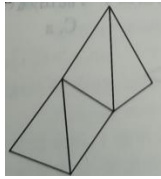


## CSP-J 初赛模拟 2

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. 以下不是属于国家顶级域名的是 ( )。  
A. .au      B. .cn      C. .com      D. .jp
2. 2 个 10 进制数 1111 和 1010 的异或运算结果的进制表示是 ( )。  
A. 101      B. 3      C. 1959      D. 5
3. 8 位二进制数中去掉符号位, 最大能表示多少字符 ( )。  
A. 127      B. 128      C. 255      D. 256
4. 向一个栈顶指针为 `hs` 的链式栈中插入一个指针 `s` 指向的结点时, 应执行 ( )。  
A. `hs->next = s;`  
B. `s->next = hs; hs = s;`  
C. `s->next = hs->next; hs->next = s;`  
D. `s->next = hs; hs = hs->next;`
5. 一棵完全二叉树, 共有 1234 个节点。其叶子节点的个数为 ( )。  
A. 615      B. 616      C. 617      D. 210
6. 某公司派赵钱孙李周五人出国学习, 选派条件是 ( )。  
A. 若赵去, 钱也去      B. 李、周两人必有一人去:  
C. 若周去, 则赵、钱也同去      D. 孙、李二人同去或同不去:  
如何选他们出国?  
A. 孙赵周去      B. 赵钱周去      C. 李周孙去      D. 钱孙去
7. 若  $(a-b)*(c+d)-e$  是中缀表达式, 则其后缀表达式是 ( )。  
A. `abcde-*+-`      B. `ab-*cd+e-`      C. `ab-cd+*e-`      D. `abcde-+*-`
8. 8 颗子弹, 编号为 1, 2, 3, 4, 5, 6, 7, 8, 从编号 1 开始按序嵌入弹夹, 以下不是正常的打出子弹的次序的是 ( )。  
A. 12345678      B. 87654321      C. 32154876      D. 32164587
9. 已知循环队列空间为 30, 队头位置编号为 12, 队尾元素下一个空位置编号为 5, 则队伍中元素个数为 ( )。  
A. 22      B. 23      C. 7      D. 8
10. 甲箱中有 200 个螺杆, 其中有 160 个 A 型螺杆; 乙箱中有 240 个螺母, 其中有 180 个 A 型的。现从甲乙两箱中各任取一个, 则能配成 A 型螺栓的概率为多少? ( )  
A.  $\frac{1}{20}$       B.  $\frac{19}{20}$       C.  $\frac{3}{5}$       D.  $\frac{15}{16}$
11. 今年信息学进复赛的同学有 6 人, 老师将他们排成一圈分发奖品, 一共有 ( ) 种排法。  
A. 60      B. 120      C. 180      D. 240
12. 设二维数组 A 的行下标为 0 至 5, 列下标为 1 至 5, F 的每个数据元素均占 2 个字节。在按行存贮的情况下, 已知数据元素 A[3][3] 的第一个字节是 2019, 则 A[4][4] 的第一个字节的地址为 ( )。  
A. 2029      B. 2025      C. 2027      D. 2031

13. 在下图中, 有( )个顶点出发存在一条路径可以遍历图中的每条边, 而且仅通历一次。



- A. 6                      B. 2                      C. 3                      D. 4

14. 有 A, B, C, D, E, F 六个绝顶聪明型又势均力敌的盗墓贼。他们都排着队, 他们每个人都想独吞财宝, 最前面的 A 如果拿了财宝, 那么体力下降, 则其后面的 B 会杀掉 A, 拿个财宝, 当然 B 拿了财宝, 体力也会下降, 一样会被 C 杀掉, 如果 B 不拿财宝, 则 C 无法杀 B, 请问 A、C、E 的最终想法是()。

- A. A 不拿 C 不拿 E 拿                      B. A 拿 C 拿 E 不拿  
C. A 不拿 C 不拿 E 不拿                      D. A 不拿 C 拿 E 拿

15. 一个有  $n$  个顶点的无向图最多有( )条边。

- A.  $n$                       B.  $n(n-1)$                       C.  $n(n-1)/2$                       D.  $2n$

二、阅读程序(程序输入不超过数组或字符串定义的范围:判断题正确的填√, 错误的填X; 除特殊说明外, 判断题 1.5 分, 选择题 3 分, 共计 40 分)

```
1
1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4 int a[1001], i, j, t, n;
5 for(i=0; i<= 1000; i++)
6 a[i]=0;
7 scanf("%d", &n);
8 for (i=1; i<=n; i++) {
9 scanf("%d", &t);
10 a[t]++;
11 }
12 for(i=1000; i>=0; i--)
13 for(j=1; j<=a[i]; j++)
14 printf("%d", i);
15 return 0;
16 }
```

●判断题

- (1)输入 10 个数字, 输出结果是从小到大。( )  
(2)若属入的数字中有两个 1, 则输出时出来第一个 1 是第一个输入的。( )  
(3)若将第 13 行的 “<=” 改为 “<”, 且输入数据为 10 2 12 33 34 2 87 22 9 0, 则输出 2。  
( )  
(4)若将第 12 行改为 for(i=0; i<=1000; i++), 则程序运行结果不变。( )。

●选择题

- (5)若将第 12 行改为 for(i=1000; i>1; i--); 第 13 行改为 for(j=a[i]; j>1; j--), 输入数据为 5 2 12 33 34 44, 则运行结果( )。  
A. 不变                      B. 则输出 2 12 33 34 44

C. 无输出

D. 则输出 44 34 33 12 2

(6) (4分) 若将第 10 行改为 `++a[t]` 或 `a[t++]`, 则输入 5 1 2 3 4 5, 输出结果为( )。

A. 1 2 3 4 5 或 5 4 3 2 1

B. 1 2 3 4 5 或无输出

C. 5 4 3 2 1 或 5 4 3 2 1

D. 5 4 3 2 1 或无输出

2.

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     const int SIZE = 100;
6     int height[SIZE], num[SIZE], n, ans;
7     cin>>n;
8     for (int i = 0; i < n; i++)
9     {
10        cin>>height[i]; num[i] = 1;
11        for (int j = 0; j < i; j++)
12        {
13            if ((height[j] < height[i]) && (num[j] >= num[i]))
14                num[i] = num[j]+1;
15        }
16    }
17    ans = 0;
18    for (int i = 0; i < n; i++)
19    {
20        if (num[i] > ans) ans = num[i];
21    }
22    cout<<ans<<endl;
23 }
```

(1) 将第 4 行的程序移动到第 2、3 行中间, 程序能正常进行。( )

(2) 第 6 行输入  $n=5$ , 则输出 `ans` 的值一定小于 5。( )

(3) 可以通过优化题目中的时间复杂度降为  $O(n\log n)$ 。( )

(4) 如果输出是 1, 则 `height` 数组中的数一定是递减的。( )

(5) 当  $n=6$  时, 输入 `height` 数组为 2 5 3 11 12 4, 输出为( )

A. 4

B. 2

C. 14

D. 6

(6) 如果将第 13 行的 `height[j]<height[i]` 改成 `height[j]>height[i]`, 则第 (5) 题的输出结果为( )

A. 4

B. 2

C. 14

D. 6

3.

```
1 #include<bits/stdc++.h>
using namespace std;
const int maxn=500000,INF=0x3f3f3f3f;
int L[maxn/2+2],R[maxn/2+2];
5 void unknown(int a[],int n,int left,int mid,int right) {
6     int n1=mid-left,n2=right-mid;
7     for(int i=0;i<n1;i++)
8         L[i]=a[left+i];
9     for(int i=0;i<n2;i++)
10        R[i]=a[mid+i];
11    L[n1]=R[n2]=INF;
12    int i=0,j=0;
13    for(int k=left;k<right;k++) {
14        if(L[i]<=R[j])
15            a[k]=L[i++];
16        else
17            a[k]=R[j++];
18    }
19 }
20 void unknownsort(int a[],int n,int left,int right){
21     if(left+1<right) {
22         int mid=(left+right)/2;
23         unknownsort(a,n,left,mid);
24         unknownsort(a,n,mid,right);
25         unknown(a,n,left,mid,right);
26     }
27 }
28 int main(){
29     int a[maxn],n;
30     cin>>n;
31     for(int i=0;i<n;i++) cin>>a[i];
32     unknownsort(a,n,0,n);
33     for(int i=0;i<n;i++) {
34         if(i) cout<<" ";
35         cout<<a[i];
36     }
37     cout<<endl;
38     return 0;
39 }
```

●判断题

- (1)将第 13 行的 “<” 改为 “<=” 不会改变运行结果。( )
- (2)将第 21 行的 “<” 改为 “<=” 不会改变运行结果。( )

- (3) 此类排序方法是高效的，但是不稳定。( )  
 (4) 将第 4 行的 2 个 “+2” 都去掉不会改变运行结果。( )

● 选择题

- (5) 本题的时间复杂的为 ( )  
 A.  $O(n)$       B.  $O(n^2)$       C.  $O(n \cdot \log n)$       D.  $O(n \cdot \sqrt{n})$   
 (6) 此题用到了( )思想。  
 A. 动态规划      B. 分治      C. 冒泡      D. 贪心

三、完善程序(单选程序, 每题 3 分, 共计 30 分)

1. 田忌赛马, 田忌每赢一次齐王的马就得 200 金币, 当然输了就扣 200 金币, 平局则金币数不变。

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int main() {
4  int n;
5  while(cin>>n&&n!=0) {
6      int tj[1001], king[1001], count=0;
7      int tj_min=0, tj_max=n-1;
8      int king_min=0, king_max=n-1;
9      for(int i=0; i<n; i++) cin>>tj[i];
10     for(int i=0; i<n; i++) cin>>king[i];
11     sort(tj, tj+n);
12     sort(king, king+n);
13     while(n--){
14         if(tj[___①___]>king[___②___]){
15             count++;
16             tj_max--;
17             king_max--;
18         }
19         else if(tj[___③___]< king[___④___]){
20             count--;
21             tj_min++;
22             king_max--;
23         }
24         else
25         {
26             if(tj[tj_min]>king[king_min]){
27                 count++;
28                 ___⑤___;
29                 ___⑥___;
30             }
31             else {
32                 if(___⑦___)
33                     count--;
34                 tj_min++;
35                 ___⑧___;

```

```

36         }
37     }
38 }
39     cout<< count * 200<< endl;
40 }
41 return 0;
42 }

```

(1)①处和②处填( )。

- A. tj\_max 和 king\_max            B. tj\_min 和 king\_max  
 C. tj\_min 和 king\_max            D. tj\_max 和 king\_min

(2)③处和④处填( )。

- A. tj\_min 和 king\_max            B. tj\_min 和 king\_min  
 C. tj\_max 和 king\_max            D. tj\_max 和 king\_min

(3)⑤和⑥处填( )。

- A. tj\_min--和 king\_min++        B. tj\_max++和 king\_min++  
 C. tj\_min++和 king\_min++        D. tj\_max++和 king\_min--

4 ⑦处填( )。

- A. tj[tj\_min]<king[king\_max]    B. tj[tj\_min] > king[king\_max]  
 C. tj[tj\_max]<king[king\_max]    D. tj[tj\_min] > king[king\_min]

(5)⑧处填()

- A. king\_max--    B. king\_maxt++  
 C. king\_min--    D. king\_min++

2. 寻路问题:N\*N矩阵, 其中0表示可以走的, 1表示无法走, 矩阵有二维数组表示, 左上角是入口, 右下角是出口, 只能横着走和竖着走, 要求找出最短路径。

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int mymax= 10000;
4 int f[4][2]= {{-1,0},{1,0},{0,-1},{0,1}}:
5 int a[20][20],v[20][20],v1[20][20];
6 int len=1;
7 int n;//矩阵的规模
8 bool check(int x1,int y1) {
9     if(x1<0||x1>=n||__①__) return false;
10    if(a[x1][y1]==1||__②__) return false;
11    return true;
12 }
13 void dfs(int x,int y) {
14    if(x==n-1&&y==n-1){

```

```

15     if(len<mymax) {
16         mymax= len;
17         memcpy(v1,v, sizeof(v1));
18     }
19     return;
20 }
21 for(int k=0; k<4; k++){
22     int x1,y1;
23     x1=x+__③__;
24     y1=y+__④__;
25     if(check(x1,y1)) {
26         __⑤__;
27         __⑥__;
28         dfs(x1,y1);
29         __⑦__;
30         v[x1][y1]=0;
31     }
32 }
33 }
34 int main () {
35     cin>> n;
36     for (int i=0;i<n;i++){
37         for(int j=0;j<n;j++)
38             cin>> a[i][j];
39     }
40     dfs (0,0) ;
41     int d=v1[n-1][n-1];
42     int x=n-1, y=n-1;
43     int k;
44     int qn[400][2];
45     qn[0][0]=n-1;
46     qn[0][1]=n-1;
47     for(k=1; ;k++){
48         x=x-f[d][0];

```

```

49     y=y-f[d][1];
50     qn[k][0]=x;
51     qn[k][1]=y;
52     d=v1[x][y];
53     if(x==0&&y==0)break;
54 }
55 for(int i=k;i>=0;i--)
56     cout<<__⑧__<<"."<<__⑨__<<endl;
57 return 0;
58 }

```

(1) ①处和②处填( )。

- |  |  |
|--|--|
| A. $y1 \leq 0 \mid  y1  > n$ 和 $v[x1][y1] > 0$ | B. $y1 < 0 \mid  y1  \geq n$ 和 $v[x1][y1] > 0$ |
| C. $y1 > 0 \&\&y1 < n$ 和 $v[x1][y1] > 0$       | D. $y1 > 0 \&\&y1 < n$ 和 $v[x1][y1] > 0$       |

(2) ③处和④处填( )。

- |                          |                          |
|--------------------------|--------------------------|
| A. $f[k][0]$ 和 $f[k][1]$ | B. $f[k][1]$ 和 $f[k][0]$ |
| C. $f[0][k]$ 和 $f[1][k]$ | D. $f[1][k]$ 和 $f[0][k]$ |

(3) ⑤处填( )。

- |                         |                       |
|-------------------------|-----------------------|
| A. $v[x1][y1] = k + 1;$ | B. $v[x1][y1] = k;$   |
| C. $v[x][y] = k;$       | D. $v[x][y] = k + 1;$ |

(4) ⑥处和⑦处填( )

- |                      |                    |
|----------------------|--------------------|
| A. $len++$ 和 $len--$ | B. $k++$ 和 $k--$   |
| C. $x1++$ 和 $x1--$   | D. $y1++$ 和 $y1--$ |

(5) ⑧处和⑨处填( )

- |                            |                            |
|----------------------------|----------------------------|
| A. $qn[i][1]$ 和 $qn[i][2]$ | B. $qn[i][0]$ 和 $qn[i][1]$ |
| C. $qn[1][i]$ 和 $qn[2][i]$ | D. $qn[0][i]$ 和 $qn[1][i]$ |