

扫码获取
答案解析

普及组 CSP-J 2025 初赛模拟卷 8

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. 在计算机的内存储器中, 每个存储单元都被赋予一个唯一的序号, 称为 ()。
A. 下标 B. 地址 C. 指针 D. 索引
2. 以下关于算法的描述中正确的是 ()。
A. 算法一定要用某种计算机语言写成程序才有价值
B. 要想实现算法, 必须先画流程图
C. 算法只需要用到数学的计算方法
D. 算法是为解决问题而采取的方法与步骤
3. 一张分辨率为 800×600 的 BMP 图片, 若每个像素用 24 位表示, 那么这张图片所占用的存储空间是 ()。
A. 1400KB B. 750KB C. 600KB D. 1000KB
4. 若某算法的计算时间表示为递推关系式: $T(N)=2T(N/2)+2N$, $T(1)=1$, 则其时间复杂度为 ()。
A. $O(\log n)$ B. $O(n^2 \log n)$ C. $O(n^2)$ D. $O(n \log n)$
5. 下列哪个特性不是数组和链表都可以实现的? ()
A. 数据元素之间的次序关系
B. 数据元素的动态添加和删除
C. 通过索引直接访问任意位置的数据元素
D. 数据可以为任意类型
6. 如果 $a = 2$, 那么经过运算 $a = \sim a + 2$, 最后 a 的值为 ()。
A. 3 B. 1 C. 0 D. 4
7. 用一个大小为 7 的数组来实现循环队列, 且 tail 和 head 的值分别为 0 和 4。当从队列中删除 2 个元素, 再加入 3 个元素后, tail 和 head 的值分别为 () 和 ()。

A. 6 3 B. 2 0 C. 3 6 D. 0 2

8. 关于二分算法, 下列说法中错误的是 ()。

- A. 二分算法可以用于二分查找、二分答案等不同应用
- B. n 个数的随机序列先排序再进行二分查找, 总时间复杂度是 $O(\log n)$
- C. 二分算法的左右区间可以左闭右闭, 也可以左闭右开
- D. 二分算法是典型的使用分治思想的算法

9. 如下代码主要表示什么数据结构? ()

```
typedef int LTDataType;
typedef struct ListNode
{
    struct ListNode* next;
    LTDataType data;
} LTNode;
```

- A. 单向链表 B. 双向链表 C. 循环链表 D. 优先队列

10. 关于字符串和字符串函数, 以下说法中错误的是 ()。

- A. `s = "ccfgesp"` 占用 8 字节内存空间
- B. 在字典序下, 字符串 `s1="123"` 比字符串 `s2="99"` 要小
- C. `s.substr(2,4)` 表示截取字符串 `s[2]~s[4]` 这一段的字符
- D. `cstring` 标准库包含了 `strcpy`、`strlen` 等函数

11. 在计算机历史上, 科学家冯·诺依曼的主要贡献是 ()。

- A. 发明了第一台计算机 ENIAC
- B. 破解了德军的 ENIGMA 密码
- C. 发明了二进制并应用到电子计算机中
- D. 提出存储程序的思想

12. 如下代码对树的操作是 ()。

```
void order(tree bt)
{
    if(bt)
    {
        cout << bt->value;
```

```

        order(bt->lchild);
        order(bt->rchild);
    }
}

```

- A. 前序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历
13. 给一排 10 个同样的玩偶的头发分别染红色和绿色, 要求任意两个绿色头发的玩偶不能相邻, 不同的染色方案共有 () 种。
A. 136 B. 140 C. 144 D. 150
14. 一棵完全二叉树共有 2026 个结点, 则该树中共有 () 个叶子结点。
A. 1014 B. 1013 C. 1012 D. 1011
15. 无向图 G 中有 2025 个度为 1 的结点, 2 个度为 2 的结点, 3 个度为 3 的结点, 4 个度为 4 的结点, 则无向图 G 有 () 条边。
A. 1025 B. 1026 C. 1027 D. 1028

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 const int N = 1e5 + 5;
04 int n, T, x, y, sum[N], is_prime[N];
05 int main() {
06     memset(is_prime, true, sizeof(is_prime));
07     for (int i = 2; i < N; ++i) {
08         if (is_prime[i]) {
09             for (int j = i + i; j < N; j += i)
10                 is_prime[j] = false;
11         }
12     }
13
14     for (int i = 1; i < N; ++i) {
15         sum[i] = sum[i - 1];

```

```
16     if (is_prime[i]) sum[i] += i;
17     }
18     scanf("%d %d", &x, &y);
19     if(x > y) swap(x, y);
20     printf("%d\n", sum[y] - sum[x - 1]);
21     return 0;
22 }
```

■ 判断题

16. 当输入为 1 5 时, 输出为 10。 ()
17. 若去除第 2 行, 程序仍能正常运行。 ()
18. (2 分) 在运行第 15 行时, 可能溢出 int 上界。 ()

■ 选择题

19. 若输入 91 95, 则输出为 ()。
- A. 0 B. 184 C. 91 D. 188
20. (4 分) 该程序的时间复杂度为 ()。
- A. $O(n)$ B. $O(n\log\log n)$ C. $O(n\log^2 n)$ D. $O(n^2)$

(2)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03 int l, r, x;
04 int restrict(int ql, int qr) {
05     return max(0, qr - max(l, ql) + 1);
06 }
07 int calc(int l, int r) {
08     if (l > r)
09         return 0;
10     x = 1;
11     while (x <= r)
12         x *= 2;
13     x /= 2;
14     return restrict(x, r) + calc(1, 2 * x - r - 1);
15 }
16 int main() {
17     scanf("%d%d", &l, &r);
```

```

18     printf("%d\n", calc(l, r));
19     return 0;
20 }

```

已知 $1 \leq l \leq r \leq 10^9$, 回答以下问题。

■ 判断题

21. 第 14 行中的 $2 * x - r - 1$ 一定比 r 小。 ()
22. 在运行第 12 行时, x 可能会溢出 `int` 的上界。 ()
23. $l=1, r=29$ 、 $l=1, r=30$ 、 $l=1, r=31$ 这三种情况的输出均一样。 ()
24. 若输入为 10 20, 则输出为 7。 ()

■ 选择题

25. 该程序的时间复杂度为 ()。
- A. $O(n)$ B. $O(\log n)$ C. $O(\log^2 n)$ D. $O(1)$
26. 若输入为 1 2007, 则输出为 ()。
- A. 1003 B. 1004 C. 1006 D. 1007

(3)

```

01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 5e3 + 5, mo = 998244353;
04 int n, j, ans1, ans2, mi, mj;
05 int C[N][N], a[N], pre[N], suf[N];
06 signed main() {
07     scanf("%d", &n);
08     C[0][0] = 1;
09     for (int i = 1; i <= n; ++i)
10         for (int j = 0; j <= i; ++j)
11             C[i][j] = (C[i - 1][j] + C[i - 1][j - 1]) % mo;
12     for (int i = 1; i <= n; ++i) {
13         scanf("%d", &a[i]);
14         if (a[i] == (n + 1) / 2) j = i;
15         else a[i] = (a[i] > (n + 1) / 2) ? 1 : -1;
16     }
17     for (int i = 1; i <= j - 1; ++i)

```



```
18     pre[i] = pre[i - 1] + a[i];
19     for (int i = n; i >= j + 1; --i)
20         suf[i] = suf[i + 1] + a[i];
21     mi = 0;
22     for (int i = 1; i <= j - 1; ++i)
23         if (!pre[i])
24             ++ans1, mi = i + 1;
25     mj = n;
26     for (int i = n; i >= j + 1; --i)
27         if (!suf[i])
28             ++ans2, mj = i - 1;
29     printf("%d %d\n", ans1 + ans2 + !(mi == mj), C[ans1 + ans2][ans1]);
30     return 0;
31 }
```

已知 $n \leq 5000$, n 为奇数, a 是长度为 n 的排列。完成下列各题。

■ 判断题

27. 若 pre 数组的最大值为 $\frac{n-1}{2}$, 则输出为 1 1。 ()
28. 第 29 行中的 $C[ans1 + ans2][ans1]$ 可以改成 $C[ans1 + ans2][ans2]$ 。 ()

■ 选择题

29. 若 $n=9$, 则输出的第一个数的最大值为 ()。
- A. 3 B. 4 C. 5 D. 6
30. 若输入为 7 1 6 2 4 5 7 3, 则输出为 ()。
- A. 3 2 B. 2 3 C. 3 3 D. 2 2
31. 若输入为 7 3 5 4 1 7 2 6, 则输出为 ()。
- A. 3 2 B. 2 3 C. 3 3 D. 2 2
32. (4 分) 若 $n=13$, 则输出的第二个数的最大值为 ()。
- A. 6 B. 10 C. 20 D. 36

三、完善程序 (单选题, 每小题 3 分, 共计 30 分)

(1) 题目描述:

给定一个长度不超过 10^4 的化学式, 计算其分子质量。(分子质量即一个化学式中原子质量之和。)

化学式可能有如下两种构成。

- 若原子只出现了一次, 则直接用大写字母表示, 如 H 代表氢元素, 原子质量为 1。若化学式为两个字母, 则首字母大写, 第二个字母小写, 如 Mg 代表镁元素, 原子质量为 24。
- 若原子出现了多次, 则用元素_{数量}代表有几个这种元素的原子, 如 C_{2} 代表有两个碳原子; H_{2}ClO_{4} 则表示 H 元素出现了 2 次, Cl 元素出现了 1 次, O 元素出现了 4 次。相对分子质量为 $1 \times 2 + 35.5 + 16 \times 4 = 101.5$ 。

编号	1	2	3	4	5	6	7	8	9	10	11	12
元素	H	C	N	O	F	P	S	Na	Mg	Al	Si	Cl
原子质量	1	12	14	16	19	31	32	23	24	27	28	35.5

```

01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 5e5 + 5;
04 const double val[N] = {0, 1, 12, 14, 16, 19, 31, 32, 23, 24, 27, 28, 35.5};
05 int n, to[N];
06 char s[N];
07 double ans;
08 int Hash(int x) {
09     if (to[s[x + 1]] >= 8) {
10         if (s[x + 1] == 'l')
11             return ①;
12         return to[s[x + 1]];
13     }
14     return to[s[x]];
15 }
16 int read(int x) {
17     int ans = 0;
18     for (int i = x; i <= n; i++) {
19         if (s[i] == '}') break;
20         ans = ②;

```

```
21     }
22     return ans;
23 }
24 int main() {
25     to['H'] = 1, to['C'] = 2, to['N'] = 3, to['O'] = 4;
26     to['F'] = 5, to['P'] = 6, to['S'] = 7;
27     to['a'] = 8, to['g'] = 9, to['l'] = 10, to['i'] = 11;
28     scanf("%s", s + 1);
29     n = strlen(s + 1)
30     for (int i = 1; i <= n; ++i) {
31         int x = Hash(i);
32         int j = i + 1 + (x >= 8);
33         if (s[j] == '_') {
34             int k = ③;
35             ans += val[x] * k;
36             while (④) ++i;
37             continue;
38         }
39         ans += val[x];
40         i += (x >= 8);
41         continue;
42     }
43     if (⑤) printf("%.0lf", ans);
44     else printf("%.1lf", ans);
45     return 0;
46 }
```

33. ①处应填 ()。

- A. $(s[x] == 'C') ? 10 : 12$ B. $(s[x] == 'A') ? 12 : 10$
C. $10 + 2 * (s[x] == 'C')$ D. $10 + 2 * (s[x] == 'A')$

34. ②处应填 ()。

- A. $ans + (int)(s[i])$ B. $ans * 10 + (int)(s[i])$
C. $ans + s[i] - '0'$ D. $ans * 10 + s[i] - '0'$

35. ③处应填 ()。

- A. $read(i+3)$ B. $read(j+1)$ C. $read(j+2)$ D. $read(i+2)$


```

01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 1e5 + 5;
04 const int mod = 1e9 + 7;
05 int n, m, sum, K, tot, f[2][1000], g[2][1000];
06 int a[N], r[N], to[N], pw[N] = {1};
07 int main() {
08     scanf("%d%d", &n, &m); --m;
09     for (int i = 1; i <= n; ++i) pw[i] = pw[i - 1] * 2 % mod;
10     for (int i = 1; i <= n; i++) scanf("%d", &a[i]);
11     for (int i = 1; i <= m; ++i) {
12         if (①) ++tot;
13         r[tot] = i, to[i] = tot;
14     }
15     for (int x = 1; x <= n; x++) {
16         int i = x & 1, k = i ^ 1;
17         for (int j = 1; j <= tot; j++)
18             f[k][j] = g[k][j] = 0;
19         if (a[x] <= m) {
20             f[i][to[a[x]]] += 1;
21             g[i][to[a[x]]] += ②;
22         }
23         for (int j = 1; j <= tot; j++) {

```

```

24         f[k][j] += f[i][j];
25         g[k][j] += g[i][j];
26         if (a[x + 1] * r[j] <= m) {
27             int mj = ③;
28             f[k][mj] += f[i][j];
29             g[k][mj] += ④;
30         }
31     }
32     sum = ⑤;
33     int rs = 0;
34     for (int j = 1; j <= tot; j++)
35         rs += g[i][j];
36     printf("%d\n", sum - rs);
37 }
38 return 0;
39 }

```

38. ①处应填 ()。

A. $i==1 || m/i!=m/(i-1)$

B. $i==1 || m/i!=m/(i+1)$

C. $m/i!=m\%(i-1)$

D. $m/i!=m/(i+1)$

39. ②处应填 ()。

A. 1

B. $a[x]$

C. sum

D. $a[x] * pw[i-1]$

40. ③处应填 ()。

A. $to[a[x+1]*r[j]]-1$

B. $to[a[x]*r[j]]$

C. $to[a[x+1]*r[j]]$

D. $to[a[x+1]*r[j]]+1$

41. ④处应填 ()。

A. $g[i][j]+f[i][j]*a[x+1]$

B. $g[i][j]+f[i][j]*a[x+1]*pw[i-1]$

C. $f[i][j]*a[x+1]$

D. $f[i][j]*a[x+1]*pw[i-1]$

42. ⑤处应填 ()。

A. $sum*2+pw[i]*a[x]$

B. $sum+a[x]$

C. $sum*2+pw[i-1]*a[x]$

D. $(sum+a[x])*pw[i]$