普及组 CSP-J 2025 初赛模拟卷 10

扫码获取答案解析

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. 以下列扩展名结尾的文件，不是多媒体文件的是（　　）。

A. mp3　　B. txt　　C. avi　　D. jpg

2. 以下关于链表和数组的描述中，错误的是（　　）。

A. 数组和链表都可以排序

B. 数组中查询元素的效率比较高

C. 链表中插入和删除元素的效率比较高

D. 向量和静态数组一样，不能动态调整数组大小

3. 与 C++语言中的 `cout << a > b ? 'a' : 'b';` 功能类似的是（　　）。

A. 顺序结构　　B. 循环结构　　C. 条件结构　　D. 递推函数

4. 下面的 C++代码中 data 占用（　　）字节内存空间。

```
union Data
{
   int no;
   double score;
   char name[6];
};
union Data data;
```

A. 4　　B. 8　　C. 18　　D. 6

5. 学号为 1 到 30 的幼儿园小朋友顺时针围成一圈，从 1 号小朋友开始按顺时针方向报数，报数从 0 开始，依次为 0,1, 2, 3, …, 28, 29, 30, 31, 32,…，一圈又一圈，问数到数字 n 的小朋友的学号是多少？（　　）

A. `n%30+1`　　B. `(n+1)%30`　　C. `(n+1)%30+1`　　D. `n%30`

6. 以下哪个不属于 STL 模板中队列的操作函数？（　　）

A. push　　B. pop　　C. empty　　D. top

7. 在 C++语言中，（　　）算法的时间复杂度是 $O(n\log n)$。

A. 插入排序　　B. 归并排序　　C. 选择排序　　D. 冒泡排序

8. 以下关于字符串的判定语句中正确的是（　　）。

A. 字符串一般以字符'0'结尾

B. 串的长度必须大于零

C. string s;中定义的 s 也可以看作字符数组，首字母是 s[0]

D. 全部都由空格字符组成的串就是空串

9. 以下算法中，（　　）算法用到了栈。

A. BFS　　B. 二分查找　　C. DFS　　D. 贪心

10. 32 位计算机系统中一个非负长整型指针变量 unsigned long long *p 占（　　）字节。

A. 1　　B. 2　　C. 8　　D. 4

11. 某山峰型数列有 1~2025 共 2025 个各不相同的数，先是奇数由小到大，后是偶数由大到小，即 1, 3, 5, 7, 9, …, 2023, 2025, 2024, 2022, 2020, …, 8, 6, 4, 2。现要对该数列进行检索，查找某正整数 x 的下标（x 为 1~2025 中的某正整数，包含 1 和 2025），最多检索（　　）次即可。

A. 2025　　B. 11　　C. 10　　D. 9

12. 在 C++程序中，lowbit(x)函数返回整数 x 在二进制表示下最低一位 1 以及后续 0 一起表示的数字，如 lowbit(12)=4。下面的表达式中，(　　)能得到相同的结果。

A. x ^ (x - 1)　B. x & (x - 1)　C. x & (~x + 1)　D. x | (x - 1)

13. 某二叉树的中序遍历序列为 BDCEAFHG，后序遍历序列为 DECBHGFA，其前序遍历序列为（　　）。

A. ABCDEFGH　　B. ABDCEFHG　　C. ABCDFEHG　　D. ABDCEFGH

14. 有 5 条线段，长度分别为 1, 3, 5, 7, 9，从中任取 3 条能构成一个三角形的概率为(　　)。

A. 1/2　　B. 3/10　　C. 1/5　　D. 2/5

15. 对于非负整数组{x,y,z}，满足 x+2y+3z=100 的非负整数解组数为（　　）个。

A. 886　　B. 885　　C. 884　　D. 883

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 = 2e4 + 5, inf = 2e9 + 7;
04 int n, a[N], ans = inf;
05 int main() {
06     scanf("%d", &n);
07     for (int i = 1; i <= n; i++)
08         scanf("%d", &a[i]);
09     sort(a + 1, a + n + 1);
10     for (int i = 2; i <= n; i += 2) {
11         int mi = inf, ma = -inf, x = 0;
12         for (int j = 1; j <= i; ++j) {
13             x = a[j] + a[i - j + 1];
14             mi = min(mi, x), ma = max(ma, x);
15         }
16         if (i ^ n)
17             ans = min(ans, max(a[n], ma) - min(a[i + 1], mi));
18         else
19             ans = min(ans, ma - mi);
20     }
21     return printf("%d\n", ans), 0;
22 }
```

■ **判断题**

16. 若将程序第 12 行中的 i 改成 i/2，程序的输出结果一定不会改变。（　　）

17. （2 分）若将程序第 10 行中的 i+=2 改成 i++，程序的输出结果一定不会改变。（　　）

18. 若将头文件#include <bits/stdc++.h>改成#include <iostream>，程序仍能正常运行。（　　）

■ 选择题

19. 若输入 4 1 3 6 7，则输出为（　　）。

A. 0　　B. 1　　C. 2　　D. 3

20.（4分）若输入 7 2 8 9 15 17 18 16，则输出为（　　）。

A. 2　　B. 3　　C. 4　　D. 5

（2）

```
01 #include <bits/stdc++.h>
02 using namespace std;
03 int n, m, k, l, r, mid;
04 int check(int g) {
05     int st = 1, ed = m, cnt = 0;
06     while (st <= n && ed >= 1) {
07         if (st * ed > g)
08             ed--;
09         else {
10             cnt += ed;
11             st++;
12         }
13     }
14     return cnt >= k;
15 }
16 int main() {
17     scanf("%d%d%d", &n, &m, &k);
18     l = 1, r = n * m;
19     while (l < r) {
20         mid = (l + r) / 2;
21         if (check(mid)) r = mid;
22         else l = mid + 1;
23     }
24     cout << l << endl;
25 }
```

已知 k≤nm，保证 n,m 同阶，完成以下问题。

■ 判断题

21. 每次运行 check 时，第 7 行必定运行 n 次。（　　）

22. 如果保证 m=1，则输出一定为 k。 ( )
23. 若将第 21 行中的 r = mid 改成 r = mid - 1，程序输出一定不变。 ( )
24. 第 24 行可以改成 cout << r << endl;。 ( )

■ 选择题

25. 该程序的时间复杂度为 ( )。

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

26. 若输入为 2 3 4，则输出为 ( )。

A. 1 B. 2 C. 3 D. 6

(3)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 10, dx[10] = {0,1,0,-1,0}, dy[10] = {0,0,1,0,-1};
04 int n, m, ans;
05 char c[N][N];
06 void dfs(int x) {
07     if (!x) return ++ans, void();
08     vector<int> v; v.clear();
09     for (int i = 1; i <= n; ++i)
10     for (int j = 1; j <= n; ++j)
11     if (c[i][j] == '.') {
12         bool flag = 0;
13             for (int k = 1; k <= 4; ++k) {
14             int tx = i + dx[k], ty = j + dy[k];
15             flag |= tx &&tx<=n &&ty &&ty<=n &&c[tx][ty] == 1;
16            }
17         if (flag)
18             v.push_back(i*10+j),c[i][j]=1,dfs(x-1),c[i][j]='#';
19     }
20     if (!v.empty())
21         for (int i = 0; i < v.size(); ++i)
22             c[v[i] / 10][v[i] % 10] = '.';
23 }
24 int main() {
25     scanf("%d%d", &n, &m);
26     for (int i = 1; i <= n; ++i)
```

```
27        scanf("%s", c[i] + 1);
28    for (int i = 1; i <= n; ++i)
29        for (int j = 1; j <= n; ++j)
30            if (c[i][j] == '.')
31                c[i][j] = 1, dfs(m - 1), c[i][j] = '#';
32    return printf("%d\n", ans), 0;
33 }
```

■ **判断题**

27. 将第18行中的c[i][j]='#'去除，结果一定不变。（　　）

28. 第6行运行的次数不超过$n^2 4^m$。（　　）

■ **选择题**

29. 若输入为2 2 #. .. ，则输出为（　　）。

A. 0　　B. 1　　C. 2　　D. 3

30. 若输入为3 5 #.# … .#. ，则输出为（　　）。

A. 2　　B. 3　　C. 4　　D. 5

31. （4分）若n=3，m=3，则输出的最大值为（　　）。

A. 16　　B. 18　　C. 22　　D. 26

32. 若n=4，m=13，则输出的最大值为（　　）。

A. 488　　B. 496　　C. 512　　D. 560

三、完善程序（单选题，每小题3分，共计30分）

（1）题目描述：

给定两个由小写字母构成的字符串 s1 和 s2，同时给定一个由数字 1,2,3…|operation|组成的排列。按该排列顺序依次删除字符串s1相应位置上的字母，在删除过程中，约定各个字符的位置不变。请计算最多可以删除几次，字符串 s1 中仍然包含字符串s2（即字符串s2仍然是字符串s1的子串）。

```
01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 2e5 + 5;
```

```
04 bool book[N];
05 char s1[N], s2[N];
06 vector <int> num;
07 int n = 1, len1, len2, ans, operation[N];
08 bool check(int x) {
09     num.clear();
10     for (int i = x + 1; i <= n; i++)
11         if (①)
12             num.push_back(operation[i]);
13     sort(num.begin(), num.end());
14     int i = 0, j = 1;
15     while (②)
16         j += ③;
17     return j == len2 + 1;
18 }
19 inline void fuc(int l, int r) {
20     if (④) return;
21     int mid = (l + r) >> 1;
22     if (check(mid)) ans = mid, fuc(mid + 1, r);
23     else fuc(l, mid - 1);
24 }
25 int main() {
26     scanf("%s", s1 + 1);
27     scanf("%s", s2 + 1);
28     len1 = strlen(s1 + 1);
29     len2 = strlen(s2 + 1);
30     while (⑤) ++n;
31     n--;
32     for (int i = 1; i <= len2; i++)
33         book[int(s2[i])] = true;
34     fuc(1, n);
35     printf("%d\n", ans);
36     return 0;
37 }
```

33. ①处应填（　　）。

A. book[s1[operation[i]]]　　B. book[s1[i]]

C. !book[s1[operation[i]]]　　D. !book[s1[i]]

34. ②处应填（　　）。

A. i<num.size()&&j<=len2　　B. i<=num.size()&&j<=len2

C. i<=num.size()&&j<len2　　D. i<num.size()&&j<len2

35. ③处应填（　　）。

A. s1[num[i++]]==s2[j]　　B. s1[num[++i]]==s2[j]

C. s1[num[i++]]==s2[j++]　　D. s1[num[++i]]==s2[j++]

36. ④处应填（　　）。

A. l>=r　　B. l==r　　C. l>r　　D. l^r

37. ⑤处应填（　　）。

A. scanf("%d",&operation[n])　　B. ~scanf("%d",&operation[n])

C. ~(cin>>operation[n])　　D. !scanf("%d",&operation[n])

（2）题目描述：

如果存在一个长度为 n 的排列（即该排列由 1, 2, 3, …, n 这 n 个数字各出现一次组成），对于所有满足 $2 \leqslant i \leqslant n-1$ 的整数 i，都有 $p_i \leqslant p_{i-1}, p_i \leqslant p_{i+1}$ 或者 $p_i \geqslant p_{i-1}, p_i \geqslant p_{i+1}$ 成立，则称这个序列为一个山峰山谷序列。

对所有长度为 n 的山峰山谷序列排序，求字典序第 k 大的排列。

$dp_{i,j,0/1}$ 表示长度为 i 的排列中第一个数为 j，其中第一个数小于/大于第二个数。

```
01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 105;
04 int n, k, dp[N][N][2], ans[N], vis[N];
05 signed main() {
06     scanf("%d%d", &n, &k);
07     dp[1][1][0] = dp[1][1][1] = 1;
08     dp[2][1][0] = dp[2][2][1] = 1;
09     for (int i = 2; i < n; ++i)
10         for (int j = 1; j <= i; ++j)
11             for (int k = 1; k <= i + 1; ++k)
12                 if (①) dp[i + 1][k][1] += dp[i][j][0];
13                 else ②;
14     for (int i = 1; i <= n; ++i) {
```

```
15        int las = 0, mk = 1;
16        if (i > 2 && ans[i - 1] < ans[i - 2])
17            mk = ③;
18        for (int j = mk; j <= n; ++j) if (!vis[j]) {
19            int cnt = 0;
20            for (int k = 1; k <= j; ++k) if (!vis[k]) cnt++;
21            int x = ④;
22            if (i == 1) x += dp[n][j][0];
23            if (x >= k) { las = j; break; }
24            ⑤;
25        }
26    ans[i] = las, vis[las] = 1;
27    }
28    for (int i = 1; i <= n; ++i)
29        printf("%d ", ans[i]);
30    return 0;
31 }
```

38. ①处应填（　　）。

A. j < k　　B. j <= k　　C. i < k　　D. i <= k

39. ②处应填（　　）。

A. dp[i+1][k][1]+=dp[i][j][0]　　B. dp[i+1][k][0]+=dp[i][j][1]

C. dp[i+1][j][1]+=dp[i][k][0]　　D. dp[i+1][j][0]+=dp[i][k][1]

40. ③处应填（　　）。

A. ans[i-1]+1　　B. ans[i-2]+1　　C. i+1　　D. ans[i-1]

41. ④处应填（　　）。

A. dp[n-i+1][cnt][ans[i-1]<j]

B. dp[n-i+1][cnt][ans[i-1]>j]

C. dp[n-i+1][j-cnt][ans[i-1]>j]

D. dp[n-i+1][j-cnt][ans[i-1]<j]

42. ⑤处应填（　　）。

A. k-=x　　B. k-=x*(n-i+1)　　C. k-=x*cnt　　D. k-=x*mk