

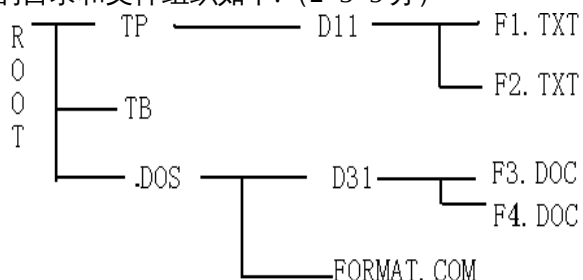
**第二届全国青少年信息学（计算机）奥林匹克分区联赛初赛试题**  
**（高中组）**

**（PASCAL 语言 竞赛用时：2 小时）**

●●全部试题答案均要求写在答卷纸上，写在试卷纸上一律无效●●

一、基础知识部分：（39 分）

1. 已知 A 盘上的目录和文件组织如下：（2+3=5 分）



其中 TP、TB、DOS、D11、D31 都是子目录名。

设当前命令提示符为 A:\TB>,请写出完成如下操作的 DOS 命令：

① 在 DOS 运行中，没有执行过 PATH 命令，现要用 DOS 子目录中的 FORMAT 命令，对插入在 B 驱动器（5.25 英寸高密）中的 360KB 软盘进行格式化工作,请写出相应的操作命令。

② 交换 F2.TXT 与 F3.DOC 两个文件的内容。

2. 请用等号或不等号联接表示下列不同进位制数值的大小。（3 分）

例如：(3)<sub>10</sub> < (4)<sub>10</sub> = (100)<sub>2</sub> < (A)<sub>16</sub>

其中圆括号外右下角的下标，表示圆括号内数的进位制。

(98.375)<sub>10</sub> (142.3)<sub>8</sub> (58.5)<sub>16</sub> (1011000.0101)<sub>2</sub>

3. 阅读下列程序段，写出程序运行后数组元素 A1，A2，…，A11 中的值。（6 分）

```
A[1] := 1; A[2] := 1; K := 1;
REPEAT
  A[K+2] := 1;
  FOR I := K+1 DOWNT0 2 DO
    A[I] := A[I] + A[I-1];
  K := K+1;
UNTIL K >= 10;
```

4. 已知：ACK (M, N) 函数的计算公式如下：（4%）

$$\text{ACK}(M, N) = \begin{cases} N+1 & M=0 \\ \text{ACK}(M-1, 1) & N=0 \\ \text{ACK}(M-1, \text{ACK}(M, N-1)) & M \neq 0 \text{ 且 } N \neq 0 \end{cases}$$

请计算：ACK (1, 3)、ACK (2, 4)、ACK (3, 3)、ACK (3, 4)

5. 有  $N \times N$  个数据组成如下方阵：（5分）

$$A_{11} \begin{pmatrix} A_{12} & A_{13} & \dots & A_{1N} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2N} \\ A_{31} & A_{32} & A_{33} & \dots & A_{3N} \\ \dots & \dots & \dots & \dots & \dots \\ A_{N1} & A_{N2} & A_{N3} & \dots & A_{NN} \end{pmatrix}$$

并已知： $A_{ij} = A_{ji}$

现将  $A_{11}$  ,  $A_{21}$  ,  $A_{22}$  ,  $A_{31}$  ,  $A_{32}$  ,  $A_{33}$  ,  $\dots$  存储在一维数组  $A[1]$  ,  $A[2]$  ,  $\dots$  ,  $A[(N \times (N+1)) / 2]$  中。

试问：任给  $i$  ,  $j$  怎样求出  $K$  来, 使得  $A[K]$  的值正好是  $A_{ij}$ , 请写出由  $i$  ,  $j$  计算  $K$  值的表达式。

6. 已知： $A_1, A_2, \dots, A_{81}$  共有 81 个数, 其中只有一个数比其它数大, 要用最少的比较运算次数, 把这个值大的数找出来 (假设两个数比较一次能决定出大于、小于或等于这三种情况) 请将以下算法补充完整：（9分）

第一步： $S1 = A_1 + A_2 + \dots + A_{27}$

$S2 = A_{28} + A_{29} + \dots + A_{54}$

第一次比较 ( $S1, S2$ ) :

$S1 > S2$  取  $K=0$

$S1 < S2$  取  $K=27$

$S1 = S2$  取  $K=54$

第二步： $S1 = A_{K+1} + A_{K+2} + \dots + A_{K+9}$

$S2 = A_{K+10} + A_{K+11} + \dots + A_{K+18}$

第二次比较 ( $S1, S2$ ) :

$S1 > S2$  取  $K=$ \_\_

$S1 < S2$  取  $K=$ \_\_

$S1 = S2$  取  $K=$ \_\_

第三步： $S1 = A_{K+1} + A_{K+2} + A_{K+3}$

$S2 = A_{K+4} + A_{K+5} + A_{K+6}$

第三次比较 ( $S1, S2$ ) :

$S1 > S2$  取  $K=$ \_\_

$S1 < S2$  取  $K=$ \_\_

$S1 = S2$  取  $K=$ \_\_

第四步： $S1 = A_{K+1}$

$S2 = A_{K+2}$

第四次比较 ( $S1, S2$ ) :

$S1 > S2$  \_\_\_\_为最大数

$S1 < S2$  \_\_\_\_为最大数,

$S1 = S2$  \_\_\_\_为最大数。

7. 下面是一个利用完全二叉树特性, 用顺序表来存储的一棵二叉树, 结点数据为字符型 (结点层次号从小到大, 同一层从左到右顺序存储, #表示空结点, @表示存储数据结

束)。

现要求画出对应该存储结构的二叉树示意图。(7分)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

A	B	C	#	#	D	E	#	#	#	#	#	G	F	@
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

二、根据题目要求，完善程序：(61分)

1. [题 目] 21分 (3+4+3+3+4+4)

积木游戏：设有  $n$  个小木块排成一排，如下图：

□□□……□

游戏开始时，每个小木块向下的一面涂有红、黄、蓝三种颜色之中的一种（约定：0表示红色，1表示黄色，2表示蓝色）。要求通过翻看与交换方式对小木块重新排列（翻看的规则为每个小木块只能看一次），最终成为下面的形状：

□□□……□	□□□……□	□□□……□
红	蓝	黄

即相同颜色的木块排列在一起，设计一个翻看与交换的方案，使得用最少的交换次数实现上面的要求。

**[算法描述]** 翻看小木块时，可以从两端进行。例如，设中间状态如下：

□□□……□	A□□……B	□□□……C	□□□……□
红	未翻过	蓝	黄

此时，可以从两个方向看，即从 A 或 B 处开始：

(1) 若看 A 则有三种可能性：

为红色，则不用交换

为蓝色，交换一次，即 A 与 B 交换

为黄色，交换两次，即 C 与 B 交换一次，然后 A 与 C 再交换一次

此时，平均交换次数为 1。

(2) 若看 B，也有三种可能性：

为蓝色，则不用交换

为红色，交换一次，即 B 与 A 交换。

为黄色，交换一次，即 B 与 C 交换。

此时，平均交换次数为  $2/3$ 。

由此可见，从 B 处翻看直到游戏结束，次数最少符合题目要求。

**[程 序]**

```
PROGRAM EXP1(INPUT,OUTPUT)
CONST N=20;
VAR I,TEM,R,B,Y:INTEGER;
A:ARRAY[1..N] OF 0..2;
BEGIN
FOR I:=1 TO N DO READ(A[I]); R:=1; ①; Y:=N;
WHILE ② DO
IF ③ THEN BEGIN
TEM:=A[R]; A[R]:=A[B];
A[B]:=TEM; R:=R+1
END
```

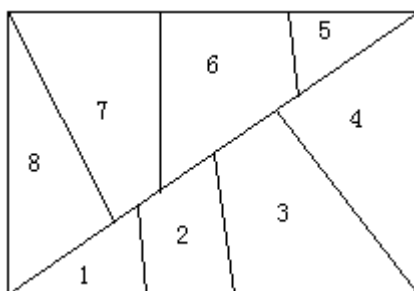
```

ELSE IF ④ THEN BEGIN
    TEM:=A[B]; A[B]:=A[Y];
    A[Y]:=TEM; ⑤; ⑥;
END
ELSE B:=B-1
FOR I:=1 TO N DO WRITE(A[I]:3)
END.

```

2. [题 目] (20 分, 每空 4 分)

4 色问题。设有下列形状的图形：(N=8)，其编号为 1, 2, …… , N。



图形之间的相邻关系用下面的邻接矩阵表示：

	1	2	3	4	5	6	7	8
1	0	1	0	0	0	0	1	1
2	1	0	1	0	0	1	1	0
3	0	1	0	1	0	1	0	0
4	0	0	1	0	1	1	0	0
5	0	0	0	1	0	1	0	0
6	0	1	1	1	1	0	1	0
7	1	1	0	0	0	1	0	1
8	1	0	0	0	0	0	1	0

其中：1——相邻，0——不相邻。

**[程序要求]** 将上面图形的每一个部分涂上红（1），黄（2），蓝（3），绿（4）四种颜色之一，要求相邻的部分有不同颜色。

输入方式：邻接矩阵。

输出方式：区域、颜色。

.....

**[算法描述]** 用数组 R:ARRAY[1..N,1..N] OF 0..1 表示相邻关系，S:ARRAY[1..N] OF INTEGER 表示颜色。

采用回溯的方法，首先给第一个图形涂上红色（1），然后在下面的图形中依次涂上其他颜色，当有矛盾时回溯解决。

**[程 序]**

```

PROGRAM EXP2(INPUT,OUTPUT);
CONST N=8;

```

```

VAR I,J,K:INTEGER;
    R:ARRAY[1..N,1..N] OF 0..1;
    S:ARRAY[1..N] OF INTEGER;
BEGIN
    FOR I:=1 TO N DO
        BEGIN
            FOR J:=1 TO N DO READ(R[I,J]); READLN
        END;
        ① ; I:=2; J:=1;
    WHILE I<=N DO
        BEGIN
            WHILE (J<=4) AND (I<=N) DO
                BEGIN
                    K:=1;
                    WHILE ② DO
                        K:=K+1;
                        IF K<I THEN ③
                            ELSE BEGIN
                                ④ ; I:=I+1; J:=1
                            END
                        END;
                    END;
                IF J>4 THEN BEGIN
                    I:=I-1; ⑤
                END;
            END;
        FOR I:=1 TO N DO WRITELN(I,'→',S[I])
    END.

```

### 3 . [题 目] (20 分, 每空 4 分)

多项式加法运算：一个仅含有  $x$  的多项式可以用下列的方式表示：

(系数, 指数), (系数, 指数),  $\dots$ , (0, 0)。

其中 (0, 0) 作为结束标志。

例如： $P(x)=4x^6-3x^3+2x^2-1$

可表示为：(4,6),(-3,3),(2,2),(-1,0),(0,0)

$Q(x)=x^4-x+1$

可表示为：(1,4),(-1,1),(1,0),(0,0)

当用上面的方式给出 2 个多项式之后，编制程序对这两个多项式进行加法运算，结果也用上面的方式给出。

例如：上面的  $P(x)$  和  $Q(x)$  相加的结果为：

$4x^6+x^4-3x^3+2x^2-x$

表示结果为：(4,6),(1,4),(-3,3),(2,2),(-1,1),(0,0)

**[算法描述]** 多项式可用数组  $P$  表示；分别以  $p1$  表示  $P$ ,  $p2$  表示  $Q$ ,  $p3$  表示结果。

处理的过程为将  $P$  复制到  $p3$ , 然后逐项检查  $Q$ , 当发现有相同的方次时, 进行系数相加；当发现没有相同方次时, 插入到  $p3$  中去。

## [程 序]

```
PROGRAM EXP3(INPUT,OUTPUT)
VAR
  X,Y,I,I1,J,J1,J2:INTEGER;
  P1,P2,P3   :ARRAY[1..20,1..2] OF INTEGER
BEGIN
  J1:=0; WRITE('INPUT P(X)='); READ(X,Y);
  WHILE X<>0 DO
    BEGIN
      J1:=J1+1; P1[J1,1]:=X; P1[J1,2]:=Y; READ(X,Y)
    END;
    J1:=J1+1; P1[J1,1]:=0; P1[J1,2]:=0;
    WRITE('INPUT Q(X)='); READ(X,Y); J2:=0;
    WHILE X<>0 DO
      BEGIN
        J2:=J2+1; P2[J2,1]:=X; P2[J2,2]:=Y; READ(X,Y)
      END;
      J2:=J2+1; P2[J2,1]:=0; P2[J2,2]:=0;
    FOR I:=1 TO J1 DO
      BEGIN
        P3[I,1]:=P1[I,1]; P3[I,2]:=P1[I,2]
      END;
      I:=1;
      WHILE ① DO
        BEGIN
          IF ② THEN
            BEGIN
              FOR J:=J1 DOWN TO 1 DO
                BEGIN
                  P3[J+1,1]:=P3[J,1]; P3[J+1,2]:=P3[J,2]
                END;
                P3[I,1]:=P2[I,1]; P3[I,2]:=P2[I,2]; J1:=J1+1
              END
            ELSE BEGIN
              I1:=1;
              WHILE P2[I,2]<P3[I1,2] DO
                ③;
                IF P2[I,2]=P3[I1,2] THEN
                  P3[I1,1]:=④
                ELSE BEGIN
                  FOR J:=J1 DOWNT0 I1 DO
                    BEGIN
                      P3[J+1,1]:=P3[J,1]; P3[J+1,2]:=P3[J,2]
                    END;

```

```

        P3[I1,1]:=P2[I,1]; P3[I1,2]:=P2[I,2];
        ⑤
    END;
END ;
I :=I+1
END ;
FOR J:=1 TO J1-2 DO WRITE ('(,P3[J,1],',',P3[J,2],'),');
    WRITELN('(',P3[J+1,1],',',P3[J+1,2],')');
END.

```