

第三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、基础部分：

<1> WPS 是属于_____类的软件;FOXBASE 是属于_____类的软件。用 FOXBASE 的命令：“CREATE GZB”，在磁盘中生成的是_____文件。

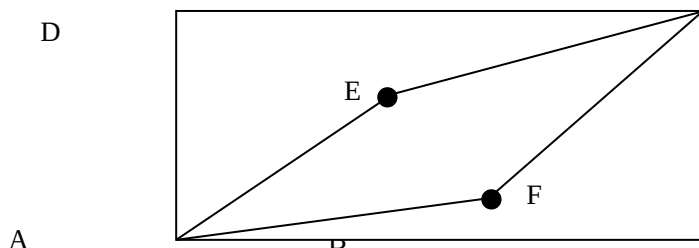
<2>在 MS DOS 的根目录中，有如下文件：TIME.EXE TIME.COM TIME.BAT
试问：C:\>TIME <回车> 执行的是什么命令？

<3> 已知 ASCII 码表中的大写字母后有 6 个其它字符，接着便是小写字母。现已知：A 字母的 ASCII 码为 $(41)_{16}$ { 表示 16 进制数 41 }，试写出如下字母用十进制表示的 ASCII 码：
 $G \rightarrow ()_{10}$ $B \rightarrow ()_{10}$ $T \rightarrow ()_{10}$

<4> 设数组 A[10..100,20..100] 以行优先的方式顺序存储，每个元素占 4 个字节，且已知 A[10, 20]的地址为 1000，则 A[50, 90]的地址是_____。

<5>一个汉字的机内码目前通常用 2 个字节来表示：第一个字节是区位码的区号加 $(160)_{10}$ ；第二个字节是区位码的位码加 $(160)_{10}$ 。
已知：汉字“却”的区位码是 4020，试写出机内码两个字节的二进制的代码：

<6> 下图中用点表示城市，点与点之间的联系表示城市间的道路：



试问：

- ① 能否找出一条从 A 城市出发，经过图中所有道路一次后又回到出发点的通路来？
 - ② 能否从 A 出发，找出去每个城市且只去一次的通路来？
- 若能，则写出通路，否则说明理由。

<7> 为了便于处理表达式，常常将普通表达式（称为中缀表示）转换为前缀 {运算符在前，如 X/Y 写为 /XY} 和后缀 {运算符在后，如 X/Y 写为 XY/} 的表达形式。
在这样的表示中可以不用括号即可确定求值的顺序，如：

$$(P+Q) * (R-S) \rightarrow *+PQ-RS \text{ 或 } \rightarrow PQ + RS -*$$

1 试将下面的表达式改写成前缀与后缀的表示形式：

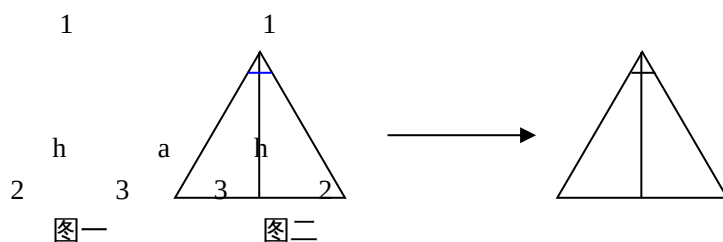
$$\langle A \rangle A+B*C/D \quad \langle B \rangle A-C*D+B \wedge E$$

② 试将下面的前缀表示还原成中缀的表示形式，同时写出后缀表示：

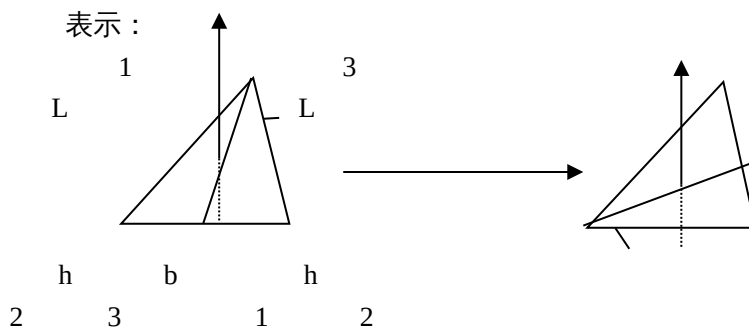
$$+\triangle A * B \triangle C \text{ \{前缀式中 } \triangle \text{ 表示一元运算符取负号，如 } \triangle A \text{ 表示 } (-A) \text{ \}}$$

<8> 一个将角编了号的正三角形可以进行如下二种运动：

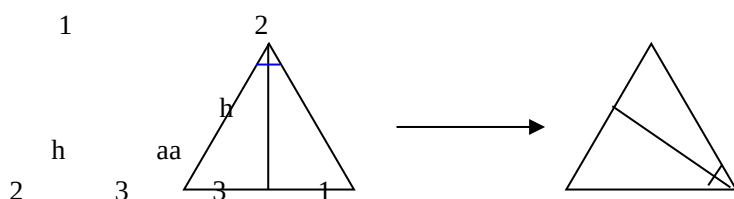
(a) 沿过顶点 1 的高 H 翻转 180° ，我们将这个运动用字母 a 来表示：



(b) 沿过三角形的外心，垂直于三角形所在平面的有向轴 L（注意：三角形翻转时 L 轴也随着翻转的），按右手法则旋转 120° （右手法则是用右手大拇指指向 L 轴的方向，由其余四指决定旋转方向的法则），我们将这样的运动用字母 b 来表示：

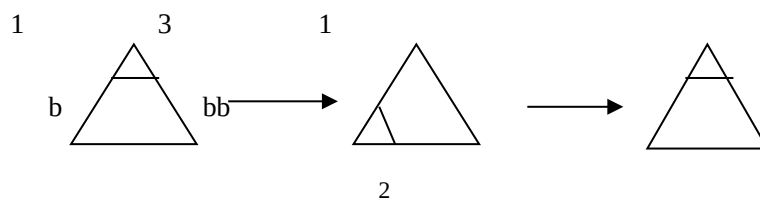


如果将 a,b 作为运算对象，并将两个运动连续进行看作是一种运算（这里不妨也称为乘法）则对图一的三角形而言，aa 的结果便成为：



若将运动前后的三角形状态简称为元素，那么三角形状态就可与运动的表达式关联。据此，请回答下列问题：

- 1 从图一的三角形的原始状态出发，可以运动出多少种不同状态的三角形，试写出最简单的运算表达式（借助于 a,b 与乘法运算）；
- 2 这样定义的乘法运算是否符合交换律与结合律？
- 3 如果将三角形的某种状态运动回到原始状态称之为该元素的逆元素，例如：



2 3 1 2 2 3

∴ bb 的逆元素为 b, 可以表示为 $(bb)^{-1} = b$

试求：(1) $a^{-1} =$ (2) $(ab)^{-1} =$ (3) $((aa)a)^{-1} =$ (4) $b^{-1} =$

二、根据题意,将以下程序补充完整

1. **[问题描述]** 一个正整数（非素数）可以表示成它的因子（1 与其本身除外）的乘积。

例如：12 有因子 2, 2, 3, 4, 6, 所以可表示为：12=2*2*3=4*3=2*6

给出任一个正整数 N，求出它所有的因子乘积的表达式（交换律得出的不同式子算同一种）。

[算法说明] 读入一个整数 N，首先求出它的所有的因子以及每个因子可能的次数。

例如：整数 48：

因子：2 3 4 6 8 12 16 24

次数：4 1 2 1 1 1 1 1

将上面的结果存入数组 A:ARRAY[0..20,1..2]中。其中：A[i,1]表示因子；A[i,2]表示次数。

然后用简单回溯的方法求出所有可能的表示。

数组 B[0..20]记录取数情况；C:ARRAY[1..20]工作单元。

[程序清单] program exp4(input,output);

var a : array[0..20,1..2] of integer;

c,b : array[0..20] of integer;

n,m,l,j,s,k,l : integer;

Begin

WRITELN; readln(n);

for i:=1 to 20 do a[i,1]:=0;

①; a[0..2]:=1; j:=0;

for i:=2 to n-1 do

begin

s:=0; m:=n;

while(m<>0) and (m mod i=0) do

begin

m:=m div i;

②;

end;

if ③ then begin

j:=j+1; ④;

a[j,2]:=⑤;

end

end;

for i:=0 to j do b[i]:=0;

whil ⑥ do

begin

k:=j;

while B[K]=A[K, 2] do k:=k-1;

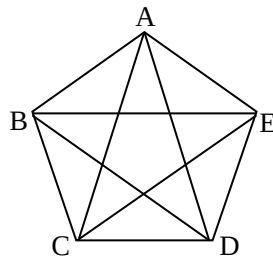
```

b[k]:=b[k]+1;
for L:= ⑦ do b[L]:=0;
s:=1;
for i:=1 to j do
  if b[i]<>0 then for L:=1 to b[i] do
    ⑧;
  if s=n then begin
    for i:=1 to j do c[i]:=b[i];
    WRITE('('); M:=1;
    for i:=1 to j do
      while(c[i]>0) and (M<>N) do
        begin
          M:=M*A[i.1];
          if M=N then write(a[i,1])
          else begin
            write(A[i,1],'*');
            c[i]:=c[i]-1;
          end;
        end;
      writeln('');
    end
  end
end
End.

```

2. **[问题描述]** 给出一个凸多边形，可以取得若干个内接三角形，同时约定内接三角形必须有一条边（仅能有一条边）与凸多边形的边相重合，例如：下面的 5 边形中，可能的内接三角形有 5 种：

$\triangle ACD$ ， $\triangle BDE$ ， $\triangle CEA$ ， $\triangle DAB$ ， $\triangle EBC$



问题：当依次给出凸多边形的每个顶点的 2 个坐标之后，找出一个面积最大的内接三角形，输出该三角形的面积与三个顶点的坐标。

[算法说明] 凸多边形的每个顶点用一对坐标(x,y)表示：

用数组 p:ARRAY[1..2*n]of point;存贮输入的顶点坐标；

同时编制一个由三角形的三个顶点计算其面积的函数 SEA。

[程序清单] program exp5(input,output)

Const n=6;

type point=record x,y:real end;

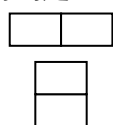
```

var p      :array[1..2*n] of point;
  i,j      :integer;
  q1,q2,q3 :point;
  smax     :real
Function Sea(p1,p2,p3:point):real;
var s1,s2,s3,p4:real;
begin
  s1:=sqrt(p1.x-p2.x)*(p1.x-p2.x)+(p1.y-p2.y)*(p1.y-p2.y)) ;
  s2:=sqrt(p1.x-p3.x)*(p1.x-p3.x)+(p1.y-p3.y)*(p1.y-p3.y)) ;
  s3:=sqrt(p2.x-p3.x)*(p2.x-p3.x)+(p2.y-p3.y)*(p2.y-p3.y)) ;
  p4:=__①__;Sea:=sqrt(p4*(p4-s1)*(p4-s2)*(p4-s3));
end;
Begin
  for i:=1 to n do readln(p[i].x, p[i].y);smax:=0;
  for i:=1 to n-1 do __②__
  for i:=1 to n do
  for j:=__③__ do
  if __④__ then
  begin
    smax:=Sea(P[i],p[i+1],p[j]);
    q1:=q[i]; q2:=__⑤__; q3:=p[j]
  end;
  writeln(smax, q1.x,q1.y,q2.x,q2.y,q3.x,q3.y)
End.

```

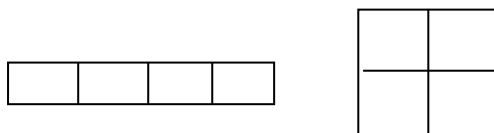
3. **[问题描述]** 拼图形：边长为 1 的正方形面积为 1，从边长为 1 的正方形出发可以用 2 个边长为 1 的正方形拼成面积为 2 的长方形：

同时约定：

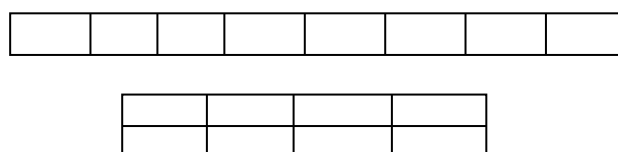


1. 边长对应相等的长方形被认为是相同的（所以左边的两个面积为 2 的长方形只看作一个长方形）。
2. 长度相等的边才能拼接，且两个边必须重合。

从面积为 2 的长方形出发，用 2 个面积为 2 的长方形可拼出面积为 4 的长方形（包括正方形），拼法如下：



同样再从面积为 4 的长方形（包括正方形）出发，可以拼成面积为 8 的长方形，拼法如下：

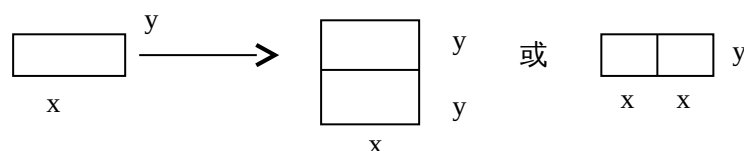


可以按上面的方法继续拼下去。

问题：输入一个数 N ，输出面积不超过 N 的所有可能拼法。例如：当 $N=20$ 时，输出 $(1,1)$ ， $(2,1)$ ， $(4,2)$ ， $(8,2)$ ， $(16,3)$ 即面积为 1 的拼法 1 种，面积为 2 的拼法 1 种，面积为 4 的拼法 2 种，面积为 8 的拼法 2 种，面积为 16 的拼法 3 种。

【算法说明】 矩形可以用三个数 x,y,s 来表示，其中 x,y 表示边长， s 表示面积，并用数组 $G[1..100,1..3]$ 表示图形。

拼接过程为： 第二种拼法：



当给出 n 之后，可能拼接的次数 r 满足： $2^r \leq N < 2^{r+1}$ （不包括面积为 1 的拼法）；用数组 $b[1..100]$ 记录各种面积可能出现的拼法。

【程序清单】

```

program exp8(input, output)
type g=record x,y,z:integer end;
var g1      :array[1..100]of g;
    i,j,n,s1,jj,j1,j2,i1 :integer;
    b        :array[1..100]of integer;
    gw       :g;
Function eg(qk:g):boolean;
var jeq      :integer;p:boolean;
begin
p:=true; jeq:=1;
while(p and (jeq<=j)) do
if ((gk.x=g1[jeq].x)and(gk.y=g1[jeq].y))
or((gk.x=g1[jeq].y)and(gk.y=g1[jeq].x))
then p:=false else jeq:=jeq+1;
eq:=p
end;
Begin
readln(n); s1:=1; jj:=1;
while ① do
begin ② ; jj:=jj+1 end;
③ ; j1:=1; j:=1;
g1[j].x:=1; g1[j].y:=1; g1[j].z:=1;
for i:=2 to jj do
begin
j2:=j;
for i1:=j1 to j2 do
begin
gw.x:=g1[i1].x*2; gw.y:=g1[i1].y; gw.z:=g1[i1].z*2;
if ④ then begin
j:=j+1; g1[j]:=gw

```

```

        end;
    gw.x:=g1[i1].x; ⑤
    if eq(gw) then begin
        j:=j+1; ⑥
    end;
end;
j1:=j2+1
end;
for i:=1 to n do b[i]:=0;
for I:=1 to j do ⑦
for I:=1 to n do if ⑧ then write(('i',',',b[i],'));
End.

```