

梦熊联盟·普及组 CSP-J 2024 初赛考前模拟卷

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）

- 以下不能用 ASCII 码表示的字符是（ ）。  
A. @                      B. ①                      C. ^                      D. ~
- 以下描述正确的是（ ）。  
A. 在 CSP-J 中，认证选手可以携带字典和词典  
B. CSP-J 的认证选手可携带已关机的手机放在自己身边的包里  
C. CSP-J 的认证选手在认证时间内上厕所的时候可携带手机  
D. CSP-J 全称是 CCF CSP 非专业级别的能力认证入门级
- 以下哪个名字可以用做 C++ 程序中的变量名？（ ）。  
A. python                  B. new                      C. class                      D. public
- 现在有表示 CSP-J/S 考试日期的整型变量 `testday = 20240921`，下面能得到月份数值 9 的代码是（ ）。  
A. `testday / 1000 % 100`  
B. `testday / 100 % 1000`  
C. `testday / 100 % 100`  
D. `testday % 100 / 100`
- 小明和小杨轮流掷硬币（假设每次掷硬币都符合古典概率模型，每次掷硬币正面朝上和反面朝上的概率都是  $1/2$ ），小明先掷，如果谁先掷到正面算谁赢。请问小明赢的概率是（ ）。  
A.  $2/3$                       B.  $1/2$                       C.  $1/3$                       D.  $3/4$
- 小明用递归的方法写了一个关于汉诺塔的程序。能够描述递归过程的数据结构是（ ）。  
A. 堆                          B. 栈                          C. 链表                      D. 队列
- 堆排序在最坏情况下运行的时间复杂度是（ ）。  
A.  $O(\log n)$                   B.  $O(n)$                       C.  $O(n^2)$                       D.  $O(n \log n)$
- 设变量 `s` 为 `double` 型且已赋值，下列哪条语句能将 `s` 中的数值保留到小数点后一位，并将第二位四舍五入？（ ）。  
A. `s = (s * 10 + 0.5) / 10.0`  
B. `s = s * 10 + 0.5 / 10.0`  
C. `s = (s / 10 + 0.5) * 10.0`  
D. `s = (int)(s * 10 + 0.5) / 10.0`

9. 关于计算机网络，下面的说法哪些是正确的？（ ）
- A. TCP 协议对应数据链路层
  - B. IP 协议对应传输层
  - C. HTTP 协议对应应用层
  - D. 接入互联网的计算机的 IP 地址已经大部分从 IPv5 升级到了 IPv6 地址
10. 以下哪个不是 C++ 里面的循环语句？（ ）
- A. while
  - B. do...while
  - C. for
  - D. switch...case
11. 若整型变量 n 的值为 25，则表达式  $n \& (n+1 \gg 1)$  的值为（ ）。
- A. 25
  - B. 26
  - C. 9
  - D. 16
12. 以下逻辑表达式里，不管 A, B 如何取值，恒为真的是？（ ）
- A.  $(\neg AVB) \wedge (AVB) \wedge AVB$
  - B.  $((\neg AVB) \vee (AV \neg B)) \wedge \neg B$
  - C.  $AV((\neg AVB) \vee (AV \neg B)) \vee \neg A$
  - D.  $((\neg AVB) \vee (AV \neg B)) \vee A \wedge \neg B$
13. 已知一棵完全二叉树共有 2024 个节点，则其叶节点数量为（ ）个。
- A. 1011
  - B. 1012
  - C. 1013
  - D. 1014
14. 将 1, 2, 3, 4, 5, 6, 7 这 7 个数字排成一排，则保证每一对相邻数字都互质的排法共有（ ）种。
- A. 576
  - B. 864
  - C. 720
  - D. 840
15. 假设 G 是一张有 n 个点 m 条边的连通图，小明想将其变成一棵 n 个节点的树，必须删去（ ）条边才能将其变成这样的一棵树？
- A.  $m-n-1$
  - B.  $m+n-1$
  - C. 1
  - D.  $m-n+1$

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  $\checkmark$ ，错误填  $\times$ ；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

1.

```

01 #include<bits/stdc++.h>
02
03 using namespace std;
04
05 unsigned int ksm(unsigned int x, unsigned int y)
06 {
07     unsigned int ans=1;
08     while(y)
09     {

```

```

10         if(y&1) ans=ans*x;
11         y>>=1;
12         x=x*x;
13     }
14     return ans;
15 }
16
17 int main()
18 {
19     int x, y;
20     cin >> x >> y;
21     cout << (int)(ksm(x,y)) << '\n';
22 }

```

保证输入的数据  $0 \leq x, y \leq 100$ , 回答下列问题。

● 判断题

- 1) 时间复杂度是  $O(\log x)$ 。 ( )
- 2) 程序空间复杂度是  $O(1)$ 。 ( )
- 3) (2分) 如果进入 ksm 函数, 则程序第 10 行代码最少执行 1 次。 ( )
- 4) (2分) 如果  $x$  为 0, 则输出恒为 0。 ( )

● 选择题

- 5) 若输入数据为 5 3, 则输出为 ( )。  
A. 1                      B. 5                      C. 25                      D. 125
- 6) 若输入数据为 2 31, 则输出为 ( )。  
A. 1073741824              B. 2147483648              C. -1                      D. -2147483648

2.

```

01 #include<bits/stdc++.h>
02
03 using namespace std;
04
05 int mwm(int x)
06 {
07     int ans=0;
08     if(x==1) return 1;
09     for(int i=2;i*i<=x;i++)
10     {
11         if(x%i==0) ; else continue;
12         ans += mwm(i)+mwm(x/i);
13     }
14     return ans+x+1;
15 }
16

```

```

17 int main()
18 {
19     int x;
20     cin>>x;
21     cout<<mwm(x)<<'\n';
22 }

```

假设数据满足  $1 \leq x \leq 20$ ，回答下列问题。

● 判断题

- 1) (2分) 存在一组输入，使得程序输出为 0。 ( )
- 2) (2分) 存在一组输入，使得程序输出为 1。 ( )
- 3) (2分) 第 11 行是为了排除  $i$  不是  $x$  因数的情况，只有当  $i$  是  $x$  因数时才会执行第 12 行。 ( )

● 选择题

- 4) 若输入 5，则程序输出为 ( )。  
A. 0                      B. 1                      C. 5                      D. 6
- 5) 若输入 18，则程序输出为 ( )。  
A. 53                      B. 54                      C. 58                      D. 59
- 6) 在数据范围内，有多少个  $mwm(x) = x+1$ ? ( )  
A. 7                      B. 8                      C. 9                      D. 10

3.

```

01 #include<cstdio>
02 typedef long long LL;
03 const int M=1e3+5;
04
05 int n;
06
07 LL b[M],ans;
08
09 int main(){
10     scanf("%d",&n);
11     if(((1+n)*n/2)&1)puts("0");
12     else{
13         for(int i=0;i<(1<<(n/2));++i){
14             int cur=0;
15             for(int j=0;(i>>j)>0;++j)if((i>>j)&1)cur+=(j+1);
16             b[cur]++;
17         }
18         for(int i=0;i<(1<<(n-n/2));++i){
19             int cur=0;
20             for(int j=0;(i>>j)>0;++j)
                if((i>>j)&1) cur+=j+n/2+1;

```

```

21         if((1+n)*n/4>=cur)
22             ans+=b[(1+n)*n/4-cur];
23     }
24     printf("%lld\n",ans/2);
25 }
26 return 0;
27 }

```

● 判断题

- 1) (2分) 第 11 行是否输出 0 取决于  $(1+n)*n/2$  的奇偶性。 ( )
- 2) (2分) 在程序运行的任意时刻, b 数组一定全部是偶数。 ( )
- 3) (2分) 此题目时间复杂度为  $O(2^n)$ 。 ( )

● 选择题

- 4) 若输入 7, 那么输出结果是 ( )。
  - A. 1
  - B. 2
  - C. 3
  - D. 4
- 5) 程序输入 5 以内的整数, 则输出的和是 ( )。
  - A. 0
  - B. 1
  - C. 2
  - D. 3

三、完善程序 (单选题, 每小题 3 分, 共计 30 分)

1. 给定一行句子, 每个词之间用空格隔开, 要么是全小写英文单词, 要么是全大写英文单词, 要么是自然数。

要求将这些单词倒序输出。而且对于每个单词, 如果是小写词, 应当转为大写; 如果是大写词, 应当转为小写; 如果是自然数, 应该倒转输出。输入以回车结尾, 该回车不在题目要求的句子中。

```

01 #include<bits/stdc++.h>
02
03 using namespace std;
04
05 int n;
06 string s[1007];
07
08 int main() {
09     while(cin >> s[++n]);
10     ①;
11     for(int i = n; i >= 1; i--) {
12         for(int j = 0; j < (int)s[i].size(); j++) {
13             if(②) s[i][j] -= ③;
14             else if(s[i][j] >= 'A' && s[i][j] <= 'Z')
15                 s[i][j] -= ④;
16         }
17         if(s[i][0] >= '0' && s[i][0] <= '9')

```



```

17         for(⑤) cout << s[i][j];
18     else cout << s[i];
19     if(i != 1) cout << " ";
20 }
21 }

```

1) ①处应填( )

- A. 不填写                      B. n--                      C. n++                      D. s[n]=0

2) ②处应填( )

- A. s[i][j] >= 'z' && s[i][j] <= 'a'  
 B. s[i][j] <= 'a' && s[i][j] >= 'z'  
 C. s[i][j] >= 'a' && s[i][j] <= 'z'  
 D. s[i][j] >= 'z' && s[i][j] >= 'z'

3) ③处应填( )

- A. 'a' - 'A'                      B. 'a' + 'A'                      C. 'A' - 'a'                      D. 'A' + 'a'

4) ④处应填( )

- A. 'a' - 'A'                      B. 'a' + 'A'                      C. 'A' - 'a'                      D. 'A' + 'a'

5) ⑤处应填( )

- A. int j = (int)s[i].size() - 1; j >= 0; j--  
 B. int j = (int)s[i].size(); j >= 0; j--  
 C. j = (int)s[i].size() - 1; j >= 0; j--  
 D. j = (int)s[i].size(); j >= 0; j--

2. 在一个餐馆中，有一台空调，每分钟可以选择上调 1 个单位的温度、下调 1 个单位的温度，或者保持不变。初始温度为  $m$ 。

餐馆有  $n$  位食客，每位食客在时间点  $t_i$  到达，他们能适应的最低温度是  $l_i$ ，最高温度是  $h_i$ 。每位食客只会在  $t_i$  时刻逗留。如果温度不在食客的适应范围内，他们就会感到不舒服。请判断空调能否调整温度，使得所有食客都感到舒服。

提示：按时间排序后，记录当前空调能达到温度的上界和下界，只要判断顾客来的时候温度要求是否符合条件就可以。

```

01 #include <cstdio>
02 #include <algorithm>
03 #define N 105
04
05 struct node {
06     int t, l, r;
07 } a[N];
08

```

```

09 inline bool cmp(const node &x, const node &y) { ① }
10
11 int main() {
12     int T;
13     scanf("%d", &T);
14     while (T--) {
15         int n, m;
16         scanf("%d%d", &n, &m);
17         for(int i=1; i<=n; ++i) scanf("%d%d%d", &a[i].t,
18                                     &a[i].l, &a[i].r);
19
20         std::sort(②);
21         int l = m, r = m;
22         int fail = 0;
23         for (int i = 1; i <= n; ++i) {
24             ③;
25             if (l > a[i].r || r < a[i].l) {
26                 ④;
27                 break;
28             }
29             ⑤;
30         }
31         if (fail) puts("NO");
32         else puts("YES");
33     }
34     return 0;
35 }

```

1) ①处应填( )

- A. return x.l < y.l;                      B. return x.r < y.r;  
C. return x.t < y.t;                      D. return x < y

2) ②处应填( )

- A. a + 1, a + n, cmp                      B. a + 1, a + n + 1, cmp  
C. a, a + n + 1                              D. a, a + n

3) ③处应填( )

- A. l += a[i].t - a[i-1].t + 1, r -= a[i].t - a[i-1].t + 1;  
B. l -= a[i].t - a[i-1].t + 1, r += a[i].t - a[i-1].t + 1;  
C. l += a[i].t - a[i-1].t, r -= a[i].t - a[i-1].t;  
D. l -= a[i].t - a[i-1].t, r += a[i].t - a[i-1].t;

4) ④处应填( )

- A. l=a[i].l    B. r=a[i].r    C. fail=0    D. fail=1

5) ⑤处应填( )

A. `l = std::max(l, a[i].l); r = std::min(r, a[i].r);`

B. `l = std::min(l, a[i].l); r = std::max(r, a[i].r);`

C. `l = std::max(l, l-a[i].l); r = std::min(r, r+a[i].r);`

D. `l = std::min(l, l+a[i].l); r = std::max(r, r-a[i].r);`