

第十六届全国青少年信息学奥林匹克联赛初赛试题

(普及组 Pascal 语言 两小时完成)

全部试题答案均要求写在答卷纸上, 写在试卷上一律无效

一、单项选择题 (共 20 题, 每题 1.5 分, 共计 30 分。每题有且仅有一个正确选项。)

1. $2E+03$ 表示 ()。

- A. 2.03 B. 5 C. 8 D. 2000

2. 一个字节 (byte) 由 () 个二进制位组成。

- A. 8 B. 16 C. 32 D. 以上都有可能

3. 以下逻辑表达式的值恒为真的是 ()。

- A. $P \vee (\neg P \wedge Q) \vee (\neg P \wedge \neg Q)$ B. $Q \vee (\neg P \wedge Q) \vee (P \wedge \neg Q)$
C. $P \vee Q \vee (P \wedge \neg Q) \vee (\neg P \wedge Q)$ D. $P \vee \neg Q \vee (P \wedge \neg Q) \vee (\neg P \wedge \neg Q)$

4. Linux 下可执行文件的默认扩展名为 ()。

- A. exe B. com C. dll D. 以上都不是

5. 如果树根算第 1 层, 那么一棵 n 层的二叉树最多有 () 个结点。

- A. $2^n - 1$ B. 2^n C. $2^n + 1$ D. 2^{n+1}

6. 提出“存储程序”的计算机工作原理的是 ()。

- A. 克劳德·香农 B. 戈登·摩尔 C. 查尔斯·巴比奇 D. 冯·诺依曼

7. 设 X 、 Y 、 Z 分别代表三进制下的一位数字, 若等式 $XY + ZX = XYX$ 在三进制下成立, 那么同样在三进制下, 等式 $XY * ZX = ()$ 也成立。

- A. YXZ B. ZXY C. XYZ D. XZY

8. Pascal 语言、C 语言和 C++ 语言都属于 ()。

- A. 面向对象语言 B. 脚本语言 C. 解释性语言 D. 编译性语言

9. 前缀表达式 “+ 3 * 2 + 5 12” 的值是 ()。

- A. 23 B. 25 C. 37 D. 65

10. 主存储器的存取速度比中央处理器（CPU）的工作速度慢得多，从而使得后者的效率受到影响。而根据局部性原理，CPU 所访问的存储单元通常都趋于聚集在一个较小的连续区域中。于是，为了提高系统整体的执行效率，在 CPU 中引入了（ ）。
A. 寄存器 B. 高速缓存 C. 闪存 D. 外存
11. 一个字长为 8 位的整数的补码是 11111001，则它的原码是（ ）。
A. 00000111 B. 01111001 C. 11111001 D. 10000111
12. 基于比较的排序时间复杂度的下限是（ ），其中 n 表示待排序的元素个数。
A. $\Theta(n)$ B. $\Theta(n \log n)$ C. $\Theta(\log n)$ D. $\Theta(n^2)$
13. 一个自然数在十进制下有 n 位，则它在二进制下的位数与（ ）最接近。
A. $5n$ B. $n \cdot \log_2 10$ C. $10 \cdot \log_2 n$ D. $10^n \log_2 n$
14. 在下列 HTML 语句中，可以正确产生一个指向 NOI 官方网站的超链接的是（ ）。
A. `欢迎访问 NOI 网站`
B. `欢迎访问 NOI 网站`
C. `<a>http://www.noi.cn`
D. `欢迎访问 NOI 网站`
15. 元素 R_1 、 R_2 、 R_3 、 R_4 、 R_5 入栈的顺序为 R_1 、 R_2 、 R_3 、 R_4 、 R_5 。如果第 1 个出栈的是 R_3 ，那么第 5 个出栈的不可能是（ ）。
A. R_1 B. R_2 C. R_4 D. R_5
16. 双向链表中有两个指针域 `llink` 和 `rlink`，分别指向该结点的前驱及后继。设 p 指向链表中的一个结点，它的左右结点均非空。现要求删除结点 p ，则下面语句序列中错误的是（ ）。
A. `p^.rlink^.llink = p^.rlink;`
 `p^.llink^.rlink = p^.llink; dispose(p);`
B. `p^.llink^.rlink = p^.rlink;`
 `p^.rlink^.llink = p^.llink; dispose(p);`
C. `p^.rlink^.llink = p^.llink;`
 `p^.rlink^.llink^.rlink = p^.rlink; dispose(p);`
D. `p^.llink^.rlink = p^.rlink;`
 `p^.llink^.rlink^.llink = p^.llink; dispose(p);`
17. 一棵二叉树的前序遍历序列是 ABCDEFG，后序遍历序列是 CBFEGDA，则根结点的左子树的结点个数可能是（ ）。

A. 2

B. 3

C. 4

D. 5

18. 关于拓扑排序, 下面说法正确的是 ()。

A. 所有连通的有向图都可以实现拓扑排序

B. 对同一个图而言, 拓扑排序的结果是唯一的

C. 拓扑排序中入度为 0 的结点总会排在入度大于 0 的结点的前面

D. 拓扑排序结果序列中的第一个结点一定是入度为 0 的点

19. 完全二叉树的顺序存储方案, 是指将完全二叉树的结点从上至下、从左至右依次存放到一个顺序结构的数组中。假定根结点存放在数组的 1 号位置, 则第 k 号结点的父结点如果存在的话, 应当存放在数组的 () 号位置。

A. $2k$

B. $2k+1$

C. $k/2$ 下取整

D. $(k+1)/2$ 下取整

20. 全国青少年信息学奥林匹克系列活动的主办单位是 ()。

A. 教育部

B. 科技部

C. 共青团中央

D. 中国计算机学会

二、问题求解 (共 2 题, 每题 5 分, 共计 10 分)

1. LZW 编码是一种自适应词典编码。在编码的过程中, 开始时只有一部基础构造元素的编码词典, 如果在编码的过程中遇到一个新的词条, 则该词条及一个新的编码会被追加到词典中, 并用于后继信息的编码。

举例说明, 考虑一个待编码的信息串: "xyx yy xyx"。初始词典只有 3 个条目, 第一个为 x , 编码为 1; 第二个为 y , 编码为 2; 第三个为空格, 编码为 3; 于是串 "xyx" 的编码为 1-2-1 (其中 - 为编码分隔符), 加上后面的一个空格就是 1-2-1-3。但由于有了一个空格, 我们就知道前面的 "xyx" 是一个单词, 而由于该单词没有在词典中, 我们就可以自适应的把这个词条添加到词典里, 编码为 4, 然后按照新的词典对后继信息进行编码, 以此类推。于是, 最后得到编码: 1-2-1-3-2-2-3-5-3-4。

现在已知初始词典的 3 个条目如上述, 则信息串 "yyxy xx yyxy xyx xx xyx" 的编码是_____。

2. 队列快照是指在某一时刻队列中的元素组成的有序序列。例如, 当元素 1、2、3 入队, 元素 1 出队后, 此刻的队列快照是 "2 3"。当元素 2、3 也出队后, 队列快照是 "", 即为空。现有 3 个正整数元素依次入队、出队。已知它们的和为 8, 则共有_____种可能的不同的队列快照 (不同队列的相同快照只计一次)。例如, "5 1"、"4 2 2"、"" 都是可能的队列快照; 而 "7" 不是可能的队列快照, 因为剩下的 2 个正整数的和不可能是 1。

三、阅读程序写结果 (共 4 题, 每题 8 分, 其中第 4 题 (1)、(2) 各 4 分, 共计 32 分)

```

1.
var
    a1, a2, a3, x : integer;

procedure swap(var a, b : integer);
var
    t : integer;
begin
    t := a;
    a := b;
    b := t;
end;

begin
    readln(a1, a2, a3);
    if a1 > a2 then
        swap(a1, a2);
    if a2 > a3 then
        swap(a2, a3);
    if a1 > a2 then
        swap(a1, a2);

    readln(x);
    if x < a2 then
        if x < a1 then
            writeln(x, ' ', a1, ' ', a2, ' ', a3)
        else
            writeln(a1, ' ', x, ' ', a2, ' ', a3)
    else
        if x < a3 then
            writeln(a1, ' ', a2, ' ', x, ' ', a3)
        else
            writeln(a1, ' ', a2, ' ', a3, ' ', x);
end.

输入:
91 2 20
77

```

2.

```
var
    n, m, i : integer;

function rSum(j : integer) : integer;
var
    sum : integer;
begin
    sum := 0;
    while j <> 0 do
    begin
        sum := sum * 10 + (j mod 10);
        j := j div 10;
    end;
    rSum := sum;
end;

begin
    readln(n, m);
    for i := n to m do
        if i = rSum(i)
            then write(i, ' ');
    end.
```

输入: 90 120

输出: _____

3.

```
var
    s : string;
    i : integer;
    m1, m2 : char;

begin
    readln(s);
    m1 := ' ';
```

```

m2 := ' ';
for i := 1 to length(s) do
  if s[i] > m1 then
    begin
      m2 := m1;
      m1 := s[i];
    end
  else if s[i] > m2 then
    m2 := s[i];
  writeln(ord(m1), ' ', ord(m2));
end.

```

输入: Expo 2010 Shanghai China

输出: _____

提示:

字符	空格	'0'	'A'	'a'
ASCII 码	32	48	65	97

4.

```
const
```

```
  NUM = 5;
```

```
var
```

```
  n : integer;
```

```
function r(n : integer) : integer;
```

```
var
```

```
  i : integer;
```

```
begin
```

```
  if n <= NUM then
```

```
    begin
```

```
      r := n;
```

```
      exit;
```

```
    end;
```

```
  for i := 1 to NUM do
```

```
    if r(n - i) < 0 then
```

```
      begin
```

```
        r := i;
```

```

        exit;
    end;
    r := -1;
end;

begin
    readln(n);
    writeln(r(n));
end.

```

(1)

输入: 7

输出: _____ (4 分)

(2)

输入: 16

输出: _____ (4 分)

四、完善程序 (前 4 空, 每空 2.5 分, 后 6 空, 每空 3 分, 共计 28 分)

1. **(哥德巴赫猜想)** 哥德巴赫猜想是指, 任一大于 2 的偶数都可写成两个质数之和。迄今为止, 这仍然是一个著名的世界难题, 被誉为数学皇冠上的明珠。试编写程序, 验证任一大于 2 且不超过 n 的偶数都能写成两个质数之和。

```

const
    SIZE = 1000;

var
    n, r, i, j, k, ans : integer;
    p : array[1.. SIZE] of integer;
    tmp : boolean;

begin
    readln(n);
    r := 1;
    p[1] := 2;
    for i := 3 to n do
        begin
            ①;

```

```

    for j := 1 to r do
        if i mod ② = 0 then
            begin
                tmp := false;
                break;
            end;
        if tmp then
            begin
                inc(r);
                ③;
            end;
        end;
    end;

    ans := 0;
    for i := 2 to (n div 2) do
        begin
            tmp := false;
            for j := 1 to r do
                for k := j to r do
                    if i + i = ④ then
                        begin
                            tmp := true;
                            break;
                        end;
                if tmp then
                    inc(ans);
            end;
        writeln(ans);
    end.

```

若输入 n 为 2010，则输出 ⑤ 时表示验证成功，即大于 2 且不超过 2010 的偶数都满足哥德巴赫猜想。

2. **(过河问题)** 在一个月黑风高的夜晚，有一群人在河的右岸，想通过唯一的一根独木桥走到河的左岸。在这伸手不见五指的黑夜里，过桥时必须借助灯光来照明，不幸的是，他们只有一盏灯。另外，独木桥上最多承受两个人同时经过，否则将会坍塌。每个人单独过桥都需要一定的时间，不同的人需要的时间可能不同。两个人一起过桥时，由于只有一盏灯，所以需要的时间是较慢的那个人单独过桥时所花的时间。现输入 n ($2 \leq n < 100$) 和这 n 个人

单独过桥时需要的时间，请计算总共最少需要多少时间，他们才能全部到达河的左岸。

例如，有 3 个人甲、乙、丙，他们单独过桥的时间分别为 1、2、4，则总共最少需要的时间为 7。具体方法是：甲、乙一起过桥到河的左岸，甲单独回到河的右岸将灯带回，然后甲、丙再一起过桥到河的左岸，总时间为 $2+1+4=7$ 。

```
const
    SIZE = 100;
    INFINITY = 10000;
    LEFT = true;
    RIGHT = false;
    LEFT_TO_RIGHT = true;
    RIGHT_TO_LEFT = false;

var
    n, i : integer;
    time : array[1..SIZE] of integer;
    pos : array[1..SIZE] of boolean;

function max(a, b : integer) : integer;
begin
    if a > b then
        max := a
    else
        max := b;
end;

function go(stage : boolean) : integer;
var
    i, j, num, tmp, ans : integer;
begin
    if (stage = RIGHT_TO_LEFT)
    then begin
        num := 0;
        ans := 0;
        for i := 1 to n do
            if pos[i] = Right then
                begin
                    inc(num);
```

```

        if time[i] > ans then
            ans := time[i];
        end;
    if ① then
    begin
        go := ans;
        exit;
    end;
    ans := INFINITY;
    for i := 1 to n - 1 do
        if pos[i] = RIGHT then
            for j := i + 1 to n do
                if pos[j] = RIGHT then
                    begin
                        pos[i] := LEFT;
                        pos[j] := LEFT;
                        tmp := max(time[i], time[j]) + ②;
                        if tmp < ans then
                            ans := tmp;
                        pos[i] := RIGHT;
                        pos[j] := RIGHT;
                    end;
                end;
            end;
        go := ans;
    end
    else if (stage = LEFT_TO_RIGHT)
    then begin
        ans := INFINITY;
        for i := 1 to n do
            if ③ then
                begin
                    pos[i] := RIGHT;
                    tmp := ④;
                    if tmp < ans then
                        ans := tmp;
                    ⑤;
                end;
            end;
        go := ans;
    end
end

```

```
        else
            go := 0;
        end;

begin
    readln(n);
    for i := 1 to n do
        begin
            read(time[i]);
            pos[i] := RIGHT;
        end;
        writeln(go(RIGHT_TO_LEFT));
    end.
```

CCF NOIP2010 普及组 (Pascal 语言) 参考答案与评分标准

一、单项选择题 (共 20 题, 每题 1.5 分, 共计 30 分)

1	2	3	4	5	6	7	8	9	10
D	A	A	D	A	D	B	D	C	B
11	12	13	14	15	16	17	18	19	20
D	B	B	B	B	A	A	D	C	D

二、问题求解 (共 2 题, 每题 5 分, 共计 10 分)

1、2-2-1-2-3-1-1-3-4-3-1-2-1-3-5-3-6 (或 22123113431213536)

2、49

三、阅读程序写结果 (共 4 题, 每题 8 分, 其中第 4 题 (1) (2) 各 4 分, 共计 32 分)

1、2 20 77 91

2、99 101 111

3、120 112

4、(1) 1

(2) 4

四、完善程序 (前 4 空, 每空 2.5 分, 后 6 空, 每空 3 分, 共计 28 分)

(说明: 以下各程序填空可能还有一些等价的写法, 各省可请本省专家审定和上机验证, 不一定上报科学委员会审查)

1、 tmp := true

p[j]

p[r] := i

p[j] + p[k] (或 p[k] + p[j])

1004

2、 num <= 2 (或 num < 3 或 num = 2)

go(LEFT_TO_RIGHT)

pos[i] = LEFT (或 LEFT = pos[i])

time[i] + go(RIGHT_TO_LEFT) (或 go(RIGHT_TO_LEFT) + time[i])

pos[i] := LEFT

本小题中, LEFT 可用 true 代替, LEFT_TO_RIGHT 可用 true 代替, RIGHT_TO_LEFT 可用 false 代替。