

【软考达人】

软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+ 免费题库



免费备考资料

PC版题库: ruankaodaren.com

中级软件设计师下午试题模拟57

试题一

【算法说明】

下面是一段插入排序的程序，将 $R[k+1]$ 插入到 $R[1 \dots k]$ 的适当位置。

```
R[0]=R[k+1];
```

```
j=k;
```

```
while (R[j]>R[0])
```

```
{
```

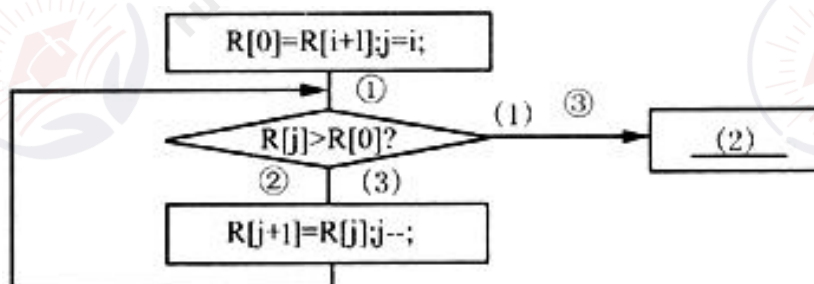
```
R[j+1]=R[j];
```

```
j--;
```

```
}
```

```
R[j+1]=R[0];
```

【流程图】



【测试用例设计】

(while循环次数为0、1、2次)

测试用例表

循环次数	输入数据					R[0]	预期结果					覆盖路径	
	j	R[i-2]	R[i-1]	R[i]	R[i+1]		j	R[i-2]	R[i-1]	R[i]	R[i+1]	约束	路径
0	i	-	-	1	2	2	i	-	-	1	2	<	(4)
	i	-	-	1	1	1	i	-	-	1	1	=	(3)
1	i	-	1	3	2	2	i-1	-	1	2	3	(7)	①②③
	i	-	2	3	2	2	i-i	-	(8)	2	(9)	>=	①②③
2	i	1	3	4	2	2	i-2	1	2	3	4	>><	(5)
	i	2	3	4	2	2	i-2	2	2	3	4	>>=	(6)

1、【问题1】

指出算法的流程图中(1)~(3)处的内容。

2、【问题2】

指出测试用例设计中(4)~(9)处的内容。

试题二

【说明】

本流程图实现从成绩文件生成学生成绩一览表。

某中学某年级的学生成绩数据(分数)登录在成绩文件10中，其记录格式见表1：

表1

学号	姓名	课程1成绩	课程2成绩	课程6成绩
----	----	-------	-------	-------	-------

由该成绩文件生成见表2的学生成绩一览表。生成的学生成绩一览表按学号升序排列。表中的名次是指该生相应课程在年级中的名次。

表2

学号	姓名	课程1		课程2			课程6	
		成绩	名次	成绩	名次	成绩	名次

流程图中的顺序文件F0是学生成绩文件，F0文件经处理1处理后产生顺序文件F，然后经过处理2至处理4对文件F进行处理和更新。在处理5中，仅对文件F的纪录进行学生成绩一览表的编排输出，不进行排序和增加名次等处理。

3、【问题1】

流程图中文件F的纪录格式设定为见表3形式：

表3

学号	姓名	课程代码	①	②
----	----	------	---	---

其中的①、②应定义为何种数据项？

4、【问题2】

简述处理2、处理3和处理4作何种处理，若有排序处理则需指明排序的键及序（升序或降序）。

【流程图】



图 3

试题三

5、 有下列关于运动会管理系统的ER图，如图所示。图中矩形表示实体，圆表示属性，双圆表示关键字属性，菱形表示实体之间的关系。假定已通过下列SQL语言建立了基本表。CREATE TABLE ATHLETE

```

  ANAME CHAR(20),
  ASEX CHAR(1),
  ATEAM CHAR(20));
CREATE TABLE ITEM
  (INO CHAR(6) NOT NULL,
  INAME CHAR(20),
  ITIME CHAR(12),
  IPLACE CHAR(20);
CREATE TABLE GAMES
  (ANO CHAR(6) NOT NULL,
  INO CHAR(6) NOT NULL,
  SCORRE CHAR(10));

```

为了答题的方便，图中的实体和属性同时给出了中英文两种文字，回答问题时只需写出英文名

即可。

【E-R图】



图 10 E-R 图

【问题】

填充下列SQL程序1~4中的(1)~(7)，使它们分别完成相应的功能：

程序1：统计参加比赛时男运动员人数。

SELECT (1)

FROM ATHLETE

WHERE ASEX='M';

程序2：查100872号运动员参加的所有项目及其比赛时间和地点。

SELECT ITEM, INO, IN A ME, ITIME, IPLACE

FROM GAMES, ITEM

WHERE (2);

AND (3);

程序3：查参加100035项目的所有运动员名单。

SELECT ANO, ANAME, ATEAM

FROM ATHLETE

WHERE (4);

(SELECT (4) (5)

FROM GAMES

WHERE GAMES ANO=ATHLETE.ANO AND INO='100035');

程序4：建立运动员成绩视图。

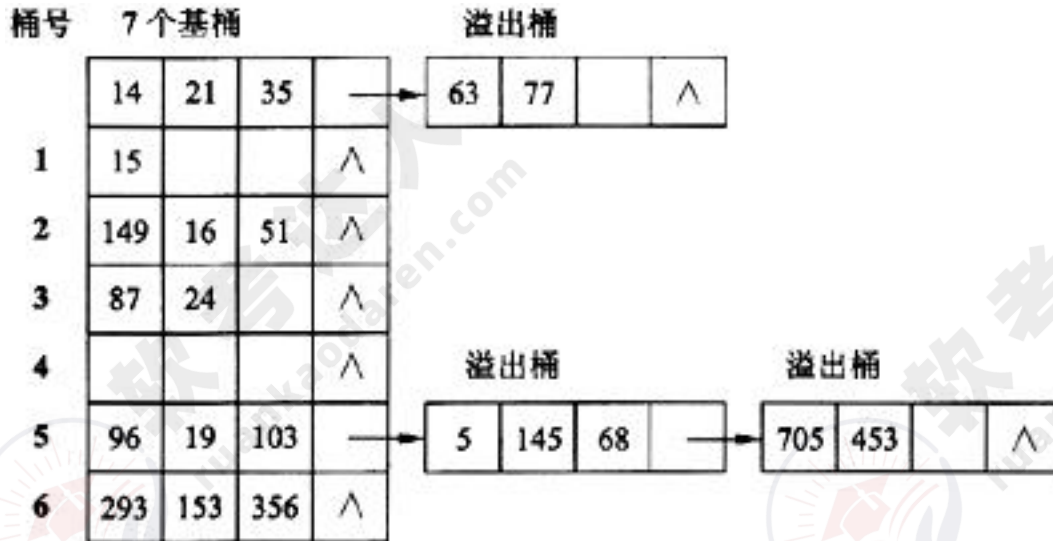
(6) ATHLETE-SCORE

AS SELECT ATHLETE, ANO, ANAME, ATEAM, INAME, SCORE

FORM (7) WHERE ATHLETE.ANO=GAMES.ANO AND GAMES.INO=ITEM.INO;

试题四

6、例如：设散列函数为 $\text{Hash}(\text{Key}) = \text{Key} \bmod 7$ ，记录的关键字序列为15, 14, 21, 87, 97, 293, 35, 24, 149, 19, 63, 16, 103, 77, 5, 153, 145, 356, 51, 68, 705, 453，建立的散列文件内容如图所示。



为简化起见，散列文件的存储单位以内存单元表示。

函数 $\text{InsertToHashTable}(\text{int NewElemKey})$ 的功能是：将元素 NewElemKey 插入散列桶中，若插入成功则返回0，否则返回-1。

采用的散列函数为 $\text{Hash}(\text{NewElemKey}) = \text{NewElemKey} \% P$ ，其中 P 为设定的基桶数目。

函数中使用的预定义符号如下：

```
#define NULLKEY  -1          /*散列桶的空闲单元标识*/
#define P      7           /*散列文件中基桶的数目*/
#define ITEMS   3          /*基桶和溢出桶的容量*/
typedef struct BucketNode{ /*基桶和溢出桶的类型定义*/
    int KcyData[ITEMS];
    struct BucketNode *Link;
}BUCKET;
BUCKET Bucket[P];          /*基桶空间定义*/
```

[函数]

```
int InsertToHashTable(int NewElemKey){
    /*将元素NewElemKey插入散列桶中，若插入成功则返回0，否则返回-1*/
    /*设插入第一个元素前基桶的所有KeyData[]、Link域已分别初始化为NULLKEY、
    NULL*/
    int Index;    /*基桶编号*/
    int i, k;
    BUCKET *s, *front, *t;
    (1);
    for(i=0; i<ITEMS; i++) /*在基桶查找空闲单元，若找到则将元素存入*/
        if(Bucket[Index].KeyData[i]=NULLKEY){
            Bucket[Index].KeyData[i]=NewElemKey; break;
        }
    if((2)) return 0;
    /*若基桶已满，则在溢出桶中查找空闲单元，若找不到则申请新的溢出桶*/
    (3);          t=Bucket[Index].Link;
    if(t!=NULL)    /*有溢出桶*/
        while (t!=NULL){
            for(k=0; k<ITEMS; k++)
                if(t->KeyData[k]=NULLKEY){ /*在溢出桶链表中找到空闲单元*/
```

```

        t->KeyData[k]=NewElemKey;    break;
    }/*if*/
    front=t;
    if(__ (4) __) t=t->Link;
    else break;
}/*while*/
}/*if*/
if(__ (5) __) { /*申请新溢出桶并将元素存入*/
    s=(BUCKET*)malloec(sizeof(BUCKET));
    if(!s) return-1;
    s->Link=NULL;
    for(k=0; k<ITEMS; k++)
        s->KeyData[k]=NULLKEY;
    s->KeyData[0]=NewElemKey;
    __ (6) __;
}/*if*/
return 0;
}/*InsertToHashTable*/

```

试题五

7、 【说明】

下面的程序先构造Point类，再顺序构造Ball类。由于在类Ball中不能直接存取类Point中的xCoordinate及yCoordinate属性值，Ball中的toString方法调用Point类中的toString方法输出中心点的值。在MovingBall类的toString方法中，super.toString调用父类Ball的toString方法输出类Ball中声明的属性值。

【Java代码】

```

//Point.java文件
public class Point{
    private double xCoordinate;
    private double yCoordinate;
    public Point(){}
    public Point(double x,double y){
        xCoordinate=x;
        yCoordinate=y;
    }
    public String toString(){
        return "("+Double.toString(xCoordinate)+","
            +Double.toString(yCoordinate)+")";
    }
    //other methods
}

//Ball.java文件
public class Ball{
    private __ (1) __; //中心点
    private double radius; //半径
    private String color; //颜色
    public Ball(){}
    public Ball(double xValue, double yValue, double r){
        //具有中心点及其半径的构造方法
    }
}

```

```

        center=__ (2) __; //调用类Point中的构造方法
        radius=r;
    }
    public Ball(double xValue, double yValue, double r, String c){
        //具有中心点、半径和颜色的构造方法
        __ (3) __; //调用3个参数的构造方法
        color=c;
    }
    public String toString(){
        return "A ball with center"+center.toString()
            +",radius "+Double.toString(radius)+" ,color"+color;
    }
    //other methods
}
class MovingBall __ (4) __{
    private double speed;
    public MovingBall(){}
    public MoringBall(double xValue, double yValue, double r, String
c, double s){
        __ (5) __; //调用父类Ball中具有4个参数的构造方法
        speed=s;
    }
    public String toString(){
        return super.toString()+" ,speed"+Double.toString(speed);
    }
    //other methods
}
public class test{
    public static void main(String args[]){
        MovingBall mb=new MovingBall(10,20,40,"green",25);
        System.out.println(mb);
    }
}

```

答案：

试题一

- 1、(1)F (2) $R[j+1]=R[0]$ (3)T
 2、(4)①③ (5)①②②③ (6)①②②③ (7) $><$ (8)1 (9)3

[解析] 本题考查用路径覆盖方法为算法设计足够的测试用例，属于基本概念的送分题。这类题拿分的关键是考生平时对于理论的理解和临场的细心。

试题二

- 3、①成绩 ②名次
 4、课程代码按升序排列、成绩按降序排列

[解析] 处理2对每个文件F进行排序。处理3对每个课程代码，确定学生名次，写入文件F的相应字段。处理4按学号(升序)、课程代码(升序)对文件F排序。

试题三

- 5、 (1)COUNT(*) (若答COUNT或COUNT，得2分)

- (2) GAMES.INO=IFEM.INO
 (3) GAMES.ANO='100872' (注：(2)、(3)可互换、无前缀得1分)
 (4) EXISTS
 (5) *或ANO或INO或SCORE或后3个列名的任意组合
 (6) CREATEVIEW
 (7) ATHLETE, ITEM, GAMES (3项可交换。)
 注：(4)、(5)也可为
 (4) ANOIN
 (5) ANO

[解析] 本题是关于系数据库标准语言——SQL (Structured Query Language) 语言的题目，由题目中给出的ER图可知3个表中，ATHLETE和ITEM是基本表，表ATHLETE的主键是运动员编号ANO，表ITEM的主键是项目编号INO，表GAMES是一个视图，以ANO、INO为外键。

程序1统计参加比赛的男运动员人数，也就是表ATHLETE中，SEX='M'的记录个数，所以要用到库函数COUNT(*)。这里要注意的是COUNT与COUNT(*)区别，COUNT的功能是对一列中的值计算个数，而COUNT(*)才是计算数据库中记录的个数。所以填空①的答案为“COUNT(*)”。

程序的2统计100872号运动员参加的所有项目及比赛时间和地点，所以SELECT后面的内容是项目编号ITEM.INO、项目名称INAME时间ITIME及地点IPLACE。统计涉及比赛表GAMES和项目表ITEM，所以FROM后面的内容为GAMES、ITEM。本题考的是连接查询，所谓连接查询指的是涉及两个以上的表的查询。由于是统计100872号运动员参加的所有项目及比赛时间和地点，所以查询条件中必然有 GAMES.INO='100872' (程序中引用到字段时，若字段名在各个表中是惟一的，则可以把字段名前的表名去掉，否则，应当加上表名作为前缀，以免引起混淆)。由于GAMES表中只有比赛的成绩，那些关于项目的数据必须从项目表ITEM中取得，所以还应该有两个表之间的关联，即 GAMES.INO=ITEM.INO。所以填空②和③可交换，不影响查询结果。

程序3要求查参加100035项目的运动员名单。分析查询表达式，必首先查询GAMES表，找出参加100035项目的那些运动员的编号ANO，即GAMES.ANO=ATHLETE.ANO AND INO='100035'，然后再根据查询到的运动员号ANO从ATHLETE表中抽取运动员的数据。所以填空④的答案为“EXISTS”或“ANOIN”，填空⑤的答案为“ANO”。

程序4要求建立运动员成绩视图。建立视图的命令为CREATE VIEW，所以填空⑥的答案一定是“CREATEVIEW”。建立的是运动员成绩视图，那么一定涉及运动员情况、运动员参加的项情况和该项目的成绩，所以要用到ATHLETE、ITEM和GAMES这3个表，因此FROM子句后为ATHLETE、GAMES、ITEM，3个表可以是任意次序，不影响结果。

试题四

- 6、(1) Index=NewElemKey % P (2) i<ITEMS
 (3) front=&Bucket[Index] (4) k==ITEMS
 (5) t==NULL, 或!t (6) front->Link=s

[解析]

本题考查元素的散列存储。

元素作散列存储时，首先用设定的散列函数计算元素的存储位置。在本题中，将元素存储在预先设定的基桶或根据需要申请的溢出桶中，只要基桶中有空闲单元，就将新元素NewElemKey插入在基桶中，若基桶中无空闲单元，则看是否存在溢出桶，若存在，则在溢出桶中查找空闲单元，若不存在溢出桶或溢出桶中无空闲单元，则申请一个溢出桶并存入新元素。

在基桶查找空闲单元时使用的桶号为Index，可知空(1)处应填入“Index= NewElemKey % P”。显然，一旦在基桶中找到空闲单元，即“Bucket[Index].KeyData[i]== NULLKEY”(0≤i<ITEMS)，则可将元素NewElemKey放入Bucket[Index].KeyData[i]，至此元素已经插入散列桶中，函数可返回，因此空(2)处应填入“i<ITEMS”；反之，若在基桶中没有找到空闲单元，则需查找溢出桶。“t=Bucket[Index].Link”，指针t首先指向桶号Index的第一个溢出桶。下面的代码即为在溢出桶中查找空闲单元。

```
if (t!=NULL) { /*有溢出桶*/
    while (t!=NULL) {
```

```

for(k=0; k<ITEMS;k++)
    if(t->KeyData[k]==NULLKEY) { /*在溢出桶链表中找到空闲单元*/
        t->KeyData[k]=NewElemKey;    break;
    } /*if*/
front=t;
    if(__ (4) __ ) t=t->Link;
    else break;
} /*while*/
} /*if*/

```

由于每个溢出桶都可以存储ITEMS个元素，所以在溢出桶中查找空闲单元与在基桶中的查找过程相同，代码如下。

```

for(k=0;k<ITEMS;k++)
    if(t->KcyData[k]==NULLKEY) { /*在溢出桶链表中找到空闲单元*/
        t->KeyData[k]=NewElemKey;    break;
    } /*if*/

```

若在指针t指向的溢出桶中找到空闲单元则插入元素，否则，由“t=t->Link”得到下一个溢出桶的指针，因此“k<ITEMS”可作为是否在当前溢出桶中找到空闲单元的判定条件。

显然，在桶号Index的基桶和其所有溢出桶都已满的情况下，t的值为空指针。此时才需要申请新的溢出桶并建立链接关系，因此在上面查找溢出桶中空闲单元时，进行指针t的后移“t=t->Link”前应先用front记录t的值，以便于后面建立链接关系。所以空(3)处应给front置初值，即“front=&Bucket[Index]”，空(4)填入“k==ITEMS”，空(5)填入“t=NULL”。空(6)处建立新申请溢出桶的链接关系“front->Link=s”。

试题五

7、(1) Point center

(2) new Point(xValue, yValue)

(3) this(xValue, yValue, r)

(4) extends Ball

(5) super(xValue, yValue, r, c)

[解析] 在类Ball的有参数构造函数中，对成员变量center通过调用Point类的构造方法初始化，而center在类Ball中尚未声明。结合注释可得空(1)是将center变量声明为Point对象引用，故空(1)应填Point。空(2)是调用Point类的构造函数，根据题意，此处应将xValue和yValue作为参数调用类Point的有参数构造函数，故空(2)应填new Point(xValue, yValue)。

根据注释，空(3)是调用类Ball的有3个参数的构造方法，而其所在方法本身就是类Ball的一个构造方法，因此可用this来调用自身的构造方法，故空(3)应填this(xValue, yValue, r)。

根据题述“在MovingBall类的toString方法中，super.toString调用父类Ball的toString方法输出类Ball中声明的属性值”，可知类MovingBall是类Ball的子类，因此空(4)应填extends Ball。

根据注释，空(5)是调用父类Ball中具有4个参数的构造方法，通过super关键字实现，故空(5)应填super(xValue, yValue, r, c)。