

第九届分区联赛提高组初赛试题

(提高组 PASCAL 语言 二小时完成)

●● 全部答案均要写在答案卷子上, 写在试卷纸上一律无效 ●●

一.单项选择题 (共 10 题, 每题 1.5 分, 共计 15 分。每题有且仅有一个正确答案.)。

1. 图灵 (Alan Turing) 是 ( )。  
A) 美国人 B) 英国人 C) 德国人 D) 匈牙利人 E) 法国人
2. 第一个给计算机写程序的人是 ( )。  
A) Alan Mathison Turing B) Ada Lovelace C) John von Neumann  
D) John Mc-Carthy E) Edsger Wybe Dijkstra
3. 十进制数 2003 等值于二进制数 ( )。  
A) 0100000111 B) 10000011 C) 110000111 D) 11111010011 E) 1111010011
4. 假设  $A=\text{true}$ ,  $B=\text{false}$ ,  $C=\text{ture}$ ,  $D=\text{ture}$ , 逻辑运算表达式  $A \wedge B \vee C \wedge D$  的值是 ( )。  
A) ture B) false C) 0 D) 1 E) NULL
5. 一个高度为  $h$  的二叉树最小元素数目是 ( )。  
A)  $2h+1$  B)  $h$  C)  $2h-1$  D)  $2h$  E)  $2h-1$
6. 已知队列 (13, 2, 11, 34, 41, 77, 5, 7, 18, 26, 15), 第一个进入队列的元素是 13, 则第五个出队列的元素是 ( )。  
A) 5 B) 41 C) 77 D) 13 E) 18
7. 下面一段程序是用 ( ) 语言书写的。  

```
int func1(int n){
    int i,sum=0;
    for(i=1;i<=n;i++)
        sum+=i*i;
    return sum;
}
```

  
A) FORTRAN B) PASCAL C) C D) PROLOG E) BASIC
8. 设全集  $E=\{1, 2, 3, 4, 5\}$ , 集合  $A=\{1, 4\}$ ,  $B=\{1, 2, 5\}$ ,  $C=\{2, 4\}$ , 则集合  $(A \cap B) \cup \sim C$  为 ( )。  
A) 空集 B)  $\{1\}$  C)  $\{3, 5\}$  D)  $\{1, 5\}$  E)  $\{1, 3, 5\}$
9. 表达式  $(1+34)*5-56/7$  的后缀表达式为 ( )。  
A)  $1+34*5-56/7$  B)  $-*+1\ 34\ 5/56\ 7$  C)  $1\ 34\ +5*56\ 7/-$   
D)  $1\ 34\ 5*+56\ 7/-$  E)  $1\ 34+5\ 56\ 7-* /$
10. 下列计算机设备, 即是输入设备, 又是输出设备的是 ( )。  
A) 键盘 B) 触摸屏 C) 扫描仪 D) 投影仪 E) 数字化仪

**二.不定项选择题（共 10 题，每题 1.5 分，共计 15 分。多选少选均不得分）。**

11. 下列分辨率的显示器显示出的图像，最清晰的是（ ）。  
A) 800\*600 B) 1024\*768 C) 640\*480 D) 1280\*1024 E) 800\*1000
12. 下列说法中，哪个（些）是错误的（ ）。  
A) 程序是指令的序列，它有三种结构：顺序、分支和循环。  
B) 数据总线决定了中央处理器 CPU 所能访问的最大内存空间的大小。  
C) 中央处理器 CPU 内部有寄存器组，用来储存数据。  
D) 不同厂家生产的 CPU 所能处理的指令集是相同的。  
E) 数据传输过程中可能会出错，奇偶校验法可以检测出数据中那一为在传输中出了差错。
13. CPU 访问内存的速度比访问下列哪个（些）存储设备要慢（ ）。  
A) 寄存器 B) 硬盘 C) 软盘 D) 高速缓存 E) 光盘
14. 下列电子邮件地址，哪个（些）是正确的（ ）。  
A) wang@hotmail.com B) cai@jcc.pc.tool.rf.edu.jp C) 162.105.111.22  
D) ccf.edu.cn E)http://www.sina.com
15. 数字图像文件可以用下列哪个（些）软件来编辑（ ）。  
A) 画笔 (Paintbrush) B) 记事簿 (Notepad) C) Photoshop D) WinRAR  
E)Midisoft
16. 下列哪个（些）软件不是操作系统软件的名字（ ）。  
A) WindowsXP B) DOS C) Linux D) OS/2 E) Arch/Info
17. 下列哪个（些）不是个人计算机的硬件组成部分（ ）。  
A) 主板 B) 虚拟内存 C) 电源 D) 硬盘 E) 总线
18. 运算式 $(2008)_{10}-(3723)_8$ 的结果是（ ）。  
A)  $(-1715)_{10}$  B)  $(5)_{10}$  C)  $(5)_{16}$  D)  $(101)_2$  E)  $(3263)_8$
19. 已知元素 (8, 25, 14, 87, 51, 90, 6, 19, 20)，问这些元素以怎样的顺序进入栈，才能使出栈的顺序满足：8 在 51 前面；90 在 87 的后面；20 在 14 的后面；25 在 6 的前面；19 在 90 的后面。（ ）。  
A) 20, 6, 8, 51, 90, 25, 14, 19, 87  
B) 51, 6, 19, 20, 14, 8, 87, 90, 25  
C) 19, 20, 90, 7, 6, 25, 51, 14, 87  
D) 6, 25, 51, 8, 20, 19, 90, 87, 14  
E) 25, 6, 8, 51, 87, 90, 19, 14, 20
20. 假设我们用  $d=(a_1, a_2, \dots, a_5)$  表示无向图 G 的 5 个顶点的度数，下面给出的哪（些）组 d 值合理（ ）。

- A) {5, 4, 4, 3, 1}    B) {4, 2, 2, 1, 1}    C) {3, 3, 3, 2, 2}  
 D) {5, 4, 3, 2, 1}    E) {2, 2, 2, 2, 2}

### 三、问题求解 (共 2 题, 每题 5 分, 共计 10 分)

1. 无向图  $G$  有 16 条边, 有 3 个 4 度顶点、4 个 3 度顶点, 其余顶点的度均小于 3, 则  $G$  至少\_\_\_\_\_个顶点。
2. 某年级学生共选修 6 门课程, 期末考试前, 必须提前将这 6 门课程考完, 每人每天只在下午至多考一门课程, 设 6 门课程为  $C_1, C_2, C_3, C_4, C_5, C_6$ ,  $S(C_i)$  为学习  $C_i$  的学生集合。已知  
 $S(C_i) \cap S(C_6) \neq \emptyset, i=1, 2, \dots, 5, S(C_i) \cap S(C_{i+1}) \neq \emptyset, i=1, 2, 3, 4, S(C_5) \cap S(C_1) \neq \emptyset$ ,  
 问至少安排\_\_\_\_\_天才能考完这 6 门课程。

### 四. 阅读程序 (共 4 题, 每题 8 分, 共计 32 分)

```
1. program Program1;
   var  a,b,c,d,sum : longint;

   begin
     read(a,b,c,d);
     a := a mod 23; b := b mod 28; c := c mod 33;
sum := a * 5544 + b * 14421 + c * 1228 - d;
     sum := sum + 21252; sum := sum mod 21252;
     if (sum = 0) then sum := 21252;
     writeln(sum);
   end.
```

输入 : 283 102 23 320

输出\_\_\_\_\_

```
2. program Program2;
   const
     u : array[1..4] of integer = (0,5,3,1);
     v : array[1..4] of integer = (0,7,6,5);
   var  a,b,c,d,e,f,x,y,z: integer;

   begin
     read(a,b,c,d,e,f);
     z := f + e + d + (c+3) div 4; y := 5 * d + u[c mod 4];
     if (b > y) then
       begin
         z := z + (b - y + 8) div 9;
         x := ((b - y + 8) div 9 * 9 - (b - y)) * 4 + 11 * e + v[c mod 4];
       end
     else
```

```

    x := (y - b) * 4 + 11 * e + v[c mod 4];
    if (a > x) then
        z := z + (a - x + 35) div 36;
        writeln(z)
    end.

```

输入：4 7 9 20 56 47

输出\_\_\_\_\_

```

3. program Program3;
var m,n: integer;  mark: Boolean;

function test(m,N:integer):integer;
var i,p: integer;  flag: boolean;
begin
    m := m - 1;  i := 0;  flag := False;
    for p:= 2*N downto (N+1) do
        begin
            i:= (i+m) mod p;
            if (i
                begin
                    test := 0;  flag := Ture;  Break;
                end
            end;
            if not(flag) then  test:=1;
        end;

begin
    read(n);  m:=1;  Mark := False;
    repeat
        if (test(m,n)=1) then
            begin  writeln(m); break;  end;
        m:= m+1;
    until Mrak;
end.

```

输入：7

输出\_\_\_\_\_

```

4. program Program4;

var m,n,i,j: integer;
    p,w,a,b: array[0..19] of integer;
begin
    read(n); m:= 0;
    for i:= 0 to n-1 do

```

```

begin  read(p[i]);  b[i]:=1; end;
for i:=0 to  n-1 do
begin
  if (i>0) then
    a[m]:=p[i]-p[i-1]
  else
    a[m]:=p[i];
    m:=m+1;
while ((m>1) and (a[m-1]=0)) do
begin  m:=m-1;  b[m]:=1; end;
  if (m>0) then
    w[i]:=b[m-1];
  else
    w[i]:=b[0];
    a[m-1]:=a[m-1]-1;
    for j:=0 to m-1 do  b[j]:=b[j]+1;
while ((m>1) and (a[m-1]=0)) do
begin
m:=m-1;  b[m]:=1;
end;
  end;
  for i:= 0 to n-1 do
begin
  write(w[i]);  write(' ');
end;
writeln(' ');
end.

```

输入：9

4 6 6 6 6 8 9 9 9 9

输出：\_\_\_\_\_

**五. 完善程序（共 2 题，第 1 题每空 3 分；第 2 题每空 2 分。共计 28 分）。**

### **1. 翻硬币**

题目描述：

一摞硬币共有  $m$  枚，每一枚都是正面朝上。取下最上面的一枚硬币，将它翻面后放回原处。然后取下最上面的 2 枚硬币，将他们一起翻面后放回原处。在取 3 枚，取 4 枚……直至  $m$  枚。然后在从这摞硬币最上面的一枚开始，重复刚才的做法。这样一直做下去，直到这摞硬币中每一枚又是正面朝上为止。例如， $m$  为 1 时，翻两次即可。

输入：仅有的一个数字是这摞硬币的枚数  $m$ ， $0 < m < 1000$ 。

输出：为了使这摞硬币中的每一枚都是朝正面朝上所必须翻的次数。

输入样例：30

输出样例：899

程 序：

```
program Program1;
var m:integer;
function solve(m: integer):integer;
var i,t,d: integer;
    flag: Boolean;
begin
    if (m = 1) then
        solve := (1)
    else begin
        d := 2*m+1;    t := 2;    i := 1;    flag := False;
        repeat
            if (t = 1) then
                begin
                    solve := (2) ;    flag := True;
                end
            else if ( (3) ) then
                begin
                    solve := i*m-1;    flag := True;
                end
            else
                t := (4) ;
                i:=i+1;
        until flag;
    end
end;
begin
    read(m); if (( (5) ) and (m<1000)) then
        writeln( (6) );
end.
```

## 2. OIM 地形

题目描述：

二维离散世界有一种地形叫 OIM(OI Mountain)。这种山的坡度只能上升('/')或下降('\'),而且两边的山脚都与地平线等高,山上所有地方都不低于地平线.例如：

$$\begin{array}{ccc} \wedge & & \wedge \\ / \quad \backslash & \text{是一座 OIM} ; & \text{而 } / \quad \backslash \quad \text{不是。} \\ & & \vee \end{array}$$

这个世界的地理学家们为了方便纪录，给 OIM 所有可能的形状用正整数编好号，而且每个正整数恰好对应一种山形。他们规定，若两座山的宽度不同，则较宽的编号较大；若宽度相同，则比较从左边开始第 1 个坡度不同的地方，坡度上升的编号较大。以下三座 OIM 的编号有小到大递增：

$\wedge$     $\wedge$     $\wedge \wedge$   
 $/ \vee \vee \vee \vee / \vee \backslash$ 。显然 $\wedge$ 的编号为1。但是地理学家在整理纪录是发觉，查找编号与山形的对应关系不是很方便。他们希望能快速地从编号得到山的形状。你自告奋勇答应他们写一个程序，输入编号，能马上输出山形。

输入：一个编号（编号大小不超过 600,000,000），  
 输出：输入编号所对应的山形，1 座山所占行数恰为它的高度，即山顶上不能有多余空行。

输入样例：15

输出样例：    $\wedge \wedge$   
                $/ \vee \backslash$

程 序：

```

program Program2;
const
  L:integer =19;  SZ: integer =50;
  UP: char = '/';  DN: char = '\';
Var
  i,nth,x,y,h,e,f:integer;
  m: array[0..1,0..38,0..19] of integer;
  pic: array[0..49,0..49] of char;

procedure init;
  var k,s,a,b,c: integer;
  begin
    for a:=0 to 1 do
      for b:=0 to 2*L do
        for c:=0 to L do
          m[a,b,c]:=0;  m[0,0,0]:=1;
        for k:=0 to 2*L-1 do
          begin
            for s:=1 to L do
              begin
                m[0,k+1,s] := m[0,k,s+1] + m[1,k,s+1];
                m[1,k+1,s]:=  (1)  ;
              end;
            m[0,k+1,0] :=m[0,k,1]+m[1,k,1];
          end;
        end;
      end;

  procedure draw(k,s,nth:integer);
  begin

```

```

if (k=0) then exit;
if ((nth-m[1,k,s])>=0) then
begin
    nth:=nth-m[1,k,s];
    if (y>h) then      (2)      ;
    pic[y,x]:=UP; y:=y+1; x:=x+1; draw(      (3)      );
end
else begin
    y:=y - 1; pic[y,x]:=DN; x:=x+1; draw(k-1,s-1,nth);
end;
end;

begin
init;
read(nth);
for e:=0 to SZ-1 do
    for f:=0 to SZ-1 do
        pic[e,f]:= ' ';
x:=0;
y:=0
h:=0;
i:=0;

while ((nth-m[0,2*i,0])>=0) do
begin
    nth:= nth-m[0,2*i,0];
    (4)      ;
end;

draw(      (5)      );
for i:=h downto x-1 do
begin
    for e:=0 to x-1 do
        write(pic[i,e]);
        writeln(' ');
end;
end.

```