

普及组 CSP-J 2024 初赛模拟卷 6

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）

- 2024 的因子与质因子分别有（ ）个。
A. 18 和 3 B. 16 和 3 C. 15 和 3 D. 16 和 4
- 使用邻接矩阵表示 N 个结点的有向图，所需要的存储空间为（ ）。
A. $N \times (N+1)$ B. N^2 C. $N \times (N-1)$ D. $N \times (N-1)/2$
- 在 C++ 程序中，表达式 $a\%b$ 与下列哪个表达式是等价的？（ ）
A. $a=\%b$ B. $a/=b$ C. $a=b\%a$ D. $a=a\%b$
- 线性表若采用链表存储结构，则要求内存中可用存储单元地址（ ）。
A. 必须连续 B. 必须不连续
C. 连续或不连续都行 D. 部分连续
- 我们输入一个新闻网站的网址便可访问该网站，其中用到的网络协议是（ ）。
A. DNS B. FTP C. SSH D. TELNET
- 以下哪个不属于 STL 中栈的操作函数？（ ）
A. empty B. front C. push D. pop
- 平面上任取 n 个整点（横坐标和纵坐标都是整数），其中一定存在两个点，它们的中点也是整点，那么 n 至少是（ ）。
A. 4 B. 5 C. 6 D. 7
- 以下哪个操作属于位运算范畴？（ ）
A. $\&\&$ B. $||$ C. $>>>$ D. $^$
- 关于树这种数据结构，下面的说法中哪个是正确的？（ ）
A. 满二叉树的结点总数一定是奇数

- B. 完全二叉树的结点总数一定是奇数
C. 树形结构只有双亲表示法和孩子表示法
D. 二叉树的遍历方法只有前序遍历法和后序遍历法
10. 以下哪个选项不属于头文件 `cmath`? ()
A. `find(iterator first, iterator last, int x)`
B. `abs(int x)`
C. `ceil(double x)`
D. `pow(double x, double y)`
11. 在 C++ 语言中, 表达式 `564|3` 的值等于 ()。
A. 7 B. 5 C. 4 D. 3
12. 定义变量 `double n`, 如果下面的代码输入为 1000, 则输出最接近 ()。

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    double n;
    cin>>n;
    cout << log10(n)-log2(n)<< endl;
    return 0;
}
```


A. 0 B. -5 C. -7 D. 7
13. 在图的广度优先搜索中, 要维护一个标识数组表示已经访问过的图的结点, 需要 () 数据结构存放结点以实现遍历。
A. 栈 B. 队列 C. 哈希表 D. 堆
14. 从一个 6×6 的棋盘 (不可旋转) 中选取不在同一行也不在同一列的两个方格, 共有 () 种方法。
A. 480 B. 450 C. 360 D. 720
15. 下列关于集合的说法哪个不正确? ()
A. 一个元素是否属于一个集合是确定的

- B. 集合中的元素两两不同
- C. 0 属于空集
- D. 集合中的元素不存在先后次序

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 main()
04 {
05     set <int> s;
06     srand(time(0));
07     int n=6;
08     int l=12;
09     int k,i=0,a[n];
10     while(s.size()<n)
11     {
12         k=rand()%10+1;
13         s.insert(k);
14     }
15     set <int>::iterator it;
16     for(it=s.begin(); it!=s.end(); it++)
17         a[i++] = *it;
18     for(int i=0; i<n; i++)
19     {
20         int x=rand()%n;
21         int y=rand()%n;
22         if(a[x] > a[y])
23             swap(a[x],a[y]);
24     }
25     for(int i=0; i<n; i++)
26         cout<<a[i]<<" ";
27     return 0;
28 }
```

■ 判断题

16. 若去掉第 6 行，程序的输出结果不受任何影响。

()

17. 若去掉第 8 行, 程序的输出结果中的数字总数不变。 ()
18. 程序的输出结果是一个从小到大排列的整数序列。 ()
19. 将第 10 行中的 `s.size()` 替换成 `s.length()`, 程序的运行结果不会改变。 ()

■ 选择题

20. 将第 13 行 `s.insert(k)` 替换成 `for(int i=1;i<=6;i++) s.insert(i)`, 则输出为 ()。
- A. 1 2 3 4 5 6 B. 6 5 4 3 2 1
- C. 1 2 3 6 5 4 D. 1~6 随机分布值
21. 将第 13 行 `s.insert(k)` 替换成 `for(int i=1;i<=7;i++) s.insert(i)`, 第 20 行替换为 `int x=i+1`, 第 21 行替换为 `int y=i`, 则输出为 ()。
- A. 1 2 3 4 5 6 B. 6 5 4 3 2 1
- C. 2 3 4 5 6 7 1 D. 2 3 4 5 6 7

(2)

```

01 #include<bits/stdc++.h>
02 using namespace std;
03 int month[13]={-1,31,28,31,30,31,30,31,31,30,31,30,31};
04 int date,ans1,ans2,y,m,d;
05 bool check1(int date){
06     char s[32];
07     sprintf(s,"%d",date);
08     if(s[0]==s[7] && s[1]==s[6] && s[2]==s[5] && s[3]==s[4])
09         return true;
10     return false;
11 }
12 bool check2(int date){
13     char s[32];
14     sprintf(s,"%d",date);
15     if(s[0]==s[2] && s[0]==s[5] && s[0]==s[7] && s[1]==s[3]
        && s[1]==s[4] && s[1]==s[6])
16         return true;
17     return false;
18 }
19 int main()
20 {
21     cin>>date;

```

```

22     y=date/10000;
23     m=date/100%100;
24     d=date%100;
25     for(int i=y; i++){
26         if(i%400==0 || (i%100!=0 && i%4==0))
27             month[2] = 29;
28         else
29             month[2] = 28;
30     int j=(i==y)?m:1;
31     for( ;j<=12;j++){
32         int k=(i==y && j==m)?d+1:1;
33         for(;k<=month[j];k++){
34             int date=i*10000+j*100+k;
35             if(check1(date) && ans1==0)
36                 ans1=date;
37             if(check2(date))
38                 return cout<<ans1<<" "<<date<<endl,0;
39         }
40     }
41 }
42 return 0;
43 }

```

注：输入为 8 位数字。

■ 判断题

22. 将第 3 行中的 -1 改为 0，程序的运行不受任何影响。 ()
23. 去掉第 15 行中的 && s[1]==s[4] && s[1]==s[6]，程序的输出不变。 ()
24. 将第 23 行 m=date/100%100 替换为 m=date%10000/100，程序的输出不变。 ()
25. 将第 30 行改为 int j; if(i==y && j==1 || i!=y && j==0) j=m，程序的输出不变。 ()

■ 选择题

26. 若输入 20240204，则输出为 ()。
- A. 20240204 20300302 B. 20300302 20400402
- C. 20400402 21211212 D. 20300302 21211212

27. 若输出 20011002 20200202, 则输入可能为 ()。

- A. 20011002 B. 20001001 C. 20020101 D. 20020202

(3)

```
01 #include<bits/stdc++.h>
02 using namespace std;
03 const int MAXN = 100;
04 bool Prime_Number_Judge(const int &num){
05     if (num <= 3)
06         return num > 1;
07     for (int i = 2; i < num; i++)
08         if (num % i == 0)
09             return false;
10     return true;
11 }
12 int& Get_Number_Size(const int &num){
13     int digit = 0, val = num;
14     while (val){
15         val /= 10;
16         digit++;
17     }
18     return digit;
19 }
20 vector<int>&Get_Digits(const int &num, vector<int> &digits){
21     int vactor_val = 0;
22     for(int num_size=Get_Number_Size(num);num_size>0;num_size--){
23         vactor_val = num % (int)pow(10.0, num_size);
24         vactor_val = vactor_val/(int)pow(10.0,num_size-1);
25         digits.push_back(vactor_val);
26     }
27     return digits;
28 }
29 vector<int>&Get_K_Adjacent(const int &num, vector<int>&adjacent){
30     vector<int> digists_number;
31     char tmp[128], buf_tmp[128];
32     Get_Digits(num, digists_number);
33     int digits = Get_Number_Size(num);
34     for (int i = 0; i < digits; i++){
35         for (int j = 0; j < digits-i; j++){
36             string buf;
```



```

37     int k = 0;
38     while (k <= i){
39         sprintf(tmp, "%d", digists_number.at(j+k));
40         buf += tmp;
41         k++;
42     }
43     for(int i=0; i<=buf.size(); i++)
44         buf_tmp[i] = buf[i];
45     adjacent.push_back(atoi(buf_tmp));
46 }
47 }
48 return adjacent;
49 }
50 int main(){
51     int count=0;
52     for (int i = 1; i < MAXN; i++){
53         if (Prime_Number_Judge(i)){
54             vector<int> buf;
55             Get_K_Adjacent(i, buf);
56             int sign = 1;
57             for (int j = 0; j < buf.size(); j++){
58                 if (!Prime_Number_Judge(buf.at(j))){
59                     sign = 0;
60                     break;
61                 }
62             }
63             if (sign)
64                 count++;
65         }
66     }
67     cout<<count;
68     return 0;
69 }

```

■ 判断题

28. 若将第 5 行 `if (num <= 3)` 替换为 `if (num < 3)`, 程序的运行结果不会改变。 ()
29. 若将第 7 行中的 `i < num` 替换为 `i*i <= num`, 程序的运行结果不会改变。 ()
30. 本程序用到的 `vector` 属于 STL。 ()
31. 若将 `MAXN=100` 替换为 `MAXN=70`, 程序的运行结果不会改变。 ()

■ 选择题

32. 运行本程序, 输出结果为 ()。

- A. 7 B. 8 C. 9 D. 10

33. 若将 $\text{MAXN} = 100$ 改为 $\text{MAXN} = 2024$, 程序的输出结果为 ()。

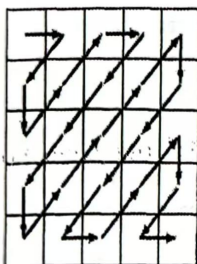
- A. 9 B. 10 C. 11 D. 12

34. (4 分) 若将 $\text{MAXN} = 100$ 改为 $\text{MAXN} = 20244202$, 程序的输出结果为 ()。

- A. 11 B. 10 C. 9 D. 大于 11 的整数

三、完善程序 (单选题, 每小题 3 分, 共计 30 分)

(1) 在图像编码的算法中, 需要对一个给定的方形矩阵进行 Z 字形扫描。给定一个 $n \times n$ 的矩阵, Z 字形扫描的过程如下图所示。



对 4×4 的矩阵

1 5 3 9

3 7 5 6

9 4 6 4

7 3 1 3

进行 Z 字形扫描后得到长度为 16 的序列:

1 5 3 9 7 3 9 5 4 7 3 6 6 4 1 3

请实现一个 Z 字形扫描的程序。给定一个 $n \times n$ 的矩阵, 输出对这个矩阵进行 Z 字形扫描的结果。

输入格式:

输入的第 1 行包含一个整数 n , 表示矩阵的大小。输入的第 2 行到第 $n+1$ 行每行包含 n 个正整数, 由空格分隔, 表示给定的矩阵。

输出格式:

输出一行, 包含 $n \times n$ 个整数, 由空格分隔, 表示输入的矩阵经过 Z 字形扫描后的结果。

输入样例:

```
4
1 5 3 9
3 7 5 6
9 4 6 4
7 3 1 3
```

输出样例:

```
1 5 3 9 7 3 9 5 4 7 3 6 6 4 1 3
```

评测用例规模与约定:

$1 \leq n \leq 500$, 矩阵元素为不超过 1000 的正整数。

```
01 #include<bits/stdc++.h>
02 using namespace std;
03 int main()
04 {
05     int n,t,x,y;
06     t=1, x=1, y=1;
07     int v[505][505], p[505][505];
08     cin >> n;
09     for(int i=1;i<=n;i++)
10         for(int j=1;j<=n;j++)
11             cin>>p[i][j];
12     printf("%d ", ①);
13     ②;
14     while(③)
15     {
16         if(y+1 <= n)
17         {
18             ++t;
19             y++;
20             v[x][y] = 1;
21             printf("%d ",p[x][y]);
22         }
23         while(x+1<=n && y-1>=1 && ④)
24         {
```

```

25      ++t;
26      v[x+1][y-1] = 1;
27      printf("%d ", p[x+1][y-1]);
28      x++;
29      y--;
30  }
31  if(x+1<=n)
32  {
33      ++t;
34      x++;
35      v[x][y] = 1;
36      printf("%d ", p[x][y]);
37  }
38  while(x-1 >= 1 && y+1 <= n && !v[x-1][y+1])
39  {
40      ++t;
41      ⑤;
42      printf("%d ", p[x-1][y+1]);
43      x--;
44      y++;
45  }
46  }
47  return 0;
48  }

```

35. ①处应填 ()。

- A. p[0][0] B. p[0][1] C. p[1][0] D. p[1][1]

36. ②处应填 ()。

- A. v[0][0] = 1 B. v[1][1] = 1 C. v[0][1] = 1 D. v[1][0] = 1

37. ③处应填 ()。

- A. t < n*n B. t <= n*n C. t < n D. t <= n

38. ④处应填 ()。

- A. v[x+1][y-1] B. v[x-1][y+1]
C. !v[x+1][y-1] D. !v[x-1][y+1]

39. ⑤处应填 ()。

A. $v[x-1][y-1] = 1$

B. $v[x+1][y+1] = 1$

C. $v[x+1][y-1] = 1$

D. $v[x-1][y+1] = 1$

(2) 你有一架天平和 N 个砝码, 这 N 个砝码的重量依次是 w_1, w_2, \dots, w_N 。请计算: 一共可以称出多少种不同的重量? 注意砝码可以放在天平两边。

输入格式:

输入的第 1 行包含一个整数 N 。第 2 行包含 N 个整数: w_1, w_2, \dots, w_N 。

输出格式:

输出一个整数代表答案。

输入样例:

3

1 4 6

输出样例:

10

样例说明:

能称出的 10 种重量是 1、2、3、4、5、6、7、9、10、11。

1=1

2=6-4 (天平一边放 6, 另一边放 4)

3=4-1

4=4

5=6-1

6=6

7=1+6

9=4+6-1

10=4+6

11=1+4+6

评测用例规模与约定:

对于 50% 的评测用例, $1 \leq N \leq 15$;

对于所有评测用例, $1 \leq N \leq 100$, N 个砝码的总重量不超过 10^5 。

```
01 #include <bits/stdc++.h>
02 using namespace std;
03 const int maxn = 105, maxv = 1e5 + 5;
```

```
04 int n, a[maxn], f[maxn][maxv], sum, ans;
05 int main()
06 {
07     cin >> n;
08     for(int i = 1; i <= n; i++)
09     {
10         cin >> a[i];
11         ①;
12     }
13     f[0][0] = 1;
14     for(int i = 1; i <= n; i++)
15     {
16         for(int ②; j >= 0; j--)//2
17         {
18             f[i][j] |= f[i - 1][j];
19             f[i][j] |= ③;
20             if( ④ )
21                 f[i][j] |= f[i - 1][j + a[i]];
22         }
23     }
24     for(int i = 1; i <= sum; i++)
25         ⑤;
26     cout << ans;
27     return 0;
28 }
```

40. ①处应填 ()。

- | | |
|----------------|----------------|
| A. sum += a[i] | B. sum += a[1] |
| C. sum += a[n] | D. sum = a[i] |

41. ②处应填 ()。

- | | |
|----------------|--------------|
| A. j = f[i][j] | B. j = sum-1 |
| C. j = n | D. j = sum |

42. ③处应填 ()。

- | | |
|----------------------|------------------------|
| A. f[i-1][j-a[i]] | B. f[i-1][abs(j-a[i])] |
| C. f[i][abs(j-a[i])] | D. f[i-1][a[i]-j] |

43. ④处应填 ()。

A. $j + a[i] < \text{sum}$

B. $j + a[i] \leq \text{sum}$

C. $j + a[i-1] < \text{sum}$

D. $j + a[i-1] \leq \text{sum}$

44. ⑤处应填 ()。

A. $\text{ans} += f[n][i-1]$

B. $\text{ans} += f[n-1][i-1]$

C. $\text{ans} += f[n][i]$

D. $\text{ans} += f[n-1][i]$