

2022 CCF 非专业级别软件能力认证第一轮

(CSP-J)入门级 C++语言试题 模拟卷 - 2

考生注意事项:

1. 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。
2. 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答案)

1. 一个 8 位二进制数, 其原码为 10010101, 则其补码是 ()
A. 10010101 B. 01101010 C. 11101010 D. 11101011
2. 请你计算 $(A3)_{16} + (233)_{10} =$ ()
A. $(394)_{10}$ B. $(1AC)_{16}$ C. $(614)_8$ D. $(110011100)_2$
3. 双向链表中有两个指针域 llink 和 rlink, 分别指向前驱及后继, 设 p 指向链中的一个结点, q 指向一个待插入的结点, 现要求在 p 前插入 q, 则正确的插入为 ()。
A. $p \rightarrow \text{llink} = q; q \rightarrow \text{rlink} = p; p \rightarrow \text{llink} \rightarrow \text{rlink} = q; q \rightarrow \text{llink} = p \rightarrow \text{llink};$
B. $q \rightarrow \text{llink} = p \rightarrow \text{llink}; p \rightarrow \text{llink} \rightarrow \text{rlink} = q; q \rightarrow \text{rlink} = p; p \rightarrow \text{llink} = q \rightarrow \text{rlink};$
C. $q \rightarrow \text{rlink} = p; p \rightarrow \text{rlink} = q; p \rightarrow \text{llink} \rightarrow \text{rlink} = q; q \rightarrow \text{rlink} = p;$
D. $p \rightarrow \text{llink} \rightarrow \text{rlink} = q; q \rightarrow \text{rlink} = p; q \rightarrow \text{llink} = p \rightarrow \text{llink}; p \rightarrow \text{llink} = q;$
4. TCP/IP 网络协议共有几层 ()
A. 3 B. 4 C. 6 D. 7
5. 中国计算机学会(CCF)成立于 () 年。
A. 1962 B. 1979 C. 1984 D. 2001
6. 如果一台计算机的内存大小为 128M, 且按字节编址, 表示该内存地址至少需要 () 位
A. 32 B. 27 C. 25 D. 16
7. 在程序运行过程中, 如果递归调用的层数过多, 会因为 () 引发错误
A. 系统分配的栈空间溢出 B. 系统分配的队列空间溢出
C. 系统分配的堆空间溢出 D. 系统分配的链表空间溢出
8. 若串 $S = \text{"copyright"}$, 其连续子串的个数是 ()
A. 72 B. 45 C. 46 D. 36
9. 中缀表达式 $(A+B*C)*(D+E)-F-G$ 的后缀表达式是 ()
A. $DE+BC*A+*F-G-$ B. $ACB*+ED+F-G-$
C. $BC*A+DE+*F-G-$ D. $ABC*+DE+*F-G-$
10. 一棵二叉树总共有 23 个结点, 其中叶子节点的个数是 11, 请问这棵二叉树最多有 () 层。规定根节点为第 1 层。
A. 11 B. 13 C. 15 D. 16

11. A,B,C 三位同学关系非常要好,老师每次安排队列时总会让 A,B,C 这 3 人排在相邻的位置。老师也知道 A 同学和 D 同学的关系非常差,因此老师不会将 A 和 D 排在相邻的位置。已知班里共有 8 名同学,问老师排队的方案数有多少种 ()
- A. 4320 B. 2560 C. 3160 D. 3840
12. 在 $[1,100]$ 区间种任选两个数 x 和 y ,同时需要满足 $x \leq y$ 且 $x+y \geq 50$ 的方案有 ()
- A. 4350 B. 4374 C. 4425 D. 4450
13. 如已知 6 个结点的二叉树的先根遍历是 1 2 3 4 5 6 (数字为结点的编号,以下同),后根遍历是 3 2 5 6 4 1,则该二叉树的可能的中根遍历是 ()
- A. 3 2 1 4 6 5 B. 3 2 1 5 4 6
C. 3 1 2 5 4 6 D. 2 3 1 4 6 5
14. 序列 $[5,4,2,6,3,1]$,如果只能进行相邻元素的两两交换,将这个序列变成有序序列,至少需要进行 () 次交换。
- A. 4 B. 6 C. 11 D. 12
15. 在一个具有 n 个顶点的无向图中,要连通全部顶点至少需要 () 条边。
- A. n B. $n + 1$ C. $n - 1$ D. $n/2$

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断正确填√，错误填×；共计40分）

1.

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  char str[1005];
5  int w1 = 0, w2 = 0;
6  int main() {
7      scanf("%s", str);
8      int n = strlen(str);
9      for(int i = 0; i < n; i++) {
10         if (str[i]>='a' && str[i]<='z') str[i]=str[i]-'a'+'A';
11         if (str[i]>='A' && str[i]<='Z') str[i]=str[i]-'A'+'a';
12     }
13     for(int i = 0; i < n; i++) {
14         if (str[i] >= 'a' && str[i] <= 'z') {
15             w1 = w1 + i % 2;
16             char ch1 = str[i] - 'a' + 'A';
17             printf("%c", ch1);
18         }
19         if (str[i] >= 'A' && str[i] <= 'Z') {
20             w2 = w2 + i % 2;
21             char ch2 = str[i] - 'A' + 'a';
22             printf("%c", ch2);
23         }
24     }
25     printf("\n%d\n", w1+w2);
26     return 0;
27 }
```

约定输入字符串长度不大于888，且不含空格。

● 判断题（每题2分）

- 1) 如果输入的字串中包含数字，程序运行时会发生错误。（ ）
- 2) 如果输入的字符全部是字母，程序运行至第27行时， $(w1+w2)*2=n$ 一定成立。（ ）
- 3) 如果输入的字符串全部都是小写字母，则输出的字符串全部都是大写字母。（ ）
- 4) 如果输入的字符串全部都是大写字母，则输出的字符串全部都是小写字母。（ ）

● 选择题（每题2.5分）

- 5) 如果输入是"A1A2A3b4b5C6C7C8d9d0"，第25行输出的结果是？（ ）
A. 0 B. 5 C. 6 D. 10
- 6) 如果程序输入的长度是19，那么第25行输出最大为？（ ）
A. 0 B. 9 C. 10 D. 19

2.

```
1  #include <iostream>
2  using namespace std;
3  int n;
4  int ff(int x) {
5      int t = 0;
6      for(int i = 2; i < x; i++) {
7          if (x % i == 0) t += ff(i);
8      }
9      if (t == 0) t = x-1;
10     return t;
11 }
12
13 int main() {
14     cin >> n;
15     cout << ff(n) << endl;
16     return 0;
17 }
```

约定输入数据 n 是区间 $[0, 10000]$ 内的整数。

● 判断题（每题2分）

- 1) 如果 $n=0$ ，程序运行时会发生错误。（ ）
- 2) 第6行，将 $i < x$ 修改为 $i * i \leq x$ ，程序运行的结果不会发生改变。（ ）
- 3) 第6行，将 $i < x$ 修改为 $i \leq x$ ，程序运行的结果不会发生改变。（ ）
- 4) 第6行，将 $i = 2$ 修改为 $i = 1$ ，程序运行的结果不会发生改变。（ ）

● 选择题（每题2.5分）

- 5) 如果输入 $n=5$ ，程序输出的结果是（ ）
A. 3 B. 4 C. 5 D. 6
- 6) 如果输入 $n=120$ ，程序输出的结果是（ ）
A. 37 B. 40 C. 119 D. 164

3.

```
1  #include <iostream>
2  using namespace std;
3
4  int n, m, f[1001], T = 0, g[1001];
5
6  int find(int x) {
7      while(x != f[x]) {
8          x = f[x];
9      }
10     return f[x];
11 }
12
13 int main() {
14     int a, b;
15     cin >> n >> m;
16     for(int i = 1; i <= n; i++) f[i] = i;
17     for(int i = 1; i <= m; i++) {
18         cin >> a >> b;
19         int la = find(a);
20         int lb = find(b);
21         if (la == lb) {
22             f[a] = a;
23             f[b] = b;
24         }
25         else {
26             f[la] = lb;
27         }
28     }
29     for(int i = 1; i <= n; i++) {
30         if (f[i] == i) T++;
31         else {
32             g[ find(i) ] ++;
33         }
34     }
35     cout << T << endl;
36     for(int i = 1; i <= n; i++) {
37         cout << g[i] << ' ';
38     }
39     return 0;
40 }
```

题目保证 n, m 在 $[1, 1000]$ 范围内。保证读入的 a, b 在 $[1, n]$ 范围内，且保证每次读入时的 $a \neq b$ 。

● 判断题（每题2分）

- 1) 第35行输出的T不可能为0。（ ）
- 2) 如果将第32行修改为 `g[f[i]]++`；程序输出的结果不会改变。（ ）
- 3) 如果输入的a均不相同，则35行输出的T不可能为n。（ ）
- 4) 如果输入的b均不相同，则35行输出的T不可能为n。（ ）

● 选择题（每题3分）

- 5) 如果 $n=100, m=51$ ，同时满足输入的a均不相同，则T最大可能是多少？（ ）
A. 49 B. 50 C. 99 D. 100
- 6) 如果 $n=100, m=51$ ，同时满足输入的a均不相同，则T最小可能是多少？（ ）
A. 1 B. 49 C. 50 D. 51

三. 完善程序（单选题，每小题 3 分，共计 30 分）

1.(修改数列) 已知数组 `a` 中有 `n` 个数字，其中仅包含数字 0 和 1 且保证 `a[n]=1`；

可以对数组进行两种修改操作：

1. 将 `a[i]=0` 修改为 `a[i]=1`；

2. 将 `a[i]=1` 修改为 `a[i]=0`；

现想要通过修改操作使得数组 `a` 前 `k` 个数字为 0，后 `n-k` 个数字为 1（ $1 \leq k \leq n$ ），输出最少操作次数是多少及 `k`。当包含多个满足要求的最少操作次数时，输出 `k` 最大的那一组答案；例如输入为

5

0 1 1 0 1

最少操作次数为 1, `k` 为 1，只需将 `a[4]` 修改为 1 即可。

数据范围 $2 \leq n \leq 1000000$, $0 \leq a[i] \leq 1$ ；

试补全程序。

```

1  #include<cstdio>
2  using namespace std;
3  int a[1000000],s[1000000];
4  int main() {
5      int n, mins;
6      scanf("%d", &n);
7      for (int i = 1; i <= n; i++) {
8          scanf("%d", &a[i]);
9          s[i] = ____ (1) ____;
10     }
11     mins = ____ (2) ____;
12     int k = n;
13     for (int i = 1; i <= n; i++) {
14         int sum1 = s[i];
15         int sum2 = ____ (3) ____;
16         if(sum1 + sum2 ____ (4) ____ mins) {
17             mins = sum1 + sum2;
18             k = ____ (5) ____;
19         }
20     }
21     cout << mins << " " << k << endl;
22     return 0;
23 }

```

- 1) (1)处应填 ()
- A. s[i-1]+a[i-1] B. a[i]+a[i-1]
C. s[i-1]+a[i] D. i
- 2) (2)处应填 ()
- A. 0 B. a[1] C. s[n] D. 1
- 3) (3)处应填 ()
- A. s[n]-s[i] B. s[n]-s[i-1]
C. (n-i)-(s[n]-s[i]) D. n-i-1
- 4) (4)处应填 ()
- A. >= B. > C. <= D. <
- 5) (5)处应填 ()
- A. i B. i-1 C. n-i-1 D. n-i

2. (二叉查找树) 二叉查找树具有如下性质：每个节点的值都大于其左子树上所有节点的值、小于其右子树上所有节点的值。试判断一棵树是否为二叉查找树。输入的第 1 行包含一个整数 n, 表示这棵树有 n 个顶点，编号分别为 1, 2, ..., n, 其中编号为 1 的是根节点。之后的第 i 行有 3 个数 value, left_child, right_child, 分别表示该节点关键字的值、左子节点的编号、右子节点的编号；如果不存在左子节点或右子节点，则用 0 代替。输出 1 表示这棵树是二叉查找树，输出 0 则表示不是。

1	#include <iostream>
---	---------------------

```

2  using namespace std;
3  const int SIZE=100;
4  const int INFINITE=1000000;
5  struct node{
6      int left_child, right_child, value;
7  };
8  node a[SIZE] ;
9  int is_bst (int root,int lower_bound,int upper_bound)
10 {
11     if (root==0) return 1;
12     int cur = ____ (1) ____;
13     if(cur>lower_bound && cur<upper_bound &&
14         (is_bst( ____ (2) ____ ,lower_bound, cur)==1) &&
15         (is_bst( ____ (3) ____ ,cur,upper_bound) ____ (4) ____ ))
16     {
17         return 1;
18     }
19     return 0;
20 }
21 int main()
22 {
23     int n;
24     cin >> n;
25     for(int i=1; i<=n; i++)
26         cin >> a[i].value >> a[i].left_child >> a[i].right_child;
25     cout << is_bst( ____ (5) ____ , -INFINITE, INFINITE);
26     return 0;
27 }

```

(1)处应填()

- A. root B. a[root] C. a[root].value D. 1

(2)处应填()

- A. left_child B. right_child
C. a[root].left_child D. a[root].right_child

(3)处应填()

- A. left_child B. right_child
C. a[root].left_child D. a[root].right_child

(4)处应填()

- A. == 0 B. == 1 C. ==0 或者空白 D. ==1 或者空白

(5)处应填()

- A. -1 B. 0 C. 1 D. 2