叉树中，指针 p 所指结点为叶子结点的条件是 **(p->lchild==NULL)&&(p->rchild==NULL)** 。
15. 二叉树的第 i 层上最多含有结点数为 2^{i-1} 。

二.简答题

1. 设给定一个权值集合 W=(1，3，5，8，10，11)，要求根据给定的权值集合构造一棵哈夫曼树并计算哈夫曼树的带权路径长度 WPL。

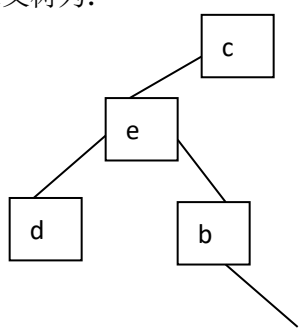
答：哈夫曼树为：



该哈夫曼树的带权路径长度 WPL=1×4+3×4+5×3+8×2+10×2+11×2=89

2. 已知一棵二叉树的先序遍历为：cedba，中序遍历为：debac，请画出该二叉树，并写出后序遍历结果。

答：该二叉树为：



后序遍历结果为：dabec

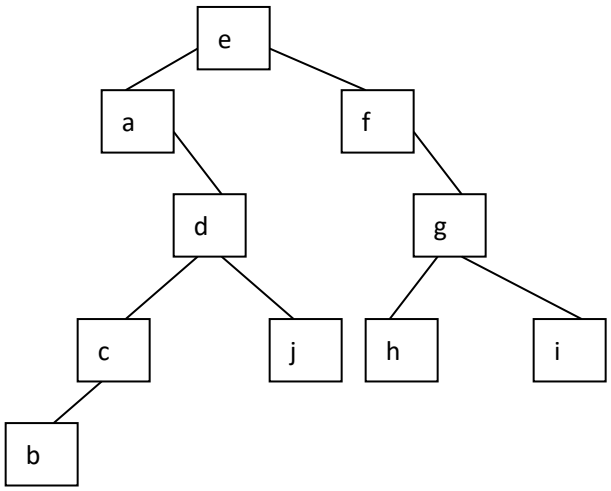
3. 二叉树结点数值采用顺序存储结构，如下图所示：

a. 画出该二叉树。

b. 写出前序遍历、中序遍历和后序遍历的结果。

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
e	a	f		d		g			c	j			h	I					b

答：a. 该二叉树为



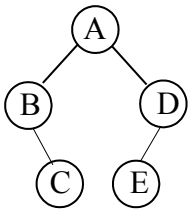
b. 前序遍历为：eadcbjfghi

中序遍历为：abcdjefhgi

后序遍历为：bcjdahigfe

4. 已知一棵二叉树的中序遍历序列为 BCAED、后序遍历序列为 CBEDA，画出这棵二叉树并写出其先序遍历的序列。

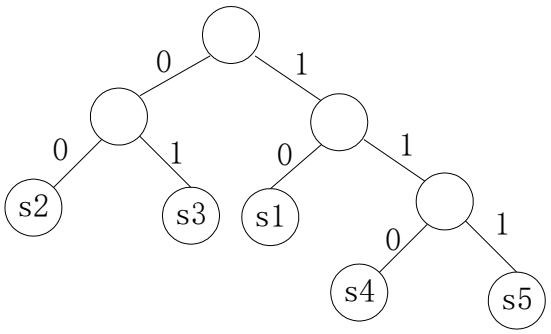
答：该二叉树为：



其先序遍历的序列为：ABCDE

5. 假设用于通讯的电文由 5 个信号{S1，S2，S3，S4，S5}组成，各信号在电文中出现的频率分别为{0.25，0.22，0.20，0.18，0.15}，请画出其对应的哈夫曼树，并给出各信号对应的哈夫曼编码。

答：其对应的哈夫曼树为：



其哈夫曼编码为：

s1 : 10

s2 : 00

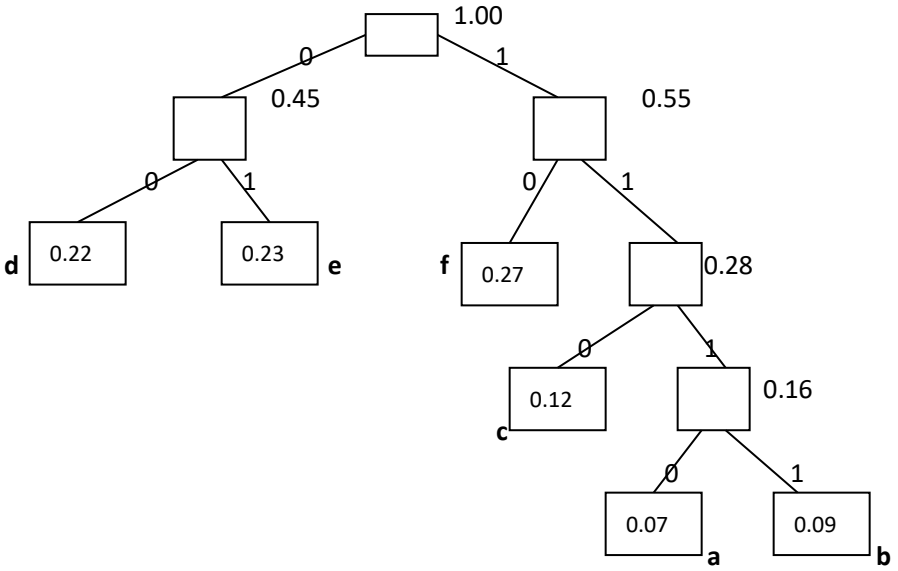
S3 : 01

S4 : 110

S5 : 111

6. 假设字符 a、b、c、d、e、f 的使用频度分别是 0.07,0.09,0.12,0.22,0.23,0.27，写出 a、b、c、d、e、f 的 Huffman（哈夫曼）编码。要求画出哈夫曼树。

答：



这六个字母的哈夫曼编码为：

a: 1110

b: 1111

c: 110

d: 00

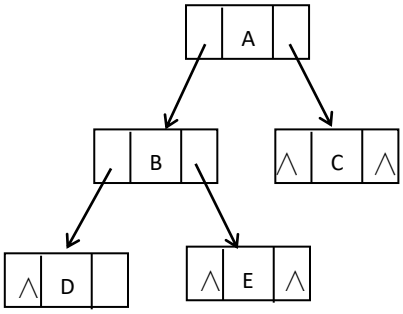
e: 01

f: 10

7. 设完全二叉树的顺序存储结构中依次存储数据 ABCDE，要求给出该二叉树的链式存储结

构并给出该二叉树的前序、中序遍历序列。

答：二叉树的链式存储结构为:



前序遍历序列为：ABDEC

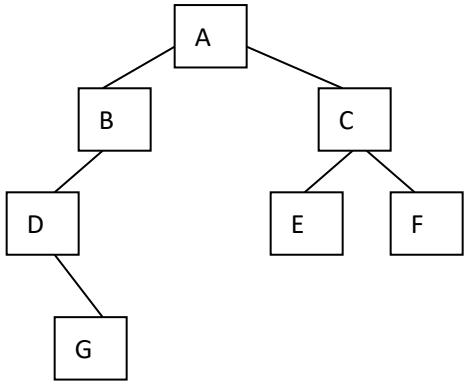
中序遍历序列为：DBEAC

8. 某二叉树中序遍历的结果是 DGBAECF，后序遍历的结果是 GDBEFCA。

（1）画出此二叉树；

（2）写出其层次遍历的结果。

答：该二叉树为：



层次遍历结果为：ABCDEFG

三.算法设计题

1. 假设二叉树的结点类型定义如下：

```
typedef struct node
{
int    data;
struct node *lchild,*rchild;
}Bstree;
```

设计一算法，统计二叉树中度为 2 的结点总数。

2. 假设二叉树的结点类型定义如下：

```
typedef struct node
{
int    data;
struct node *lchild,*rchild;
```

}Bstree;

设计一算法，查找元素 x 所在结点，若找到，返回该结点的指针，否则返回一个空指针。

3. 假设二叉树的结点类型定义如下：

```
typedef struct node
{
int    data;
struct node *lchild,*rchild;
}Bstree;
```

设计一算法，统计叶子结点总数。

4. 假设二叉树的结点类型定义如下：

```
typedef struct node
{
int    data;
struct node *lchild,*rchild;
}Bstree;
```

设计一算法，求二叉树的深度。

解答:

1. 设计一算法，统计二叉树中度为 2 的结点总数。

```
int count(Bstree* bt)
      {   if (bt==NULL) return 0;
          if(bt->lchild!=NULL && bt->rchild!=NULL)    return 1;
          else    return(count(bt->lchild)+count(bt->rchild));
      }
```

不对，应为：

```
int n=0;
void count(BiTree bt )
     { if(bt!=NULL)
        {if(bt->lchild!=NULL&&bt->rchild!=NULL)
                    n=n+1;
              count(bt->lchild);
              count(bt->rchild);   }
      }
```

或

```
int count(BiTree bt)
{     if (bt==NULL) return 0;
     if ((bt->lchild==NULL)&&(bt->rchild==NULL))      return 0;
     if ((bt->lchild!=NULL)&&(bt->rchild==NULL))      return count(bt->lchild);
     if ((bt->lchild==NULL)&&(bt->rchild!=NULL))      return count(bt->rchild);
     if ((bt->lchild!=NULL)&&(bt->rchild!=NULL))      return
count(bt->rchild)+count(bt->lchild)+1;
}
```

2. 设计一算法，查找元素 x 所在结点，若找到，返回该结点的指针，否则返回一个空指针。

```
BiTree Search(Bstree* bt, datatype x)
{ Bstree * p=NULL;
   if ( bt )
      { if ( bt->data == x) return bt;
         if ( bt->Lchild != NULL)
            { p=Search(bt->lchild,x);
               if(p) return p; }
         if ( bt->rchild != NULL)
            { p=Search(bt->rchild,x);
               if(p) return p; }
       }
   return NULL;
}
```

3. 设计一算法，统计叶子结点总数。

```
int countleaf1(Bstree* bt)
      {   if (bt==NULL) return 0;
          if(bt->lchild==NULL && bt->rchild==NULL)     return 1;
          else    return(countleaf1(bt->lchild)+countleaf1(bt->rchild));
}
```

4. 设计一算法，求二叉树的深度。

```
int DeepBiTree(Bstree *T)                          //求二叉树的深度
{
        int ldeep,rdeep;
   if(!T)
     return 0;                                     //空二叉子树深度为 0
else
{
     ldeep = DeepBiTree(T->lchild);
     rdeep = DeepBiTree(T->rchild);
}
if(ldeep > rdeep)                                  //ldeep 就是每个“二叉树”的左子树深度。
     return ldeep + 1;                             //rdeep 就是每个“二叉树”的右子树深度。
else
     return rdeep + 1; }
```

第七章　树

一.填空题

1. 用孩子兄弟法存储树时，树的先序遍历结果与其对应的二叉树的 **先序** 遍历结果相同。
2. 由树转换成的二叉树，其根结点的右子树总是 **空** 。

3. 树的后序遍历结果与其对应的二叉树的 **中序** 遍历结果相同。
4. 设结点 A 有 3 个兄弟结点且结点 B 为结点 A 的双亲结点，则结点 B 的度数为 **4** 。
5. 树是由 n(n>=0)个结点构成的有限集合。当 n=0 时称为 **空树** 。
6. 树的度是树中所有结点的度的 **最大值** 。
7. 设森林 T 中有三棵树，第一，二，三棵树的结点个数分别为 n1,n2,n3,将森林转换成二叉树后，其根结点的左子树上有 **n1-1** 个结点。
8. 树的根结点无前趋，其它结点 **有且仅有一个** 前趋。
9. 树形结构与线性结构的区别是 **树形结构有分支而线性结构无分支** 。
10. 零棵或有限棵不相交的树的集合称为 **森林** 。
11. 可以根据根结点有无右分支，将一棵二叉树转换为树或 **森林** 。
12. 树转换成的二叉树，二叉树的根结点 **无右子树** 。
13. 树和森林的先根遍历与所转换成的二叉树的先序遍历结果 **一样** 。
14. 树和森林的后根遍历与所转换成的二叉树的中序遍历结果 **一样** 。
15. 森林转换后的二叉树，二叉树的根结点 **一定有右子树** 。

二.简答题

1. 一棵度为 2 的树与一棵二叉树的区别是什么？

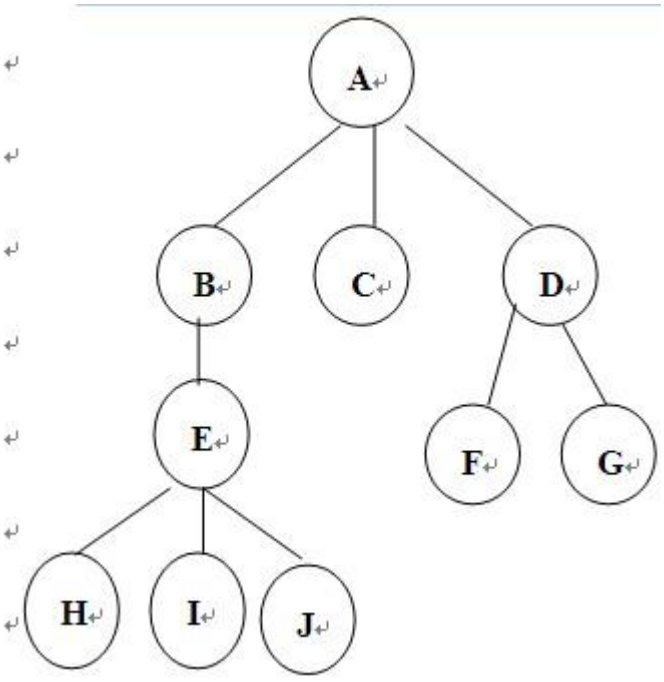
答：一棵度为 2 的树与一棵二叉树的区别在于: 度为 2 的树有两个分支，没有左右之分；一棵二叉树也有两个分支，但有左右之分，左右不能交换。

树与二叉树的区别：

（1）二叉树的一个结点至多有两个子树，树形则不然。

（2）树中的结点只有一个子树时，无须区分其是左子树还是右子树；而二叉树中的结点只有一个子树时，必需确定其是左子树还是右子树。

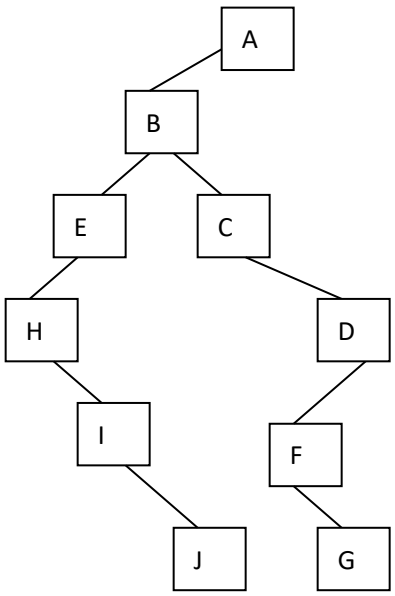
2. 写出下面树的各种遍历结果，并将其转换为对应的二叉树。



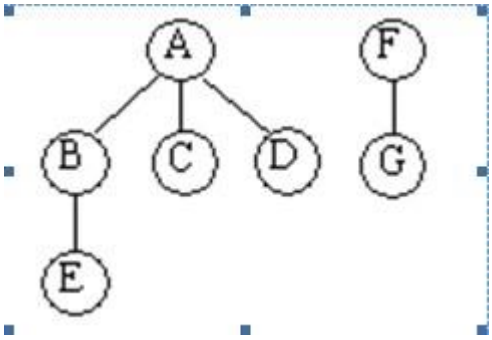
答：先序遍历结果为：ABEHIJCDFG

后序遍历结果为：HIJEBCFGDA

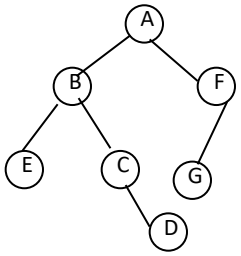
对应的二叉树为：



3. 将下列森林转化为二叉树，并写出对该森林进行的各种遍历序列.

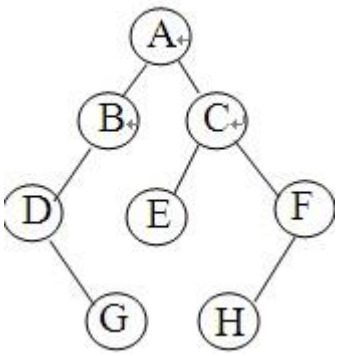


答：其二叉树为：

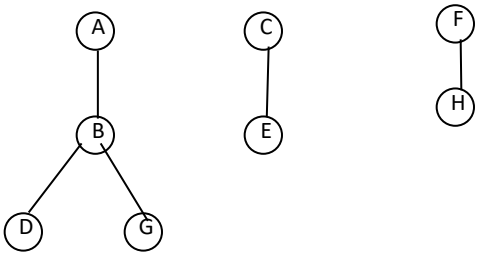


森林的先序遍历序列为：ABECDFG
森林的后序遍历序列为：EBCDAGF

4. 设下列二叉树是某森林对应的二叉树，请画出对应的森林，写出森林的先序和后序遍历。

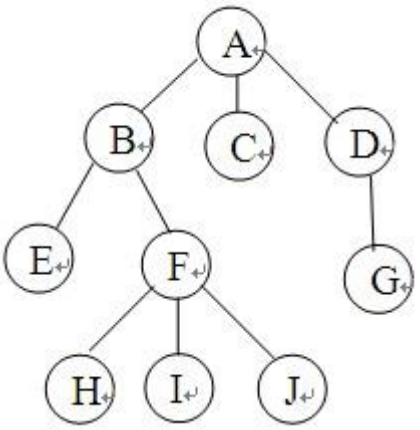


答：对应的森林为：

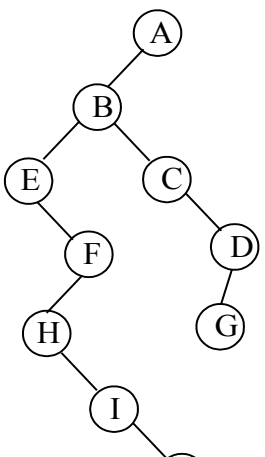


森林的先序遍历序列为：ABDGCEFH
森林的后序遍历序列为：DGBAECHF

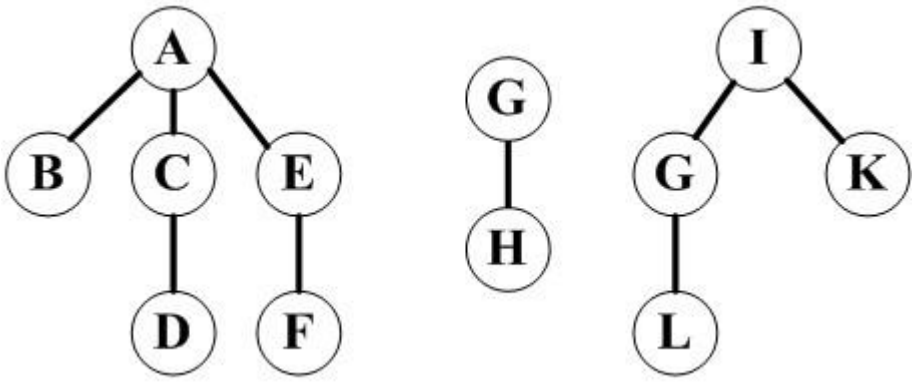
5. 写出下图所示的树的先序和后序遍历序列，并将其转换成对应的二叉树。



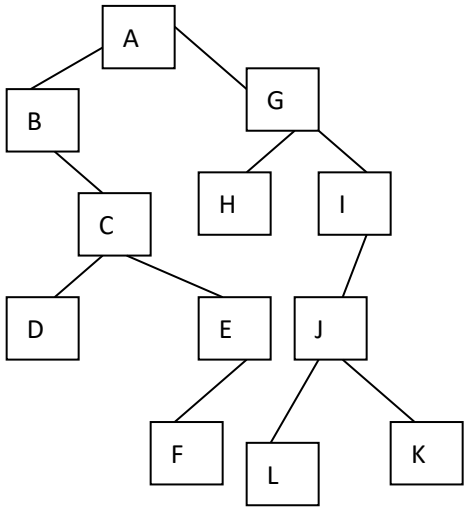
答：先序遍历序列为：ABEFHIJCDG
后序遍历序列为：EHIJFBCGDA



6. 将如下图所示的森林转换成对应的二叉树，并写出二叉树的先序、中序和后序遍历序列。



答：对应的二叉树为：



其二叉树的先序遍历序列为：ABCDEFGHIJLK
中序遍历序列为：BDCFEAHGLJKI
后序遍历序列为：DFECBHLKJIGA

7. 树都有哪些存储方法？请分别描述。

答：树有双亲数组存储方法、指向孩子的链表存储方法、双亲孩子链表存储方法、孩子兄弟链表存储方法等。详见书上 P131- P134。

8. 树都有哪些表示方法？请分别描述。

答：树有直观表示法、嵌套集合表示法、凹入表示法（类似书的目录）、广义表示法四种表示方法。详见书上 P130。

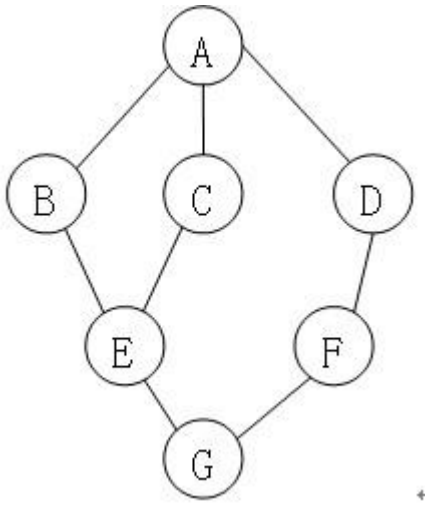
第八章　图基础

一.填空题

1. 无向图的邻接矩阵是 **对称** 矩阵。
2. 图的存储结构有邻接矩阵和 **邻接链表** 两种。
3. 在一个具有 n 个顶点的有向图中，最多可以有 **n(n-1)** 条边。
4. 具有 N 个顶点的无向图中，边至少有 **0** 条。
5. 在一个具有 n 个顶点的有向图中，最多可以有 **n(n-1)** 条边。
6. 依据有向图的邻接矩阵，顶点 i 的入度为 **邻接矩阵第 i 列非零元素(或非∞) 的个数** 。
7. 有 10 个顶点的无向图至少有 **9** 条边才能确保其是连通的。
8. 图形结构与树形结构的区别是 **图形结构有回路而树形结构无回路** 。
9. 图是由顶点和 **边（弧）**构成。
10. 图的深度优先搜索遍历要用 **栈** 实现。
11. 图可分为分有向图、无向图、有向网图和 **无向网图** 四种。
12. 图的广度优先搜索遍历类似于树的 **层次遍历** 。
13. 所谓图的邻接矩阵表示，就是用一维数组存储图中顶点的信息，用称作邻接矩阵的矩阵表示图中 **各顶点之间的邻接关系** 。
14. 图的广度优先搜索遍历要用 **队列** 实现。
15. 图的深度优先搜索遍历类似于树的 **先序遍历** 。

二.简答题

1. 画出下图邻接矩阵存储结构，并根据邻接矩阵写出从顶点 A 开始分别进行深度优先搜索遍历和广度优先搜索遍历的结果。

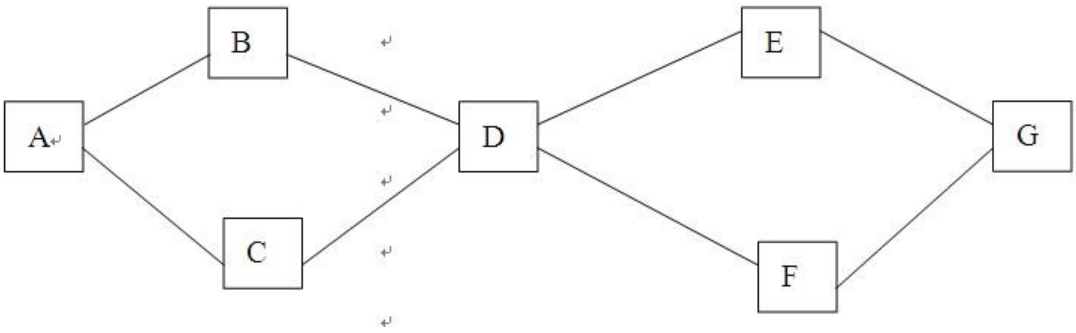


答：

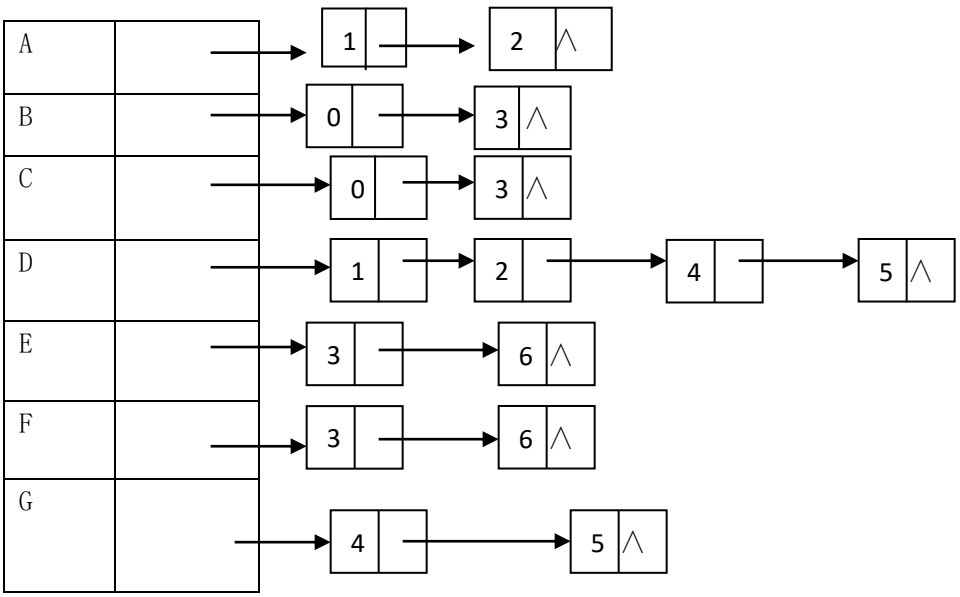
$$\begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

深度优先搜索遍历结果为：ABECGFD
广度优先搜索遍历结果为：ABCDEFG

2. 画出下图的邻接链表存储结构，并写出根据该邻接链表从顶点 A 开始进行深度优先搜索和广度优先搜索遍历的结果。

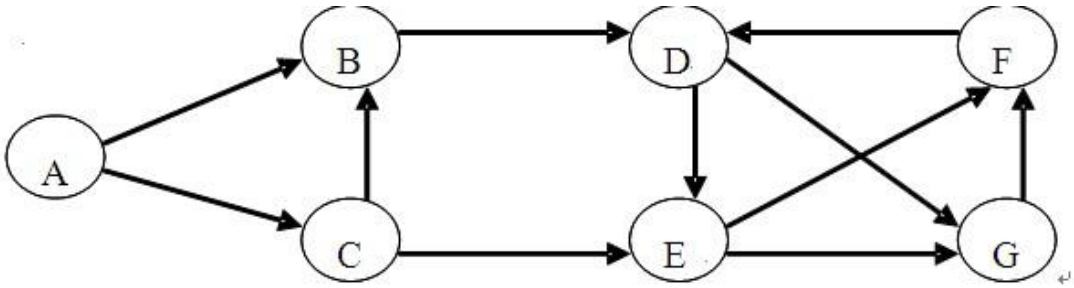


答：其邻接链表存储结构为：

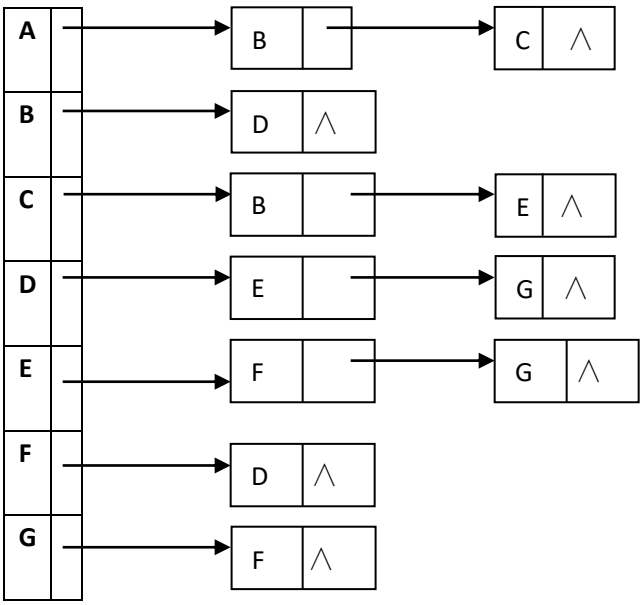


深度遍历结果为：ABDCEGF
广度遍历结果为：ABCDEFG

3. 给出下图的邻接链表，并根据邻接链表分别写出从顶点 A 出发遍历该图的深度优先搜索和广度优先搜索序列。



答：该图邻接链表为：



深度优先搜索遍历序列：ABDEFGC

广度优先搜索遍历序列：ABCDEGF

4. 已知某图的邻接矩阵存储如下图所示。请根据邻接矩阵

(1) 画出该图；

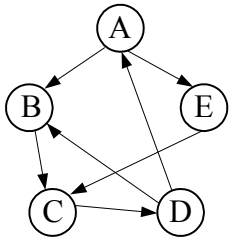
(2) 写出从顶点 A 出发深度优先遍历该图的序列。

A	B	C	D	E

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

答：深度优先遍历该图的序列为：A、B、C、D、E/ A、E、C、D、B

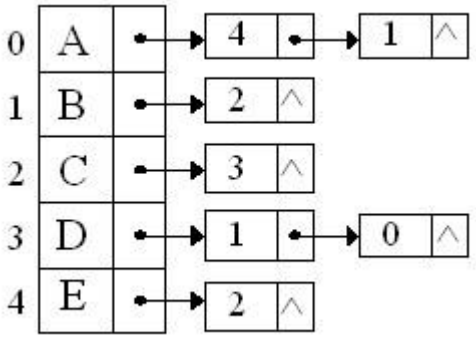
其逻辑关系图为：



5. 已知某图的邻接链表如下图所示。请根据邻接链表

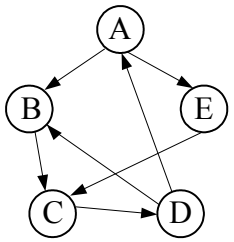
(1) 写出从顶点 A 出发深度优先遍历该图的序列；

(2) 画出该图。

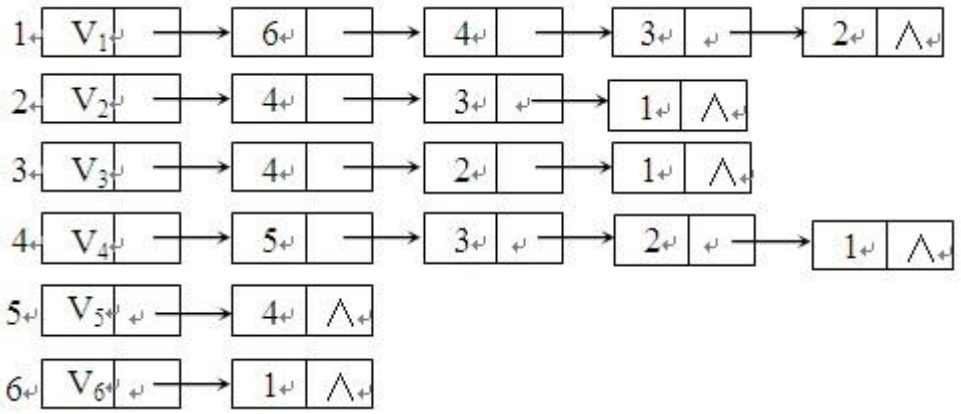


答：深度优先遍历该图的序列为：A、E、C、D、B

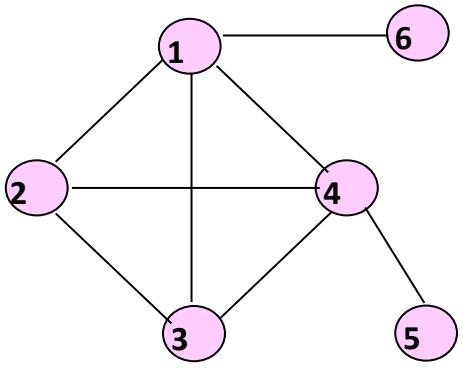
其逻辑关系图为：



6. 已知一个图的邻接链表如下所示，请画出该图并写出其深度优先和广度优先遍历的序列



答：

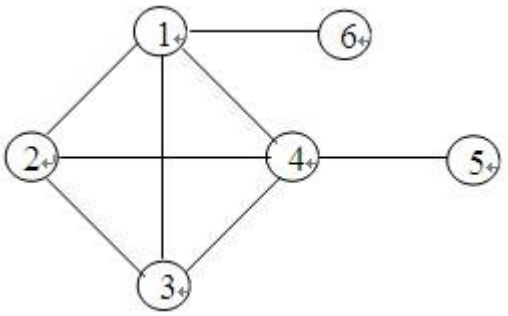


深度：V1,V6,V4,V5,V3,V2
广度：V1,V6,V4,V3,V2,V5

7. 已知一个图的邻接矩阵如下所示，顶点用 V1,V2,V3,V4,V5,V6 表示，请画出该图并写出其深度优先和广度优先遍历的序列。

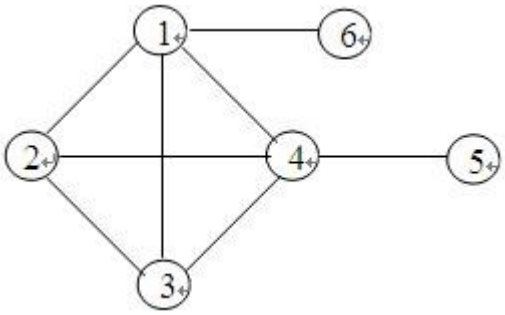
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

答:



深度优先遍历的序列: V1,V2,V3,V4,V5,V6
广度优先遍历的序列: V1,V2,V3, V4, V6,V5

8. 设无向图 G 如下图所示，要求给出该图的邻接矩阵，并根据邻接矩阵写出其深度优先和广度优先遍历的序列。



答: 该图的邻接矩阵为:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

深度优先遍历的序列为: V1,V2,V3,V4,V5,V6
广度优先遍历的序列为: V1,V2,V3, V4, V6,V5

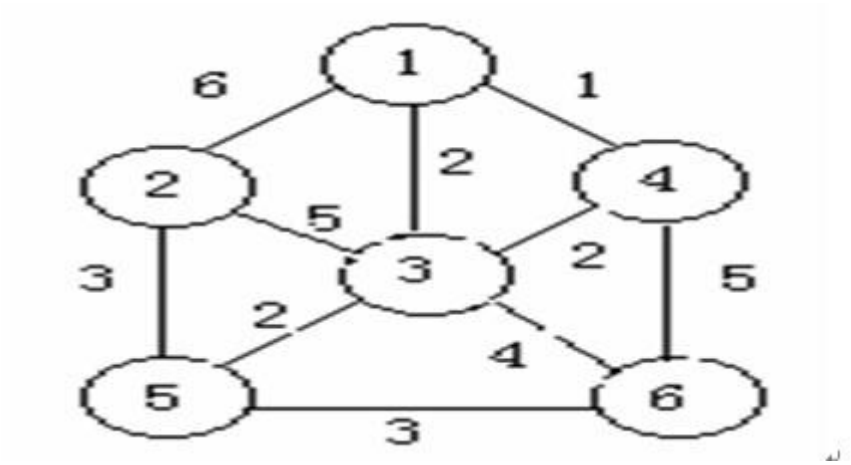
第九章　图应用

一.填空题

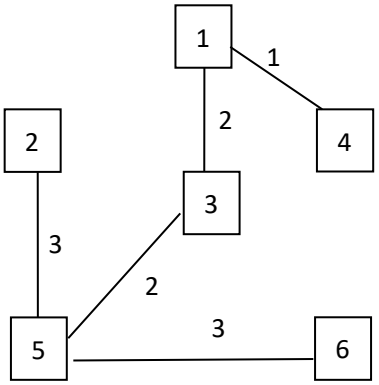
1. 普里姆（Prim）算法适合于求 **边稠密的网** 的最小生成树。
2. 迪杰斯特拉(Dijkstra)算法是按 **路经长度递增的次序** 产生最短路径。
3. 如果含 n 个顶点的图形形成一个环，则它有 **n** 棵生成树。
4. 普里姆（Prim）算法是用来求 **最小生成树** 。
5. 若连通图的顶点个数为 n，则该图的生成树的边数为 **n-1** 。
6. 设一个连通图 G 中有 n 个顶点 e 条边，则其最小生成树上有 **n-1** 条边。
7. 克鲁斯卡尔（Kruskal）算法适合于求 **边稀少的网** 的最小生成树。
8. 包含无向图 G 所有顶点的的极小连通子图称为 G 的生成树。
9. 把所有 n-1 条边的权值之和最小的生成树叫 **最小生成树** 。
10. Prim 算法开始只有一个顶点，而 kruskal 算法开始包含 **所有顶点** 。
11. 弗洛伊德(Floyd)算法是用来求 **任意两点之间的最短路径** 。
12. 从一个源点到其它各顶点的最短路径叫 **单源最短路经** 。
13. 克鲁斯卡尔(Kruskal)算法是一种按照 **网中边的权值递增的顺序** 构造最小生成树的方法。
14. 所生成的当前最小生成树中的顶点， Prim 算法连通，而 kruskal 算法 **不一定** 连通。
15. 迪杰斯特拉(Dijkstra)算法是用来求 **单源最短路经** 。

二.简答题

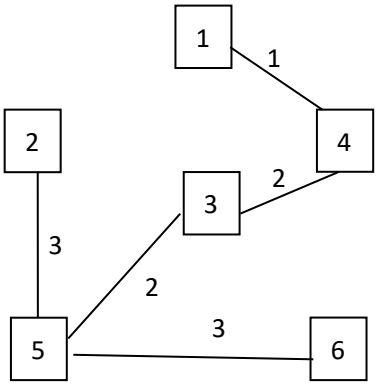
1. 试用 Prim 方法画出下图从顶点 1 开始的最小生成树。



答：**其最小生成树为：**

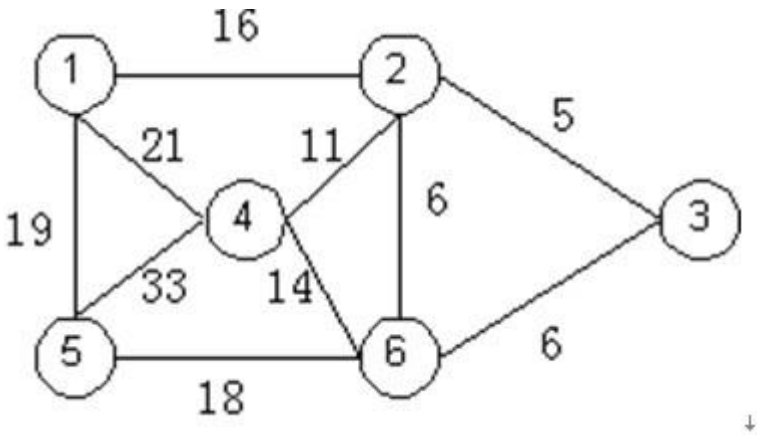


或：

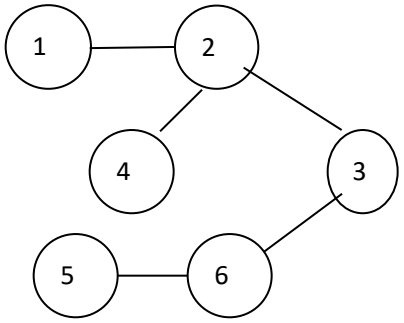


(不唯一)

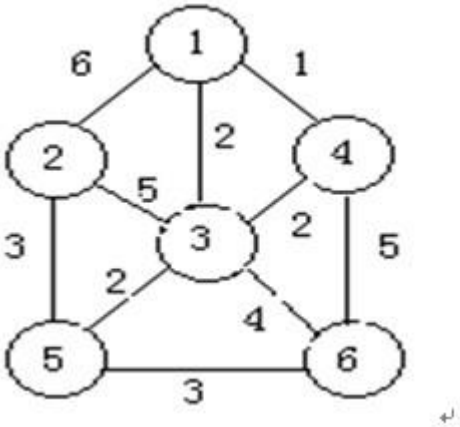
2. 试用普里姆（Prim）方法画出下图从顶点 1 开始的最小生成树



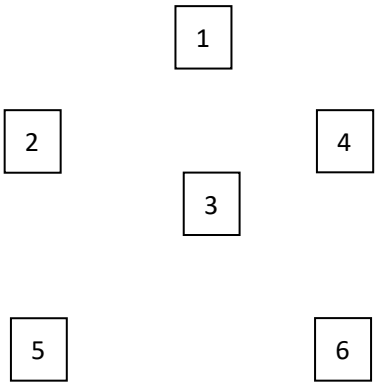
答: 从顶点 1 开始的最小生成树为:

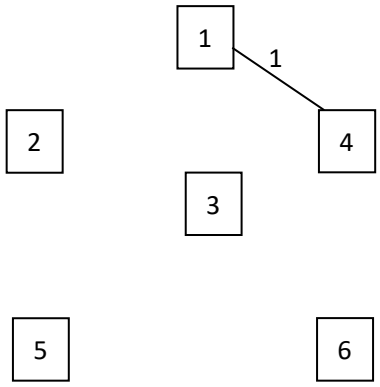


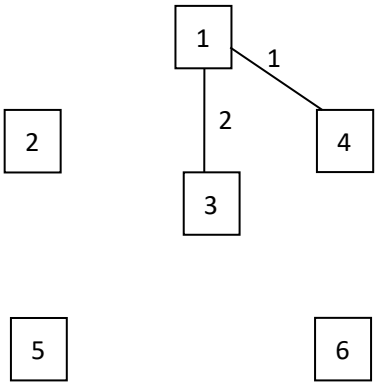
3. 设无向图 G 如下图所示，要求用 kruskal 算法画出该图的该图的最小生成树。

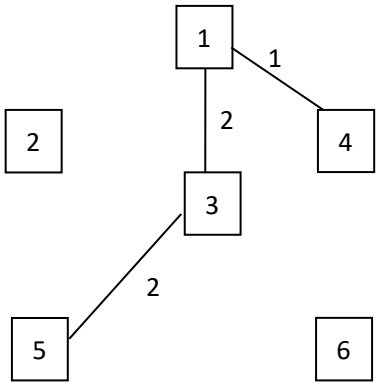


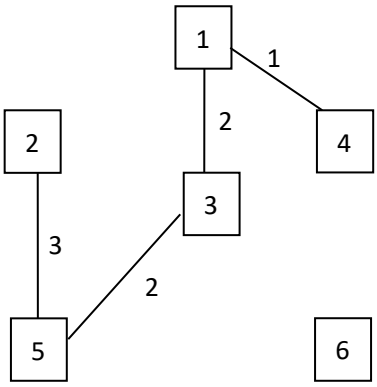
答：

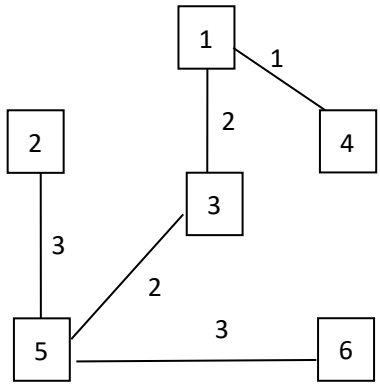




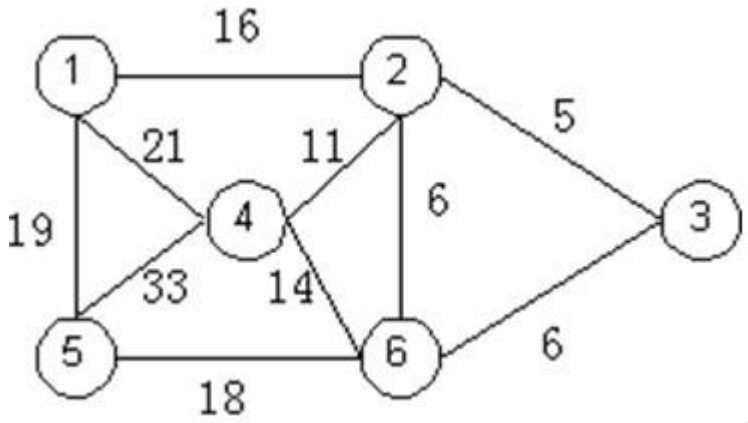




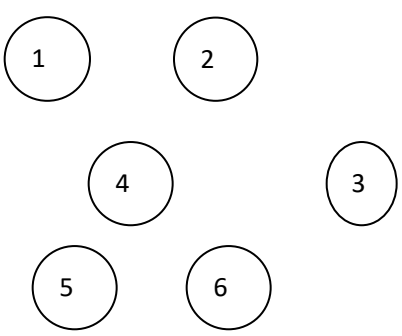


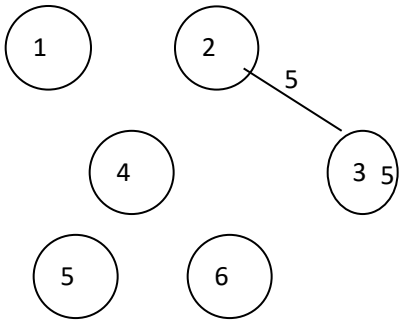


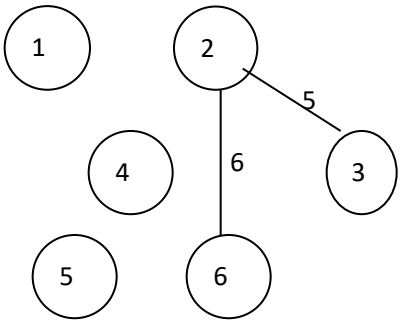
4. 设无向图 G 如下图所示，要求用 kruskal 算法画出该图的该图的最小生成树。

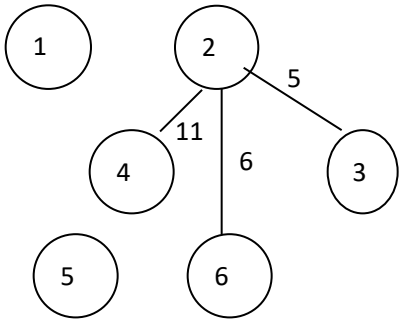


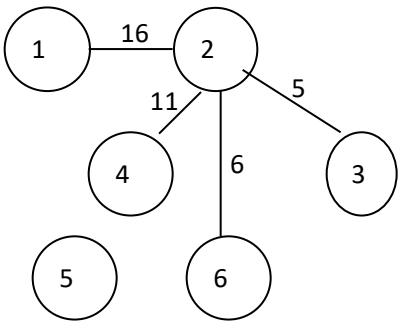
答：

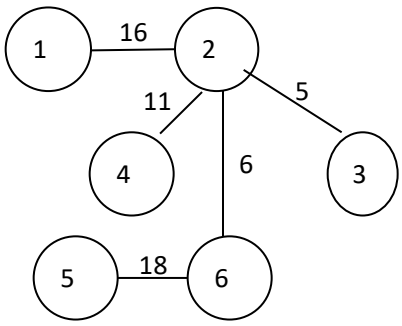




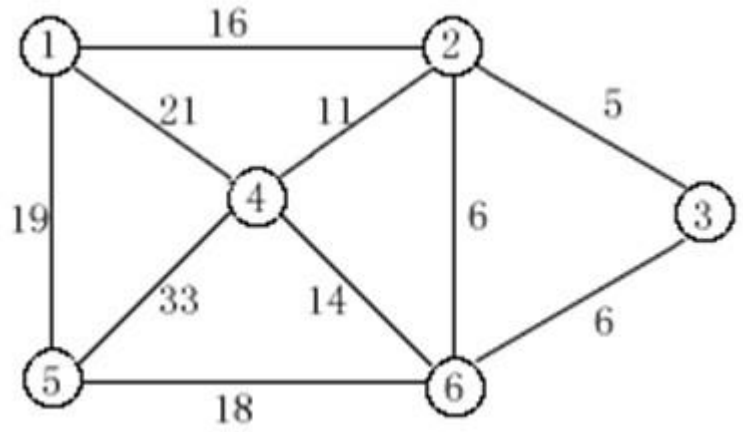




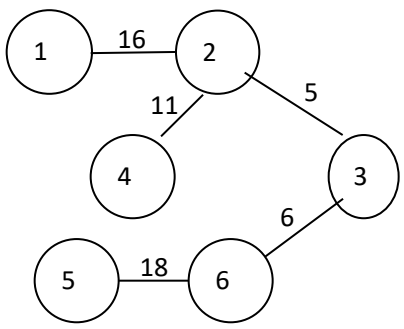


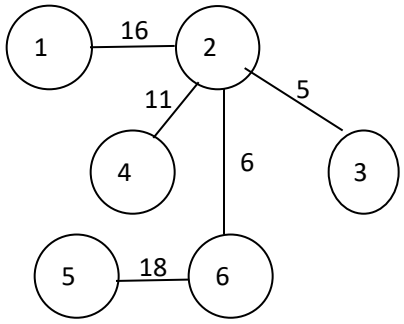


5. 下图表示一个地区的通讯网，边表示城市间的通讯线路，边上的权表示架设线路花费的代价，如何选择能沟通每个城市且总代价最省的 n-1 条线路，画出所有可能的选择



答：有两种：





6. 对下图所示的有向网，执行迪杰斯特拉（Dijkstra）算法可得到顶点 A 到其余各顶点的最短距离及最短路径，试在表 1 的（1）、（2）、（3）、（4）处填上适当的内容，以说明该算法的执行过程

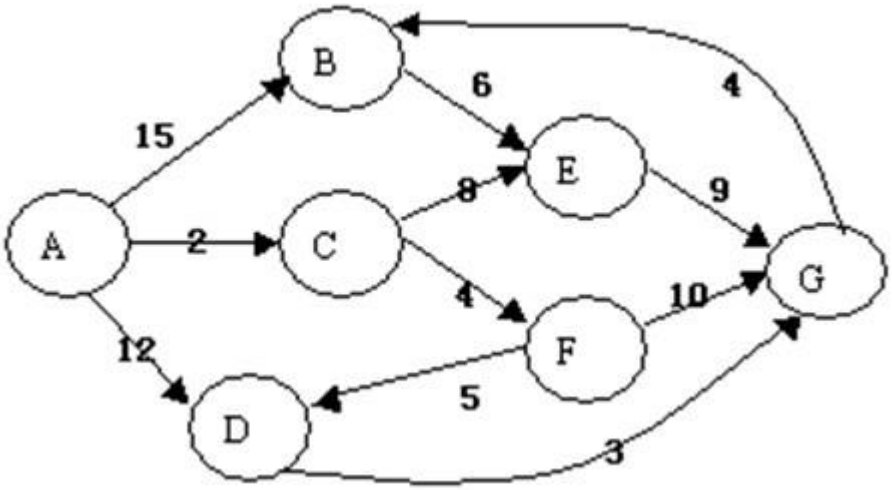


表 1　从顶点 A 到其它各顶点的最短距离及最短路径的迪杰斯特拉算法求解过程

终点 / 距离及路径	B	C	D	E	F	G	S{终点集}
K=1	15 (A, B)	(1)	12 (A, D)				{A，C}
K=2	15 (A, B)		12 (A, D)	(2)	6 (A, C, F)		{A, C, F}
K=3	15 (A, B)		11 (A, C, F, D)	10 (A, C, E)		16 (A, C, F, G)	{A, C, F, E}
K=4	15 (A, B)		11 (A, C, F, D)			16 (A, C, F, G)	(3)
K=5	15 (A, B)					(4)	{A, C, F, E, D, G}
K=6	15 (A, B)						{A, C, F, E, D, G, B}

答：

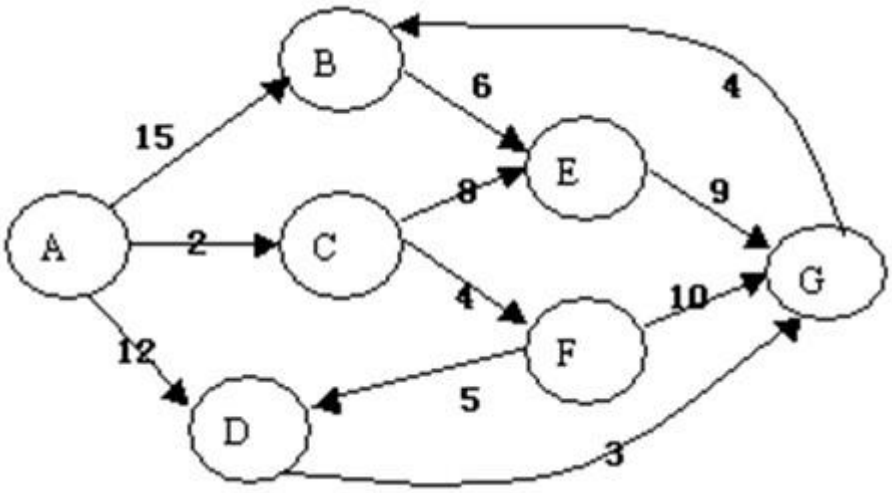
(1) 2(A, C)

(2) 10(A, C, E)

(3) {A, C, F, E, D}

(4) 14(A, C, F, D, G)

7. 试用迪杰斯特拉（Dijkstra）算法写出下图从顶点 A 到其它顶点的最短路径。



答：(A, C)，2

(A, C, F)，6

(A, C, E)，10

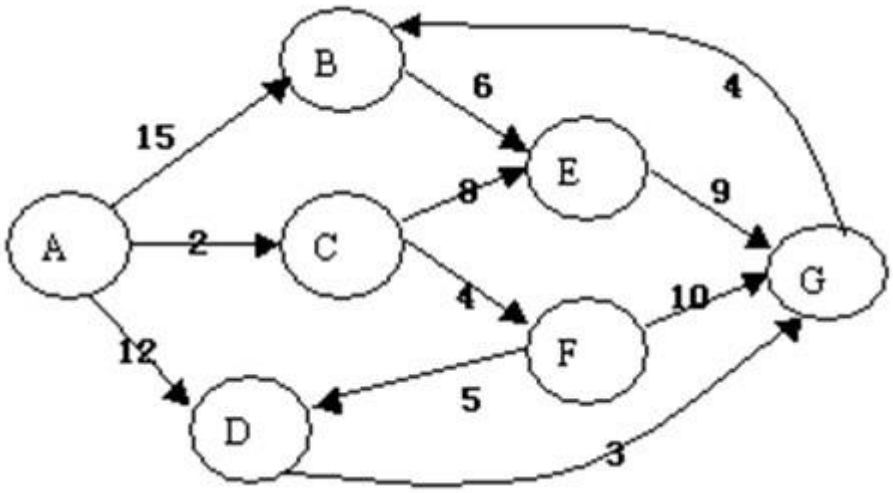
(A, C, F，D)，11

(A,C,F,D,G)，14

(A,B)，15

8. 试用弗洛伊德(Floyd)算法写出下图从顶点 A 到其它顶点的最短路径

（该题有问题，有点大材小用，求 A 到其它顶点的最短路径用迪杰斯特拉（Dijkstra）算法足够，没必要用弗洛伊德(Floyd)算法，考试不会出这道题）



答：

(A,C)，2

(A,C,F)，6

(A,C,E)，10

(A,C,F，D)，11

(A,C,F,D,G)，14

(A,B)，15

第十章　排序

一.填空题

1. 稳定的排序方法是指在排序中，关键字值相等的不同记录间的前后相对位置 **不变** 。

2. 希尔排序法、快速排序法、堆排序法和二路归并排序法四种排序法中，要求辅助空间最多的是 **二路归并排序** 。

3. 分别采用堆、快速、插入、归并排序法对初始状态为递增序列的表按递增顺序排序，最费时的是 **快速** 算法。

4. 在直接插入、冒泡、快速排序和简单选择排序方法中，平均时间复杂度最低的排序方法是 **快速排序** 。

5. 快速排序的平均时间复杂度为 **O(nlog2n)** 。

6. 具有 12 个记录的序列，采用冒泡排序最少的比较次数是 **11** 。

7. 设有 1000 个无序元素，希望用较快速度挑选出其中前 10 个最大元素，在快速排序，堆

排序，基数排序，直接插入，归并排序这些方法中，采用 **堆排序** 方法最好。

8. 外排序的基本方法是 **归并** 。

9. 快速排序的最坏情况，其待排序的初始排列是 **已经按其排序码从小到大排好序** 。

10. 分别采用堆、快速、插入、归并排序法对初始状态为递增序列的表按递增顺序排序，最费时的是 **快速** 算法。

11. 对 n 个元素按某关键字用堆排序方法由小到大排序，应使用 **大** 顶堆。

12. 内排序指待排序列 **在内存中** 所进行的排序。

13. 外排序指排序过程中需访问 **外存** 的排序。

14. 在直接插入、冒泡、快速排序和简单选择排序方法中，具有稳定性的排序方法有 **直接插入、冒泡** 。

15. 在冒泡、快速、直接插入三种排序方法中，排序的趟数与数据表的初始排列顺序无关的是 **直接插入** 排序方法。

16. 当数据表初态基本有序的情况下，在冒泡、快速和简单选择排序方法中应选择 **冒泡** 排序方法，从而使得排序的趟数最少。

二.简答题

1. 对于一组给定的关键字{49，38，27，15，94，53}，写出以第一个关键字作为基准元素运用快速排序方法进行升序排序时第一次划分的结果，如果要实现排序，共需进行几次划分

答：第一次划分的结果为：[15 38 27] 49 [94 53]

要实现排序，共需进行 4 次划分。

2. 什么是排序算法的稳定性？举出一个稳定的排序算法和不稳定的排序算法的例子。

答：排序算法的稳定性是指待排序范围中多个关键字值相同的记录，使用某种排序算法进行排序后其相对次序与排序前相比没有改变。冒泡排序算法只进行元素间的顺序移动，所以是一个稳定的排序方法，而快速排序就是一种不稳定的排序方法（可用 3,2,2′ 序列来验证）。

3. 对于一组给定的关键字{49，38，27，15，94，53，81}，写出用冒泡排序法进行升序排列的各趟结果。

答：

初始状态	49	38	27	15	94	53	81
第一趟	38	27	15	49	53	81	94
第二趟	27	15	38	49	53	81	94
第三趟	15	27	38	49	53	81	94
第四趟	15	27	38	49	53	81	94

4. 设一组初始记录关键字序列为(19，21，16，5，18，23)，要求给出以 19 为基准的一趟快速排序结果以及第 2 趟直接选择排序后的结果。

答：（1）[18 5 16] 19 [21 23]

（2）5，16， 21， 19，18，23

5. 对于一组给定的关键字序列{58,12,33,20,65,77}，分别写出用冒泡排序法和快速排序法进行升序排列的各趟结果。

答：

冒泡排序各趟结果：

初始状态　58　12　33　20　65　77
第一趟　12　33　20　58　65　77
第二趟　12　20　33　58　65　77
第三趟　12　20　33　58　65　77
结果　12　20　33　58　65　77

快速排序各趟结果：
初始状态　58　12　33　20　65　77
第一趟　[20　12　33]　58　[65　77]
分别快速排序　[12]　20　[33]　58　65　[77]
结果　12　20　33　58　65　77

6. 对于给定的一组数据：3、6、1、2、4、5，请分别写出直接插入排序和快速排序的各趟运行结果

答：

直接插入排序各趟结果：
初态　[3]　6　1　2　4　5
第一趟　[3　6]　1　2　4　5
第二趟　[1　3　6]　2　4　5
第三趟　[1　2　3　6]　4　5
第四趟　[1　2　3　4　6]　5
第五趟　[1　2　3　4　5　6]

快速排序各趟结果：
初始状态　3　6　1　2　4　5
第一趟　[2　1]　3　[6　4　5]
分别快速排序　[1]　2　3　[5　4]　6
对[5 4]快速排序　1　2　3　[4]　5　6
结果　1　2　3　4　5　6

7. 对于一组给定的关键字序列{58,12,33,20,65,77}，分别写出用冒泡排序法和简单选择排序法进行升序排列的各趟结果。

答：

冒泡排序各趟结果：
初始状态　58　12　33　20　65　77
第一趟　12　33　20　58　65　77
第二趟　12　20　33　58　65　77
第三趟　12　20　33　58　65　77

简单选择排序各趟结果：
初始状态　58　12　33　20　65　77
第一趟　12　58　33　20　65　77

第二趟　12　20　33　58　65　77
第三趟　12　20　33　58　65　77
第四趟　12　20　33　58　65　77
第五趟　12　20　33　58　65　77

8. 对于一组给定的关键字{53,87,12,61,35,48}，写出分别用直接插入法和简单选择法进行升序排列的各趟结果

答：

直接插入排序各趟结果：

初态　[53]　87　12　61　35　48
第一趟　[53　87]　12　61　35　48
第二趟　[12　53　87]　61　35　48
第三趟　[12　53　61　87]　35　48
第四趟　[12　35　53　61　87]　48
第五趟　[12　35　48　53　61　87]

简单选择排序各趟结果：

初态　53　87　12　61　35　48
第一趟　[12]　87　53　61　35　48
第二趟　[12　35]　53　61　87　48
第三趟　[12　35　48]　61　87　53
第四趟　[12　35　48　53]　87　61
第五趟　[12　35　48　53　61]　87

9. 对于一组给定的关键字{53,87,12,61,35,48}，写出分别用冒泡排序法和直接插入排序法进行升序排列的各趟结果

答：

冒泡排序各趟结果：

初态　53　87　12　61　35　48
第一趟　53　12　61　35　48　87
第二趟　12　53　35　48　61　87
第三趟　12　35　48　53　61　87
第四趟　12　35　48　53　61　87

直接插入排序各趟结果：

初态　[53]　87　12　61　35　48
第一趟　[53　87]　12　61　35　48
第二趟　[12　53　87]　61　35　48
第三趟　[12　53　61　87]　35　48
第四趟　[12　35　53　61　87]　48
第五趟　[12　35　48　53　61　87]

10. 对于一组给定的关键字{53,87,12,61,35,48}，写出分别用冒泡排序法和直接插入排序法进行降序排列的各趟结果。

答：

冒泡排序各趟结果：

初态	53	87	12	61	35	48
第一趟	87	53	61	35	48	12
第二趟	87	61	53	48	35	12
第三趟	87	61	53	48	35	12

直接插入排序各趟结果：

初态	[53]	87	12	61	35	48
第一趟	[87	53]	12	61	35	48
第二趟	[87	53	12]	61	35	48
第三趟	[87	61	53	12]	35	48
第四趟	[87	61	53	35	12]	48
第五趟	[87	61	53	48	35	12]

11. 对于一组给定的关键字序列{45,19,81,58,12,33,20,65,77,28}，写出用简单选择排序法进行降序排列的各趟结果。

答：

初始状态	45	19	81	58	12	33	20	65	77	28
第一趟	81	19	45	58	12	33	20	65	77	28
第二趟	81	77	45	58	12	33	20	65	19	28
第三趟	81	77	65	58	12	33	20	45	19	28
第四趟	81	77	65	58	12	33	20	45	19	28
第五趟	81	77	65	58	45	33	20	12	19	28
第六趟	81	77	65	58	45	33	20	12	19	28
第七趟	81	77	65	58	45	33	28	12	19	20
第八趟	81	77	65	58	45	33	28	20	19	12
第九趟	81	77	65	58	45	33	28	20	19	12

12. 对于一组给定的关键字序列{45,19,81,58,12,33,20,65,77,28}，写出用直接插入排序法进行降序排列的各趟结果。

答：

初始状态	[45]	19	81	58	12	33	20	65	77	28
第一趟	[45	19]	81	58	12	33	20	65	77	28
第二趟	[81	45	19]	58	12	33	20	65	77	28
第三趟	[81	58	45	19]	12	33	20	65	77	28
第四趟	[81	58	45	19	12]	33	20	65	77	28
第五趟	[81	58	45	33	19	12]	20	65	77	28
第六趟	[81	58	45	33	20	19	12]	65	77	28
第七趟	[81	65	58	45	33	20	19	12]	77	28
第八趟	[81	77	65	58	45	33	20	19	12]	28
第九趟	[81	77	65	58	45	33	28	20	19	12]

13. 对于一组给定的关键字序列{45,19,81,58,12}，分别写出用直接插入排序法和快速排序法进行降序排列的各趟结果

答：

直接插入排序各趟结果：

初始状态　[45]　19　81　58　12
第一趟　　[45　19]　81　58　12
第二趟　　[81　45　19]　58　12
第三趟　　[81　58　45　19]　12
第四趟　　[81　58　45　19　12]

快速排序各趟结果：

初始状态　　45　19　81　58　12
第一趟　　[58　81]　45　[19　12]
分别快速排序　[81]　58　45　19　[12]
结果　　81　58　45　19　12

14. 对于一组给定的关键字序列{45,19,81,58,12}，分别写出用直接插入排序法和快速排序法进行降序排列的各趟结果

答：

直接插入排序各趟结果：

初始状态　[45]　19　81　58　12
第一趟　　[45　19]　81　58　12
第二趟　　[81　45　19]　58　12
第三趟　　[81　58　45　19]　12
第四趟　　[81　58　45　19　12]

快速排序各趟结果：

初始状态　　45　19　81　58　12
第一趟　　[58　81]　45　[19　12]
分别快速排序　[81]　58　45　19　[12]
结果　　81　58　45　19　12

15. 对于一组给定的关键字序列{45,19,81,58,12}，分别写出用直接插入排序法和快速排序法进行升序排列的各趟结果。

答：

直接插入排序各趟结果：

初始状态　[45]　19　81　58　12
第一趟　　[19　45]　81　58　12
第二趟　　[19　45　81]　58　12
第三趟　　[19　45　58　81]　12
第四趟　　[12　19　45　58　81]

快速排序各趟结果：

初始状态　　45　19　81　58　12

第一趟	[12	19]	45	[58	81]
分别快速排序	12	[19]	45	58	[81]
结果	12	19	45	58	81

三.算法设计题

1. 假设二叉树的结点类型定义如下：

```
typedef struct node
{
int    data;
struct node *lchild,*rchild;
}Bstree;
```

给定一组整型值，-1 作为结束，写出二叉排序树的生成过程。

2. 写出用直接插入排序进行升序排序的算法。
3. 写出用冒泡排序进行升序排序的算法。
4. 在一按关键字升序排列的整型线性表中插入一个元素，插入后使得该线性表依然有序。要求用直接插入排序算法中一趟插入的思想。

解答:

1. 给定一组整型值，-1 作为结束，写出二叉排序树的生成过程。

```
void insert(BiTree t, Bstree *s)     //将 s 插入到 t 为根的二叉排序树中
{   if(s->data<t->data)
    {   if (t->lchild==NULL)
                t->lchild=s;
        else        insert(t->lchild,s);
     }
   else
    {   if (t->rchild==NULL)
                t->rchild=s;
        else
                insert(t->rchild,s);
     }
}
Bstree* bstreecreate() /*二叉排序树的生成算法*/
{   int x;
    Bstree *s,*t;
    t=NULL;
    printf("input the elements of bstree,end flag is -1\n");
    scanf("%d",&x);
    while(x!=-1)
       {   s=(BiTNode*)malloc(sizeof(BiTNode));
           s->data=x;
           s->lchild=s->rchild=NULL;
           if (t==NULL)
```

```
                t=s;
            else
                insert(t,s);
            scanf("%d",&x);
        }
    return (t);
}
```

2. 写出用直接插入排序进行升序排序的算法。

```
void    ins_sort（datatype R[ ],int n）
{   int i;
    for( i=2;i<=n;i++)
      { R[0]=R[i];
        j=i-1;
        while( R[0].key<R[j].key )
          { R[j+1]=R[j] ;
            j- -;
          }
        R[j+1]=R[0] ;
        }
  }
```

3. 写出用冒泡排序进行升序排序的算法。

```
    void bubble_sort(datatype R[ ],int n)
    {   int i=1,j,swap=1;
      while (i<n&&swap==1)
      {   swap=0;
          for( j=n;j<=i+1;j--)
          if( R[j].key<R[j-1].key)then
                        {   R[0]=R[j];
                            R[j]=R[j-1];
                            R[j-1]=R[0];
                            swap=1;
                        }
          i++;
      }//while
}
```

4.
```
void    ins_sort（int R[ ],int *n，int x）
{       int i;
        R[0]=x;
        i=*n;
        while( R[0]<R[i])
            { R[i+1]=R[i] ;
```

```
                i- -;
            }
        R[i+1]=R[0] ;
        (*n)++;
    }
```

第十一章　查找

一.填空题

1. 折半查找（即二分查找）只能在 **有序的顺序表** 上进行。

2. 设有序查找表中有 13 个元素。采用折半查找方法进行查找，查找成功时最少比较 **1** 次，最多比较 **4** 次。

3. 如果在一棵二叉树中,每个结点的值都大于它的左子树上所有结点的值,而小于它的右子树上所有结点的值，这样的一棵二叉树称为 **二叉排序树** 。

4. 在顺序表（8,11,15,19,25,26,30,33,42,48,50）中，用二分（折半）法查找关键码值 20，需做的关键码比较次数为 **4** 。

5. 已知有序表为(12,18,24,35,47,50,62,83,90,115,134)当用二分法查找 18 时，需 **4** 次查找成功。

6. 已知一棵二叉排序树，通过 **中序遍历** ，可得到结点的有序序列。

7. 设一哈希表表长 M 为 100 ，用除留余数法构造哈希函数，即 H(K)=K MOD P（P<=M）， 为使函数具有较好性能，P 应选 **质数** 。

8. 不包括对数据元素的插入和删除操作的查找称为 **静态** 查找。

9. 包括对数据元素的插入和删除操作的查找称为 **动态** 查找

10. 分块查找的前提是 **块间有序** 。

11. 根据元素的关键码确定元素存储位置的存储方式叫 **散列** 存储。

12. 在哈希查找中，元素关键字值与其在哈希表中存放位置的对应关系称为 **哈希函数** 。

13. 在哈希查找中，不同关键字值对应到同一哈希地址上的现象称为 **冲突** 。

14. 在有序表（41,62,75,77,82,95,100）上进行二分查找，查找关键字为 82 的数据元素需要比较的次数是 **3** 次。

15. 在结点数确定的二叉排序树上进行查找的平均查找长度与二叉树的形态有关，最好的情况是二叉排序树为 **平衡二叉树** 树的时候。

二.简答题

1. 什么是二叉排序树？给定一组关键字序列，其中没有相同的关键字，若关键字顺序不同，生成的二叉排序树会不会不同？依次输入关键字{50,28,73,91,56,18,34,86}，画出所生成的二叉排序树，并写出其中序遍历序列。

答：二叉排序树或者是一棵空树，或者是一棵具有如下特征的非空二叉树：

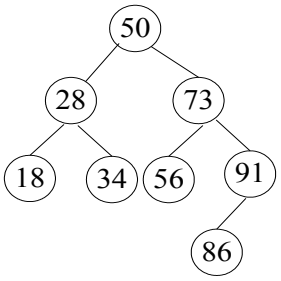
(1)若它的左子树非空，则左子树上所有结点的关键字均小于根结点的关键字；

(2)若它的右子树非空，则右子树上所有结点的关键字均大于等于根结点的关键字；

(3)左、右子树本身又都是一棵二叉排序树。

给定一组关键字序列，其中没有相同的关键字，若关键字顺序不同，生成的二叉排序树会不同（会生成不同的二叉排序树）。

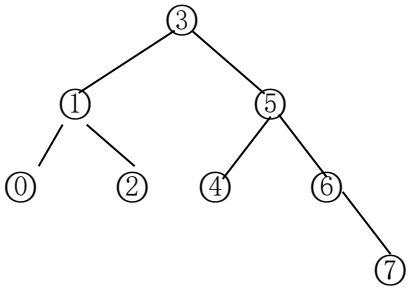
依次输入关键字{50,28,73,91,56,18,34,86}，所生成的二叉排序树为：



其中序遍历序列为：18、28、34、50、56、73、86、91

2. 画出描述 n=8（具有 8 个记录）的折半查找（即二分查找）过程判定树，并计算查找成功时的平均查找长度 ASL(假定查找每个记录的概率相等)。

答：n=8 的折半查找过程判定树为：



查找成功时的平均查找长度 ASL 为：（1×1＋2×2＋4×3＋1×4）/8＝21/8

3. 设一组初始记录关键字集合为(25，10，8，27，32，68)，哈希表地址区间为 0~7，哈希函数 H(k)=k mod 7，用线性探测再散列作为解决冲突的方法设计哈希表，将其内容填入下表，并计算 ASL。

哈希表

地址	0	1	2	3	4	5	6	7
关键字								

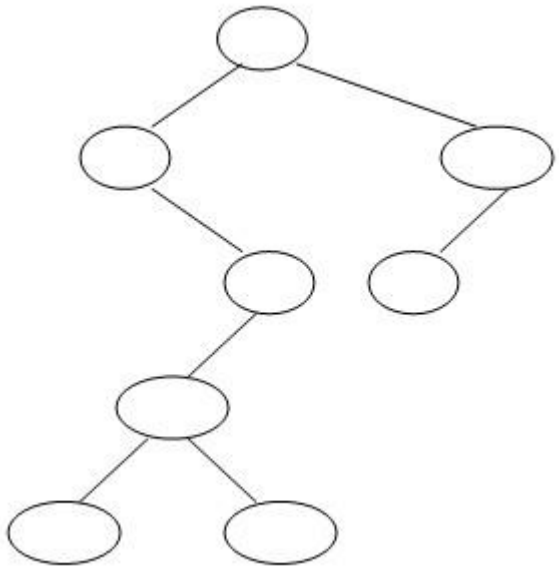
答：

哈希表

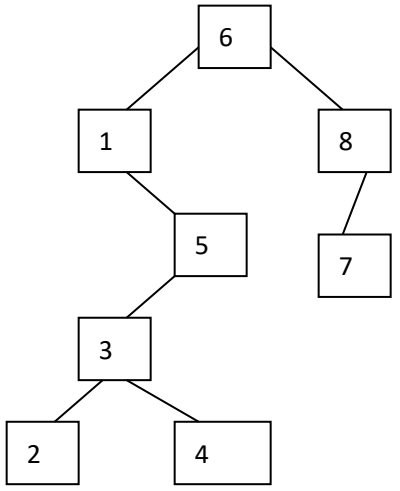
地址	0	1	2	3	4	5	6	7
关键字		8		10	25	32	27	68

平均查找长度 ASL=(1+1+1+1+2+3)/6=1.5

4. 一棵排序二叉树结构如下，各结点的值从小到大依次为 1~8，请标出各结点的值。



答：该排序二叉树为：



5. 给定关键码序列 11，78，10，1，3，2，4，21，分别用线性探测法和拉链法处理冲突，试画出它们对应的散列存储形式，并求出每一种查找的成功平均查找长度。散列函数 Hash(key)=key%11。

答：（1）用线性探测法处理冲突

散列存储形式：

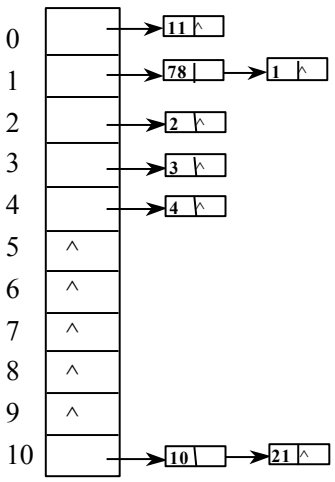
0	1	2	3	4	5	6	7	8	9	10
11	78	1	3	2	4	21				10

线性探查的散列表

平均查找长度 ASL=(1+1+1+2+1+3+2+8)/8=2.375

（2）用拉链法处理冲突

散列存储形式：



拉链法的哈希表

平均查找长度 ASL=(1*6+2*2)/8=1.25

6. 关键字集合为{47，7，29，11，16，92，22，8，3，13}，地址区间为 0~10，构造合理的哈希函数，用线性探测再散列法处理冲突，画出哈希表并求出平均查找长度 ASL 。（2+4+2=8 分）

哈希函数：

哈希表：

地址	0	1	2	3	4	5	6	7	8	9	10
关键字											

平均查找长度 ASL=

答：

哈希函数：

Hash(k)=k%11

地址表

关键字值	47	7	29	11	16	92	22	8	3	13
哈希地址	3	7	7	0	5	4	0	8	3	2

哈希表：

地址	0	1	2	3	4	5	6	7	8	9	10
关键字	11	22	13	47	92	16	3	7	29	8	

平均查找长度 ASL=(1+1+2+1+1+1+2+2+4+1)/10=1.6

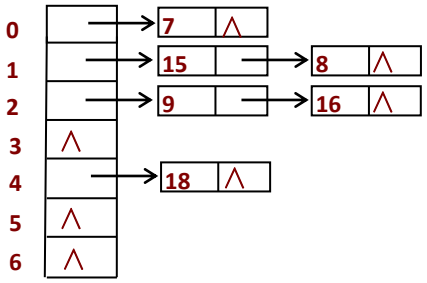
7. 设有关键字序列（7，8，9，16，15，18），采用除留取余法构造哈希函数 hash(key)=key%7；用链地址法处理冲突，将其散列到地址空间 0~6 之中，（1）求出其对应哈希地址；（2）画出对应的哈希表；（3）求出其平均查找长度。（1+5+2=8 分）

答：

（1）地址表：

关键字	7	8	9	16	15	18
地址	0	1	2	2	1	4

（2）哈希表



（3）ASL=(1+1+2+1+2+1)/6=4/3

8. 已知一个哈希表如下所示，其地址空间为 0~10，哈希函数为 Hash(k)=k%11，冲突处理方法为线性探测再散列。

地址	0	1	2	3	4	5	6	7	8	9	10
关键字值	66	45	78	32	54	48		62	30	18	21

回答以下问题：

（1）在此哈希表上，在哪些地址上发生了冲突？（3 分）

（2）在查找元素 32、54 和 21 时各需要进行多少次比较？（3 分）

（3）计算等概率条件下查找成功时的平均查找长度。（2 分）

答：.

关键字	66	45	78	32	54	48	62	30	18	21
哈希地址	0	1	1	10	10	4	7	8	7	10

（1）在 1、10、7 地址存在冲突

（2）查找 32，5 次比较；

查找 54,6 次比较

查找 21,1 次比较

(3)ASL=(1+1+2+5+6+2+1+1+3+1)/10=2.3

9. 哈希表的地址空间为 0~6,哈希函数为 H(k) = k%7,采用二次探测再散列法处理冲突，如将关键字序列(9,11,8,13,15,10)依次存放到哈希表中，填写下面的地址表和生成的哈希表，并计算出在此哈希表上进行查找的平均查找长度。 (2+4+2=8 分)

地址表

关键字值	9	11	8	13	15	10
哈希地址						

哈希表

地址	0	1	2	3	4	5	6
关键字值							

平均查找长度 ASL=

答：

地址表

关键字值	9	11	8	13	15	10
哈希地址	2	4	1	6	1	3

哈希表

地址	0	1	2	3	4	5	6
关键字值	15	8	9	10	11		13

平均查找长度 ASL=(1+1+1+1+3+1)/6=4/3

10. 设有关键字序列（13，10，6，14，21，17），试用除留取余法构造哈希函数，用线性探查再散列将其散列到地址空间 0~6 之中，请问：

①除留取余法中除数 p 如何选取，可以减少冲突？你因此用除留取余法构造的哈希函数为?

②画出对应的地址表和哈希表。

③求出等概率条件下的平均查找长度（2+4+2=8 分）

地址表：

关键字	13	10	6	14	21	17
地址						

哈希表：

地址	0	1	2	3	4	5	6
关键字							

答：

① p 选取不大于表长的最大质数，Hash(k)=k%7

②（13，10，6，14，21，17）

地址表：

关键字	13	10	6	14	21	17

地址	6	3	6	0	0	3

哈希表：

地址	0	1	2	3	4	5	6
关键字	6	14	21	10	17		13

③ASL=(1+1+2+2+3+2)/6=11/6

11. 哈希表的地址空间为 0~9,哈希函数为 H(k) = k%7,采用线性探测再散列法处理冲突，如将关键字序列(9,11,16,10,15,12,,24,20,18)依次存放到哈希表中，填写下面的地址表和生成的哈希表，并计算出在此哈希表上进行查找的平均查找长度。 (2+4+2=8 分)

地址表

关键字值	9	11	16	10	15	12	24	20	18
哈希地址									

哈希表

地址	0	1	2	3	4	5	6	7	8	9
关键字值										

答：

地址表

关键字值	9	11	16	10	15	12	24	20	18
哈希地址	2	4	2	3	1	5	3	6	4

哈希表

地址	0	1	2	3	4	5	6	7	8	9
关键字值		15	9	16	11	10	12	24	20	18

平均查找长度 ASL=(1+1+2+3+1+2+5+3+6)/9=8/3

12. 假设哈希表的地址空间为 0-6,哈希函数为 H(k) = k%7,采用二次探测再散列法处理冲突，如将关键字序列(26,72,35,8,18,60)依次存放到哈希表中，(1)根据哈希函数计算每个关键字对应的哈希地址并画出生成的哈希表；(2)计算出在此哈希表上进行查找的平均查找长度。（6+2=8 分）

地址表

关键字	26	72	35	8	18	60
地址						

哈希表

地址	0	1	2	3	4	5	6
关键字							

ASL=

答：

地址表

关键字	26	18	35	12	25	60
地址	5	4	0	5	4	4

哈希表

地址	0	1	2	3	4	5	6
关键字	35	25	60		18	26	12

ASL=(1+1+1+2+5+6)/6=8/3

以上错，应为：

地址表

关键字	26	72	35	8	18	60
地址	5	2	0	1	4	4

哈希表

地址	0	1	2	3	4	5	6
关键字	35	8	72	60	18	26	

ASL=(1+1+1+1+1+3)/6=4/3

13. 哈希表的地址空间为 0~9,哈希函数为 H(k) = k%7,采用二次探测再散列法处理冲突，如将关键字序列(9,11,16,10,15,12,24,20,18)依次存放到哈希表中，填写下面的地址表和生成的哈希表，并计算出在此哈希表上进行查找的平均查找长度。 (2+4+2=8 分)

地址表

关键字值	9	11	16	10	15	12	24	20	18
哈希地址									

哈希表

地址	0	1	2	3	4	5	6	7	8	9
关键字值										

平均查找长度 ASL=

答：

地址表

关键字值	9	11	16	10	15	12	24	20	18
哈希地址	2	4	2	3	1	5	3	6	4

哈希表

地址	0	1	2	3	4	5	6	7	8	9
关键字值		15	9	16	11	12	20	10	18	24

平均查找长度 ASL=(1+1+2+4+1+1+5+1+4)/9=20/9

14. 已知一个哈希表如下所示，其地址空间为 0~10，哈希函数为 Hash(k)=k%11，冲突处理方法为二次探测再散列。

冲突处理方法为二次探测再散列。

地址	0	1	2	3	4	5	6	7	8	9	10
关键字值	66	45	78	54	48		18	62	30	32	21

回答以下问题：

（1）在此哈希表上，在哪些地址上发生了冲突？（2 分）

（2）在查找元素 32、54 和 21 时各需要进行多少次比较？（3 分）

（3）计算等概率条件下查找成功时的平均查找长度。（3 分）

答：

关键字	66	45	78	32	54	48	62	30	18	21
哈希地址	0	1	1	10	10	4	7	8	7	10

（1）在 1、10、7 地址存在冲突

（2）查找 32，3 次比较；

查找 54,4 次比较

查找 21,1 次比较

(3)ASL=(1+1+2+3+4+1+1+1+3+1)/10=1.8

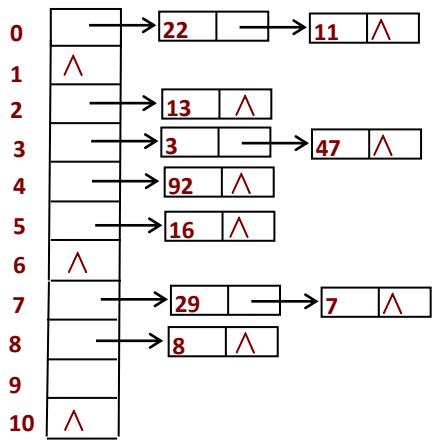
15. 关键字集合为{47，7，29，11，16，92，22，8，3，13}，地址区间为 0~10，构造合理的哈希函数，用链地址法处理冲突，画出哈希表并求出平均查找长度 ASL 。

答：

哈希函数：

Hash(k)=k%11

关键字	47	7	29	11	16	92	22	8	3	13
哈希地址	3	7	7	0	5	4	0	8	3	2



ASL=(1+2+1+1+2+1+1+1+2+1)/10=1.3

三.算法设计题

1. 写出在顺序表中进行二分查找的非递归算法。
2. 写出在顺序表中进行二分查找的递归算法。
3. 假设二叉树的结点类型定义如下：

```
typedef struct node
{
int    data;
struct node *lchild,*rchild;
}Bstree;
```

写出在二叉排序树中进行查找的算法。

4. 假设二叉排序树 t 的各个元素值为整型且均不相同，请定义结点类型并设计一个算法按递减次序输出各元素的值。

解答:

1.
```
int Binary_Search(S_TBL *tbl,keytype kx)
    { int low,high,mid;
      low=1;high=tbl->last;
      while(low<=high)
          { mid=(low+high)/2;      if(kx<tbl->data[mid].key)
            high=mid-1;      else if(kx>tbl->data[mid].key)
            low=mid+1;       else return mid;
          }
       return 0;
      }
```

2. 写出在顺序表中进行二分查找的递归算法。

```
int BSearch(S_TBL *tbl,keytype kx，int low,int high)
/*在下界为 low，上界为 high 的线性表 tbl 中折半查找关键字为 kx 的元素*/
{ int mid;
   if(low>high) return -1;      mid=(low+high)/2;      if(kx== tbl->data[mid].key) return mid;
   if(kx< tbl->data[mid].key) return(BSearch(tbl,kx,low,mid-1));
   return(BSearch(tbl,kx,mid+1,high));
}
```

3. 写出在二叉排序树中进行查找的算法。

```
BiTNode* SearchData(BiTree t, KeyType kx)
    /*在二叉排序树 t 上查找关键码为 kx 的元素,找到返回其所在及结点的指针，否则返回一个空指针 */
        {BiTNode* q=t;
            while(q)
                { if    (kx>q->data.key)
                            q=q->rchild;                          /*  在右子树中查找  */
                  else if (kx<q->data.key)
                            q=q->lchild;                          /*  在左子树中查找  */
                       else return q ;                            /*  查找成功  */
                }
             return    NULL;                                      /*  查找不成功  */
        }
```

4. typedef struct node

```
{      int    data;
       struct node *lchild,*rchild;
}Bstree;

void bianli(Bstree* bt)
       {    if (bt!=NULL)
                {    bianli(bt->rchild);
                     printf("%5d",bt->data);
                     bianli(bt->lchild);
                 }
       }
```