

## 第七届分区联赛提高组初赛

(提高组 PASCAL 语言 二小时完成)

一、选择一个正确答案代码(A/B/C/D)，填入每题的括号内(每题 1.5 分，多选无分，共 30 分)

- 1、中央处理器 CPU 能访问的最大存储器容量取决于( )  
A)地址总线 B)数据总线 C)控制总线 D)内存容量
- 2、计算机软件保护法是用来保护软件( )的。  
A)编写权 B)复制权 C)使用权 D)著作权
- 3、64KB 的存储器用十六进制表示，它的最大的地址码是( )  
A)10000 B)FFFF C)1FFFF D)EFFFF
- 4、在树型目录结构中，不允许两个文件名相同主要指的是( )  
A)同一个磁盘的不同目录下 B)不同磁盘的同一个目录下  
C)不同磁盘的不同目录下 C)同一个磁盘的同一个目录下
- 5、下列设备哪一项不是计算机输入设备( )  
A)鼠标 B)扫描仪 C)数字化仪 D)绘图仪
- 6、在计算机硬件系统中，cache 是( )存储器  
A)只读 B)可编程只读 C)可擦除可编程只读 D)高速缓冲
- 7、若我们说一个微机的 CPU 是用的 PII300，此处的 300 确切指的是( )  
A)CPU 的主时钟频率 B)CPU 产品的系列号  
C)每秒执行 300 百万条指令 D)此种 CPU 允许最大内存容量
- 8、Email 邮件本质上是一个( )  
A)文件 B)电报 C)电话 D)传真
- 9、2KB 的内存能存储( )个汉字的机内码  
A)1024 B)516 C)2048 D)218
- 10、以下对 Windows 的叙述中，正确的是( )  
A)从软盘上删除的文件和文件夹，不送到回收站  
B)在同一个文件夹中，可以创建两个同类、同名的文件  
C)删除了某个应用程序的快捷方式，将删除该应用程序对应的文件  
D)不能打开两个写字板应用程序
- 11、运算式 $(2047)_{10} - (3FF)_{16} + (2000)_8$ 的结果是( )  
A) $(2048)_{10}$  B) $(2049)_{10}$  C) $(3746)_8$  D) $(1AF7)_{16}$
- 12、TCP/IP 协议共有( )层协议  
A)3 B)4 C)5 D)6
- 13.若已知一个栈的入栈顺序是 1, 2, 3, ..., n, 其输出序列为 P1, P2, P3, ..., Pn, 若 P1 是 n, 则 Pi 是( )  
A)i B)n-1 C)n-i+1 D)不确定
- 14.计算机病毒是( )  
A)通过计算机传播的危害人体健康的一种病毒  
B)人为制造的能够侵入计算机系统并给计算机带来故障的程序或指令集合  
C)一种由于计算机元器件老化而产生的对生态环境有害的物质  
D)利用计算机的海量高速运算能力而研制出来的用于疾病预防的新型病毒
- 15.下面关于算法的错误说法是( )  
A)算法必须有输出 B)算法必须在计算机上用某种语言实现

- C)算法不一定有输入 D)算法必须在有限步执行后能结束
- 16.[x]补码=10011000，其原码为( )  
A)01100111 B)11101000 C)11100110 D)01100101
- 17.以下哪一个不是栈的基本运算( )  
A)删除栈顶元素 B)删除栈底的元素  
C)判断栈是否为空 D)将栈置为空栈
- 18.在顺序表(2, 5, 7, 10, 14, 15, 18, 23, 35, 41, 52)中，用二分法查找 12，所需的  
关键码比较的次数为( )  
A)2 B)3 C)4 D)5
- 19.一棵二叉树的高度为 h，所有结点的度为 0，或为 2，则此树最少有( )个结点  
A) $2^h-1$  B) $2h-1$  C) $2h+1$  D) $h+1$
- 20.无向图  $G=(V, E)$ ，其中  $V=\{a,b,c,d,e,f\}$   $E=\{(a,b),(a,e),(a,c),(b,e),(c,f),(f,d),(e,d)\}$ ，对该图  
进行深度优先遍历,得到的顶点序列正确的是( )  
A)a,b,e,c,d,f B)a,c,f,e,b,d C)a,e,b,c,f,d D)a,b,e,d,f,c

## 二、问题求解(5+7=12 分)

- 1.已知一棵二叉树的结点名为大写英文字母，其中序与后序遍历的顺序分别为：CBGEAFHDIJ 与 CGEBHFJIDA 则该二叉树的先序遍历的顺序为：
- 2.平面上有三条平行直线，每条直线上分别有 7，5，6 个点，且不同直线上三个点都不在同一条直线上。问用这些点为顶点，能组成多少个不同四边形？

## 三、阅读程序，写出程序正确的运行结果(4+7+8+9=28 分)

### 1.PROGRAM GAO7\_1 :

```
FUNCTION ACK(M, N : INTEGER) : INTEGER ;
BEGIN
  IF M=0 THEN ACK := N+1
    ELSE IF N=0 THEN ACK:=ACK(M-1, 1)
      ELSE ACK:=ACK(M-1, ACK(M, N-1))
END ;
BEGIN WRITELN(ACK(3, 4)) ; READLN ; END.
```

输出

### 2.PROGRAM GAO7\_2 ;

```
VAR P, Q, S, T : INTEGER ;
BEGIN
  READLN(P) ;
  FOR Q := P+1 TO 2*P DO
  BEGIN
    T:=0 ; S:=(P*Q)MOD(Q-P) ;
    IF S=0 THEN BEGIN T:=P+Q+(P*Q)DIV(Q-P) ; WRITE(T : 4) ; END ;
  END ;
END.
```

输入 12      输出

### 3.PROGRAM GAO7\_3 ;

```

VAR I , J , H , M , N , K : INTEGER ;
    B  : ARRAY[1..10]OF INTEGER ;
BEGIN
    READLN(N) ;
    FOR I:=1 TO 10 DO
        BEGIN
            M :=N ; J := 11 ;
            WHILE M>0 DO
                BEGIN J:=J-1 ; B[J] :=M MOD 10 ; M:=M DIV 10      END ;
            FOR H:=J TO 10 DO    N:=N+B[H] ;
            END ;
            WRITELN(N) ;
        END.
    输入 1234      输出 :

```

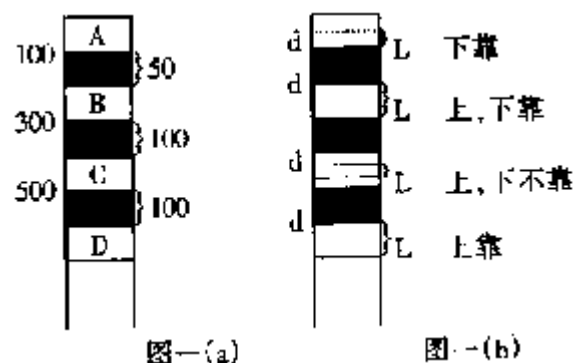
```

4.PROGRAM GAO7_4 ;
VAR X , Y1 , Y2 , Y3 : INTEGER ;
BEGIN
    READLN(X) ; Y1 :=0 ; Y2 := 1 ; Y3 := 1 ;
    WHILE Y2<=X DO
        BEGIN
            Y1 :=Y1+1 ; Y3 :=Y3+2 ; Y2 :=Y2+Y3
        END ;
        WRITELN(Y1) ;
    END.
    输入 : 23420      输出 :

```

#### 四、完善程序(每空 3 分，共 30 分)

1.存储空间的回收算法。设在内存中已经存放了若干个作业 A , B , C , D。其余的空间为可用的(如图一中(a))。



此时，可用空间可用一个二维数组  $dk[1..100, 1..2]$  表示，(如下表一中(a))，其中： $dk[i, 1]$  对应第  $i$  个可用空间首址， $dk[i, 2]$  对应第  $i$  个可用空间长度如上图， $dk$ ：

100	50
300	100
50	100

表一(a)

0	0
100	50
300	100
500	100
10000	0

表一(b)

现某个作业释放一个区域，其首址为  $d$ ，长度为  $L$ ，此时将释放区域加入到可用空间表中。要求在加入时，若可用空间相邻时，则必须进行合并。因此出现下面的 4 种情况(如上图一(b)所示)。

- (1)下靠，即回收区域和下面可用空间相邻，例如， $d=80$ ， $L=20$ ，此时成为表二中的(a)。
- (2)上靠，例如， $d=600$ ， $L=50$ ，此时表成为表二中的(b)。
- (3)上、下靠，例如， $d=150$ ， $L=150$ ，此时表成为表二中的(c)。
- (4)上、下不靠，例如， $d=430$ ， $L=20$ ，此时表成为表二中的(d)。

80

70	
300	100
50	100

表二(a)(下靠)

100	50
300	100
500	150

表二(b)(上靠)

100	300
500	100

表二(c)(上，下靠)

100	50
300	100
430	20
500	100

表二(d)(上，下不靠)

程序说明：对数组  $dk$  预置 2 个标志，即头和尾标志，成为表二中的(b)，这样可使算法简单， $sp$  为  $dk$  表末地址。

程序清单：

```

PROGRAM GAO7_5 ;
  VAR  I , J , SP , D , L : INTEGER ;
        DK   : ARRAY[0..100 , 1..2]OF INTEGER ;
  BEGIN
    READLN(SP) ;
    FOR I:=1 TO SP DO
      READLN(DK[I , 1] , DK[I , 2]) ;
      DK[0 , 1]:=0 ; DK[0 , 2]:=0 ; ① ;
      DK[SP , 1]:=10000 ; DK[SP , 2]:=0 ; READLN(D , L) ; I:=1 ;
    WHILE DK[I , 1]<D DO I:=I+1 ; ② ;
    IF(DK[I , 1]+DK[I , 2]=D)THEN
      IF(D+L=DK[I+1 , 1])THEN
        BEGIN
          DK[I , 2]:= ③ ;
          FOR J:=I+1 TO SP-1 DO
            DK[J]:=DK[J+1] ;
        
```

```

        SP := SP-1 ;
        END
    ELSE DK[I, 2]:=DK[I, 2]+L
ELSE IF(D+L=DK[I+1, 1])THEN
    BEGIN
        DK[I+1, 1]:= ④ ; DK[I+1, 2]:=DK[I+1, 2]+L
    END

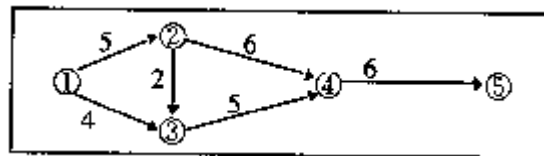
    ELSE BEGIN
        FOR J:=SP DOWNT0 I+1 DO    DK[J+1]:=DK[J] ;
        ⑤:=D ;    DK[I+1, 2]:=L ; SP:=SP+1 ;
    END ;
    FOR I:=1 TO SP-1 DO    WRITELN(DK[I, 1]:4, DK[I, 2]:4) ; READLN ;
END.

```

## 2.求关键路径

设有一个工程网络如下图表示(无环路的有向图)：

其中，顶点表示活动，①表示工程开始，⑤表示工程结束(可变，用 N 表示)，边上的数字表示活动延续的时间。



如上图，活动①开始 5 天后活动②才能开始工作，而活动③则要等①、②完成之后才能开始，即最早也要 7 天后才能工作。

在工程网络中，延续时间最长的路径称为关键路径。上图中的关键路径为：①—②—③—④—⑤共 18 天完成。

关键路径的算法如下：

### 1.数据结构：

$R[1..N, 1..N]$  OF INTEGER ;      表示活动的延续时间，若无连线，则用-1 表示；

$EET[1..N]$       表示活动最早可以开始的时间

$ET[1..N]$       表示活动最迟应该开始的时间

关键路径通过点 J，具有如下的性质： $EET[J]=ET[J]$

### 2.约定：

结点的排列已经过拓扑排序，即序号前面的结点会影响序号后面结点的活动。

程序清单：

```

PROGRAM GAO7_6 ;
VAR I, J, N, MAX, MIN, W, X, Y : INTEGER ;
    R : ARRAY[1..20, 1..20] OF INTEGER ;
    EET, ET : ARRAY[1..20] OF INTEGER ;
BEGIN
    READLN(N)
    FOR I:=1 TO N DO
        FOR J:=1 TO N DO

```

```

    R[I , J]:=-1 ;
    READLN(X , Y , W) ; {输入从活动 X 到活动 Y 的延续时间，以 0 为结束}
WHILE X<>0 DO
    BEGIN
        R[X , Y]:=W ; ①
    END ;
    EET[1] : =0 ; {认为工程从 0 天开始}
    FOR I : =2 TO N DO
        BEGIN
            MAX:=0 ;
            FOR J:=1 TO N DO
                IF R[J , I]<>-1 THEN
                    IF ② THEN MAX:=R[J , I]+EET[J] ;
            EET[I]:=MAX ;
        END ;
        ③
    FOR I : =N-1 DOWNT0 1 DO
        BEGIN
            MIN : =10000 ;
            FOR J : =1 TO N DO
                IF R[I , J]<>-1 THEN
                    IF ④ THEN MIN : =ET[J] - R[I , J] ;
            ET[I] : =MIN ;
        END ;
        WRITELN(EET[N]) ;
        FOR I : =1 TO N-1 DO
            IF ⑤ THEN WRITE(I , '→') ;
        WRITE(N) ; READLN
    END.

```