

2024 LUOGU 非专业级别收容能力认证第一轮

(SCP-S1) 提高级 C++语言试题

认证时间：2024 年 8 月 18 日 14:30~16:30

考生注意事项：

- 试题纸共有 13 页，满分 100 分。请在洛谷作答，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。
- 试题由洛谷网校学术组命制，欢迎报名洛谷网校第一轮课程。课程内容包含专题讲解、真题讲评与本试题讲评。<https://class.luogu.com.cn/course/yugu24acs>

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. 下列编译命令，是 CSP-S 第二轮评测时用于编译题目英文名为 luogu 的传统型试题的选手代码的是（ ）。
 - gcc -std=c++14 -O2 -o * luogu.cpp
 - g++ -std=c++14 -O2 -o * luogu.cpp
 - g++ -O2 -o * luogu.cpp
 - g++ -std=c++14 -o * luogu.cpp
2. 在 2023 年的 CSP-S 第二轮认证中，有四位同学对于某一题得 0 分的情况提出了申诉。根据 CCF NOI 竞赛委员会的相关规定，下列申诉中可能被接受的是（ ）。
 - 小明在提交的程序中并未删除用于输出调试信息的代码。
 - 小红在提交的程序中使用了随机化算法。
 - 小黄提交的程序在考场的 Windows 电脑上正常编译，而在最终评测时编译失败。
 - 小陈在与评测环境相同的电脑上运行程序，使用了 850 毫秒运行得出正确答案，而该题的时间限制为 1 秒。程序运行并未超出内存限制。
3. 大型语言模型（LLM）指使用大量文本数据训练的深度学习模型，它们可以生成自然语言文本或理解语言文本的含义。下列人工智能技术的应用中，不属于 LLM 的是（ ）。
 - ChatGPT
 - Stable Diffusion
 - Gemini
 - 文心一言
4. $(47)_{10}$ 与下列（ ）选项表示的数值一致？

- A. $(56)_8$
- B. $(1100110)_2$
- C. $(142)_5$
- D. $(2D)_{16}$

5. 定义 a 在模 p 意义下的乘法逆元为同余方程 $ax \equiv 1 \pmod{p}$ 的解 x 。请计算 13 在模 29 意义下的乘法逆元是（ ）。

- A. 11
- B. 10
- C. 8
- D. 9

6. 仿照二项式系数，定义三项式系数 T_n^k 为 $(1+x+x^2)^n$ 的展开式中 x^k 项的系数。即：
 $(1+x+x^2)^n = T_n^0 + T_n^1x^1 + T_n^2x^2 + \dots + T_n^{2n}x^{2n}$ 。现要计算 $T_8^0 + T_8^1 + T_8^2 + \dots + T_8^{16}$ 的值，其结果为（ ）？

- A. 4096
- B. 6561
- C. 8192
- D. 15625

7. 当在二叉排序树中插入一个新结点时，若树中不存在与待插入结点关键字相同的结点，且待插入结点的关键字小于根结点的关键字，则新结点将成为根结点的（ ）？

- A. 左子树的某一叶子结点
- B. 右子树的某一叶子结点
- C. 左子树的某一非叶子结点
- D. 右子树的某一非叶子结点

8. 一个盒子里有 4 个红球、5 个蓝球和 6 个绿球。现在从盒子里随机抽取 3 个球，则这 3 个球中恰好有 1 个红球、1 个蓝球和 1 个绿球的概率是（ ）。

- A. $136/455$
- B. $24/91$
- C. $5/66$
- D. $1/11$

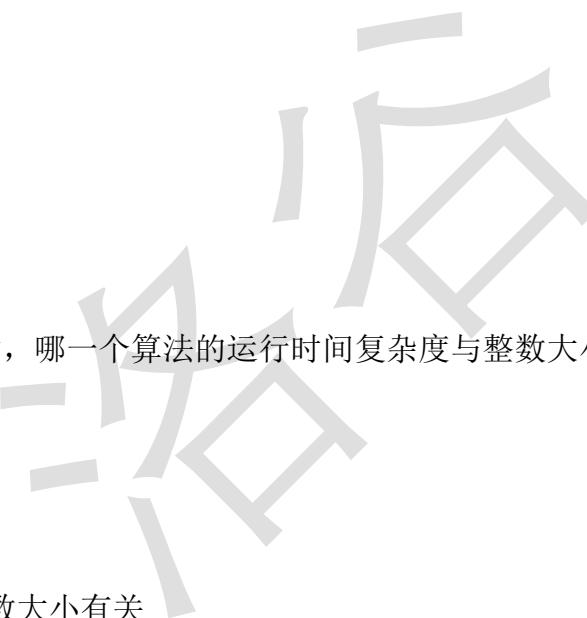
9. 对前缀表达式 `- + 10 * 2 3 + 5 / 6 2` 进行求值，得到的结果是（ ）。

- A. 8

- B. 9
- C. 10
- D. 11

10. 已知一个循环队列的存储空间为数组 `data q[21]` 且不浪费空间。且在队列的定义中，头指针指向队列的开头，尾指针指向队列末尾的下一个元素。现在若头指针和尾指针分别指向下标 8 和 3，则队列当前的长度为（ ）？

- A. 5
- B. 6
- C. 17
- D. 16

11. 在使用下列算法对整数数列进行排序时，哪一个算法的运行时间复杂度与整数大小无关（ ）。


- A. 基数排序
- B. 计数排序
- C. 希尔排序
- D. 以上排序算法的运行时间复杂度均与整数大小有关

12. 假设变量 `a, b, c` 均为绝对值不超过 10^9 的 32 位有符号整型变量，下列关于这三个变量，说法正确的是（ ）。

- A. 若执行代码 `auto d = a * b;` 则 `d` 的变量类型是 64 位有符号整型变量。
- B. 逻辑表达式 `max(a, b) * 1LL * c == max(1LL * a * c, 1LL * b * c)` 的运算结果一定为真。
- C. 执行代码 `cout << sizeof(a);` 后，输出的结果为 4。
- D. `for (unsigned i = a; i < b; i++)` 可能是一个死循环。

13. 对于以下代码，假设执行单次 `rand()` 函数的时间复杂度为 $O(1)$ ，那么 `calc` 函数的运行时间复杂度为？（ ）

```
unsigned int ans = 0;
void val(int n) {
    for (int i = 1; i <= n; ++ i)
        for(int j = 0; j < n; j += i)
            ans += rand();
}
```

```

void calc(int n) {
    if (n <= 1) val(n);
    else calc(n / 2), calc(n / 2), val(n);
}

```

- A. $\Theta(n)$
- B. $\Theta(n \log n)$
- C. $\Theta(n \log^2 n)$
- D. $\Theta(n \log^3 n)$

14. 给定一个无向图 G , 其中包含以下顶点和边:

顶点集合: $V = \{A, B, C, D, E\}$

边集合: $E = \{(A, B), (A, C), (B, C), (B, D), (C, D), (C, E)\}$

请问以下哪一个选项正确描述了图 G 的双连通性? ()。

- A. 图 G 是点双连通的, 因为删除任意一个顶点后, 图仍然连通。
- B. 图 G 不是点双连通的, 因为存在割点。
- C. 图 G 是边双连通的, 因为存在至少两个顶点的简单环。
- D. 图 G 不是边双连通的, 因为删除任意一条边后, 图不再连通。

15. 使用哈希表存储元素 19, 53, 49, 114, 为了让哈希表不发生哈希冲突, 可以使用 ()

选项的哈希函数? (定义: $\lfloor x \rfloor$ 为不超过 x 的最大整数, 如 $\lfloor 3.14 \rfloor = 3$)

- A. $f(x) = x \bmod 17$
- B. $f(x) = \lfloor \sqrt{x} \rfloor$
- C. $f(x) = \lfloor 100 \lg x \rfloor$
- D. $f(x) = x^2 \bmod 13$

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 T, 错误填 F; 除特殊说明外, 判断题 1.5 分, 选择题 3 分, 共计 40 分)

(1)

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 set <int> s1 = {1, 2, 3}, s2 = {2, 3, 1, 1},
4           s3 = {1, 2, 100}, s4 = {3, 4, 5};
5 map <set <int>, int> mp1, mp2;
6 set <int> intersection(set <int> a, set <int> b) {
7     if (a.size() > b.size()) swap(a, b);

```

```

8     set <int> res = a;
9     for (int i : a)
10        if (b.count(i) == 0)
11            res.erase(i);
12     return res;
13 }
14 int main() {
15     cout << (s1 == s2) << endl;
16     mp1[s3] = 9; mp1[s1] = 1;
17     mp2[s2] = 1; mp2[s4] = 5;
18     cout << (mp1 < mp2) << ' ';
19     cout << mp2[s3] << ' ';
20     cout << (mp1 <= mp2) << endl;
21     mp1.clear();
22     int n, l, t;
23     cin >> n;
24     for (int i = 1; i <= n; i++) {
25         set <int> tmp;
26         for (cin >> l; l; l--) {
27             cin >> t;
28             tmp.emplace(t);
29         }
30         mp1[tmp] = i;
31     }
32     set <int> res = mp1.begin()->first;
33     for (auto i : mp1)
34         res = intersection(i.first, res);
35     cout << res.size();
36     return 0;
37 }

```

假设输入的 n, l, t 满足： n 是不超过 100 的正整数， l 和 t 是不超过 1000 的正整数，完成下面的判断题和单选题。

- 判断题

16. 代码运行后，输出的第一行是 0。 ()

17. 设集合 a, b 的元素个数分别为 p, q ，则执行函数 `intersection(a, b)` 的时间复杂度为

$\Theta(\min(p, q) \log \max(p, q))$ 。 ()

18. 将第 32 行的 `begin()`换成 `rbegin()`, 运行结果可能改变。()

19. 将第 33 行的 `auto`换成 `pair <set <int>, int>`, 运行结果不变。()

• 单选题

20. 输出的第二行是()。

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

21. 记所有输入的 1 之和为 L, 则该程序最坏情况下的时间复杂度是()?

- A. $\Theta(L \log L)$ B. $\Theta(L \log^2 L)$
C. $\Theta(nL \log L)$ D. $\Theta(nL \log n \log L)$

(2)

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 constexpr int mod = 998244353;
4 int n, k, a[12], b[12], ans[12], fa[12];
5 int findfa(int u) { return u == fa[u] ? u : fa[u] = findfa(fa[u]); }
6 int functionUnknown(int a[], int n) {
7     if (n <= 1) return 0;
8     int i = n - 1, j, k;
9     while (true) {
10         j = i; --i;
11         if (a[i] < a[j]) {
12             for (k = n; a[i] >= a[--k];);
13             swap(a[i], a[k]);
14             reverse(a + j, a + n);
15             return 1;
16         }
17         if (!i) {
18             reverse(a, a + n);
19             return 0;
20         }
21     }
22     return -1;
23 }
```

```

24 int F(int x) {
25     int ans = 0;
26     for (int i = k - 1; ~i; --i)
27         ans = (111 * ans * x % mod + a[i]) % mod;
28     return ans;
29 }
30 int main() {
31     scanf("%d %d", &n, &k);
32     for (int i = 0; i < k; ++i) scanf("%d", a + i);
33     for (int m = 1; m <= n; ++m) {
34         for (int i = 0; i < m; ++i) b[i] = i;
35         do {
36             for (int i = 0; i < m; ++i) fa[i] = i;
37             int res = m;
38             for (int i = 0, u, v; i < m; ++i) {
39                 u = findfa(i); v = findfa(b[i]);
40                 if (u == v) continue;
41                 --res; fa[u] = v;
42             }
43             int flag = 0;
44             for (int i = 0; i < m; ++i)
45                 if (b[i] == i) flag = 1;
46             if (flag) continue;
47             ans[m] = (ans[m] + F(res)) % mod;
48         } while(functionUnknown(b, m));
49         printf("%d%c", ans[m], " \n"[m == n]);
50     }
51     return 0;
52 }

```

假设输入的 n, k 是不超过 10 的正整数，输入的 $a[i]$ 是不超过 10^8 的正整数，完成下面的判断题和单选题：

- 判断题

22. `functionUnknown` 函数的返回值不可能为 -1。()

23. 当输入的 $k=1$ 时，输出的数值均能表示为 $t \times a[0]$ 的形式，其中 t 是一个整数。()

24. 如果把第 34 行整行改为 $b[m - 1] = m - 1$, 则程序输出将会被改变。()

• 单选题

25. 在主函数中, `findfa` 函数被调用的次数与以下哪个选项同阶()。

- | | |
|--------------------|--------------------|
| A. $\Theta(n!)$ | B. $\Theta(n!n)$ |
| C. $\Theta(n!n^2)$ | D. $\Theta(n!n^3)$ |

26. 对于以下的输入数据, 输出结果为()。

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

- | | |
|---------------------|---------------------|
| A. 0 15 30 261 1500 | B. 0 12 24 297 1788 |
| C. 0 15 30 477 2940 | D. 0 15 30 864 5520 |

27. `functionUnknown` 的功能接近于 C++ STL 库中的以下() 函数。

- | | |
|----------------------------------|----------------------------------|
| A. <code>next_permutation</code> | B. <code>prev_permutation</code> |
| C. <code>is_sorted</code> | D. <code>is_permutation</code> |

(3)

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 unsigned f(unsigned x) {
4     x ^= x << 3;
5     x ^= x >> 5;
6     return x;
7 }
8 unsigned g(unsigned x) {
9     return (x >> 5) ^ (x >> 2) ^ x ^ (x << 3);
10 }
11 int main() {
12     unsigned y, l, r;
13     cin >> y >> l >> r;
14     for (unsigned long long i = l; i <= r; i++)
15         if (f(i) == y)
16             cout << i % 11 << endl;
17     return 0;
18 }
```

除第 28 题外，假设输入的 y, l, r 均为不超过 $2^{32} - 1$ 的非负整数，完成下面的判断题和单选题：

• 判断题

28. 若输入 2024 1 -1，那么程序将不会输出任何数字。（ ）

29. 若把第 14 行 i 的类型改为 `unsigned int`，则程序可能无法结束运行。（ ）

30. 对于所有符合要求的输入，输出的数字不可能超过一个。（ ）

• 选择题

31. 若从所有 `unsigned int` 当中随机均匀选取一个 x ，则 $f(x)=g(x)$ 成立的概率和（ ）的差最小。

- A. 0 B. $\frac{1}{15}$ C. $\frac{1}{2}$ D. 1

32. (4 分) 定义 $f^n(x)$ 表示对 x 用函数 f 作用 n 次的结果，例如 $f^2(x) = f(f(x))$ 。

假设 `unsigned` 类型具有 ω 个二进制位，且已知存在某种计算 $f^n(x)$ 的算法的时间复杂度为 $\Theta(T(n)P(\omega))$ ，其中 $P(\omega)$ 是一个关于 ω 的幂函数。则 $T(n)$ 最小是（ ）？

- A. $\Theta(1)$ B. $\Theta(\log n)$ C. $\Theta(\log^2 n)$ D. $\Theta(n)$

33. 当输入为 50 1 4294967295 时，输出的第一个数是（ ）。

- A. 1 B. 8 C. 9 D. 10

三、完善程序（单选题，每小题 3 分，共计 30 分）

(1) (最小垄断集) 问题：你有一张 n 个结点（结点编号从 1 到 n ）， m 条边的无向无权图 $G=(V,E)$ ，其中 V 为点集， E 为边集。称 V 的一个子集 S 垄断了图 G ，当且仅当 $\forall(u,v) \in E$ ， u,v 中必然恰有一个结点在 S 内。求垄断了 G 的子集 S 至少有多少个元素，或报告不存在这样的子集。试补全程序。

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int n, m, is[100005];
4 vector <int> g[100005];
5 queue <int> q;
6 int main() {
7     cin >> n >> m;
```

```

8  for (int i = 1; i <= m; i++) {
9      int u, v;
10     cin >> u >> v;
11     g[u].emplace_back(v);
12     g[v].emplace_back(u);
13 }
14 int tot = 0;
15 for (int k = 1; k <= n; k++) {
16     if (①) continue;
17     q.emplace(k);
18     is[k] = 1;
19     int cnt[3] = ②;
20     while (q.size()) {
21         for (int i = 0; i < g[q.front()].size(); i++) {
22             int to = g[q.front()][i];
23             if (is[to] == 0) {
24                 is[to] = ③;
25                 cnt[is[to]]++;
26                 q.emplace(to);
27             } else if (④) {
28                 puts("No such subset."); // 报告不存在这样的子集
29                 return 0;
30             }
31         }
32         q.pop();
33     }
34     tot += ⑤;
35 }
36 cout << tot;
37 return 0;
38 }

```

34. ①处应填 ()

- A. `is[k]` B. `is[k] == 0` C. `q.empty()` D. `tot > k`

35. ②处应填 ()

- A. `{0, 0, 0}` B. `{0, 0, 1}`

C. {1, 0, 0}

D. {0, 1, 0}

36. ③处应填 ()

A. is[q.front()]

B. 3 - is[q.front()]

C. !is[q.front()]

D. q.front()

37. ④处应填 ()。

A. is[to] == is[q.front()]

B. is[to] != is[q.front()]

C. is[to] && is[q.front()]

D. abs(is[to] - is[q.front()]) == 1

38. ⑤处应填 ()。

A. cnt[1]

B. min(cnt[0], cnt[1])

C. min(cnt[1], cnt[2])

D. cnt[1] + cnt[2]

(2) (最长公共前缀) 定义 $LCP(s, t)$ 为两个字符串的最长公共前缀。例如: $s=abcde$, $t=abcabc$, 则 $LCP(s, t)=abc$ 。定义 $|s|$ 为字符串 s 的长度, 在上一例中 $|s|=5$ 。

现在给定了 n 个字符串 s_1, s_2, \dots, s_n 。现在需要统计所有同时满足如下两个条件的

$|LCP(s_i, s_j)|$ 之和对 998244353 取模的值: 条件 1: $i < j$ 。条件 2: $s_i < s_j$ 。

若 $1 \leq n \leq 2 \times 10^5$, $1 \leq |s_i| \leq 20$, 字符串仅由小写字母构成, 试补全程序。

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 const int maxn = 2e5 + 9, mod = 998244353;
4 char s[maxn][22];
5 int n;
6 int tmp[27][maxn], a[maxn];
7 int dfs(int l, int r, int p) {
8     if (l >= r || p >= 20) return 0;
9     int tail[27], ans = 0;
10    memset(tail, 0, sizeof(tail));
11    for (int i = l; i <= r; ++i) {
12        int ch = ①;
13        ②;
14        for (int j = 0; j < ch; ++j)
15            ③;
16    }
17    int tot = l, pos = l;
```

```

18     for (int i = 0; i < 27; ++i)
19         for (int j = 0; j < tail[i]; ++j)
20             a[tot++] = tmp[i][j];
21     for (int i = 0; i < 27; ++i) {
22         ④;
23         pos += tail[i];
24     }
25     return ans;
26 }
27 int main() {
28     scanf("%d", &n);
29     for (int i = 1; i <= n; ++i) scanf("%s", s[i]);
30     for (int i = 1; i <= n; ++i) a[i] = i;
31     printf("%d\n", ⑤);
32     return 0;
33 }

```

39. ①处可以填 ()

- A. $s[i][p] ? s[i][p] - 'a': 0$
- B. $s[i][p] ? s[i][p] - 'a' + 1 : 0$
- C. $s[a[i]][p] ? s[a[i]][p] - 'a': 0$
- D. $s[a[i]][p] ? s[a[i]][p] - 'a' + 1 : 0$

40. ②处应填 ()

- A. $\text{tmp}[ch][tail[ch]++] = a[i]$
- B. $\text{tmp}[ch][++tail[ch]] = a[i]$
- C. $\text{tmp}[ch][tail[ch]++] = i$
- D. $\text{tmp}[ch][++tail[ch]] = i$

41. ③处应填 ()

- A. $\text{ans} = (\text{ans} + 111 * (p - 1) * \text{tail}[j] \% \text{mod}) \% \text{mod}$
- B. $\text{ans} = (\text{ans} + 111 * p * \text{tail}[j] \% \text{mod}) \% \text{mod}$
- C. $\text{ans} = (\text{ans} + 111 * (p + 1) * \text{tail}[j] \% \text{mod}) \% \text{mod}$
- D. $\text{ans} = (\text{ans} + 111 * (p + 2) * \text{tail}[j] \% \text{mod}) \% \text{mod}$

42. ④处应填（ ）。

- A. `ans = (ans + dfs(pos + 1, pos + tail[i], p + 1)) % mod`
- B. `ans = (ans + dfs(pos + 1, pos + tail[i], p - 1)) % mod`
- C. `ans = (ans + dfs(pos, pos + tail[i] - 1, p + 1)) % mod`
- D. `ans = (ans + dfs(pos, pos + tail[i] - 1, p - 1)) % mod`

43. ⑤处应填（ ）。

- A. `dfs(0, n - 1, 0)`
- B. `dfs(1, n, 0)`
- C. `dfs(1, n, 1)`
- D. `dfs(1, n - 1, 1)`

广告 祝贺洛谷计划学员在 NOI2024 获得 73 枚奖牌（9 金 37 银 27 铜）的好成绩

1. 第一轮（初赛课程）<https://class.luogu.com.cn/course/yugu23acs>

2024 年 CSP 第一轮（初赛）课程系统的梳理 CSP J/S 第一轮测试的题型和常考内容，并提供模拟赛和讲评用于查缺补漏。对于希望熟悉第一轮考点、提升第一轮能力的同学均可报名。

本套试题的讲评将在这个课程中获得。此外之前的回放也可以获得。

2. NOIP 计划和进阶计划

- 计划包括集中授课、资料阅读与题单作业布置、定期模拟比赛讲评，从学习过进阶算法开始，达到 NOIP 一等奖较高分的目的。<https://class.luogu.com.cn/course/yugu24nob>
- 进阶计划计划包括集中授课、题单作业布置、定期模拟比赛讲评，学习 NOI 大纲 5-7 级算法，目标 CSP-S 和 NOIP 一等奖。<https://class.luogu.com.cn/course/yugu24tgc>
 - 进阶计划寒假亦有开班。



3. 基础提高衔接计划 <https://class.luogu.com.cn/course/yugu24xjb> 寒假也有开班

计划包括集中授课、题单作业布置、定期模拟比赛讲评，巩固算法基础和举一反三能力，目标 CSP-J 高分，为提高级打基础。

4. 洛谷秋令营（基础组·提高组）9月公开

面向已经掌握基础/进阶算法学员，通过讲题和模拟增加经验，提升 CSP-J/S 应试能力。

5. 洛谷省选计划（每年 50+ 学员进省队）第二轮后公开

A 组前期：题单+模拟比赛+金牌选手答疑；A 组后期：更多的模拟比赛+金牌选手答疑

B 组：授课（省选重要知识点）+模拟比赛+金牌选手答疑

请关注公众号获得最新的课程资讯

解析：关注公众号，回复“SCP2024”获得，考后一周内公开。



2024 LUOGU 非专业级别收容能力认证第一轮

(SCP-S1) 提高级 C++语言试题参考答案

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	2	3	4	5	6	7	8	9	10
B	D	B	C	D	B	A	B	A	D
11	12	13	14	15					
C	C	C	B	C					

二、阅读程序（判断题正确填 T，错误填 F；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

第一题	判断题					选择题	
	1	2	3	4	5	6	
	F	F	F	T	D	A	
第二题	判断题					选择题	
	1	2	3	4	5	6	
	T	F	F	B	C	A	
第三题	判断题					选择题	
	1	2	3	4	5 (4 分)	6	
	F	T	T	B	B	D	

三、完善程序（单选题，每小题 3 分，共计 30 分）

第一题					第二题				
1	2	3	4	5	1	2	3	4	5
A	D	B	A	C	D	A	B	C	B

解析：关注公众号，回复“SCP2024”获得

