

# CSP-J 2025

## 第一轮真题 · 试卷与答案

作答时间: 120 分钟      姓名: \_\_\_\_\_

满分 100 分, 共 43 个作答点。判断题填写 A 或 B; 参考答案附于卷末。

### 一、单项选择 (共 30 分)

#### 第 1 题 (2 分)

一个 32 位无符号整数可以表示的最大值, 最接近下列哪个选项?

- |                    |                       |
|--------------------|-----------------------|
| A. $4 \times 10^9$ | B. $3 \times 10^{10}$ |
| C. $2 \times 10^9$ | D. $2 \times 10^{10}$ |

#### 第 2 题 (2 分)

在 C++ 中, 执行 `int x = 255; cout << (x & (x - 1));` 后, 输出的结果是?

- |        |        |
|--------|--------|
| A. 255 | B. 254 |
| C. 128 | D. 0   |

#### 第 3 题 (2 分)

函数 `calc(n)` 的定义如下, 则 `calc(5)` 的返回值是多少? ( )

```
1 int calc(int n) {  
2     if (n <= 1) return 1;  
3     if (n % 2 == 0) return calc(n / 2) + 1;  
4     else return calc(n - 1) + calc(n - 2);  
5 }
```

- |      |      |
|------|------|
| A. 5 | B. 6 |
| C. 7 | D. 8 |

#### 第 4 题 (2 分)

用 5 个权值 10, 12, 15, 20, 25 构造哈夫曼树, 该树的带权路径长度是多少?

- |        |        |
|--------|--------|
| A. 176 | B. 186 |
| C. 196 | D. 206 |

第 5 题 (2 分)

在一个有向图中，所有顶点的入度之和等于所有顶点的出度之和，这个总和等于？

- A. 顶点数
  - B. 边数
  - C. 顶点数 + 边数
  - D. 顶点数  $\times 2$
- 

第 6 题 (2 分)

从 5 位男生和 4 位女生中选出 4 人组成一个学习小组，要求学习小组中男生和女生都有。有多少种不同的选法？

- A. 126
  - B. 121
  - C. 120
  - D. 100
- 

第 7 题 (2 分)

假设  $a, b, c$  都是布尔变量，逻辑表达式  $(a \ \&\& \ b) \ || \ (!c \ \&\& \ a)$  的值与下列哪个表达式不始终相等？

- A.  $a \ \&\& \ (b \ || \ !c)$
  - B.  $(a \ || \ !c) \ \&\& \ (b \ || \ !c) \ \&\& \ (a \ || \ a)$
  - C.  $a \ \&\& \ (!b \ || \ c)$
  - D.  $!(!a \ || \ !b) \ || \ (a \ \&\& \ !c)$
- 

第 8 题 (2 分)

已知  $f[0] = 1, f[1] = 1$ ，并且对于所有  $n \geq 2$  有  $f[n] = (f[n-1] + f[n-2]) \% 7$ 。那么  $f[2025]$  的值是多少？

- A. 2
  - B. 4
  - C. 5
  - D. 6
- 

第 9 题 (2 分)

下列关于 C++ string 类的说法，正确的是？

- A. string 对象的长度在创建后不能改变。
  - B. 可以使用 + 运算符直接连接一个 string 对象和一个 char 类型的字符。
  - C. string 的 length() 和 size() 方法返回的值可能不同。
  - D. string 对象必须以 '\0' 结尾，且这个结尾符计入 length()。
-

### 第 10 题 (2 分)

考虑以下 C++ 函数:

```
1 void solve(int &a, int b) {  
2     a = a + b;  
3     b = a - b;  
4     a = a - b;  
5 }  
6 int main() {  
7     int x = 5, y = 10;  
8     solve(x, y);  
9 }
```

在 main 函数调用 solve 后,  $x$  和  $y$  的值分别是?

- A. 5, 10
  - B. 10, 5
  - C. 10, 10
  - D. 5, 5
- 

### 第 11 题 (2 分)

一个  $8 \times 8$  的棋盘, 左上角坐标为  $(1, 1)$ , 右下角为  $(8, 8)$ 。一个机器人从  $(1, 1)$  出发, 每次只能向右或向下走一格。要到达  $(4, 5)$ , 有多少种不同的路径?

- A. 20
  - B. 35
  - C. 56
  - D. 70
- 

### 第 12 题 (2 分)

某同学用冒泡排序对数组  $\{6, 1, 5, 2, 4\}$  进行升序排序, 请问需要进行多少次元素交换?

- A. 5
  - B. 6
  - C. 7
  - D. 8
- 

### 第 13 题 (2 分)

十进制数  $720_{10}$  和八进制数  $270_8$  的和用十六进制表示是多少?

- A.  $388_{16}$
  - B.  $3DE_{16}$
  - C.  $288_{16}$
  - D.  $990_{16}$
- 

### 第 14 题 (2 分)

一棵包含 1000 个结点的完全二叉树, 其叶子结点的数量是多少?

- A. 499
  - B. 512
  - C. 500
  - D. 501
-

第 15 题 (2 分)

给定一个初始为空的整数栈  $S$  和一个空的队列  $P$ 。我们按顺序处理输入的整数队列  $A: 7, 5, 8, 3, 1, 4, 2$ 。对于队列  $A$  中的每一个数，执行以下规则：如果该数是奇数，则将其压入栈  $S$ ；如果该数是偶数，且栈  $S$  非空，则弹出一个栈顶元素，并加入到队列  $P$  的末尾；如果该数是偶数，且栈  $S$  为空，则不进行任何操作。当队列  $A$  中的所有数都处理完毕后，队列  $P$  的内容是什么？

- A. 5,1,3

B. 7,5,3
- C. 3,1,5

D. 5,1,3,7

二、阅读程序 (共 40 分)

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

第 16 题—第 21 题

```
1  #include <algorithm>
2  #include <cstdio>
3  #include <cstring>
4  inline int gcd(int a, int b) {
5      if (b == 0)
6          return a;
7      return gcd(b, a % b);
8  }
9  int main() {
10     int n;
11     scanf("%d", &n);
12     int ans = 0;
13     for (int i = 1; i <= n; ++i) {
14         for (int j = i + 1; j <= n; ++j) {
15             for (int k = j + 1; k <= n; ++k) {
16                 if (gcd(i, j) == 1 && gcd(j, k) == 1
17                     && gcd(i, k) == 1) {
18                     ++ans;
19                 }
20             }
21         }
22     }
23     printf("%d\n", ans);
24     return 0;
25 }
26
```

第 16 题 (1 分) (1 分) 当输入为 2 时，程序并不会执行第 16 行的判断语句。 ( )

- A. 正确
- B. 错误

第 17 题 (1.5 分) 将第 16 行中的 `&& gcd(i,k)==1` 删去不会影响程序运行结果。 ( )

- A. 正确
- B. 错误

第 18 题 (1.5 分) 当输入的  $n \geq 3$  的时候, 程序总是输出一个正整数。 ( )

- A. 正确 B. 错误

第 19 题 (3 分) 将第 7 行的  $\text{gcd}(b, a\%b)$  改为  $\text{gcd}(a, a\%b)$  后, 程序可能出现的问题是 ( )。

- A. 输出的答案大于原答案。 B. 输出的答案小于原答案。  
C. 程序有可能陷入死循环。 D. 可能发生整型溢出问题。

第 20 题 (3 分) 当输入为 8 的时候, 输出为 ( )。

- A. 37 B. 42  
C. 35 D. 25

第 21 题 (3 分) 调用  $\text{gcd}(36, 42)$  会返回 ( )。

- A. 6 B. 252  
C. 3 D. 2

#### 第 22 题—第 27 题

```
1 #include <algorithm>
2 #include <cstdio>
3 #include <cstring>
4 #define ll long long
5 int n, k;
6 int a[200007];
7 int ans[200007];
8 int main() {
9     scanf("%d%d", &n, &k);
10    for (int i = 1; i <= n; ++i) {
11        scanf("%d", &a[i]);
12    }
13    std::sort(a + 1, a + n + 1);
14    n = std::unique(a + 1, a + n + 1) - a - 1;
15    for (int i = 1, j = 0; i <= n; ++i) {
16        for (; j < i && a[i] - a[j + 1] > k; ++j)
17            ;
18        ans[i] = ans[j] + 1;
19    }
20    printf("%d\n", ans[n]);
21    return 0;
22 }
```

第 22 题 (1.5 分) 当输入为 3 1 3 2 1 时, 输出结果为 2。 ( )

- A. 正确 B. 错误

---

**第 23 题 (1.5 分)** 假设输入的  $n$  为正整数，输出的答案一定小于等于  $n$ ，大于等于 1。 ( )

A. 正确

B. 错误

---

**第 24 题 (1.5 分)** 将第 14 行的  $n = \text{std::unique}(a + 1, a + n + 1) - a - 1$ ；删去后，有可能出现与原本代码不同的输出结果。 ( )

A. 正确

B. 错误

---

**第 25 题 (3 分)** 假设输入的  $a$  数组和  $k$  均为正整数，执行第 18 行代码时，一定满足的条件不包括 ( )。

A.  $j < i$

B.  $a[i] - a[j] > k$

C.  $j < n$

D.  $a[j] < a[i]$

---

**第 26 题 (3 分)** 当输入的  $n = 100$ 、 $k = 2$ 、 $a = \{1, 2, \dots, 100\}$  时，输出为 ( )。

A. 34

B. 100

C. 50

D. 33

---

**第 27 题 (3 分)** 假设输入的  $a$  数组和  $k$  均为正整数，但  $a$  数组不一定有序，则若误删去第 13 行的  $\text{std::sort}(a + 1, a + n + 1)$ ；，程序有可能出现的问题有 ( )。

A. 输出的答案比原本答案更大

B. 输出的答案比原本答案更小

C. 出现死循环行为

D. 以上均可能发生

---

第 28 题—第 33 题

```
1  #include <algorithm>
2  #include <cstdio>
3  #include <cstring>
4  #define ll long long
5  int f[5007][5007];
6  int a[5007], b[5007];
7  int n;
8  int main() {
9      scanf("%d", &n);
10     for (int i = 1; i <= n; ++i) {
11         scanf("%d", &a[i]);
12     }
13     for (int i = 1; i <= n; ++i) {
14         scanf("%d", &b[i]);
15     }
16     for (int i = 1; i <= n; ++i) {
17         for (int j = 1; j <= n; ++j) {
18             f[i][j] = std::max(f[i][j], std::max(f[i - 1][j], f[i][j - 1]));
19             if (a[i] == b[j]) {
20                 f[i][j] = std::max(f[i][j], f[i - 1][j - 1] + 1);
21             }
22         }
23     }
24     printf("%d\n", f[n][n]);
25     return 0;
26 }
```

第 28 题 (1.5 分) 当输入 4 1 2 3 4 1 3 2 2 时, 输出为 2。 ( )

- A. 正确 B. 错误

第 29 题 (1.5 分) 当程序运行完毕后, 对于所有的  $1 \leq i, j \leq n$ , 都一定有  $f[i][j] \leq f[n][n]$ 。 ( )

- A. 正确 B. 错误

第 30 题 (1.5 分) 将第 18 行的  $f[i][j] = \text{std::max}(f[i][j], \text{std::max}(f[i-1][j], f[i][j-1]))$ ; 删去后, 并不影响程序运行结果。 ( )

- A. 正确 B. 错误

第 31 题 (3 分) 输出的答案满足的性质有 ( )。

- A. 小于等于  $n$  B. 大于等于 0  
C. 不一定大于等于 1 D. 以上均是

---

**第 32 题 (3 分)** 如果在 16 行的循环前加上以下两行:

```
cpp std::sort(a+1, a+n+1); std::sort(b+1, b+n+1);
```

则答案会 ( )。

- |                 |                 |
|-----------------|-----------------|
| <b>A.</b> 变大或不变 | <b>B.</b> 变小或不变 |
| <b>C.</b> 一定变大  | <b>D.</b> 不变    |
- 

**第 33 题 (3 分)** (本题在原卷中被删除) 如果输入的  $a = \{1, 2, \dots, n\}$ , 而且  $b$  数组中数字均为  $1 \sim n$  中的正整数, 则上述代码等价于下面哪个问题: ( )。

- |                          |                            |
|--------------------------|----------------------------|
| <b>A.</b> 求 $b$ 数组去重后的长度 | <b>B.</b> 求 $b$ 数组的最长上升子序列 |
| <b>C.</b> 求 $b$ 数组的长度    | <b>D.</b> 求 $b$ 数组的最大值     |
-

### 三、完善程序 (共 30 分)

(单选题, 每小题 3 分, 共计 30 分)

#### 第 33 题 (完善①, 原卷重号) —第 37 题

(1) (字符串解码) “行程长度编码” (Run-Length Encoding) 是一种无损压缩算法, 常用于压缩重复字符较多的数据, 以减少存储空间。假设原始字符串不包含数字字符。压缩规则如下: i) 如果原始字符串中一个字符连续出现  $N$  次 ( $N \geq 2$ ), 在压缩字符串中它被表示为 “字符 + 数字  $N$ ”。例如, 编码 A12 代表 12 个连续的字符 A。ii) 如果原始字符串中一个字符只出现 1 次, 在压缩字符串中它就表示为该字符本身。例如, 编码 B 代表 1 个字符 B。

以下程序实现读取压缩字符串并输出其原始的、解压后的形式。试补全程序。

```
1  #include <cctype>
2  #include <iostream>
3  #include <string>
4  using namespace std;
5
6  int main() {
7      string z;
8      cin >> z;
9      string s = "";
10
11     for (int i = 0; i < z.length(); ) {
12         char ch = z[i];
13
14         if (___①___ && isdigit(z[i + 1])) {
15             i++;
16             int count = 0;
17             while (i < z.length() && isdigit(z[i])) {
18                 count = ___②___;
19                 i++;
20             }
21             for (int j = 0; j < ___③___; ++j) {
22                 s += ch;
23             }
24         } else {
25             s += ___④___;
26             ___⑤___;
27         }
28     }
29
30     cout << s << endl;
31     return 0;
32 }
33
```

#### 第 33 题 (完善①, 原卷重号) (3 分) ①处应填 ( )

- |                         |                    |
|-------------------------|--------------------|
| A. $i < z.length()$     | B. $i - 1 \geq 0$  |
| C. $i + 1 < z.length()$ | D. $isdigit(z[i])$ |

---

第 34 题 (3 分) ②处应填 ( )

A. `count + (z[i] - '0')`

B. `count * 10 + (z[i] - '0')`

C. `z[i] - '0'`

D. `count + 1`

---

第 35 题 (3 分) ③处应填 ( )

A. `count - 1`

B. `count`

C. `10`

D. `z[i] - '0'`

---

第 36 题 (3 分) ④处应填 ( )

A. `z[i+1]`

B. `ch`

C. `z.back()`

D. `(char)z[i] + 1`

---

第 37 题 (3 分) ⑤处应填 ( )

A. `i--`

B. `i = i + 2`

C. `i++`

D. `// 不执行任何操作`

---

### 第 38 题—第 42 题

(2) (精明与糊涂) 有  $N$  个人, 分为两类: i) 精明人: 永远能正确判断其他人是精明还是糊涂; ii) 糊涂人: 判断不可靠, 会给出随机的判断。

已知精明人严格占据多数, 即如果精明人有  $k$  个, 则满足  $k > N/2$ 。

你只能通过函数 `query(i, j)` 让第  $i$  个人判断第  $j$  个人: 返回 `true` 表示判断结果为“精明人”; 返回 `false` 表示判断结果为“糊涂人”。你的目标是, 通过这些互相判断, 找出至少一个百分之百确定的精明人。同时, 你无需关心 `query(i, j)` 的内部实现。

以下程序利用“精明人占多数”的优势。设想一个“消除”的过程, 让人们互相判断并进行抵消。经过若干轮抵消后, 最终留下的候选人必然属于多数派, 即精明人。

例如, 假设有三人 0, 1, 2。如果 0 说 1 是糊涂人, 而 1 也说 0 是糊涂人, 则 0 和 1 至少有一个是糊涂人。程序将同时淘汰 0 和 1。由于三人里至少有两个精明人, 我们确定 2 是精明人。

试补全程序。

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int N;
6  bool query(int i, int j);
7
8  int main() {
9      cin >> N;
10
11     int candidate = 0;
12     int count = __①__;
13     for (int i = 1; i < N; ++i) {
14         if (__②__) {
15             candidate = i;
16             count = 1;
17         } else {
18             if (__③__) {
19                 __④__;
20             } else {
21                 count++;
22             }
23         }
24     }
25     cout << __⑤__ << endl;
26     return 0;
27 }
```

### 第 38 题 (3 分) ①处应填 ( )

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

---

第 39 题 (3 分) ②处应填 ( )

- |                              |                                              |
|------------------------------|----------------------------------------------|
| A. <code>count &lt; 0</code> | B. <code>count == 1</code>                   |
| C. <code>count == 0</code>   | D. <code>query(candidate, i) == false</code> |
- 

第 40 题 (3 分) ③处应填 ( )

- A. `query(candidate, i) == false`  
B. `query(i, candidate) == true`  
C. `query(candidate, i) == false && query(i, candidate) == false`  
D. `query(candidate, i) == false || query(i, candidate) == false`
- 

第 41 题 (3 分) ④处应填 ( )

- |                         |                               |
|-------------------------|-------------------------------|
| A. <code>count--</code> | B. <code>break</code>         |
| C. <code>count++</code> | D. <code>candidate = i</code> |
- 

第 42 题 (3 分) ⑤处应填 ( )

- |                           |                       |
|---------------------------|-----------------------|
| A. <code>N - 1</code>     | B. <code>count</code> |
| C. <code>candidate</code> | D. <code>0</code>     |
-

# 参考答案

满分 100 分。判断题 A 表示正确，B 表示错误。

| 题号     | 分值  | 答案 | 题号                 | 分值  | 答案 |
|--------|-----|----|--------------------|-----|----|
| 第 1 题  | 2   | A  | 第 2 题              | 2   | B  |
| 第 3 题  | 2   | B  | 第 4 题              | 2   | B  |
| 第 5 题  | 2   | B  | 第 6 题              | 2   | C  |
| 第 7 题  | 2   | C  | 第 8 题              | 2   | D  |
| 第 9 题  | 2   | B  | 第 10 题             | 2   | C  |
| 第 11 题 | 2   | B  | 第 12 题             | 2   | B  |
| 第 13 题 | 2   | A  | 第 14 题             | 2   | C  |
| 第 15 题 | 2   | A  | 第 16 题             | 1   | A  |
| 第 17 题 | 1.5 | B  | 第 18 题             | 1.5 | B  |
| 第 19 题 | 3   | B  | 第 20 题             | 3   | D  |
| 第 21 题 | 3   | A  | 第 22 题             | 1.5 | A  |
| 第 23 题 | 1.5 | A  | 第 24 题             | 1.5 | B  |
| 第 25 题 | 3   | B  | 第 26 题             | 3   | A  |
| 第 27 题 | 3   | B  | 第 28 题             | 1.5 | A  |
| 第 29 题 | 1.5 | A  | 第 30 题             | 1.5 | B  |
| 第 31 题 | 3   | D  | 第 32 题             | 3   | A  |
| 第 33 题 | 3   | B  | 第 33 题 (完善①, 原卷重号) | 3   | C  |
| 第 34 题 | 3   | B  | 第 35 题             | 3   | B  |
| 第 36 题 | 3   | B  | 第 37 题             | 3   | C  |
| 第 38 题 | 3   | B  | 第 39 题             | 3   | C  |
| 第 40 题 | 3   | D  | 第 41 题             | 3   | A  |
| 第 42 题 | 3   | C  |                    |     |    |