

几门级 CSP-J 第 6 套初赛模拟试题

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)

- 在计算机内部用来传送、存储、加工处理的数据或指令都是以()形式进行的。
A. 二进制码 B. 八进制码 C. 十进制码 D. 智能拼音码
- 计算机软件系统通常分为()。
A. 硬件系统和软件系统 B. 高级软件和一般软件
C. 系统软件和应用软件 D. 军用软件和民用软件
- 关于软盘读写孔,正确的说法是()。
A. 从该孔读信息 B. 从该孔写信息
C. 当该孔处于开状态时,不能删除盘中文件 D. 该孔没有作用
- 一棵二叉树的中序遍历为 DGBAECHF,后序遍历为 GDBEHFCA,则前序遍历是()。
A. ABCDFGHE B. ABDGCEFH C. ACBGDHEF D. ACEFHBGD
- 下列叙述中错误的是()。
A. 微机应避免置于强磁场之中
B. 微机使用时间不宜过长,而应隔几个小时关机一次
C. 微机应避免频繁关开,以延长其使用寿命
D. 微机应经常使用,不宜长期闲置不用
- 计算机网络最主要的优点是()。
A. 运算速度快 B. 共享资源 C. 精度高 D. 存储容量大
- 下列 4 个不同进制表示的数中,最大的一个数是()。
A. $(220.1)_{10}$ B. $(11011011.1)_2$ C. $(334.1)_8$ D. $(DC.1)_{16}$
- 为了区分汉字与 ASCII 码,计算机中汉字编码的最高位为()。
A. 1 B. 0 C. -1 D. 2
- 一般来说,TCP/IP 的 IP 提供的服务是()。
A. 运输层服务 B. 会话层服务 C. 表示层服务 D. 网络层服务
- 通信时,模拟信号也可以用数字信道来传输,能实现模拟信号与数字信号之间转换功能的是()。
A. D/A B. A/D C. Modem D. Codec
- 一个栈的输入顺序为 1、2、3、4、5,下列序列中可能是栈的输出序列是()。
A. 54312 B. 24135 C. 21543 D. 12534
- 不属于 Internet 的功能是()。
A. 聊天 B. 远程教育 C. 查询资料 D. 传送能量
- 下列描述计算机病毒的特性中,()不是正确的。
A. 潜伏性 B. 传染性 C. 高速性 D. 危害性
- 下面不属于网络操作系统的是()。
A. UNIX B. WINDOWS NT C. NETWARE D. DOS
- 根据《计算机软件保护条例》,中国公民开发的软件享有著作权的是()。
A. 不论何人、何地及发表与否的软件 B. 只有公开发表的软件
C. 只有单位发表的软件 D. 只有个人在国内公开发表的软件

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 1.5 分,选择题 4 分,共计 40 分)

1.

```
01 #include<bits/stdc++.h>
02 using namespace std;
```

```

03 const int Maxn=10005;
04 int n,b[Maxn];
05 inline void mergesort(int*a,int l,int r) {
06     if (l==r) return;
07     int mid=l+r>>1;
08     mergesort(a,l,mid),mergesort(a,mid+1,r);
09     int i=l,j=mid+1,cnt=0;
10     while (i<=mid && j<=r) {
11         if (a[i]<=a[j]) b[++cnt]=a[i++];
12         else b[++cnt]=a[j++];
13     }
14     while (i<=mid) b[++cnt]=a[i++];
15     while (j<=r) b[++cnt]=a[j++];
16     for (i=l;i<=r;i++) a[i]=b[i-l+1];
17 }
18 int a[Maxn];
19 int main(void) {
20     cin>>n;
21     for (int i=1;i<=n;i++) cin>>a[i];
22     mergesort(a,1,n);
23     for (int i=1;i<=n;i++) cout<<a[i]<<(i==n?'\\n':' ');
24     return 0;
25 }

```

●判断题

- (1)该算法中“int * a”没有传值。()
- (2)该算法会换行。()
- (3)该算法中 mergesort 函数时间复杂度为 $O(n \log n)$ 。()
- (4)如果输入为“5 4 3 9 7 8”则输出为 3 4 7 8 9 \\n。()

●选择题

- (5)下面哪句与“i==n ? '\\n' : ' '”相同()。
- A. i != 1 ? '\\n' : ' '
- B. "\\n"[i==n]
- C. "\\n"[i!=n]
- D. ' '
- (6)该算法的最劣复杂度与哪个排序算法相同()。
- A. 快速排序 B. 选择排序 C. 计数排序 D. 堆排序

2.

```

01 #include<bits/stdc++.h>
02 using namespace std;
03 int i,j,k,n,m,f[10010],p1,p2,p3;
04 int find(int k){
05     if (f[k]==k) return k;
06     return f[k]=find(f[k]);
07 }
08 int main()
09 {
10     cin>>n>>m;
11     for(i=1;i<=n;i++) f[i]=i;

```



```

12     for(i=1;i<=m;i++){
13         cin>>p1>>p2>>p3;
14         if(p1==1)
15             f[find(p2)]=find(p3);
16         if(p1==2)
17             if(find(p2)==find(p3))
18                 printf("Y\n");
19             else
20                 printf("N\n");
21     }
22     return 0;
23 }

```

●判断题

- (1)该算法中 p1 的作用是确定操作类型。()
- (2)去掉 for(i=1;i<=n;i++) f[i]=i;对该算法没有影响。()
- (3)输入 2 2 1 1 2 2 1 2 输出为 Y。()
- (4)输入 2 1 2 1 2 输出为 N。()

●选择题

- (5)该算法时间复杂度为()。
- A. $O(m \log n)$ B. $O(nm)$ C. $O(n+m)$ D. $O(nm^2)$
- (6)把 return f[k]=find(f[k]);改成 return find(f[k]);最差时间复杂度为()。
- A. $O(m \log n)$ B. $O(nm)$ C. $O(n+m)$ D. $O(nm^2)$

3.

```

01 int  t,x[100],a[100];
02 void work(int d,int i,int n){
03     int k;
04     if(n==1)
05     {
06         for(k=0;k<d;k++)
07             printf("%3d",a[k]);
08             printf("\n");
09     }
10     else
11         for(k=i;k<t;k++)
12             if(n% x[k]==0)
13             {
14                 a[d]=x[k];
15                 work(d+1,k,n/x[k]);
16             }
17 }
18 int  main(){
19     int i,k,n;
20     cin>>n;
21     for(i=n;i>1;i--) if(n%i==0) x[t++]=i;
22     work(0,0,n);
23 }

```

● 判断题

- (1) for ($i=n; i>1; i--$) if ($n\%i==0$) $x[t++] = i$; 的作用是求出 n 的所有因数。()
- (2) 该程序的作用是对 n 进行质因数分解。()
- (3) printf("%3d", $a[k]$); 中 3 对程序没有影响。()
- (4) 去掉 if($n\%x[k]==0$) 对程序有影响。()

● 选择题

- (5) 如果输入为 2, 那么输出为()。
- A. 2 B. 2 1 C. 1 2 D. 2 2
- (6) 如果输入为 72, 那么输出的非回车字符有多少行()。
- A. 14 B. 15 C. 16 D. 17

三、完善程序(单选题, 每小题 3 分, 共计 30 分)

1. (坐标统计) 输入 n 个整点在平面上的坐标。对于每个点, 可以控制所有位于它左下方的点(即 x, y 坐标都比它小), 它可以控制的点的数目称为“战斗力”。依次输出每个点的战斗力, 最后输出战斗力最高的点的编号(如果若干个点的战斗力并列最高, 输出其中最大的编号)

```

01 #include<iostream>
02 using namespace std;
03 const int SIZE=100;
04 int x[SIZE], y[SIZE], f[SIZE];
05 int n, i, j, max_f, ans;
06 int main()
07 {
08     cin>>n;
09     for(i=1; i<=n; i++) cin>>x[i]>>y[i];
10     max_f=0;
11     for(i=1; i<=n; i++)
12     {
13         f[i]=[ ① ];
14         for(j=1; j<=n; j++)
15         {
16             if(x[j]<x[i] && [ ② ])
17                 [ ③ ];
18         }
19         if([ ④ ])
20         {
21             max_f=f[i];
22             [ ⑤ ];
23         }
24     }
25     for(i=1; i<=n; i++) cout<<f[i]<<endl;
26     cout<<ans<<endl;
27     return 0;
28 }

```

- (1) ①处应该填()。

A. 0
C. i

B. 1

D. INT_MAX

(2) ②处应该填()。

- A. $y[j] \leq y[i]$ B. $y[j] < y[i]$ C. $y[j] > y[i]$ D. $y[j] \geq y[i]$

(3) ③处应该填()。

- A. $ans = i$ B. $f[i]++$ C. $f[i]--$ D. $ans++$

(4) ④处应该填()。

- A. $(f[i] > \max_f)$ B. $(f[i] \geq \max_f)$ C. $(f[i] < \max_f)$ D. $(f[i] \leq \max_f)$

(5) ⑤处应该填()。

- A. $ans++$ B. $ans += \max_f$ C. $ans = i$ D. $ans = \max_f$

2. (排列数) 输入两个正整数 n, m ($1 < n < 20, 1 < m < n$), 在 $1 \sim n$ 中任取 m 个数, 按字典序从小到大输出所有这样的排列。

例如:

输入:

3 2

输出:

1 2

1 3

2 1

2 3

3 1

3 2

```
01 #include<iostream>
02 #include<cstring>
03 using namespace std;
04 const int SIZE=25;
05 bool used[SIZE];
06 int data[SIZE];
07 int n,m,i,j,k;
08 bool flag;
09 int main()
10 {
11     cin>>n>>m;
12     memset(used,false,sizeof(used));
13     for(i=1;i<=m;i++)
14     {
15         data[i]=i;
16         used[i]=true;
17     }
18     flag=true;
19     while(flag)
20     {
21         for(i=1;i<=m-1;i++)
22             cout<<data[i]<<" ";
23         cout<<data[m]<<endl;
24         flag=[ ① ];
25         for(i=m;i>=1;i--)
26         {
27             [ ② ];
```



```

28     for(j=data[i]+1;j<=n;j++)
29         if(!used[j])
30         {
31             used[j]=true;
32             data[i]=[ ③ ];
33             flag=true;
34             break;
35         }
36     if(flag)
37     {
38         for(k=i+1;k<=m;k++)
39             for(j=1;j<=[ ④ ];j++)
40                 if(!used[j])
41                 {
42                     data[k]=j;
43                     used[j]=true;
44                     break;
45                 }
46         [ ⑤ ];
47     }
48 }
49 }
50 return 0;
51 }

```

(1) ①处应该填()。

- A. false B. true C. 1 D. -1

(2) ②处应该填()。

- A. used[i]=true B. data[i]=i
C. used[data[i]]=true D. used[data[i]]=false

(3) ③处应该填()。

- A. j B. i C. true D. false

(4) ④处应该填()。

- A. n B. m C. i D. j

(5) ⑤处应该填()。

- A. return 0 B. exit C. continue D. break