



扫码获取  
答案解析

## 普及组 CSP-J 2025 初赛模拟卷 5

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)

- 十进制数 2025 的十六进制表示是 ( )。  
A. 07D9                      B. 07E9                      C. 07F9                      D. 07F1
- 以下关于计算机竞赛 IOI 的描述正确的是 ( )。  
A. IOI 非英语国家参赛选手可以在比赛中携带电子词典  
B. IOI 参赛选手可携带已关机的手机放在自己座位后面的包里  
C. IOI 参赛选手在比赛时间内去厕所的时候可携带手机  
D. IOI 全称是国际信息学奥林匹克竞赛
- 以下不能用 ASCII 码表示的字符是 ( )。  
A. @                      B. ①                      C. ^                      D. ~
- 设变量  $s$  为 `double` 型且已赋值, 下列哪条语句能将  $s$  中的数值保留到小数点后一位, 并将第二位四舍五入? ( )  
A.  $s = (x * 10 + 0.5) / 10.0$       B.  $s = s * 10 + 0.5 / 10.0$   
C.  $s = (s / 10 + 0.5) * 10.0$       D.  $s = (\text{int})(s * 10 + 0.5) / 10.0$
- 以下不属于 STL 链表中的函数的是 ( )。  
A. `sort`                      B. `empty`                      C. `push_back`                      D. `resize`
- 小明写了一个程序, 在这里用到的数据结构是 ( )。

```
#include <iostream>
using namespace std;
int k;
int f(int a)
{
    if(a-k > 0 && (a-k) % 2 == 0)
        return f((a+k)/2) + f((a-k)/2);
```

```
        else
            return 1;
    }
    int main()
    {
        int n;
        cin >> n >> k;
        if((n+k) & 1)
        {
            cout << 1 << endl;
            return 0;
        }
        cout << f(n);
        return 0;
    }
```

A. 树                      B. 栈                      C. 链表                      D. 队列

7. 小明想求  $n$  个不同正整数的全排列，他设计的程序采用 DFS 方法的时间复杂度是 ( )。

A.  $O(\log n)$                       B.  $O(n!)$                       C.  $O(n^2)$                       D.  $O(n \log n)$

8. 在下列排序算法中，( ) 是不稳定的排序算法。

A. 归并排序                      B. 插入排序                      C. 选择排序                      D. 冒泡排序

9. 一台 32 位操作系统的计算机运行 C++，下列说法中错误的是 ( )。

A. double 类型的变量占用 8 字节内存空间  
B. bool 类型的变量占用 1 字节内存空间  
C. long long 类型变量的取值范围比 int 类型变量的大一倍  
D. char 类型的变量也可以作为循环变量

10. 若整型变量  $n$  的值为 25，则表达式  $n \& (n+1 >> 1)$  的值是 ( )。

A. 25                      B. 26                      C. 9                      D. 16

11. 一群学生参加学科夏令营，每名同学至少参加一个学科的考试。已知有 100 名学生参加了数学考试，50 名学生参加了物理考试，48 名学生参加了化学考试，学生总数

是参加至少两门考试学生的两倍,也是参加三门考试学生数的三倍,则学生总数为 ( )。

- A. 90                      B. 96                      C. 108                      D. 120

12. 以下不是 C++ 中的循环语句的是 ( )。

- A. while                      B. do...while                      C. for                      D. switch...case

13. 二叉树 T, 已知其后序遍历序列为 4 2 7 5 6 3 1, 中序遍历序列为 4 2 1 5 7 3 6, 则其前序遍历序列为 ( )。

- A. 1 2 5 7 6 3 4                      B. 1 2 4 3 5 7 6  
C. 1 4 2 7 5 3 6                      D. 1 4 7 2 3 5 6

14. 一个六位数是完全平方数,且最后三位数字都是 4, 这样的六位数有 ( ) 个。

- A. 2                      B. 3                      C. 4                      D. 5

15. 用三种颜色给  $1 \times 4$  的长方形方格区域涂色,在每种颜色至少用 1 次的前提下,相邻方格不涂同一种颜色的概率为 ( )。

- A.  $1/3$                       B.  $2/3$                       C.  $1/2$                       D.  $4/9$

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,判断题每题 1.5 分,选择题每题 3 分,共计 40 分)

(1)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 int solve(vector<int> &nums)
05 {
06     map<int, int> cnt;
07     int tot = 0;
08     for(auto v: nums)
09     {
10         cnt[v]++;
11         tot += v;
12     }
13     int ans = -1E5;
```

```
14     for(auto v: nums)
15     {
16         cnt[v]--;
17         if((tot - v) % 2 == 0 && cnt[(tot - v) / 2] > 0)
18             ans = max(ans, v);
19         cnt[v]++;
20     }
21     return ans;
22 }
23
24 int main()
25 {
26     int n;
27     cin >> n;
28     vector<int> a(n);
29     for(int i = 0; i < n; i++)
30         cin >> a[i];
31     cout << solve(a) << endl;
32     return 0;
33 }
```

#### ■ 判断题

16. 若程序输入 5 -2 -1 -3 -6 4, 则程序输出 4。 ( )
17. 对于第 17 行的代码, 如果不判断  $(tot - v) \% 2 == 0$ , 则程序依然可以得到正确的结果。 ( )
18. 若将头文件 `<bits/stdc++.h>` 换为 `<iostream>`, 程序依然可以正常运行。 ( )

#### ■ 选择题

19. 若输入 8 6 -31 50 -35 41 37 -42 13, 则输出是 ( )。
- A. 13                      B. -35                      C. -31                      D. -100000
20. 如果去除第 19 行的代码, 对于输入 4 2 3 5 10, 输出是 ( )。
- A. 2                      B. 5                      C. 10                      D. -100000

#### (2)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
```

```

04 const int inf = 0x3f3f3f3f;
05
06 int solve(vector<string> &words, string target)
07 {
08     const int n = target.length();
09     set<string> s;
10     for(auto v: words)
11         for(int i = 1; i <= v.length(); i++)
12             s.insert(v.substr(0, i));
13     vector<int> dp(n + 1, inf);
14     dp[0] = 0;
15     for(int i = 0; i < n; i++)
16         for(int j = 0; j <= i; j++)
17             if(s.find(target.substr(j, i - j + 1)) != s.end())
18                 dp[i + 1] = min(dp[i + 1], dp[j] + 1);
19     return dp[n] != inf ? dp[n] : -1;
20 }
21
22 int main()
23 {
24     int n;
25     cin >> n;
26     vector<string> a(n);
27     for(int i = 0; i < n; i++)
28         cin >> a[i];
29     string t;
30     cin >> t;
31     cout << solve(a, t) << endl;
32     return 0;
33 }

```

假设第 17 行的 find 函数的时间复杂度为  $O(\log n)$ , substr 函数的时间复杂度视为  $O(1)$ , 完成下面的问题。

#### ■ 判断题

21. 若输入 3 abc aaaaa bcdef aabcdabc, 则输出为 2。 ( )
22. 若将第 18 行中的  $dp[i + 1]$  改为  $dp[i]$ , 则可能出现编译错误。 ( )
23. (2 分) 该程序的输出一定小于或等于输入的  $n$ 。 ( )



## ■ 选择题

24. 当输入的  $a$  数组为{"abababab", "ab"},  $t$ ="ababaababa"时, 程序的输出为 ( )。
- A. 0                      B. 1                      C. 2                      D. 3
25. 若删除第 17 行的代码, 则当输入的  $a$  数组为{"abababab", "ab"},  $t$ ="ababa"时, 程序的输出为 ( )。
- A. 1                      B. 2                      C. 3                      D. 4
26. (4 分) 这段代码的时间复杂度为 ( )。
- A.  $O(n)$                   B.  $O(n\log n)$               C.  $O(n^2)$                   D.  $O(n^2\log n)$

(3)

```

01 #include <bits/stdc++.h>
02 using namespace std;
03
04 int calc(int n, int presses)
05 {
06     set<int> seen;
07     for (int i = 0; i < (1 << 4); i++)
08     {
09         vector<int> pressArr(4);
10         for (int j = 0; j < 4; j++)
11             pressArr[j] = (i >> j) & 1;
12         int sum = 0;
13         for(int j = 0; j < 4; j++)
14             sum += pressArr[j];
15         if (sum % 2 == presses % 2 && sum <= presses)
16         {
17             int status = pressArr[0] ^ pressArr[2] ^ pressArr[3];
18             if (n >= 2)
19                 status |= (pressArr[0] ^ pressArr[1]) << 1;
20             if (n >= 3)
21                 status |= (pressArr[0] ^ pressArr[2]) << 2;
22             if (n >= 4)
23                 status |= (pressArr[0] ^ pressArr[1] ^ pressArr[3]) <<
24                             3;
25             seen.insert(status);
26         }
27     }
28 }

```

```

26     }
27     return seen.size();
28 }
29
30 int main()
31 {
32     int n, presses;
33     cin >> n >> presses;
34     cout << calc(n, presses) << endl;
35     return 0;
36 }

```

假设  $1 \leq n, presses \leq 4$ , 回答下面的问题。

#### ■ 判断题

27. 若输入为 2 1, 则程序的输出为 3。 ( )
28. 对于第 7 行代码, 变量  $i$  的上界为 16。 ( )
29. 对于任意的输入, 程序的输出不会大于 8。 ( )

#### ■ 选择题

30. 当输入为  $n=3, presses=2$  时, 程序的输出为 ( )。
- A. 5                      B. 6                      C. 7                      D. 8
31. (4 分)若删除第 18~19 行的代码, 当输入为  $n=3, presses=2$  时, 程序的输出为 ( )。
- A. 1                      B. 2                      C. 3                      D. 4
32. 上述代码的时间复杂度为 ( )。
- A.  $O(1)$                   B.  $O(\log n)$               C.  $O(n)$                   D.  $O(n \log n)$

### 三、完善程序 (单选题, 每小题 3 分, 共计 30 分)

#### (1) 题目描述:

输入  $n$  ( $1 \leq n \leq 2 \times 10^5$ ) 和长为  $n$  的数组  $a$  ( $1 \leq a[i] \leq n$ )。

你可以多次执行如下操作: 选择两个下标  $i$  和  $j$ , 满足  $a[i]=a[j]$ 。删除下标  $[i, j]$  中的元素。删除后, 数组长度减小  $i-j+1$ 。

输出你最多可以删多少个。

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 const int inf = 0x3f3f3f3f;
05
06 void solve()
07 {
08     int n;
09     cin >> n;
10     vector<int> a(n + 1);
11     for(int i = 1; i <= n; i++)
12         cin >> a[i];
13     vector<int> dp(n + 1), ①;
14     for(int i = 1; i <= n; i++)
15     {
16         dp[i] = max(dp[i], ②);
17         dp[i] = max(dp[i], ③);
18         lst[a[i]] = max(lst[a[i]], ④);
19     }
20     cout << ⑤ << endl;
21     return;
22 }
23
24 int main()
25 {
26     int t = 1;
27     cin >> t;
28     while(t--)
29         solve();
30     return 0;
31 }
```

33. ①处应填 ( )。

- A. `lst(n+1)`  
C. `lst(n+1,inf)`

- B. `lst(n+1,0)`  
D. `lst(n+1,-inf)`

34. ②处应填 ( )。

- A. `dp[i - 1]`      B. `dp[i] - 1`

- C. `dp[i + 1]`      D. `dp[i] + 1`





```
24     ans = max(ans, lst + 1);
25 }
26 cout << ⑤ << endl;
27 }
28
29 int main()
30 {
31     int t = 1;
32     cin >> t;
33     while(t--)
34         solve();
35     return 0;
36 }
```

38. ①处应填 ( )。

- A.  $i < 20$       B.  $i \leq 20$       C.  $i > 20$       D.  $i \neq 20$

39. ②处应填 ( )。

- A.  $!(a[i] \ \& \ (1 \ll j))$       B.  $a[i] \ \& \ (1 \ll j)$   
C.  $a[j] \ \& \ (1 \ll i)$       D.  $!(a[j] \ \& \ (1 \ll i))$

40. ③处应填 ( )。

- A.  $\min(\text{lst}, \text{cnt})$       B.  $\max(\text{lst}, \text{cnt})$   
C.  $\text{cnt}$       D.  $\text{lst} + \text{cnt}$

41. ④处应填 ( )。

- A.  $\text{lst} < n$       B.  $\text{lst} \neq n$       C.  $\text{lst} == n$       D.  $\text{lst} > n$

42. ⑤处应填 ( )。

- A.  $\text{ans}$       B.  $\min(\text{ans}, n)$   
C.  $\min((\text{ans} == 0 ? 1 : \text{ans}), n)$       D.  $(\text{ans} == 0 ? 1 : \text{ans})$