

NOI'95 “同创杯”全国青少年信息学（计算机）奥林匹克竞赛

分区联赛初赛试题（初中组） 竞赛用时：2 小时

答 题 要 求

一、全部试题答案均应写在答卷纸上，写在试卷纸上一概无效。

二、算法描述中，可以使用下列过程、函数或算符：

(1) 算术运算：+，-，×，÷，DIV，MOD

整数除（DIV）：是取二整数相除的商的整数部分。如：11 DIV 2 = 5

取模（MOD）：是取二整数相除的余数。如：11 MOD 2 = 1

(2) 关系运算：>，<，=，<>，>=，<=

(3) 逻辑运算：AND，OR，NOT

(4) 函数：

ABS(X)：求 X 的绝对值。如：ABS (3.14) =3.14 ABS(-3.14)=3.14

SQR(X)：求 X 的平方值。如：SQR (3) =9 SQR (-15) =225

SQRT(X)：求 X 的平方根值。如：SQRT(9)=3 SQRT(225)=15

TRUNC(X)：去掉 X 的小数部分：如 TRUNC(6.3)=6 TRUNC(-7.9)=-7

ROUND(X)：函数值是小数四舍五入后的整数值。

如：ROUND(3.14)=3 ROUND(3.16)=4 ROUND(-3.14)=-4

ORD(X)：函数值是字符在 ASCII 码中的序号。

如：ORD('A')=65 ORD('B')=66 ORD('Z')=90 ORD('0')=48

CHR(X)：X 表示 ASCII 码中的序号，函数值是该序号代表的字符值。

如：CHR(48)='0' CHR(65)='A' CHR(90)='Z'

(5) 过程：

DEC(A,[X])：变量递减，A 为有序变量，X 缺省时为 1。

INC(A,[X])：变量递增，A 为有序变量，X 缺省时为 1。

一、基础题：

<1> 执行① C>DIR 命令后，屏幕上显示如下画面：

```
FORMAT COM 12145
```

```
SYS COM 4878
```

```
PUC BAT 126
```

```
XCOPY EXE 11216
```

```
4 FILE (S) 123456 bytes free
```

接着又顺序执行了如下几条 DOS 命令：

② C>DIR> DF.TXT //表示将列表显示的目录作为文件写盘 //

```
1 C>TYPE DF.TXT
```

```
2 C>DIR
```

试问:执行命令③和④ 在屏幕上显示的结果是否与①相同?

<2> 请将以下程序段表示的计算公式写出来（假设 X 的值已给出）

```
E := 1 ;  
A := 1 ;  
  
FOR N := 1 TO 10 DO  
    A := A * X / N ;  
    E := E + A ;  
ENDFOR ;
```

写出所表示的公式。

<3> 列举一个算法，使算法的解能对应相应的问题。

例如，设问题为：学生答题，答对一题可得 10 分，答错一题则要扣去 5 分，输入答对的题数 (M) 与答错的题数 (N)，求最后得分 (S) 是多少？

列举出相应算法为：

```
X := 10 ;  
Y := 5 ;  
READ (M, N) ;  
S := X * M - Y * N ;
```

现有以下问题：用五角钱换成 5 分、2 分与 1 分的硬币，可有多少种换法？

请列出该问题的算法。

<4> 已知如下 $N*(N+1)/2$ 个数据，按行的顺序存入数组 A[1]，A[2]，……中：

```
a11  
a21  a22  
a31  a32  a33  
.....  
an1  an2  an3  .....  ann
```

其中：第一个下标表示行 第二个下标表示列。

若： $a_{ij}(i \geq j, i=1, 2, \dots, n)$ 存贮在 A[k] 中，试问：

- (1) k 和 i, j 之间的关系如何表示？
- (2) 给定 k 值 ($k \leq n*(n+1)/2$) 后，写出能决定相应的 i, j 值的算法。

<5> 有红、黄、黑、白四色球各一个，放置在一个内存编号为 1、2、3、4 四个格子的盒中，每个格子放置一只球，它们的顺序不知。甲、乙、丙三人猜测放置顺序如下：

甲：黑编号 1，黄编号 2；

乙：黑编号 2，白编号 3；

丙：红编号 2，白编号 4。

结果证明甲乙丙三人各猜中了一半。

写出四色球在盒子中放置情况及推理过程。

二、根据题目要求，补充完善以下伪代码程序：

<1> 求出所有满足下列条件的二位数：将此二位数的个位数字与十位数字进行交换，可得到一个新的数，要求新数与原数之和小于 100。

程序要求：每行输出 6 个满足条件的数。

【算法提要】 分解每一个二位数，然后重新组成一个新数，当满足条件时，用计数器来统计个数。

```
程序：      K := 0;
            FOR i := _____①_____ TO 99 DO
              X := _____②_____; Y := _____③_____;
              J := x * 10 + y;
              IF _____④_____ THEN
                K := k + 1;
                Write(I : 4);
                _____⑤_____ THEN WRITELN;
              ENDIF
            ENDFOR;
```

<2> 找出小于 33 的 6 个正整数，用这些整数进行加法运算，使得包括原来的整数在内能组成尽可能多的不同整数。

例如：用 2，3，5 这三个数能可组成下面的数

2, 3, 5

$2 + 3 = 5$, 但 5 已经存在

$2 + 5 = 7$, $3 + 5 = 8$, $2 + 3 + 5 = 10$

所以用 2，3，5 能组成 6 个不同的数。

程序要求：输出所选的这 6 个数，以及能组成不同整数的个数。

【算法提要】：选择的这 6 个数，用来组成数时应该尽可能不重复，引入数组 A 保存找出的这 6 个整数。

```
程序：      A[1] := 1; t := 0;
            For i := 2 to 6 do
              _____①_____;
              for j := 1 to i - 1 do
                s := _____②_____;
              ENDFOR;
              a[i] := _____③_____;
            ENDFOR;
            FOR i:=1 TO 6 DO
              T := _____④_____ WRITE(a[i], ' ');
            ENDFOR;
            Writeln('能组成不同整数的个数：', t)
            End.
```

<3> 求出 2~1000 之间长度最长的、成等差数列的素数（质数）。

例如：在 2~50 之间的全部素数有

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47

其中公差为 1 的素数数列为 2, 3，其长度为 2

公差为 2 的素数数列为 3, 5, 7，其长度为 3

.....

程序要求：输出满足条件的素数数列。

[算法提要]：首先用筛选法求出此范围内的全部素数，存放在数组 B 中，然后用 2 个变量 i,j，逐步求出满足条件的素数数列。

```
程序:  Max := 0; num := 1000;
      For i := 2 to num do
        b[i] := i;
      ENDFOR
      For i := 2 to _____①_____ do
        If _____②_____ then
          k := i + i;
          While k <= num do
            B[k] := 0;
            K := k + i
          ENDWHILE;
        ENDIF;
      ENDFOR;
      For i := 2 to num - 1 do
        If _____③_____ then
          J := 1;
          D[j] := b[i];
          For i1 := _____④_____ do
            If b[i1] <> 0 then
              Delta := _____⑤_____;
              k := delta;
              While (i+k<=num)and _____⑥_____ do
                j := j + 1;
                d[j] := i + k;
                k := k + delta
              ENDWHILE;
              If j > max then begin
                Max := j;
                C := d //数组 D 的每个元素值
                           分别送放数组 C 的相应
                           元素中去//
              ENDIF;
              J := 1
            ENDIF;
          ENDFOR;
        ENDIF;
      ENDFOR;
      writeln('The max length is: ', max);
      write('The string is: ');
      for i := 1 to max do write(c[i], ' ');
      writeln
    ENDFOR;
```

<4> 求出二个整形数组错位相加的最大面积。

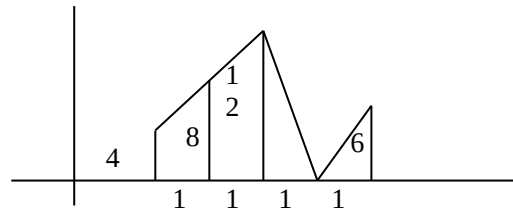
1. 数组面积的定义：(限定数组头尾不为0)

设有一个数组 $C = (4, 8, 12, 0, 6)$

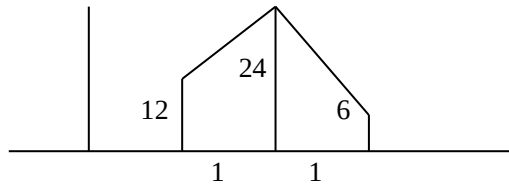
则 C 的面积为：

$$S_c = (4+8)/2 + (8+12)/2 + 12/2 + 6/2$$

也就是说， S_c = 各梯形面积之和 (其中梯形的高约定为 1，三角形作为梯形的特殊情况处理)。



又如 $D = (12, 24, 6)$ 时，其面积的定义为 $S_d = (12+24)/2 + (24+6)/2$



2. 数组错位相加的定义

设有 2 个正整数的数组 a, b ，长度为 n ，当 $n=5$ 时：

$a = (34, 26, 15, 44, 12)$ $b = (23, 46, 4, 0, 18)$

对 a, b 进行错位相加，可能有下列情况

$$\begin{array}{r} 34 \ 26 \ 15 \ 44 \ 12 \\ +) \qquad \qquad \qquad 23 \ 46 \ 4 \ 0 \ 18 \\ \hline 34 \ 26 \ 15 \ 44 \ 12 \ 23 \ 46 \ 4 \ 0 \ 18 \end{array}$$

或：

$$\begin{array}{r} 34 \ 26 \ 15 \ 44 \ 12 \\ +) \qquad \qquad \qquad 23 \ 46 \ 4 \ 0 \ 18 \ - \\ \hline 34 \ 26 \ 15 \ 44 \ 35 \ 46 \ 4 \ 0 \ 18 \end{array}$$

或：

$$\begin{array}{r} 34 \ 26 \ 15 \ 44 \ 12 \\ +) \qquad \qquad \qquad 23 \ 46 \ 4 \ 0 \ 18 \\ \hline 34 \ 26 \ 15 \ 67 \ 58 \ 4 \ 0 \ 18 \end{array}$$

或：……

最后有：

$$\begin{array}{r} \qquad \qquad \qquad 34 \ 26 \ 15 \ 44 \ 12 \\ +) \ 23 \ 46 \ 4 \ 0 \ 18 \qquad \qquad \qquad - \\ \hline 23 \ 46 \ 4 \ 0 \ 18 \ 34 \ 26 \ 15 \ 44 \ 12 \end{array}$$

可以看到：由于错位不同，相加的结果也不同。

程序要求：找出一个错位相加的方案，使得输出的数组面积为最大。

[算法提要]：设 a, b 的长度为 10，用 a, b : array[1..10] of integer 表示，其结果用数组 C, D : array[1..30] of integer 表示。

错位相加的过程可以从开始不重叠，然后逐步重叠，再到最后的不重叠。

梯形面积的计算公式为：(上底+下底)×高÷2

其中由于约定高为 1，故可写为(上底+下底)÷2。

```

程序：  n = 10;
.....  Function sea : real; {计算数组 C 面积}
Begin
  J1 := 1;
  While _____①_____ do
    j1 := j1 + 1;
  ENDWHILE;
  If j1 = 3 * n then sea := 0
    Else begin
      J2 := 3 * n;
      While _____②_____ do
        j2 := j2 - 1;
        If j1 = j2 then sea := 0
          Else begin
            J3 := c[j1] + c[j2];
            For j4 := j1 + 1 to j2 - 1 do
              INC(j3,c[j4]*2);
            ENDFOR;
            Sea := j3 / 2
          end
        ENDIF;
      End;
    End;

```

//主程序//

```

For i := 1 to n do read(a[i]); endfor;
For j := 1 to n do read(b[j]); endfor;
_____③_____;
for i := 1 to 2 * n + 1 do
  for j := 1 to 3 * n do _____④_____ endfor;
  for j := 1 to n do c[j + n] := a[j] endfor;
  for j := 1 to n do
    _____⑤_____;
  endfor;
  p := sea;
  if p > s then begin
    d := c;
    s := p
  end;
endif;
endfor;
for I := 1 to 3 * n do write(d[I], ' '); endfor;
write(s);
End. //主程序结束//

```