

2021 CCF 非专业级别软件能力认证第一轮
(CSP-J1) 入门级 Pascal 语言试题
认证时间：2021 年 9 月 19 日 14:30~16:30

考生注意事项：

- 试题纸共有 12 页，答题纸共有 1 页，满分 100 分。请在答题纸上作答，写在试题纸上的
一律无效。
- 不得使用任何电子设备（如计算器、手机、电子词典等）或查阅任何书籍资料。

一、单项选择题（共 15 题，每题 2 分，共计 30 分；每题有且仅有一个正确选项）

1. 以下不属于面向对象程序设计语言的是（ ）。

- A. C++
- B. Python
- C. Java
- D. C

2. 以下奖项与计算机领域最相关的是（ ）。

- A. 奥斯卡奖
- B. 图灵奖
- C. 诺贝尔奖
- D. 普利策奖

3. 目前主流的计算机储存数据最终都是转换成（ ）数据进行储存。

- A. 二进制
- B. 十进制
- C. 八进制
- D. 十六进制

4. 以比较作为基本运算，在 N 个数中找出最大数，最坏情况下所需要的最少的比较次数为（ ）。

- A. N^2
- B. N
- C. $N-1$

D. $N+1$

5. 对于入栈顺序为 a, b, c, d, e 的序列，下列 () 不是合法的出栈序列。

A. a, b, c, d, e

B. e, d, c, b, a

C. b, a, c, d, e

D. c, d, a, e, b

6. 对于有 n 个顶点、 m 条边的无向连通图 ($m > n$)，需要删掉 () 条边才能使其成为一棵树。

A. $n-1$

B. $m-n$

C. $m-n-1$

D. $m-n+1$

7. 二进制数 101.11 对应的十进制数是 ()。

A. 6.5

B. 5.5

C. 5.75

D. 5.25

8. 如果一棵二叉树只有根结点，那么这棵二叉树高度为 1。请问高度为 5 的完全二叉树有 () 种不同的形态？

A. 16

B. 15

C. 17

D. 32

9. 表达式 $a*(b+c)*d$ 的后缀表达式为()，其中“*”和“+”是运算符。

A. $**a+bcd$

B. $abc+*d*$

C. $abc+d**$

D. $*a*+bcd$

10. 6 个人，两个人组一队，总共组成三队，不区分队伍的编号。不同的组队情况有（ ）种。

A. 10

B. 15

C. 30

D. 20

11. 在数据压缩编码中的哈夫曼编码方法，在本质上是一种（ ）的策略。

A. 枚举

B. 贪心

C. 递归

D. 动态规划

12. 由 1, 1, 2, 2, 3 这五个数字组成不同的三位数有（ ）种。

A. 18

B. 15

C. 12

D. 24

13. 考虑如下递归算法

```
solve(n)
    if n<=1 return 1
    else if n>=5 return n*solve(n-2)
    else return n*solve(n-1)
```

则调用 `solve(7)` 得到的返回结果为（ ）。

A. 105

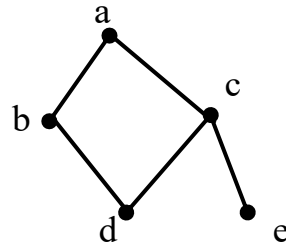
B. 840

C. 210

D. 420

14. 以 a 为起点，对右边的无向图进行深度优先遍历，则 b、c、d、e 四个点中有可能作为最后一个遍历到的点的个数为（ ）。

- A. 1
- B. 2
- C. 3
- D. 4



15. 有四个人要从 A 点坐一条船过河到 B 点，船一开始在 A 点。该船一次最多可坐两个人。已知这四个人中每个人独自坐船的过河时间分别为 1, 2, 4, 8，且两个人坐船的过河时间为两人独自过河时间的较大者。则最短（ ）时间可以让四个人都过河到 B 点（包括从 B 点把船开回 A 点的时间）。

- A. 14
- B. 15
- C. 16
- D. 17

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断题正确填√，错误填×；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

(1)

```
01 var
02   n, i, res: longint;
03   a: array[0..999] of longint;
04
05 function f(x: longint): longint;
06   begin
07     res := 0;
08     while (x <> 0) do
09       begin
10         inc(res);
11         x := x and (x - 1);
12       end;
13     f := res;
14   end;
15
16 function g(x: longint): longint;
17   begin
18     g := x and (-x);
19   end;
20
```

```

21 begin
22     read(n);
23     for i := 0 to n - 1 do read(a[i]);
24     for i := 0 to n - 1 do
25         write(f(a[i]) + g(a[i]), ' ');
26     writeln;
27 end.

```

• 判断题

16. 输入的 n 等于 1001 时，程序不会发生下标越界。（ ）

17. 输入的 $a[i]$ 必须全为正整数，否则程序将陷入死循环。（ ）

18. 当输入为“5 2 11 9 16 10”时，输出为“3 4 3 17 5”。（ ）

19. 当输入为“1 511998”时，输出为“18”。（ ）

20. 将源代码中 g 函数的定义（16-19 行）移到主程序的后面，程序可以正常编译运行。（ ）

• 单选题

21. 当输入为“2 -65536 2147483647”时，输出为（ ）。

- A. “65532 33” B. “65552 32” C. “65535 34” D. “65554 33”

(2)

```

01 var
02     base: array[0..63] of shortint;
03     table: array[0..255] of shortint;
04     i, len: longint;
05     str, ret: string;
06
07 procedure init;
08     begin
09         for i := 0 to 25 do base[i] := ord('A') + i;
10         for i := 0 to 25 do base[26 + i] := ord('a') + i;
11         for i := 0 to 9 do base[52 + i] := ord('0') + i;
12         base[62] := ord('+'); base[63] := ord('/');
13
14         for i := 0 to 255 do table[i] := $ff;
15         for i := 0 to 63 do table[base[i]] := i;
16         table[ord('=')] := 0;
17     end;
18

```

```

19 function decode(str: string): string;
20     begin
21         len := length(str);
22         i := 1;
23         while (i <= len) do
24             begin
25                 ret := ret + chr((table[ord(str[i])] shl 2) or
26                                     (table[ord(str[i + 1])] shr 4));
27                 if (str[i + 2] <> '=') then
28                     ret := ret + chr((table[ord(str[i + 1])] and
29                                         $0f) shl 4 or table[ord(str[i + 2])] shr 2);
30                 if (str[i + 3] <> '=') then
31                     ret := ret + chr(table[ord(str[i + 2])] shl 6 or
32                                         table[ord(str[i + 3])]);
33                 i := i + 4;
34             end;
35         decode := ret;
36     end;
37
38 begin
39     init;
40     writeln(longint(table[0]));
41     read(str);
42     writeln(decode(str));
43 end.

```

● 判断题

22. 输出的第二行一定是由小写字母、大写字母、数字和“+”、“/”、“=”构成的字符串。（ ）

23. 可能存在输入不同，但输出的第二行相同的情形。（ ）

24. 输出的第一行为“-1”。（ ）

● 单选题

25. 设输入字符串长度为 n ，decode 函数的时间复杂度为（ ）。

- A. $\theta(\sqrt{n})$ B. $\theta(n)$ C. $\theta(n \log n)$ D. $\theta(n^2)$

26. 当输入为“Y3Nx”时，输出的第二行为（ ）。

- A. “csp” B. “csq” C. “CSP” D. “Csp”

27. (3.5 分) 当输入为“Y2NmIDIwMjE=”时，输出的第二行为（ ）。

- A. “ccf2021” B. “ccf2022” C. “ccf 2021” D. “ccf 2022”

(3)

```
01 const
02     n = 100000;
03 var
04     i, j, k, x, m: longint;
05     a, b, c, d, f, g: array[0..n] of longint;
06
07 procedure init;
08     begin
09         f[1] := 1; g[1] := 1;
10         for i := 2 to n do
11             begin
12                 if (a[i] = 0) then
13                     begin
14                         b[m] := i; inc(m);
15                         c[i] := 1; f[i] := 2;
16                         d[i] := 1; g[i] := i + 1;
17                     end;
18                 j := 0;
19                 while (j < m) and (b[j] * i <= n) do
20                     begin
21                         k := b[j];
22                         a[i * k] := 1;
23                         if (i mod k = 0) then
24                             begin
25                                 c[i * k] := c[i] + 1;
26                                 f[i * k] := f[i] div c[i * k]
27                                     * (c[i * k] + 1);
28                                 d[i * k] := d[i];
29                                 g[i * k] := g[i] * k + d[i];
30                                 break;
31                             end
32                         else
33                             begin
34                                 c[i * k] := 1;
35                                 f[i * k] := 2 * f[i];
36                                 d[i * k] := g[i];
37                                 g[i * k] := g[i] * (k + 1);
38                             end;
39                         inc(j);
40                     end;
41                 end;
42             end;
43 begin
44     init;
45
46     read(x);
47     writeln(f[x], ' ', g[x]);
```

48 end.

假设输入的 x 是不超过 1000 的自然数，完成下面的判断题和单选题：

● 判断题

28. 若输入不为“1”，把第 9 行删去不会影响输出的结果。（ ）

29. (2 分) 第 26 行的“ $f[i] \text{ div } c[i * k]$ ”可能存在无法整除而向下取整的情况。（ ）

30. (2 分) 在执行完 `init()` 后， f 数组不是单调递增的，但 g 数组是单调递增的。（ ）

● 单选题

31. `init` 函数的时间复杂度为（ ）。

- A. $\theta(n)$ B. $\theta(n \log n)$ C. $\theta(n\sqrt{n})$ D. $\theta(n^2)$

32. 在执行完 `init()` 后， $f[1], f[2], f[3] \dots f[100]$ 中有（ ）个等于 2。

- A. 23 B. 24 C. 25 D. 26

33. (4 分) 当输入为“1000”时，输出为（ ）。

- A. “15 1340” B. “15 2340” C. “16 2340” D. “16 1340”

三、完善程序（单选题，每小题 3 分，共计 30 分）

(1) (Josephus 问题) 有 n 个人围成一个圈，依次标号 0 至 $n-1$ 。从 0 号开始，依次 0,1,0,1,... 交替报数，报到 1 的人会离开，直至圈中只剩一个人。求最后剩下人的编号。

试补全模拟程序。

```
01 const
02     MAXN = 1000000;
03 var
04     F: array[0..MAXN-1] of longint;
05     n, i, p, c, ans: longint;
06 begin
07     read(n);
08     while (①) do
09         begin
10             if (F[i] = 0) then
11                 begin
12                     if (②) then
13                         begin
```



```

14                                     F[i] := 1;
15                                     ③;
16                                     end;
17                                     ④;
18                                     end;
19                                     ⑤;
20                                     end;
21     ans := -1;
22     for i := 0 to n - 1 do
23         if (F[i] = 0) then
24             ans := i;
25         writeln(ans);
26 end.

```

34. ①处应填 ()

- A. $i < n$ B. $c < n$ C. $i < n - 1$ D. $c < n - 1$

35. ②处应填 ()

- A. $i \bmod 2 = 0$ B. $i \bmod 2 = 1$ C. $p \neq 0$ D. $p = 0$

36. ③处应填 ()

- A. $\text{inc}(i)$ B. $i := (i + 1) \bmod n$
C. $\text{inc}(c)$ D. $p := p \text{ xor } 1$

37. ④处应填 ()

- A. $\text{inc}(i)$ B. $i := (i + 1) \bmod n$
C. $\text{inc}(c)$ D. $p := p \text{ xor } 1$

38. ⑤处应填 ()

- A. $\text{inc}(i)$ B. $i := (i + 1) \bmod n$
C. $\text{inc}(c)$ D. $p := p \text{ xor } 1$

(2) (矩形计数) 平面上有 n 个关键点, 求有多少个四条边都和 x 轴或者 y 轴平行的矩形, 满足四个顶点都是关键点。给出的关键点可能有重复, 但完全重合的矩形只计一次。

试补全枚举算法。

```

01 type
02     point = record
03         x, y, id: longint;
04     end;

```

```

05 const
06     MAXN = 1000;
07 var
08     A: array[0..MAXN-1] of point;
09     n, i, j, ans: longint;
10
11 function op(a, b, c: boolean): boolean;
12     begin
13         if a then op := b else op := c;
14     end;
15
16 function equals(a, b: point): boolean;
17     begin
18         equals := (a.x = b.x) and (a.y = b.y);
19     end;
20
21 function cmp(a, b: point): boolean;
22     begin
23         cmp := ①;
24     end;
25
26 procedure sort(n: longint);
27     var
28         i, j: longint;
29         t: point;
30     begin
31         for i := 0 to n - 1 do
32             for j := 1 to n - 1 do
33                 if (cmp(A[j], A[j - 1])) then
34                     begin
35                         t := A[j];
36                         A[j] := A[j - 1];
37                         A[j - 1] := t;
38                     end;
39             end;
40
41 function unique(n: longint): longint;
42     var
43         i, t: longint;
44     begin
45         t := 0;
46         for i := 0 to n - 1 do
47             if (②) then
48                 begin
49                     A[t] := A[i];
50                     inc(t);
51                 end;
52         unique := t;
53     end;

```

```

54
55 function binary_search(n, x, y: longint): boolean;
56     var
57         p: point;
58         l, r, mid: longint;
59     begin
60         p.x := x;
61         p.y := y;
62         p.id := n;
63         l := 0; r := n - 1;
64         while (l < r) do
65             begin
66                 mid := ③;
67                 if (④) then
68                     l := mid + 1
69                 else
70                     r := mid;
71             end;
72         binary_search := equals(A[l], p);
73     end;
74
75 begin
76     read(n);
77     for i := 0 to n - 1 do
78         begin
79             read(A[i].x, A[i].y);
80             A[i].id := i;
81         end;
82     sort(n);
83     n := unique(n);
84     ans := 0;
85     for i := 0 to n - 1 do
86         for j := 0 to n - 1 do
87             if (⑤) and binary_search(n, A[i].x, A[j].y) and
                binary_search(n, A[j].x, A[i].y) then
88                 inc(ans);
89     writeln(ans);
90 end.

```

39. ①处应填 ()

- A. op(a.x != b.x, a.x < b.x, a.id < b.id)
- B. op(a.x != b.x, a.x < b.x, a.y < b.y)
- C. op(equals(a, b), a.id < b.id, a.x < b.x)
- D. op(equals(a, b), a.id < b.id, op(a.x != b.x, a.x < b.x, a.y < b.y))

40. ②处应填 ()

- A. $(i = 0) \text{ or cmp}(A[i], A[i - 1])$
- B. $(t = 0) \text{ or equals}(A[i], A[t - 1])$
- C. $(i = 0) \text{ or not cmp}(A[i], A[i - 1])$
- D. $(t = 0) \text{ or not equals}(A[i], A[t - 1])$

41. ③处应填 ()

- A. $r - (r - 1) \text{ div } 2 + 1$
- B. $(1 + r + 1) \text{ shr } 1$
- C. $(1 + r) \text{ shr } 1$
- D. $1 + (r - 1 + 1) \text{ div } 2$

42. ④处应填 ()

- A. $\text{not cmp}(A[\text{mid}], p)$
- B. $\text{cmp}(A[\text{mid}], p)$
- C. $\text{cmp}(p, A[\text{mid}])$
- D. $\text{not cmp}(p, A[\text{mid}])$

43. ⑤处应填 ()

- A. $A[i].x = A[j].x$
- B. $A[i].id < A[j].id$
- C. $(A[i].x = A[j].x) \text{ and } (A[i].id < A[j].id)$
- D. $(A[i].x < A[j].x) \text{ and } (A[i].y < A[j].y)$