

普及组 CSP-J 2025 初赛模拟卷 7



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. 八进制数 2025 用二进制表示是 ()。
A. 10000000101 B. 10000010101 C. 10000001101 D. 10000100101
2. C++语言的创始人是 ()。
A. 林纳斯·托瓦茨 B. 丹尼斯·里奇
C. 吉多·范罗苏姆 D. 本贾尼·斯特劳斯特卢普
3. 以下设备中, () 不是输出设备。
A. 扫描仪 B. 触摸屏 C. 绘图仪 D. 音箱
4. 当执行以下 C++程序段后输出结果为 ()。

```
char c1 = '2' + '0';  
char c2 = '2' + '5';  
cout << c1 << c2 << endl;
```


A. 2025 B. 27 C. bg D. ch
5. 应用二分算法的思想, 在一个有 n 个数的有序序列中查找某个指定的数 m , 其程序时间复杂度为 ()。
A. $O(n\log n)$ B. $O(n)$ C. $O(\log n)$ D. $O(m\log n)$
6. 贝希要参加 CSP-J 比赛, 在 CCF 官网注册时需设置登录密码, 下列选项中 () 最安全。
A. 12345678 B. abcd1234 C. 20010911 D. F1@CcfGq6dh
7. 双向链表的优点是 ()。
A. 查找速度快 B. 插入和删除方便
C. 节省内存 D. 后进先出

8. 小明买了一块 1TB 的固态硬盘, 相当于 () MB 的存储容量。
A. 2^{10} B. 2^{20} C. 2^{30} D. 2^{40}
9. 下面 () 没有用到有关人工智能的技术。
A. 智能手机设置的闹钟定时叫我起床
B. 智能手环收集患者的数据并上传至医疗系统云端进行分析
C. 国家围棋队棋手和围棋机器人下围棋
D. 大学校园用人脸识别门禁系统控制人员出入
10. 下列选项中 () 不是 C++ 标准库 string 类的函数。
A. substr B. size C. replace D. strcmp
11. 有一个 2025 位的正整数, 它的各位数字按照如下规则排列:
123456789123456789123456789..., 请问这个数被 9 除的余数是多少? ()
A. 3 B. 2 C. 0 D. 1
12. 九宫格数独游戏是一种训练推理能力的数字谜题游戏。九宫格分为九个小宫格, 某小九宫格如图所示, 小明需要在 9 个小格子中填上 1 至 9 中不重复的整数。小明通过推理已经得到了 4 个小格子中的准确数字, 其中, a、b、c、d、e 这 5 个数字未知, 且 b 和 d 为奇数, 则 $a+b>5$ 的概率为 ()。
- | | | |
|---|---|---|
| 9 | a | 7 |
| b | c | d |
| 4 | e | 5 |
- A. $3/5$ B. $1/2$ C. $2/3$ D. $1/3$
13. 四位同学进行篮球传球练习, 要求每个人接球后再传给别人。开始时甲同学发球, 并作为第一次传球, 第五次传球后, 球又回到甲同学手中, 则不同的传球方法有 () 种。
A. 60 B. 65 C. 70 D. 75
14. 字符串 CCCSSSPPP 共有 () 种不同的非空子串。
A. 45 B. 36 C. 37 D. 39

15. 向一个栈顶指针为 head 的链式栈中插入一个指针 p 指向的结点时, 应执行 ()。
- A. head->next = p;
 - B. p->next = head; head = p;
 - C. p->next = head->next; head->next = p;
 - D. p->next = head; head = head->next;

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 int t, p, a, b, c;
04 int f(int a,int b){
05     if(a % b == 0) return 0;
06     return b - a % b;
07 }
08 void solve(){
09     scanf("%d%d%d", &p, &a, &b, &c);
10     printf("%d\n", min(min(f(p,a), f(p,b)), f(p,c)));
11 }
12 int main(){
13     scanf("%d", &t);
14     while(t--) solve();
15     return 0;
16 }
```

■ 判断题

16. 若程序输入
- ```
1
2 6 10 9
```
- 则最终输出为 4。 ( )
17. (2 分) 若将第 5 行删除, 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 <stdio.h>, 程序仍能正常运行。 ( )

## ■ 选择题

19. 若程序输入 2 9 5 4 8 10 9 9 9, 则输出是 ( )。

- A. 1 8                      B. 1 1                      C. 0 8                      D. 0 1

20. (4 分) 若将第 10 行的输出内容改为  $f(f(f(p, a), b), c)$ , 则输入

1  
2 6 10 9

时, 输出是 ( )。

- A. 3                      B. 4                      C. 5                      D. 6

## (2)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 2e5 + 5;
04 int a, b, n, k;
05 char s[N];
06 void solve(int &a, int &b, int k) {
07 while (k--)
08 putchar(a ? --a, '1' : (--b, '0'));
09 }
10 int main() {
11 cin >> a >> b >> k; n = a + b;
12 for (int i = 1; i <= a; ++i) s[i] = '1';
13 for (int i = 1; i <= b; ++i) s[i + a] = '0';
14 while ((s[n - k] == '0' || s[n] == '1') && n > k)
15 --n;
16 if (n <= k + 1 && k) return printf("No\n"), 0;
17 printf("Yes\n");
18 if (!k) return printf("%s\n%s", s + 1, s + 1), 0;
19 a -= 2, b -= 1;
20 int A = a, B = b;
21 printf("11");
22 solve(A, B, k - 1);
23 putchar('0');
24 solve(A, B, a + b - k + 1);
25 A = a, B = b;
26 printf("\n10");
27 solve(A, B, k - 1);
28 putchar('1');
```

```

29 solve(A, B, a + b - k + 1);
30 return 0;
31 }

```

### ■ 判断题

21. 去掉第 6 行的两个 &, 程序的输出一定不改变。 ( )
22. 如果  $a=0$ ,  $k \neq 0$ , 则必定输出 No。 ( )
23. 当  $a+b < k+3$  时, 必定输出 No。 ( )
24. 若输出 Yes, 则输出的两个数在二进制下的差一定有  $k$  位 1。 ( )

### ■ 选择题

25. 若输入为 2 4 3, 则输出为 ( )。
- A. Yes 101000 100001                      B. Yes 110000 100010
- C. Yes 110000 100001                      D. No
26. 若输入为 2 3 4, 则输出为 ( )。
- A. Yes 10100 10001                      B. Yes 11000 10001
- C. Yes 10001 00011                      D. No

(3)

```

01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 2e5 + 5;
04 int n, m, ans, pos[2][N];
05 char a[N], b[N];
06 int main() {
07 scanf("%d%d%s%s", &n, &m, a, b);
08 reverse(a, a + n); reverse(b, b + m);
09 for (int i = 0, now = 0; i < n && now < m; ++i)
10 if (a[i] == b[now])
11 pos[0][now++] = i;
12 for (int i = n - 1, now = m - 1; ~i && ~now; --i)
13 if (a[i] == b[now])
14 pos[1][now--] = i;
15 for (int i = 1; i < m; ++i)
16 ans = max(pos[1][i] - pos[0][i - 1], ans);
17 printf("%d", ans);

```

```

18 return 0;
19 }

```

假设  $m \leq n \leq 200000$ , 完成下列各题。

### ■ 判断题

27. 若  $m$  不为  $n$  的子序列, 则输出必定为 0。 ( )
28. 若将第 8 行删除, 程序输出结果一定不会改变。 ( )

### ■ 选择题

29. 若输入为 5 3 abaab abb, 则输出为 ( )。
- A. 1                      B. 2                      C. 3                      D. 4
30. 若  $a = \text{"ababdc"}$ ,  $b$  的长度为 5, 则使答案取到最大值的  $b$  可能有 ( ) 个。
- A. 3                      B. 4                      C. 6                      D. 7
31. (4 分) 当  $a$  为 "1010101" 时,  $b$  的长度为 3,  $b$  的每一位上要么是 0, 要么是 1。总共有 8 种情况, 对应 8 个输出。这 8 个输出的和为 ( )。
- A. 18                      B. 24                      C. 30                      D. 36
32. 当  $a$  为 "1010101" 时,  $b$  的长度为 4,  $b$  的每一位上要么是 0, 要么是 1。总共有 16 种情况, 对应 16 个输出。这 16 个输出的和为 ( )。
- A. 32                      B. 38                      C. 46                      D. 52

### 三、完善程序 (单选题, 每小题 3 分, 共计 30 分)

#### (1) 题目描述:

给定一个数组  $\{a\}$  表示一排蘑菇的数量。有一个篮子。每次到一个新的  $a_i$  时, 篮子中会增加  $a_i$  个蘑菇。如果篮子里的蘑菇超过  $x$  个, 则篮子里的蘑菇会清空。询问有多少组  $[l, r]$ , 使得从  $l$  采摘到  $r$ , 蘑菇数量不为 0。

```

01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 2e5 + 5;
04 int n, x, a[N], cnt[N], dp[N];

```

```
05 int main() {
06 cin >> n >> x;
07 for (int i = 1; i <= n; i++)
08 cin >> a[i];
09 int l = 1, r = 0, sum = 0;
10 while(l <= n) {
11 while (①)
12 ②;
13 cnt[l] = r;
14 ③;
15 }
16 sum = 0;
17 for (int i = n; i >= 1; i--) {
18 if (cnt[i] == n + 1) continue;
19 dp[i] = ④;
20 sum += dp[i];
21 }
22 cout << ⑤ << endl;
23 return 0;
24 }
```

33. ①处应填 ( )。

A.  $r \leq n \&\& \text{sum} \leq x$

B.  $l \leq n \&\& \text{sum} > x$

C.  $\text{sum} \leq x$

D.  $\text{sum} > x$

34. ②处应填 ( )。

A.  $\text{sum} += a[++r]$

B.  $\text{sum} += a[r++]$

C.  $\text{sum} -= a[++l]$

D.  $\text{sum} -= a[l++]$

35. ③处应填 ( )。

A.  $\text{sum} -= a[r--]$

B.  $\text{sum} -= a[--r]$

C.  $\text{sum} -= a[++l]$

D.  $\text{sum} -= a[l++]$

36. ④处应填 ( )。

A.  $\text{dp}[\text{cnt}[i]] + 1$

B.  $\text{dp}[\text{cnt}[i] + 1] + 1$

C.  $\text{dp}[\text{cnt}[i + 1]] + 1$

D.  $\text{dp}[\text{cnt}[i + 1] + 1] + 1$

- (2) 题目描述:

```

01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N = 5005;
04 char s[N];
05 int n, f[N][N], dp[N][N];
06 bool check (int l, int r) {
07 if (①) return f[l][r];
08 if (l >= r) return f[l][r] = 1;
09 if (s[l] ^ s[r]) return f[l][r] = 0;
10 return f[l][r] = ②;
11 }
12 int main() {
13 memset (f, -1, sizeof (f));
14 scanf ("%s", s + 1); n = strlen (s + 1);
15 for (int i = 1; i <= n; ++i) ③ ;
16 for (int l = 2; l <= n; ++l) {
17 for (int i = 1; i <= n - l + 1; ++i) {
18 int j = i + l - 1;
19 dp[i][j] = ④;
20 if (check (i, j)) ⑤;
21 }
22 }
23 int T; scanf ("%d", &T);
24 while (T --) {
25 int x, y; scanf ("%d%d", &x, &y);
26 printf ("%d\n", dp[x][y]);
27 }
28 return 0;
29 }

```

38. ①处应填 ( )。

- A. `~f[l][r]`      B. `!f[l][r]`      C. `r > l`      D. `r - l > 1`

39. ②处应填 ( )。

- A. `check(l + 1, r - 1)`      B. `check(l + 1, r) + 1`  
C. `f[l + 1][r - 1] + 1`      D. `f[l + 1][r] + 1`

40. ③处应填 ( )。

- A. `dp[i][i] = 1`  
B. `dp[i][i+1] = 1`  
C. `dp[i][i+1] = (s[i] == s[i+1]) + 2`  
D. `dp[i][i+1] = s[i] == s[i + 1]`

41. ④处应填 ( )。

- A. `dp[i+1][j] + dp[i][j-1] + dp[i+1][j-1]`  
B. `dp[i+1][j] + dp[i][j-1] - dp[i+1][j-1]`  
C. `dp[i+1][j] + dp[i][j-1] + (s[i] == s[j])`  
D. `dp[i+1][j] + dp[i][j-1] - (s[i] == s[j])`

42. ⑤处应填 ( )。

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