

## 普及组 Pascal 语言试题

### 选手注意:

- 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）

- CCF NOIP2016 初赛普及组 Pascal 语言试题  
第 1 页，共 10 页

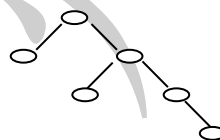
9. 以下是 32 位机器和 64 位机器的区别的是 ( )。

- A. 显示器不同
- B. 硬盘大小不同
- C. 寻址空间不同
- D. 输入法不同

10. 以下关于字符串的判定语句中正确的是 ( )。

- A. 字符串是一种特殊的线性表
- B. 串的长度必须大于零
- C. 字符串不可以用数组来表示
- D. 空格字符组成的串就是空串

11. 一棵二叉树如右图所示, 若采用顺序存储结构, 即用一维数组元素存储该二叉树中的结点(根结点的下标为 1, 若某结点的下标为  $i$ , 则其左孩子位于下标  $2i$  处、右孩子位于下标  $(2i+1)$  处), 则图中所有结点的最大下标为 ( )。



- A. 6
- B. 10
- C. 12
- D. 15

12. 若有如下程序段, 其中  $s$ 、 $a$ 、 $b$ 、 $c$  均已定义为整型变量, 且  $a$ 、 $c$  均已赋值 ( $c$  大于 0)。

```
s := a;  
for b := 1 to c do  
    s := s + 1;
```

则与上述程序段修改  $s$  值的功能等价的赋值语句是 ( )。

- A.  $s := a + b;$
- B.  $s := a + c;$
- C.  $s := s + c;$
- D.  $s := b + c;$

13. 有以下程序:

```
var  
    k, n: longint;  
begin  
    k := 4; n := 0;  
    while n < k do  
        begin  
            inc(n);  
            if n mod 3 <> 0 then  
                continue;  
            dec(k);  
        end;  
        writeln(k, ', ', n);  
    end.
```

程序运行后的输出结果是 ( )。

- A. 2,2
- B. 2,3
- C. 3,2
- D. 3,3

14. 给定含有  $n$  个不同的数的数组  $L = \langle x_1, x_2, \dots, x_n \rangle$ 。如果  $L$  中存在  $x_i (1 < i < n)$  使得  $x_1 < x_2 < \dots < x_{i-1} < x_i > x_{i+1} > \dots > x_n$ , 则称  $L$  是单峰的, 并称  $x_i$  是  $L$  的

“峰顶”。现在已知  $L$  是单峰的，请把 a-c 三行代码补全到算法中使得算法正确找到  $L$  的峰顶。

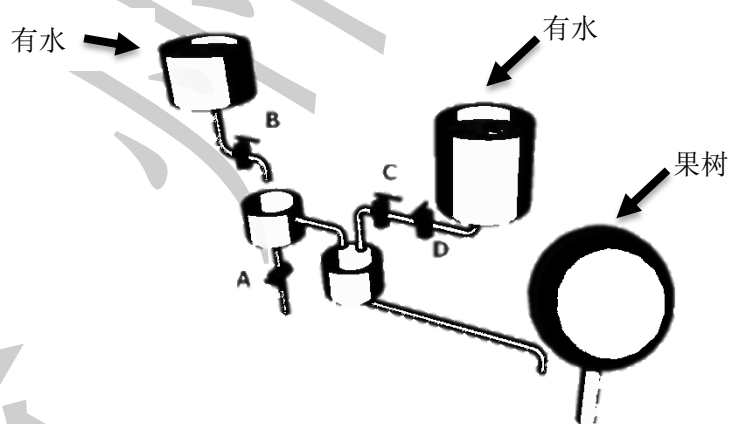
- a. `Search(k+1, n)`
- b. `Search(1, k-1)`
- c. `return L[k]`

`Search(1, n)`

- 1.  $k \leftarrow \lfloor n/2 \rfloor$
- 2. `if  $L[k] > L[k-1]$  and  $L[k] > L[k+1]$`
- 3. `then _____`
- 4. `else if  $L[k] > L[k-1]$  and  $L[k] < L[k+1]$`
- 5. `then _____`
- 6. `else _____`

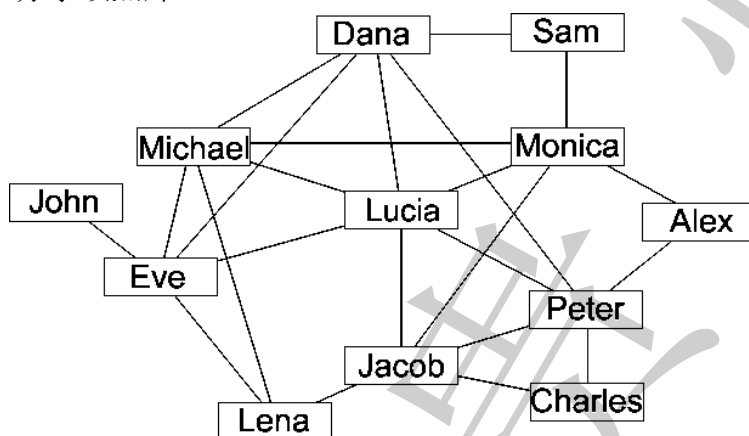
正确的填空顺序是（ ）。

- A. c, a, b                      B. c, b, a                      C. a, b, c                      D. b, a, c
15. 设简单无向图  $G$  有 16 条边且每个顶点的度数都是 2，则图  $G$  有（ ）个顶点。
- A. 10                      B. 12                      C. 8                      D. 16
16. 有 7 个一模一样的苹果，放到 3 个一样的盘子中，一共有（ ）种放法。
- A. 7                      B. 8                      C. 21                      D.  $3^7$
17. 下图表示一个果园灌溉系统，有 A、B、C、D 四个阀门，每个阀门可以打开或关上，所有管道粗细相同，以下设置阀门的方法中，可以让果树浇上水的是（ ）。



- A. B 打开，其他都关上
- B. AB 都打开，CD 都关上
- C. A 打开，其他都关上
- D. D 打开，其他都关上

18. Lucia 和她的朋友以及朋友的朋友都在某社交网站上注册了账号。下图是他们之间的关系图，两个人之间有边相连代表这两个人是朋友，没有边相连代表不是朋友。这个社交网站的规则是：如果某人 A 向他（她）的朋友 B 分享了某张照片，那么 B 就可以对该照片进行评论；如果 B 评论了该照片，那么他（她）的所有朋友都可以看见这个评论以及被评论的照片，但是不能对该照片进行评论（除非 A 也向他（她）分享了该照片）。现在 Lucia 已经上传了一张照片，但是她不想让 Jacob 看见这张照片，那么她可以向以下朋友（ ）分享该照片。



- A. Dana, Michael, Eve  
B. Dana, Eve, Monica  
C. Michael, Eve, Jacob  
D. Micheal, Peter, Monica
19. 周末小明和爸爸妈妈三个人一起想动手做三道菜。小明负责洗菜、爸爸负责切菜、妈妈负责炒菜。假设做每道菜的顺序都是：先洗菜 10 分钟，然后切菜 10 分钟，最后炒菜 10 分钟。那么做一道菜需要 30 分钟。注意：两道不同的菜的相同步骤不可以同时进行。例如第一道菜和第二道的菜不能同时洗，也不能同时切。那么做完三道菜的最短时间需要（ ）分钟。
- A. 90      B. 60      C. 50      D. 40
20. 参加 NOI 比赛，以下不能带入考场的是（ ）。
- A. 钢笔      B. 适量的衣服      C. U 盘      D. 铅笔

二、问题求解（共 2 题，每题 5 分，共计 10 分；第一题全部答对得 5 分，没有部分分；第二题第一空 2 分，第二空 3 分）

1. 从一个  $4 \times 4$  的棋盘（不可旋转）中选取不在同一行也不在同一列上的两个方格，共有\_\_\_\_\_种方法。

2. 约定二叉树的根节点高度为 1。一棵结点数为 2016 的二叉树最少有  
\_\_\_\_\_个叶子结点；一棵结点数为 2016 的二叉树最小的高度值是  
\_\_\_\_\_。

三、阅读程序写结果（共 4 题，每题 8 分，共计 32 分）

1. var

```
    tmp, max, min, sum, count: longint;
begin
    count := 0;
    read(tmp);
    if tmp = 0 then
        exit;
    max := tmp; min := tmp; sum := tmp;
    inc(count);
    while tmp <> 0 do
    begin
        read(tmp);
        if tmp <> 0 then
        begin
            inc(sum, tmp);
            inc(count);
            if tmp > max then
                max := tmp;
            if tmp < min then
                min := tmp;
        end;
    end;
    writeln(max, ',', min, ',', sum div count);
end.
```

输入: 1 2 3 4 5 6 0 7

输出: \_\_\_\_\_

2. var

```
    i, x, y: longint;
begin
    i := 100; x := 0; y := 0;
    while i > 0 do
```

```

begin
    dec(i);
    x := i mod 8;
    if x = 1 then
        inc(y);
    end;
    writeln(y);
end.

```

输出: \_\_\_\_\_

3. var

```

a: array[1..6] of longint = (1, 2, 3, 4, 5, 6);
pi, pj, t, i: longint;
begin
    pi := 1;
    pj := 6;
    while pi < pj do
    begin
        t := a[pi];
        a[pi] := a[pj];
        a[pj] := t;
        inc(pi);
        dec(pj);
    end;
    for i := 1 to 6 do
        write(a[i], ',');
    writeln;
end.

```

输出: \_\_\_\_\_

4. const

```

max = 100;
var
    i, length1, length2: longint;
    s1, s2: string[max];
begin
    s1 := 'I have a dream.';
    s2 := 'I Have A Dream.';
    length1 := length(s1);
    length2 := length(s2);
    for i := 1 to length1 do

```

```

        if (s1[i] >= 'a') and (s1[i] <= 'z') then
            s1[i] := chr(ord(s1[i]) - ord('a') + ord('A'));
    for i := 1 to length2 do
        if (s2[i] >= 'a') and (s2[i] <= 'z') then
            s2[i] := chr(ord(s2[i]) - ord('a') + ord('A'));
    if s1 = s2 then
        writeln('=')
    else if s1>s2 then
        writeln('>')
    else
        writeln('<');
end.

```

输出: \_\_\_\_\_

#### 四、完善程序（共 2 题，每题 14 分，共计 28 分）

1. （读入整数）请完善下面的程序，使得程序能够读入两个 `longint` 范围内的整数，并将这两个整数分别输出，每行一个。（第一、五空 2.5 分，其余 3 分）  
输入的整数之间和前后只会出现空格或者回车。输入数据保证合法。

例如：

输入：

123 -789

输出：

123

-789

var

a,b: longint;

function readint: longint;

var

num: longint; // 存储读取到的整数

negative: longint; // 负数标识

c:char; // 存储当前读取到的字符

begin

num := 0; negative := 0;

read(c);

while ((c < '0') or (c > '9')) and (c <> '-') do

\_\_\_\_\_ (1);

if (c = '-') then

negative := 1

```

else
    (2);
read(c);
while (3) do
begin
    (4);
    read(c);
end;
if negative = 1 then
    (5);
exit(num);
end;

begin
    a := readint();
    b := readint();
    writeln(a);
    writeln(b);
end.

```

2. (郊游活动) 有  $n$  名同学参加学校组织的郊游活动，已知学校给这  $n$  名同学的郊游总经费为  $A$  元，与此同时第  $i$  位同学自己携带了  $M_i$  元。为了方便郊游，活动地点提供  $B(\geq n)$  辆自行车供人租用，租用第  $j$  辆自行车的价格为  $C_j$  元，每位同学可以使用自己携带的钱或者学校的郊游经费，为了方便账务管理，每位同学只能为自己租用自行车，且不会借钱给他人，他们想知道最多有多少位同学能够租用到自行车。(第四、五空 2.5 分，其余 3 分)

本题采用二分法。对于区间  $[l, r]$ ，我们取中间点  $mid$  并判断租用到自行车的人数能否达到  $mid$ 。判断的过程是利用贪心算法实现的。

```

const
    maxn = 1000000;
type
    arr = array[1..MAXN] of longint;
var
    n, B, A, l, r, ans, mid, i: longint;
    M, C: arr;

function check(nn: longint): boolean;
var
    count, i, j: longint;
begin
    count := 0;
    i := (1);

```



```

j := 1;
while i <= n do
begin
    if (2) then
        inc(count, C[j] - M[i]);
    inc(i);
    inc(j);
end;
exit((3));
end;

procedure sort(var a: arr; l: longint; r: longint);
var
    i, j, x, y: longint;
begin
    i := l; j := r; x := a[(l + r) div 2];
    while i <= j do
    begin
        while a[i] < x do inc(i);
        while a[j] > x do dec(j);
        if i <= j then
        begin
            y := a[i]; a[i] := a[j]; a[j] := y;
            inc(i); dec(j);
        end;
    end;
    if i < r then sort(a, i, r);
    if l < j then sort(a, l, j);
end;

begin
    read(n, B, A);
    for i := 1 to n do
        read(M[i]);
    for i := 1 to B do
        read(C[i]);
    sort(M, 1, n);
    sort(C, 1, B);
    l := 0;
    r := n;
    while l <= r do
    begin
        mid := (l + r) div 2;

```

```
    if (4) then
    begin
        ans := mid;
        l := mid + 1;
    end
    else
        r := (5);
    end;
    writeln(ans);
end.
```