2023 LGR 非专业级别软件能力认证第一轮

（SCP-J）入门级 C++语言模拟试题

认证时间：2023 年 8 月 14 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. 网址 www.luogu.com.cn 当中，顶级域名是（　　）。

A. www
B. cn
C. com.cn
D. luogu

2. 以下物品可以携带进 CSP 第二轮测试考场的是（　　）。

A. 可以发出巨大响声的发光机械键盘
B. 写有 kkksc03 签名的《深入浅出程序设计竞赛 基础篇》的封皮
C. 印有 dzd 照片（已设法获得其授权）的文化衫 T 恤
D. 具有拍照功能的游标卡尺

3. 将元素 a,b,c,d,e,f 依次入栈，则以下选项中出栈序列不可能是（　　）。

A. b,d,c,f,e,a　　B. f,e,d,c,b,a
C. d,c,e,b,f,a　　D. d,c,f,b,e,a

4. 「流程结构」是编程中用于控制程序执行流程的一种方式．它包括顺序结构、分支结构和循环结构．在一些诗歌作品中，也有对「流程结构」的体现．下列诗歌片段中体现循环结构的是（　　）。

A. 如果还能找到你，就让风儿告诉你。

——《Artificial Emotions》

B. 只要我还能够行走，只要我还能够张望，只要我还能够呼吸，就一直走向前方。

——《Песня отревожной молодости》

C. 昔闻洞庭水，今上岳阳楼。

——《登岳阳楼》

D．啊如果我在，战斗中牺牲，啊朋友再见吧，再见吧，再见吧！如果我在，战斗中牺牲，你一定把我来埋葬。

——《Bella Ciao》

5． 已知两个二进制整数 a,b： $a = 1010001010_{(2)}; b = 1110100110_{(2)}$

则表达式 (a&b)^(a|b) 的值为（　　）。

A．$0011011010_{(2)}$
B．$0100101100_{(2)}$
C．$0011010010_{(2)}$
D．$0100101000_{(2)}$

6． 一个有 10 个节点的有向图，要使得每一个满足 $1 \le i,j \le 10, i \ne j$ 的点对 (i,j) 都存在一条从 i 到达 j 的路径，至少需要连（　　）条有向边。

A．9　　B．10
C．19　　D．20

7． 观察下列代码

```
int  a[] = {5, 4, 3, 2, 1};
auto p = a + 3;
auto q = &p;
(*q) ++;
auto k = *p;
```

其中，k 的类型以及 k 的值分别为（）。

A．int 类型，值为 1
B．int 类型，值为 3
C．int 指针类型，值为 a 数组的下标为 3 的元素的地址
D．int 指针类型，值为 a 数组的下标为 4 的元素的地址

8． 中缀表达式 a+(b+c)-d*e/f 改写成后缀表达式为（　　）。

A．a b + c + d e * f / -
B．a b c + + d e * f / -
C．a b + c + d * e f / -
D．a b c + + d * e / f -

9． 一张大小为 6114×8192 的 24 位彩色图片，使用 .bmp 格式存储，占用的空间大小约为（　　）。

A．144 MiB　　B．288 MiB
C．1152 MiB　　D．48 MiB

10．以下程序片段的时间复杂度为（　　）。

```
int cnt = 0;
 for(int i = 1;i <= n;i ++){
     for(int j = 1;j <= n;j += i){
```

```
            for(int k = 1;k <= n;k += j){
                ++ cnt;
            }
        }
    }
```

提示：

$$\frac{n}{1^1}+\frac{n}{2^1}+\frac{n}{3^1}+\cdots+\frac{n}{n^1}\approx C_1\times\log n$$

$$\frac{n}{1^2}+\frac{n}{2^2}+\frac{n}{3^2}+\cdots+\frac{n}{n^2}\approx C_2$$

其中，C_1, C_2 均为常数。

A. $\Theta(n^2)$　　B. $\Theta(n^2\log n)$

C. $\Theta(n\log n)$　　D. $\Theta(n\log^2 n)$

11. 依次抛出四个六面骰子，按照抛出顺序将骰子上的数值记为 a, b, c, d。则 `a < b; b > c; c < d` 同时成立的概率为（　　）。

A. 95/648　　B. 4/27

C. 5/27　　D. 1/6

12. 十进制小数 0.3，转写成八进制为（　　）。

A. 0.3　　B. 0.2314631…

C. 0.2046204…　　D. 0.3333333…

13. 观察如下代码片段：

```
union U{
    bool flag1, flag2, flag3, flag4, flag5;
    signed   short a;
    unsigned short b;
    enum E{
        CardA = 0, CardB = 1,
        CardC = 2, CardD = 142857
    } e;
} u;
```

其中，`sizeof(u)` 的值为（　　）。

A. 4　　B. 8

C. 13　　D. 16

14. 已知某种可用来维护序列的数据结构，支持 $\Theta(\log n)$ 向某个位置后面插入元素、$\Theta(n)$ 查询某个元素的排名，$\Theta(n\log n)$ 遍历整个序列，那么用上述三种操作实现插入排序的时间复杂度最坏为（　　）。

A. $\Theta(n^2)$　　B. $\Theta(\mathrm{n}^2\log\mathrm{n})$

C. $\Theta(\mathrm{n}\log\mathrm{n})$　　D. $\Theta(\mathrm{n}\log^2\mathrm{n})$

15. 今年是 CCF（中国计算机学会）第（　　）次举办 CSP-J/S（计算机非专业级别的软件能力认证）？

A. 27　　B. 28

C. 5　　D. 4

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 2 分，选择题 3 分，共计 40 分）

(1)

```
#include <iostream>
#include <cstring>
using namespace std;
string s, t;
string a, b;
int main(){
    getline(cin, s);
    getline(cin, a);
    getline(cin, b);
    for(int i = 0;i < s.size();i ++){
        bool flag = true;
        if(i + a.size() <= s.size()){
            for(int j = 0;j < a.size();j ++){
                char p = s[i + j], q = a[j];
                if('a' <= p && p <= 'z') p = p - 'a' + 'A';
                if('a' <= q && q <= 'z') q = q - 'a' + 'A';
                if(p != q)
                    flag = false;
            }
        } else
            flag = false;

        if(flag == true){
            t += b;
            i += a.size() - 1;
        } else
            t += s[i];
    }
    cout << t << endl;
    return 0;
}
```

保证输入的三行字符串的长度均不超过 10^3，且每个字符串均由 ASCII 可视字符或空格组成且非空。完成下面的判断题和单选题：

- 判断题

16. 保持 s,b 的字母大小写不变，将 a 里的小写英文字母改写成大写、将大写英文字母改写成小写，输出结果不变。（ ）

17. 每次调用 s.size()的复杂度是 O(1) 的，但若是把 s.size()替换成 s.length() 则调用的复杂度将会变成 O(l)，其中 l 是 s 当前的长度。这是因为 s.length() 作为 strlen() 函数的 string 版本，会每次重新计算 s 最后一个元素的位置。（ ）

18. 当输入的字符串 s,a,b 均由小写字母 a 或 b 组成，记 s,a,b 的长度分别为 n,m,k，则程序的时间复杂度最坏为 O(nm+nk)。（ ）

19. 当输入的字符串 s,a,b 均由小写字母 a 组成，记 s,a,b 的长度分别为 n,m,k 且有 n>m>k，那么上述程序的总复杂度为 O(n)。（ ）

- 单选题

20. 针对下列输入数据，程序的输出为？。

```
National Olympiad in Informatics A154
A
C
```

A. National Olympiad in Informatics C154
B. NCtionCl OlympiCd in InformCtics C154
C. National!
D. Nctioncl Olympicd in Informctics C154

21. 针对下列输入数据，程序的输出为？。

```
abaabaaabaaaabaaaaababababab
aa
ab
```

A. abABbABBbABBBbABBBBbababab
B. ababbabbbabbbbabbbbbbababab
C. ababbababababbababababababab
D. 程序陷入死循环/非正常退出，无法正常输出

(2)

```
01 #include<iostream>
02 using namespace std;
03 int a[100005], b[100005], n, m;
04 void very_quick_sort(int l, int r, int p, int q){
05     if(l >= r || p > q){    // ①
06         return;
07     }
08     int mid = (l + r) / 2;
```

```
09      int p0 = p - 1;
10      int q0 = q + 1;
11      for(int i = p;i <= q;i ++){
12          if(a[i]  > mid) b[++ p0] = a[i];
13              else        b[-- q0] = a[i];
14      }
15      for(int i = p;i <= q;i ++)
16          a[i] = b[i];
17      very_quick_sort(mid + 1, r, p, p0);
18      very_quick_sort(l, mid, q0, q);
19  }
20  int main(){
21      cin >> n >> m;
22      for(int i = 1;i <= n;i ++)
23          cin >> a[i];
24      very_quick_sort(1, m, 1, n);
25          // ②
26      for(int i = 1;i <= n;i ++)
27          cout << a[i] << " ";
28      cout << endl;
29      return 0;
30  }
```

保证输入的 n 不超过 10^5，m 不超过 10^9，且 $1 \le a_1, a_2, \cdots, a_n \le m$。完成下面的判断题和单选题

- 判断题

22. (1 分)上述代码实现了一种排序算法，可以将 a 数组按照从小到大的顺序排序。（　　）

23. 如果在程序开始之前向 b 数组里写入数据（保证不会发生数组越界），则上述代码的输出不会发生变化。（　　）

24. 若 $n = m$，存在某种数据构造方式，使得上述代码运行的时间复杂度为 $O(n^2)$，这是因为算法本身是对快速排序的改进，但是这种改进不能避免由于对数组的划分不够均等而在极端数据下导致复杂度发生退化。（　　）

25. 如果将 ① 处的 l >= r 条件删除（同时删除 || 使得程序能正常编译运行，下同），程序的时间复杂度不会发生变化；而将 p > q 条件删除，程序在某些数据下的运行效率将会明显降低。（　　）

- 单选题

26. 不认为 n, m 同阶，即可能出现 n 远大于 m 或者 m 远大于 n 的情况。则该程序的最坏时间复杂度为（　　）。

A. $\Theta(n^2 + m^2)$　　B. $\Theta(\mathrm{m} \log \mathrm{m})$

C. Θ(m log n)　　D. Θ(n log m)

27. 若输入数据为：

```
10 10
10 4 5 2 2 3 1 5 8 3
```

那么在程序执行到 ② 位置时， b 数组 内的值为 （　　）。

A. [10,8,3,5,1,3,2,2,5,4]
B. [3,5,1,3,2,2,5,4,10,8]
C. [10,8,5,5,4,3,3,2,2,1]
D. [1,2,2,3,3,4,5,5,8,10]

(3)

```
01 #include <iostream>
02 #include <vector>
03 #include <algorithm>
04 using namespace std;
05 const int mod = 1000000000 + 7;
06 int w0[100005];
07 int w1[100005];
08 int w2[100005];
09 int n, m, k, f[100005], d[100005], id[100005];
10 vector <int> e[100005];
11 void dfs(int u, int fa){
12     d[u] = d[fa] + 1;
13     f[u] = fa;
14
15     for(auto &v : e[u]) if(v != fa){
16         dfs(v, u);
17     }
18 }
19 bool cmp(int a, int b){
20     return d[a] < d[b];
21 }
22 int main(){
23     cin >> n >> m >> k;
24     for(int i = 2;i <= n;i ++){
25         int u, v;
26         cin >> u >> v;
27         e[u].push_back(v);
28         e[v].push_back(u);
29     }
30     dfs(1, 0);                    // ①
31     for(int i = 1;i <= m;i ++){
32         int x, w;
```

```
33          cin >> x >> w;
34          w1[x] = (w1[x] + w) % mod;
35          w2[x] = (w2[x] + w) % mod;
36      }
37      for(int i = 1;i <= n;i ++)
38          id[i] = i;
39      sort(id + 1, id + 1 + n, cmp);
40      for(int i = 1;i <= k;i ++){
41          for(int j = n;j >= 1;j --){
42              int x = id[j];
43              for(auto &y : e[x]) if(y != f[x]){
44                  w1[y] = (w1[y] + w1[x]) % mod;
45              }
46              w1[x] = 0;
47          }
48          for(int x = 1;x <= n;x ++)
49              w1[x] = (w1[x] - w0[x] + mod) % mod,
50              w0[x] = 0;
51          for(int j = 1;j <= n;j ++){ // ②
52              int x = id[j];
53              if(f[x]){
54                  w1[f[x]] = (w1[f[x]] + w2[x]) % mod;
55                  w2[f[x]] = (w2[f[x]] + w2[x]) % mod;
56                  w0[x] = (w0[x] + w2[x]) % mod;
57                  w2[x] = 0;
58              }
59          }
60      }
61      for(int i = 1;i <= n;i ++)
62          cout << w1[i] << " ";
63      return 0;
64  }
```

保证输入的 n,m 不超过 10^5，k 不超过 20，且 $1\le x_i\le n$，$0\le w_i<10^9+7$。**完成下面的判断题和单选题：**

- 判断题

28. 如果更改 ① 处 dfs(1, 0) 为 dfs(n, 0)，则输出结果可能有变化。(　　)

29. （1 分）如果 k=n，那么输出结果均为 0。(　　)

30. 如果更改 ② 处 for(int j=1;j< n;j++) 为 for(int j=n;j>=1;j--)，那么对于任意合法的输入数据，更改前后程序的输出均相同。(　　)

- 单选题

31. （2 分）该程序的时间复杂度为（　　）。

A. $\Theta(nmk)$　　B. $\Theta(nk+m)$

C. $\Theta(km+n)$　　D. $\Theta(n+m+2^k)$

32. 对于以下的输入数据，输出结果为（　　）。

```
5 2 1
1 2
2 3
3 4
3 5
1 5
3 2
```

A. 0 7 0 2 2　　B. 2 7 2 3 2

C. 5 2 1 1 1　　D. 0 2 1 1 2

33. 对于以下的输入数据，输出结果为（　　）。

```
9 9 2
1 2
1 7
2 3
2 4
7 8
4 5
4 6
8 9 （接续右栏）
```

```
1 1
2 10
3 100
4 1000
5 10000
6 100000
7 1000000
8 10000000
9 100000000
```

A. 10001100 1110000 1001 101 100010 10010 100000010 1 1000000

B. 0 0 1 1 10 10 0 1 1000000

C. 11001110 1111100 1001 110101 100010 10010 110000010 100000001 1000000

D. 11001111 1111112 1121 111121 112010 112010 111000012 112000001 121000000

三、完善程序（单选题，每小题 3 分，共计 30 分）

(1)（序列问题）给定序列 a_n，求有多少对 (i,j) 满足 $a_i<a_j$。测试数据满足 $n\le10^6$，$a_i\le10^9$。

提示：对于任意的 $a_i\neq a_j$，可以发现 (i,j) 或 (j,i) 能对答案产生 1 的贡献，因此我们只需要用总的对数减去 $a_i=a_j$ 的 (i,j) 数量，就能得到答案。

试补全程序。

```
01  #include<bits/stdc++.h>
02  using namespace std;
03  const int Maxn = ①;
04  int n, a[Maxn];
```

```
05  long long s[Maxn], ans;
06  bool check(int l, int r) {
07      if (s[r] - s[l - 1] == ②) return true;
08      return false;
09  }
10  int cmp(int &x, int &y){
11      return a[x] < a[y];
12  }
13  int main() {
14      cin >> n;
15      for (int i = 1; i <= n; i ++)
16          cin >> a[i];
17      sort (a + 1, a + n + 1, cmp);
18      for (int i = 1; i <= n; i ++)
19          s[i] = s[i - 1] + a[i];
20      ans = ③;
21      for (int i = 1; i <= n;) {
22          int l = i, r = n, pos = n;
23          while (④) {
24              int mid = (l + r) >> 1;
25              if(check(l, mid))
26                  l = mid + 1, pos = mid;
27              else
28                  r = mid - 1;
29          }
30          ans -= ⑤;
31          i = pos + 1;
32      }
33      cout << ans;
34  }
```

34. ① 处应填（　　）。

A. 1e6 + 7　　B. 1000000
C. 1e6　　D. 1000000000000

35. ② 处应填（　　）。

A. (r - l + 1) * a[l]　　B. s[r] - s[l - 1]
C. 1LL * (r-l+1) * a[l]　　D. a[1] = 0

36. ③ 处应填（　　）。

A. n * (n + 1) / 2
B. 1ll * n * (n + 1) / 2
C. n * (n - 1) / 2
D. 1ll * n * (n - 1) / 2

37. ④ 处应填（　　）。

A. l < r　　B. l <= r

C. l <= r + 1　　D. l > r

38. ⑤ 处应填（　　）。

A. (pos - l + 1) * (pos - l) / 2

B. 1ll * (pos - l + 1) * (pos - l) / 2

C. (pos - i + 1) * (pos - i) / 2

D. 1ll * (pos - i + 1) * (pos - i) / 2

(2)（异或和）给定序列 a_n，求其所有子区间异或和的和。某个区间的异或和的意思是将这个区间内的所有数字分别进行异或计算。其中 $n \le 10^5$，$0 \le a_i \le 10^9$。

提示：对每一位独立计算，对右端点扫描线，并用异或前缀和辅助统计。

试补全程序。

```
01  #include<bits/stdc++.h>
02  using namespace std;
03  const int N = 1e5 + 7;
04  int n, a[N], cnt[2];
05  long long ans;
06  int main () {
07      cin >> n;
08      for (int i = 1; i <= n; i ++)
09          cin >> a[i], ①;
10      for (int bit = ②; ③; bit --) {
11          cnt[0] = cnt[1] = 0;
12          for (int i = 0; i <= n; i ++) {
13              cnt[④] ++;
14              ans += 1LL* ⑤;
15          }
16      } cout << ans;
17  }
```

39. ① 处应填（　　）。

A. a[i - 1] ^= a[i]　　B. a[i] ^= a[i - 1]

C. a[i - 1] += a[i]　　D. a[i] += a[i - 1]

40. ② 处应填（　　）。

A. n - 1　　B. 29

C. n　　D. n + 1

41. ③ 处应填（　　）。

A. bit　　B. bit >= n

C. bit - 1　　D. ~bit

42. ④ 处应填（　　）。

A. (a[i] >> bit) & 1　　B. a[i] & (1 << bit)

C. a[i] & (1LL << bit)　　D. (a[i] >> bit) ^ 1

43. ⑤ 处应填（　　）。

A. cnt[(a[i] >> bit) & 1] << i

B. cnt[(a[i] >> bit) & 1 ^ 1] << bit

C. cnt[(a[i] >> bit) & 1] << bit

D. cnt[(a[i] >> bit) & 1 ^ 1] << i

试题到此结束