普及组 CSP-J 2025 初赛模拟卷 3

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. 如果 a 和 b 都是 char 类型的变量，下列哪个语句不符合 C++语法？（　　）

A. `b = ++a;`　　B. `b = 'a'++;`

C. `b = 'a' + '1';`　　D. `b = a++;`

2. 泛洪填充算法属于（　　）算法。

A. 贪心　　B. 二分　　C. 动态规划　　D. 搜索

3. 在下列代码的横线处填写（　　），可以使得输出是“5 8”。

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    Int x = 8, y = 5;
    _________;
    x = x ^ y;
    y = x ^ y;
    cout << x << " " << y << endl;
    return 0;
}
```

A. `x = x ^ y`　　B. `y = x ^ y`

C. `a = x + y`　　D. `x = x + y`

4. 小写字母 a 的 ASCII 码值为 97，小写字母 z 的 ASCII 码值用八进制数表示为（　　）。

A. 170　　B. 174　　C. 172　　D. 171

5. 从 n 个正整数 1, 2, …, n 中任意取出两个不同的数，若取出的两数之和等于 5 的概率为 1/14，则 n 为（　　）。

A. 6　　B. 7　　C. 8　　D. 9

6. 下面不可以用作 C++程序中的变量名的是（　　）。

A. cstr　　B. cint　　C. pops　　D. this

7. 设有 n 个数和 m 个桶，桶排序算法（桶内采用插入排序）在最坏情况下的时间复杂度是（　　）。

A. $O(nm)$　　B. $O(n+m)$　　C. $O(n^2)$　　D. $O(n\log n)$

8. 一个二维数组定义为 `long long a[5][8];`，则这个二维数组占用内存空间的大小为（　　）字节。

A. 320　　B. 160　　C. 80　　D. 40

9. 下列关于 C++语言中自定义函数的叙述，正确的是（　　）。

A. 自定义函数的参数可以是结构体类型

B. 自定义函数的参数不能超过五个

C. 自定义函数必须有返回值

D. 自定义函数定义必须写在调用它的函数前面，否则会发生编译错误

10. 为了防范计算机病毒，保护个人隐私和信息安全，下列做法中正确的是（　　）。

A. 每六个月更换一次计算机的登录密码，密码采用大小写英文字母和数字混合的形式，位数多于 8 位

B. 手机收到提示中奖的短信，点开链接看看是否真的中奖了

C. 将个人的私密照片和视频发到同学间建立的 QQ 群

D. 借用同学的 U 盘将下载的网络游戏安装包复制到自己的笔记本计算机中

11. 下列代码可以用来求最长上升子序列（LIS）的长度，如果输入是 5 1 7 3 5 9，则输出是（　　）。

```
#include <bits/stdc++.h>
using namespace std;
int a[2025],dp[2025];
int main()
{
    int n,i,j,ret = -1;
    cin >> n;
    for(i=1; i<=n; ++i)
    {
```

```
        cin >> a[i];
        dp[i] = 1;
    }
    for(i=1; i<=n; ++i)
        for(j=1; j<i; ++j)
            if(a[j] < a[i])
                dp[i] = max(dp[i],dp[j]+1);
    for(i=1; i<=n; ++i)
    {
        ret = max(ret,dp[i]);
        cout << dp[i] << " ";
    }
    cout << ret << endl;
    return 0;
}
```

A. 9 7 5 1 1 9　B. 1 2 2 3 4 4　C. 1 3 5 7 9 9　D. 1 1 1 1 1 1

12. 已知逻辑表达式 A=true，B=C=D=false，则以下逻辑表达式中取值为真的是（　　）。

A. (C∧D∨¬A)∨(A∧C∨D)　　B. ¬((A∧B∨C)∧(D∨B))

C. (A∧(B∨C∨D))∨(A∧D)　　D. (A∨(C∨D))∧(B∨C)

13. 某二叉树的前序遍历序列为 ABDFCEGH，中序遍历序列为 BFDAGEHC，则下列说法中正确的是（　　）。

A. 树的高度为 3

B. 点 A 的右子树共有 4 个结点

C. 树可能有 4 个叶子结点或者 2 个叶子结点

D. 以上说法都不对

14. 设 p 为 2~100 范围内的质数，p^3+7p^2 为完全平方数，则 p 的取值有（　　）种不同的可能。

A. 2　　B. 1　　C. 3　　D. 0

15. 在图的广度优先搜索中，既要维护一个标志数组来标志已访问的结点，还需使用（　　）结构存放结点以实现遍历。

A. 栈　　B. 堆　　C. 队列　　D. 哈希表

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 bool isValid(string s) {
05     stack<char> stk;
06     for (char ch: s) {
07         if (ch == '(' || ch == '[' || ch == '{') {
08             stk.push(ch);
09         } else {
10             if (stk.empty()) {
11                 return false;
12             }
13             if (ch == ')' && stk.top() != '(') {
14                 return false;
15             }
16             if (ch == ']' && stk.top() != '[') {
17                 return false;
18             }
19             if (ch == '}' && stk.top() != '{') {
20                 return false;
21             }
22             stk.pop();
23         }
24     }
25     return stk.empty();
26 }
27
28 int main() {
29     string s;
30     cin >> s;
31     if(isValid(s))
32         cout << "Valid" << endl;
33     else
34         cout << "Invalid" << endl;
35     return 0;
36 }
```

■ 判断题

16. 若程序输入({[]})，则程序输出 Valid。 （ ）

17. 若将第 10~12 行代码删除，则程序依然可以正常运行。 （ ）

18. 若删除头文件<bits/stdc++.h>，则只需要添加<iostream>头文件就可以通过编译。 （ ）

■ 选择题

19. 若输入((({{[]}})))，则输出是什么？（ ）

A. Valid　　B. Invalid　　C. invalid　　D. valid

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. $O(n\sqrt{n})$

（2）

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 int main() {
05     int n;
06     cin >> n;
07     vector<int> a(n);
08     for(int i = 0; i < n; i++)
09         cin >> a[i];
10     vector<int> dp(n + 1, 0);
11     for(int i = 0; i < n; i++) {
12         if(i >= 2)
13             dp[i] = max(dp[i - 1], dp[i - 2] + a[i]);
14         else
15             dp[i] += a[i];
16         if(i >= 1)
17             dp[i] = max(dp[i], dp[i - 1]);
18     }
19     int ans = 0;
20     for(int i = 0; i < n; i++) {
21         ans = max(ans, dp[i]);
22     }
23     cout << ans << endl;
24     return 0;
25 }
```

■ 判断题

21. 若输入 5 2 7 9 3 1，则输出为 12。 ( )

22. 这段代码对应的状态转移方程为 dp[i] = max(dp[i-1], dp[i-2]+a[i]),i>=2;初值为 dp[0] = a[0], dp[1] = a[1]。 ( )

23. （2 分）在主函数中，访问 dp[n]不会发生越界错误。 ( )

■ 选择题

24. 当输入的 a 数组为{2, 1, 1, 2}时，程序的输出为 ( )。

A. 1 B. 2 C. 3 D. 4

25. 若将第 13 行改为 dp[i] = max(dp[i-1], dp[i-2] - a[i]);，则当输入的 a 数组为{10, 1, 0, 25, 3}时，程序的输出为 ( )。

A. 1 B. 10 C. 35 D. 25

26. （4 分）当输入的 a 数组为{0, 2, 3, 0, 5, 6, 0, 8, 9}时，程序的输出为 ( )。

A. 18 B. 33 C. 34 D. 2

(3)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 int bfs(vector<vector<int>>& grid) {
05     const int m = grid.size(), n = grid[0].size();
06     const int dx[] = {-1, 0, 1, 0}, dy[] = {0, -1, 0, 1};
07     using pii = pair<int, int>;
08     queue<pii> q;
09     vector<vector<bool>> vis(m + 1, vector<bool>(n + 1));
10     int ans = 0, tot = 0, cnt = 0;
11     for(int i = 0; i < m; i++)
12         for(int j = 0; j < n; j++) {
13             if(grid[i][j] == 2)
14                 q.push({i, j}), vis[i][j] = 1;
15             tot += (grid[i][j] != 0);
16         }
17     while(!q.empty()) {
18         int cur = q.size();
19         for(int i = 1; i <= cur; i++) {
```

```
20          auto u = q.front();
21          int x = u.first, y = u.second;
22          q.pop();
23          cnt++;
24          for(int j = 0; j < 4; j++) {
25              int cx = x + dx[j], cy = y + dy[j];
26              if(cx < 0 || cx >= m || cy < 0 || cy >= n ||
                        vis[cx][cy] || grid[cx][cy] == 0)
27                  continue;
28              if(grid[cx][cy] == 1)
29                  q.push({cx, cy}), vis[cx][cy] = 1;
30          }
31      }
32      ans++;
33  }
34  if(tot == cnt)
35      return max(0, ans - 1);
36  else
37      return -1;
38 }
39
40 int main() {
41  int m, n;
42  cin >> m >> n;
43  vector<vector<int>> a(m, vector<int>(n));
44  for(int i = 0; i < m; i++)
45      for(int j = 0; j < n; j++)
46          cin >> a[i][j];
47  cout << bfs(a) << endl;
48  return 0;
49 }
```

■ **判断题**

27. 当输入的 a 数组为{{2,1,1}, {1,1,0}, {0,1,1}}时，程序的输出为 4。（　　）
28. bfs 函数的时间复杂度为 $O(nm)$。（　　）
29. 由代码可知，格子中的一个 2 可以把八个方向上的 1 都变为 2。（　　）

■ **选择题**

30. 当输入的 a 数组为{{2,1,1}, {0,1,1}, {1,0,1}}时，程序的输出为（　　）。

A. -1　　B. 2　　C. 3　　D. 4

31.（4 分）如果删掉 bfs 函数中与 vis 数组相关的内容，不可能发生的结果是（　　）。

A. 不影响结果，答案依然正确

B. 某个格子会重复进入队列，但答案依然正确

C. 某个格子会重复进入队列，将得到不正确的答案

D. 无法控制格子的入队次数，内存超限

32. 当输入的 a 数组为{{0,2}}时，程序的输出为（　　）。

A. 0　　B. 1　　C. -1　　D. 发生运行时错误

三、完善程序（单选题，每小题 3 分，共计 30 分）

（1）题目描述：

给定一个长度小于或等于 10^6 的只包含小写英文字母的字符串 s，输出有多少个子串满足以 heavy 开头并且以 metal 结尾。

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 int main() {
05     string s;
06     cin >> s;
07     long long cnt = ①, ans = 0;
08     for(int i = 4; ②; i++) {
09         auto cur = ③;
10         if(④)
11             cnt++;
12         else if(cur == "metal")
13             ⑤;
14     }
15     cout << ans << endl;
16     return 0;
17 }
```

33. ①处应填（　　）。

A. 0　　B. 1E9　　C. -1E9　　D. -2E9

34. ②处应填（　　）。

A. i < s.length()　　B. i <= s.length()

C. i < s.length() - 1　　D. i >= s.length()

35. ③处应填（　　）。

A. s.substr(i, 5)　　B. s.substr(i - 5, 5)

C. s.substr(i - 4, 5)　　D. s.substr(i - 5)

36. ④处应填（　　）。

A. cur == heavy　　B. cur == "heavy"

C. cur != heavy　　D. cur != "heavy"

37. ⑤处应填（　　）。

A. ans++　　B. ans += cnt　　C. cnt += ans　　D. cnt++

（2）题目描述：

输入 l 和 r（$1 \leqslant l \leqslant r \leqslant 10^{18}$）。如果整数 x 的首位数字等于末位数字，那么称 x 是合法数字。例如 101,477474,9 是合法数字，而 47,253,1020 不是合法数字。输出[l,r]中有多少个合法数字？

（提示：考虑最低位和最高位之间的关系。）

```
01 #include <bits/stdc++.h>
02 using namespace std;
03 long long l,r;
04 long long fir(long long g) {
05     while(g>=10) ①;
06     return g;
07 }
08 long long fin(long long g) {
09     return g%10;
10 }
11 long long solve(long long n) {
12     if(n<=9)return n;
13     else {
14         long long base=(②);
15         long long first=fir(n);
16         bool flag=(③);
```

```
17        return ④;
18    }
19 }
20 int main() {
21    cin>>l>>r;
22    cout<<⑤<<endl;
23    return 0;
24 }
```

38. ①处应填（　　）。

A. g /= 10　　B. g -= 10　　C. g %= 10　　D. g ^= 10

39. ②处应填（　　）。

A. n % 10 + 9　　B. n % 10　　C. n / 10　　D. n / 10 + 9

40. ③处应填（　　）。

A. fir(n)<fin(n)　　B. fir(n)<=fin(n)

C. fir(n)>fin(n)　　D. fir(n)>=fin(n)

41. ④处应填（　　）。

A. base + flag　　B. base

C. flag　　D. base - flag

42. ⑤处应填（　　）。

A. solve(r)　　B. solve(l)

C. solve(r)-solve(l-1)　　D. solve(r)-solve(l)