



普及组 CSP-J 2025 初赛模拟卷 4

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)

- 正整数 2025 与 1800 的最大公约数是 ()。
A. 15 B. 25 C. 45 D. 225
- 表达式 `(( '0' == 0 ) + 's' + 5 + 2.0 )` 的结果类型为 ()。
A. double B. int C. char D. bool
- 对一个 int 类型的值, 执行以下哪个操作后, 一定会变回原来的值? ()
A. 左移 5 位, 再右移 5 位 B. 右移 5 位, 再左移 5 位
C. 按位或 15, 再按位与 15 D. 按位异或 15, 再按位异或 15
- 在数组 `H[x]` 中, 若存在 $(i < j) \ \&\& \ (H[i] > H[j])$, 则称 $(H[i], H[j])$ 为数组 `H[x]` 的一个逆序对。对于序列 27, 4, 1, 59, 3, 26, 38, 15, 在不改变顺序的情况下, 去掉 () 会使逆序对的个数减少 4。
A. 1 B. 3 C. 26 D. 15
- 如果字符串 `s` 在字符串 `str` 中出现, 则称字符串 `s` 为字符串 `str` 的子串。设字符串 `str = "oiers"`, 则 `str` 的非空子串的数目是 ()。
A. 17 B. 16 C. 15 D. 14
- 以下哪种排序算法的平均时间复杂度最好? ()
A. 插入排序 B. 归并排序 C. 选择排序 D. 冒泡排序
- 如果 `x` 和 `y` 均为 int 类型的变量, 且 `y` 的值不为 0, 那么能正确判断 “`x` 是 `y` 的 2 倍” 的表达式是 ()。
A. `(x >> 2 == y)` B. `(x - 2*y) % 2 != 0`
C. `(x / y == 2)` D. `(x == 2 * y)`
- 表达式 `a*(b+c)-d` 的后缀表达式为 ()。
A. `abcd*+-` B. `abc**d-` C. `abc**d-` D. `++abcd`

9. 关于计算机网络, 下列说法中正确的是 ()。
- A. SMTP 和 POP3 都是电子邮件发送协议
B. IPv6 地址是从 IPv4、IPv5 一路升级过来的
C. 计算机网络是一个在协议控制下的多机互连系统
D. 192.168.0.1 是 A 类地址
10. 下列哪种语言不是面向对象的语言? ()
- A. Java B. C++ C. Python D. Fortran
11. 信息学奥赛的所有课程和课程间的先修关系构成一个有向图 G , 我们用有向边 $\langle A, B \rangle$ 表示课程 A 是课程 B 的先修课, 则找到某门课程 C 的全部先修课, 下面哪种方法不可行? ()
- A. BFS B. DFS C. 枚举 D. BFS+DFS
12. 一个字长为 8 位的整数的补码为 11111001, 则它的原码是 ()。
- A. 00000111 B. 10000110 C. 10000111 D. 11111001
13. 元素 A, B, C, D, E, F 入栈的顺序为 A, B, C, D, E, F , 如果第一个出栈的是 C , 则最后一个出栈的不可能是 ()。
- A. A B. B C. D D. F
14. 一个三位数等于它的各位数字的阶乘之和, 则此三位数的各位数字之和为 ()。
- A. 9 B. 10 C. 11 D. 多于一种情况
15. 在一个非连通无向图 G 中有 36 条边, 则该图至少有 () 个顶点。
- A. 8 B. 9 C. 10 D. 7

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 using i64 = long long;
05
06 const i64 k = 3;
07 const i64 mod = 8;
08
09 i64 toint(string s)
10 {
11     sort(s.begin(), s.end());
12     i64 ans = 0;
13     for(int i = 0; i < s.length(); i++)
14         ans = (ans * k + (s[i] - 'a' + 1)) % mod;
15     return ans;
16 }
17
18 vector<vector<string>> solve(vector<string>& str)
19 {
20     map<i64, vector<string>> mp;
21     for(auto s: str)
22         mp[toint(s)].push_back(s);
23     vector<vector<string>> ans;
24     for(auto v: mp)
25         ans.push_back(v.second);
26     return ans;
27 }
28
29 int main()
30 {
31     int n;
32     cin >> n;
33     vector<string> vec(n);
34     for(int i = 0; i < n; i++)
35         cin >> vec[i];
36     auto ans = solve(vec);
37     for(auto v: ans)
38         for(int i = 0; i < v.size(); i++)
39             cout << v[i] << " \n"[i == v.size() - 1];
40     return 0;
41 }
```

假设 $1 \leq n \leq 10^3$, $1 \leq \text{vec}[i].\text{length}() \leq 10^3$, 回答下面的问题。

■ 判断题

16. 若程序输入 6 eat tea tan ate nat bat, 则程序输出 bat (换行) eat tea ate (换行) tan nat (换行)。 ()
17. 对于这段代码, `toint("aaf") != toint("atmoa")`。 ()
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 4 aad zpf zpz yyl, 则输出是什么? ()
- A. aad (换行) zpf (换行) zpz (换行) yyl (换行)
- B. aad zpf (换行) zpz yyl (换行)
- C. aad zpf zpz (换行) yyl (换行)
- D. aad zpf zpz yyl (换行)
20. 这个程序的时间复杂度是多少? ()
- A. $O(n)$ B. $O(n^2)$ C. $O(n\log n)$ D. $O(n^2\log n)$

(2)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 int calc(vector<vector<int>> &grid)
05 {
06     int n = grid.size(), m = grid[0].size();
07     vector<int> dp(m);
08     dp[0] = (grid[0][0] == 0);
09     for(int i = 0; i < n; i++)
10         for(int j = 0; j < m; j++)
11             {
12                 if(grid[i][j] == 1)
13                     {
14                         dp[j] = 0;
15                         continue;
16                     }
17                 if(j - 1 >= 0 && grid[i][j - 1] == 0)
18                     dp[j] += dp[j - 1];
19             }
20     return dp[m - 1];
```

```

21 }
22
23 int main()
24 {
25     int n, m;
26     cin >> n >> m;
27     vector<vector<int>> a(n, vector<int>(m));
28     for(int i = 0; i < n; i++)
29         for(int j = 0; j < m; j++)
30             cin >> a[i][j];
31     cout << calc(a) << endl;
32     return 0;
33 }

```

■ 判断题

21. 若输入 3 3 0 0 0 0 1 0 0 0 0, 则输出为 2。 ()
22. 若 $f[i][j]$ 表示从 $(0,0)$ 走到 (i,j) 的路径数, 则在第 10~19 行的循环中, $f[i][j] = dp[j]$ 。 ()
23. (2 分) 若将第 27 行的代码改为 `vector<vector<int>> a(n+1, vector<int>(m+1))`, 则当输入的 $n=3, m=3$ 时, `calc` 函数中的 $n=3, m=3$ 。 ()

■ 选择题

24. 当输入的 a 数组为 $\{\{0,0,1\}, \{1,1,0\}, \{0,1,0\}, \{1,0,1\}, \{0,0,0\}\}$ 时, 程序的输出为 ()。
- A. 0 B. 1 C. 2 D. 3
25. 若删除第 12~16 行的代码, 则当输入的 a 数组为 $\{\{0,0,0\}, \{0,1,0\}, \{0,0,0\}\}$ 时, 程序的输出为 ()。
- A. 1 B. 2 C. 3 D. 4
26. (4 分) 当输入的 a 数组为 $\{\{0,0,2\}, \{0,1,2\}, \{5,3,4\}\}$ 时, 程序的输出为 ()。
- A. 0 B. 1 C. 2 D. 3

(3)

```

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

```



```
03
04 using i64 = long long;
05
06 int cmp(string v1, string v2)
07 {
08     using i64 = long long;
09     int i = 0, j = 0;
10     while(i < v1.length() || j < v2.length())
11     {
12         i64 num1 = 0, num2 = 0;
13         while(i < v1.length() && v1[i] != '.')
14             num1 = num1 * 10 + (v1[i++] - '0');
15         while(j < v2.length() && v2[j] != '.')
16             num2 = num2 * 10 + (v2[j++] - '0');
17         if(num1 > num2)
18             return 1;
19         else if(num1 < num2)
20             return -1;
21         i++, j++;
22     }
23     return 0;
24 }
25
26 int main()
27 {
28     int n;
29     cin >> n;
30     vector<string> s(n);
31     for(int i = 0; i < n; i++)
32     {
33         cin >> s[i];
34         if(s[i][0] == '.')
35         {
36             cout << "err" << endl;
37             return 0;
38         }
39     }
40     for(int i = 0; i < n; i++)
41         for(int j = 0; j < n; j++)
42             cout << cmp(s[i], s[j]) << " \n"[j == n - 1];
```

```

43     return 0;
44 }

```

假设 $f[i][j] = \text{cmp}(s[i], s[j])$, 完成下面的问题。

■ 判断题

27. 任取 $0 \leq i < n$, 都有 $f[i][i] = 0$ 。 ()
28. 若输入 3 1.0.1 2.1 1.1.0, 则 $f[0][1] = 1$ 。 ()
29. 任取 $0 \leq i, j < n$, 都有 $f[i][j] + f[j][i] = 0$ 。 ()

■ 选择题

30. 当输入的 s 数组为{"1.2.3", "4.5", ".2"}时, 程序输出中第一行第二个数为 ()。
- A. -1 B. 0 C. 1 D. 不存在
31. (4分)若删除第 34~38 行代码, 则当输入的 s 数组为{"1.2.3", "4.5", ".2"}时, $f[0][2]$ 的值为 ()。
- A. -1 B. 0 C. 1 D. 未计算
32. 阅读代码可知, 当两个点之间的数为 () 时, cmp 函数将无法得到正确的结果。
- A. 1×10^9 B. 2×10^9 C. 4×10^{18} D. -1

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 ($3 \leq n \leq 2 \times 10^5$) 和长为 n 的数组 a ($1 \leq a[i] \leq 1 \times 10^9$)。你需要从 a 中恰好删除一个数, 得到长为 $n-1$ 的数组 a' 。然后生成一个长为 $n-2$ 的数组 b , 其中 $b[i] = \text{GCD}(a'[i], a'[i+1])$ 。你需要让数组 b 是非降序列, 即 $b[i] \leq b[i+1]$ 。能否做到? 输出 YES 或 NO。

(提示: 枚举 i , 考察删除 $a[i]$ 后对 b 数组产生的影响。)

```

01 #include <bits/stdc++.h>
02 using namespace std;
03
04 int gcd(int x, int y)

```

```
05 {
06     return !y ? x : gcd(y, x % y);
07 }
08
09 void solve()
10 {
11     int n;
12     cin >> n;
13     vector<int> a(n + 1), b(n + 2);
14     for(int i = 1; i <= n; i++)
15         cin >> a[i];
16     b[n] = b[n + 1] = ①;
17     for(int i = 1; i < n; i++)
18         b[i] = gcd(a[i], a[i + 1]);
19     vector<int> pre(n + 1), suf(n + 2);
20     pre[0] = 1;
21     for(int i = 1; i <= n; i++)
22         pre[i] = pre[i - 1] && (②);
23     suf[n + 1] = 1;
24     for(int i = n; i >= 1; i--)
25         suf[i] = suf[i + 1] && (b[i] <= b[i + 1]);
26     bool flag = ③;
27     for(int i = 2; i < n; i++)
28     {
29         int cur = ④;
30         if(⑤ && b[i - 2] <= cur && cur <= b[i + 1])
31             flag = true;
32     }
33     cout << (flag ? "YES\n" : "NO\n");
34     return;
35 }
36
37 int main()
38 {
39     int t = 1;
40     // cin >> t;
41     while(t--)
42         solve();
43 }
```


33. ①处应填 ()。

- A. 0 B. 3E9 C. -1E9 D. 2E9

34. ②处应填 ()。

- A. $b[i] \geq b[i-1]$ B. $b[i] \leq b[i-1]$
C. $b[i] > b[i-1]$ D. $b[i] < b[i-1]$

35. ③处应填 ()。

- A. $pre[n-2] \& suf[2]$ B. $pre[n-2] | suf[2]$
C. $pre[n-2] \wedge suf[2]$ D. $pre[n-2] - suf[2]$

36. ④处应填 ()。

- A. $\gcd(a[i], b[i])$ B. $\gcd(a[i], a[i+1])$
C. $\gcd(a[i-1], a[i+1])$ D. $\gcd(a[i-1], a[i])$

37. ⑤处应填 ()。

- A. $pre[i-2] \&\& suf[i+1]$ B. $pre[i-1] \&\& suf[i+1]$
C. $pre[i-2] || suf[i+1]$ D. $pre[i-1] || suf[i+1]$

(2) 题目描述:

输入 n ($1 \leq n \leq 2 \times 10^5$) 和长为 n 的数组 a ($1 \leq a[i] \leq 1 \times 10^6$)。

对于数组 B , 如果满足 $B[0] + 1 = \text{len}(B)$, 那么称数组 B 为“块”。对于数组 A , 如果可以将其划分成若干个“块”, 那么称数组 A 是合法的。

例如 $A = [3, 3, 4, 5, 2, 6, 1]$ 是合法的, 因为 $A = [3, 3, 4, 5] + [2, 6, 1]$, 这两段都是块。

把数组 a 变成合法数组, 至少要删除多少个元素?

(提示: 令 $dp[i]$ 表示将 $a[i]$ 到 $a[n]$ 变成合法数组最少要删除的元素个数。)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 const int inf = 0x3f3f3f3f;
05
06 int main()
07 {
```

```
08  int n;  
09  cin >> n;  
10  vector<int> a(n + 1), dp(①, inf);  
11  for(int i = 1; i <= n; i++)  
12      cin >> a[i];  
13  dp[n + 1] = ②;  
14  for(int i = n; i >= 1; i--)  
15  {  
16      dp[i] = ③;  
17      if(i + a[i] + 1 <= n + 1)  
18          dp[i] = min(dp[i], ④);  
19  }  
20  cout << ⑤ << endl;  
21  return 0;  
22 }
```

38. ①处应填 ()。

- A. n-1 B. n C. n+1 D. n+2

39. ②处应填 ()。

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

40. ③处应填 ()。

- A. dp[i+1] B. dp[i-1] C. dp[i+1]+1 D. dp[i-1]+1

41. ④处应填 ()。

- A. dp[i+a[i]+1] B. dp[i+a[i]]
C. dp[a[i]+1] D. dp[i+a[i]-1]

42. ⑤处应填 ()。

- A. dp[n] B. dp[1] C. dp[n-1] D. dp[0]