

第十五届全国青少年信息学奥林匹克联赛初赛试题

(提高组 Pascal 语言 二小时完成)

● ● 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 ● ●

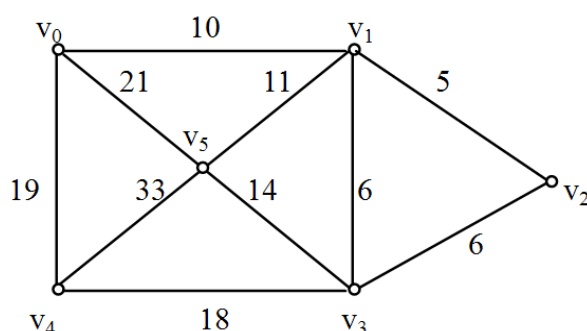
一．单项选择题 (共 10 题，每题 1.5 分，共计 15 分。每题有且仅有一个正确答案。)

- 1、关于图灵机下面的说法哪个是正确的：
A) 图灵机是世界上最早的电子计算机。
B) 由于大量使用磁带操作，图灵机运行速度很慢。
C) 图灵机只是一个理论上的计算模型。
D) 图灵机是英国人图灵发明的，在二战中为破译德军的密码发挥了重要作用。
- 2、关于 BIOS 下面的说法哪个是正确的：
A) BIOS 是计算机基本输入输出系统软件的简称。
B) BIOS 里包含了键盘、鼠标、声卡、图形界面显示器等常用输入输出设备的驱动程序。
C) BIOS 一般由操作系统厂商来开发完成。
D) BIOS 能提供各种文件拷贝、复制、删除以及目录维护等文件管理功能。
- 3、已知大写字母 A 的 ASCII 编码为 65 (十进制)，则大写字母 J 的十六进制 ASCII 编码为：
A) 48 B) 49 C) 50 D) 以上都不是
- 4、在字长为 16 位的系统环境下，一个 16 位带符号整数的二进制补码为 1111111111101101。其对应的十进制整数应该是：
A) 19 B) -19 C) 18 D) -18
- 5、一个包含 n 个分支结点 (非叶结点) 的非空满 k 叉树， $k \geq 1$ ，它的叶结点数目为：
A) $nk + 1$ B) $nk - 1$ C) $(k+1)n - 1$ D) $(k-1)n + 1$
- 6、表达式 $a*(b+c)-d$ 的后缀表达式是：
A) $abcd*+-$ B) $abc+*d-$ C) $abc*+d-$ D) $-+*abcd$
- 7、最优前缀编码，也称 Huffman 编码。这种编码组合的特点是对于较频繁使用的元素给与较短的唯一编码，以提高通讯的效率。下面编码组合哪一组不是合法的前缀编码。
A) (00, 01, 10, 11)
B) (0, 1, 00, 11)
C) (0, 10, 110, 111)
D) (1, 01, 000, 001)
- 8、快速排序平均情况和最坏情况下的算法时间复杂度分别为：
A) 平均情况 $O(n \log_2 n)$ ，最坏情况 $O(n^2)$
B) 平均情况 $O(n)$ ，最坏情况 $O(n^2)$

- C) 平均情况 $O(n)$ ，最坏情况 $O(n\log_2 n)$
 D) 平均情况 $O(\log_2 n)$ ，最坏情况 $O(n^2)$

9、左图给出了一个加权无向图，从顶点 V_0 开始用 prim 算法求最小生成树。则依次加入最小生成树的顶点集的顶点序列为：

- A) $V_0, V_1, V_2, V_3, V_5, V_4$
 B) $V_0, V_1, V_5, V_4, V_3, V_2$
 C) $V_1, V_2, V_3, V_0, V_5, V_4$
 D) $V_1, V_2, V_3, V_0, V_4, V_5$



10、全国信息学奥林匹克的官方网站为参与信息学竞赛的老师同学们提供相关的信息和资源，请问全国信息学奥林匹克官方网站的网址是：

- A) <http://www.noi.com/> B) <http://www.noi.org/>
 C) <http://www.noi.cn/> D) <http://www.xinxixue.com/>

二．不定项选择题（共 10 题，每题 1.5 分，共计 15 分。每题正确答案的个数不少于 1。多选或少选均不得分）。

1、关于 CPU 下面哪些说法是正确的：

- A) CPU 全称为中央处理器（或中央处理单元）。
 B) CPU 能直接运行机器语言。
 C) CPU 最早是由 Intel 公司发明的。
 D) 同样主频下，32 位的 CPU 比 16 位的 CPU 运行速度快一倍。

2、关于计算机内存下面的说法哪些是正确的：

- A) 随机存储器（RAM）的意思是当程序运行时，每次具体分配给程序的内存位置是随机而不确定的。
 B) 一般的个人计算机在同一时刻只能存/取一个特定的内存单元。
 C) 计算机内存严格说来包括主存（memory）、高速缓存（cache）和寄存器（register）三个部分。
 D) 1MB 内存通常是指 1024×1024 字节大小的内存。

3、关于操作系统下面说法哪些是正确的：

- A. 多任务操作系统专用于多核心或多个 CPU 架构的计算机系统的管理。
 B. 在操作系统的管理下，一个完整的程序在运行过程中可以被部分存放在内存中。
 C. 分时系统让多个用户可以共享一台主机的运算能力，为保证每个用户都得到及时的响应通常会采用时间片轮转调度的策略。
 D. 为了方便上层应用程序的开发，操作系统都是免费开源的。

4、关于计算机网络，下面的说法哪些是正确的：

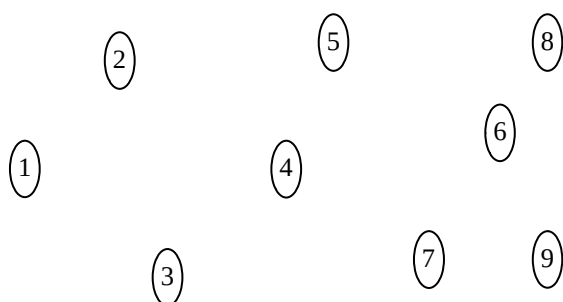
- A) 网络协议之所以有很多层主要是由于新技术需要兼容过去老的实现方案。

- B) 新一代互联网使用的 IPv6 标准是 IPv5 标准的升级与补充。
- C) TCP/IP 是互联网的基础协议簇，包含有 TCP 和 IP 等网络与传输层的通讯协议。
- D) 互联网上每一台入网主机通常都需要使用一个唯一的 IP 地址，否则就必须注册一个固定的域名来标明其地址。
- 5、关于 HTML 下面哪些说法是正确的：
- A) HTML 全称超文本标记语言，实现了文本、图形、声音乃至视频信息的统一编码。
- B) HTML 不单包含有网页内容信息的描述，同时也包含对网页格式信息的定义。
- C) 网页上的超链接只能指向外部的网络资源，本网站网页间的联系通过设置标签来实现。
- D) 点击网页上的超链接从本质上就是按照该链接所隐含的统一资源定位符（URL）请求网络资源或网络服务。
- 6、若 3 个顶点的无权图 G 的邻接矩阵用数组存储为 $\{\{0, 1, 1\}, \{1, 0, 1\}, \{0, 1, 0\}\}$ ，假定在具体存储中顶点依次为: v_1, v_2, v_3 关于该图，下面的说法哪些是正确的：
- A) 该图是有向图。
- B) 该图是强连通的。
- C) 该图所有顶点的入度之和减所有顶点的出度之和等于 1。
- D) 从 v_1 开始的深度优先遍历所经过的顶点序列与广度优先的顶点序列是相同的。
- 7、在带尾指针（链表指针 clist 指向尾结点）的非空循环单链表中每个结点都以 next 字段的指针指向下一个节点。假定其中已经有 2 个以上的结点。下面哪些说法是正确的：
- A) 如果 p 指向一个待插入的新结点，在头部插入一个元素的语句序列为：
 $p \wedge \text{next} := \text{clist} \wedge \text{next}; \text{clist} \wedge \text{next} := p;$
- B) 如果 p 指向一个待插入的新结点，在尾部插入一个元素的语句序列为：
 $p \wedge \text{next} := \text{clist}; \text{clist} \wedge \text{next} := p;$
- C) 在头部删除一个结点的语句序列为：
 $p := \text{clist} \wedge \text{next}; \text{clist} \wedge \text{next} := \text{clist} \wedge \text{next} \wedge \text{next}; \text{dispose}(p);$
- D) 在尾部删除一个结点的语句序列为。
 $p := \text{clist}; \text{clist} := \text{clist} \wedge \text{next}; \text{dispose}(p);$
- 8、散列表的地址区间为 0-10,散列函数为 $H(K)=K \bmod 11$ 。采用开地址法的线性探查法处理冲突，并将关键字序列 26, 25, 72, 38, 8, 18, 59 存储到散列表中，这些元素存入散列表的顺序并不确定。假定之前散列表为空，则元素 59 存放在散列表中的可能地址有：
- A) 5 B) 7 C) 9 D) 10
- 9、排序算法是稳定的意思是关键码相同的记录排序前后相对位置不发生改变，下列哪些排序算法是稳定的：
- A) 插入排序 B) 基数排序 C) 归并排序 D) 冒泡排序
- 10、在参加 NOI 系列竞赛过程中，下面哪些行为是被严格禁止的：
- A) 携带书写工具，手表和不具有通讯功能的电子词典进入赛场。

- B) 在联机测试中通过手工计算出可能的答案并在程序里直接输出答案来获取分数。
- C) 通过互联网搜索取得解题思路。
- D) 在提交的程序中启动多个进程以提高程序的执行效率。

三．问题求解（共 2 题，每空 5 分，共计 10 分）

1．拓扑排序是指将有向无环图 G 中的所有顶点排成一个线性序列，使得图中任意一对顶点 u 和 v ，若 $\langle u, v \rangle \in E(G)$ ，则 u 在线性序列中出现在 v 之前，这样的线性序列成为拓扑序列。如下的有向无环图，对其顶点做拓扑排序，则所有可能的拓扑序列的个数为。



2．某个国家的钱币面值有 1, 7, 7^2 , 7^3 共计四种，如果要用现金付清 10015 元的货物，假设买卖双方各种钱币的数量无限且允许找零，那么交易过程中至少需要流通____张钱币。

四．阅读程序写结果（共 4 题，每题 8 分，共计 32 分）

1．

```
var
  a, b: integer;

function work(a, b: integer): integer;
begin
  if a mod b <> 0 then
    work := work(b, a mod b)
  else
    work := b;
end;

begin
  read(a, b);
  writeln(work(a, b));
end.
```

输入：123 321

输出：_____

2.

```
var
  a, b: array[0..3] of integer;
  i, j, tmp: integer;
begin
  for i := 0 to 3 do
    read(b[i]);
  for i := 0 to 3 do
    begin
      a[i] := 0;
      for j := 0 to i do
        begin
          inc(a[i], b[j]);
          inc(b[a[i] mod 4], a[j]);
        end;
      end;
    end;
  tmp := 1;
  for i := 0 to 3 do
    begin
      a[i] := a[i] mod 10;
      b[i] := b[i] mod 10;
      tmp := tmp * (a[i] + b[i]);
    end;
  writeln(tmp);
end.
```

输入：2 3 5 7

输出：_____

3.

```
const
  y = 2009;
  maxn = 50;

var
  n, i, j, s: longint;
```

```

c: array[0..maxn, 0..maxn] of longint;

begin
  s := 0;
  read(n);
  c[0, 0] := 1;
  for i := 1 to n do
    begin
      c[i, 0] := 1;
      for j := 1 to i - 1 do
        c[i, j] := c[i-1, j-1] + c[i-1, j];
      c[i, i] := 1;
    end;
  for i := 0 to n do
    s := (s + c[n, i]) mod y;
  write(s);
end.

```

输入：17

输出：_____

4.

```

var
  n, m, i, j, k, p: integer;
  a, b: array[0..100] of integer;
begin
  read(n, m);
  a[0] := n;
  i := 0;
  p := 0;
  k := 0;
  repeat
    for j := 0 to i - 1 do
      if a[i] = a[j] then
        begin
          p := 1;
          k := j;
          break;
        end;
    if p <> 0 then
      break;
    b[i] := a[i] div m;
    a[i+1] := (a[i] mod m) * 10;
    inc(i);
  until

```

```

until a[i] = 0;
write(b[0], '.');
for j := 1 to k - 1 do
    write(b[j]);
if p <> 0 then
    write('(');
for j := k to i - 1 do
    write(b[j]);
if p <> 0 then
    write(')');
writeln;
end.

```

输入：5 13

输出：_____

五．完善程序（前 5 空，每空 2 分，后 6 空，每空 3 分，共 28 分）

1. **（最大连续子段和）** 给出一个数列（元素个数不多于 100），数列元素均为负整数、正整数、0。请找出数列中的一个连续子数列，使得这个子数列中包含的所有元素之和最大，在和最大的前提下还要求该子数列包含的元素个数最多，并输出这个最大和以及该连续子数列中元素的个数。例如数列为 4，-5，3，2，4 时，输出 9 和 3；数列为 1 2 3 -5 0 7 8 时，输出 16 和 7。

```

var
    a: array[1..100] of integer;
    n, i, ans, len, tmp, beg: integer;

begin
    read(n);
    for i := 1 to n do
        read(a[i]);
    tmp := 0;
    ans := 0;
    len := 0;
    beg := ①;
    for i := 1 to n do
        begin
            if tmp + a[i] > ans then
                begin
                    ans := tmp + a[i];
                    len := i - beg;
                end
            else if (②) and (i - beg > len) then

```

```

        len := i - beg;
    if tmp + a[i] _____ ③ then
    begin
        beg := _____ ④;
        tmp := 0;
    end
    else
        _____ ⑤;
    end;
    writeln(ans, ' ', len);
end.

```

2. **(寻找等差数列)** 有一些长度相等的等差数列（数列中每个数都为 0~59 的整数），设长度均为 L，将等差数列中的所有数打乱顺序放在一起。现在给你这些打乱后的数，问原先，L 最大可能为多大？先读入一个数 n（1≤n≤60），再读入 n 个数，代表打乱后的数。输出等差数列最大可能长度 L。

```

var
    hash: array[0..60] of integer;
    n, x, ans, maxnum, i: integer;

function work(now: integer): boolean;
var
    ok: boolean;
    first, second, delta, i: integer;
begin
    while ((_____ ①) and (hash[now]=0)) do
        inc(now);
    if now > maxnum then
    begin
        work := true;
        exit;
    end;
    first := now;
    for second := first to maxnum do
        if hash[second] > 0 then
        begin
            delta := _____ ②;
            if first + delta * _____ ③ > maxnum then
                break;
            if delta = 0 then
                ok := (_____ ④)
            else

```



```

begin
    ok := true;
    for i := 0 to ans - 1 do
        ok := _____ ⑤ _____ and (hash[first+delta*i]>0);
    end;
    if ok then
        begin
            for i := 0 to ans - 1 do
                dec(hash[first+delta*i]);
            end;
            if work(first) then
                begin
                    work := true;
                    exit;
                end;
            for i := 0 to ans - 1 do
                inc(hash[first+delta*i]);
            end;
        end;
        work := false;
    end;

begin
    fillchar(hash, sizeof(hash), 0);
    read(n);
    maxnum := 0;
    for i := 1 to n do
        begin
            read(x);
            inc(hash[x]);
            if x > maxnum then
                maxnum := x;
        end;
    for ans := n downto 1 do
        if (n mod ans = 0) and _____ ⑥ _____ then
            begin
                writeln(ans);
                break;
            end;
    end;
end.

```