

全国入门组 CSP-J 初赛模拟试题 (3)

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. 在网络上, 若某台电脑的设备及数据可由其他电脑共享, 这台电脑称为()。

- A.个人计算机 B.副机 C.服务器 D.主机

2. 下列抢占不同数制的数中, 最大的一个数是()。

- A.十进制数 220.1 B.二进制数 11011011 C.八进制数 334.1 D.十六进制数 DC.1

3. 字母在计算机中是以编码形式表示的, 通用的编码是 ASCII 码, 字母 “A” 的 ASCII 码 65, 字母 “E” 的 ASCII 码是 ()。

- A.05 B.52 C.69 D.68

4. 连接 internet 上的每台计算机必须有 1 个 () 地址, 该地址共含()字节。前面若干个子节表示(); 后面若干个子节表示()。为了避免使用数字, 人们经常用字母替代, 这些名字称为()。

- A.IP、四、哈地址、IP 地址、网
B.网络、四、IP 地址、网内计算机地址、域名
C.网络、不超过十、网页、网址、网名
D.IP、四、网络地址、网内计算机地址、域名

5. 在 TCP/IP 协议中, TCP 和 IP 分别提供什么服务()。

- A.传输层、网络层 B.链路层、网络层 C.传输层、会话层 D.物理层、链路层

6. 一棵树 T 有 2 个度数为 2 的结点、有 1 个度数为 3 的结点、有 3 个度数为 4 的结点, 那么树 T 有()个树叶。

- A.14 B.6 C.18 D.7

7. 在一个图中, 所有顶点的度数之和等于所有边数的()倍。

- A.1/2 B.1 C.2 D.4

8. 下列 IP 地址中止确的是()。

- A.202.300.12.4 B.192.168.0.3 C.100:128:35:91 D.111-102-35-21

9. 设有 100 个顶点, 利用二分法查找时, 最大比较次数是()。

- A.50 B.10 C.25 D.7

10. 一棵二叉树的中序遍历序列为:DGBAECHF, 后序遍历序列为:GDBEHFCA, 则前序遍历的序列是()。

- A.ABCDFGHE B.ABDGCEF C.ACBDGHEF D.ACEFHBGD

11. 下面合法的电子邮件地址是()。

- A.http://ybw.online.sh.cn B.ftp://ybw.onlinc.sh.cn
C.ybw@online.sh.cn D.http.ybw.online.sh.cn

12. Internet 给我们提供了资源共享、浏览、检索信息和远程登录等多种服务, 下面几个选项中用于远程登录的是()。

- A.Telnet B.E_mail C.TCP/IP D.WWW

13. 在数据结构中, 链表是()。

- A.顺序存储的线性表结构 B.非顺序存储的线性表结构
C.非顺序存储的非线性表结构 D.顺序存储的非线性表结构

14. C++ 程序运行时, 是在哪个存储器上进行的?()。

- A.硬盘 B.RAM C.ROM D.CACHE

15. 当 $A \geq B \ \&\& \ B \geq C$ 的取值为真时, 表达式 $A > C \ || \ B == C$ 的值()。

- A.为真 B.无法判定结果的真假
C.也有可能为假 D.只有当 A、B、C 都相等时才为真

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

1、

```
1  #include<iostream>
2  #include<cstdio>
3  using namespace std;
4  int i,j,n;
5  int x[101],y[101];
6  int main()
7  {
8      cin>>n;
9      for(i=1; i<=n; i++) cin>>x[i];
10     for(i=1; i<=n-1; i++)
11         for (j=i+1; j<=n; j++)
12             if(x[i]>x[j])
13                 y[j]++;
14             else if (x[i]<x[j])
15                 y[i]++;
16     for(i=1; i<=n; i++)
17         printf("%5d ",y[i]);
18     cout<<endl;
19     return 0;
20 }
```

判断题

- 1) (1 分)把第 13 行与第 15 行互换位置，结果不会改变。()
- 2) (1 分)第 14 行把 `if(x[i]<x[j])` 删掉效果一样。()
- 3) 第 11 行把 `i+1` 改成 `1`，数组 `y` 每个元素的值增加 1 倍。()
- 4) 数组 `y[i]`中存的是 `x[i]`在数列中从大到小的次序。()

选择题

- 5) 此程序的时间复杂度是()
A. $O(\log n)$ B. $O(n \log n)$ C. $O(n^2)$ D. $O(n)$
- 6) 此程序如果 `n` 输入 4，然后输入 2 4 1 3，输出结果是()。
A.1 2 3 4 B.2 0 3 1 C.4 3 2 1 D.1 3 0 2

2、

```
1  #include<iostream>
2  #include<cstdio>
3  using namespace std;
4  int n,i, x;
5  int d[11];
6  int main()
7  {
8      cin>>n;
9      for(i=1; i<=n; i++)
```

```

10     {
11         cin>>x;
12         d[x]=d[x]+1;
13     }
14     d[0]=0;
15     for(i=1; i<=10; i++)
16         d[i]=d[i-1]+d[i];
17     for(i=1; i<=10; i++)
18         if(d[i]!=d[i-1])
19             printf("%3d:%4d ",i,d[i-1]+1);
20     return 0;
21 }

```

判断题

- 1) (1分)输入的 x 值应在[0, n-1]的范围内。()
- 2) (1分)把第 12 行改成 “d[x]++”，不影响程序运行结果。()
- 3) 对任意在[1,9]之间的数 i，都有 $d[i-1]<d[i]<d[i+1]$ 。()
- 4) 把第 18 行改成 “if(d[i]> d[i-1])”，程序运行的结果会发生改变。()

选择题

- 5) 输入 n=10 后，接着输入:1 3 4 5 2 4 3 5 2 3 后，程序输出结果为()。
 A.1:1 2:2 3:3 4:2 5:3 B.1:1 2:2 3:4 4:7 5:9
 C.1:0 2:1 3:2 4:1 5:2 D.1:1 2:2 3:3 4:4 5:5
- 6) 把程序第 19 行改成 “print(“%3d” ,i);”后，输入 n=10 后，接首输入: 1 3 4 1 5 4 4 3 5 1 后，程序输出结果为()。
 A.1 2 3 4 5 B.3 2 3 2 C.4 3 4 3 D.1 3 4 5

3、

```

1  #include<iostream>
2  #include<cstdio>
3  using namespace std;
4  int j,i,m;
5  int a[10];
6  int main()
7  {
8      for(i=2; i<=6; i++)
9          a[i]=i+1;
10     do
11     {
12         m=2;
13         for (i=3; i<=6; i++)
14             if(a[m]>a[i]) m=i;
15         a[m]=a[m]+m;
16         m=1;
17         for(i=2; i<=5; i++)
18             for(j=i+1; j<=6; j++)

```

```

19             if (a[i]<a[j]) m=0;
20
21         }while(m==0);
22         printf("%d",a[2]);
23         return 0;
24     }

```

判断题

- 1) 程序结束时，a[2]的值一定是数组 a 中的最大值。()
- 2) 第 21 行 “m==0” 成立时，数组 a[i] ($2 \leq i \leq 6$) 从大到小排序:()
- 3) 程序输出时，a 数组满足:对任意的 $2 \leq i < 6$ ，有 $a[i] > a[i+1]$ 。()
- 4) 删除第 16 行代码 “m=1”程序结果会发生改变。()

选择题

- 5) 程序的输出结果为()
A.58 B.59 C.61 D.60
- 6) 此程序的时间复杂度是()。
A. $O(n^3)$ B. $O(n \log n)$ C. $O(n^2)$ D. $O(n)$

三、完善程序（单选题，每题 3 分，共计 30 分）

1、菲波拉契数列为 1,1,2,3,5,8,13,21,...其元素产生的规则是前两个数为 1，从第三个数开始每个数等于它前面两个数之和。已知任意一个正整数可以表示为若干个互不相同的菲波拉契数之和。例如: $36=21+13+2$ 。

下面的程序是由键盘输入一个正整数 n，输出组成 n 的互不相同的菲波拉契数。
算法说明:(1)寻找小于等于 n 的最大菲波拉契数 a，并以 a 作为组成 n 的一个数。
(2)若 $n \neq a$ ，则以 $n-a$ 作为 n 的新值，重复步骤(1)。若 $a=n$ ，则结束。

```

#include<iostream>
#include<cstdio>
using namespace std;
int n;
bool first;
int find(int n) {
    int a,b,c;
    a=1;
    b=1;
    do {
        c=a+b;
        ___(1)___;
    } while(b<n);
    if(___ (2) ___)
        return b;
    else
        ___(3)___;
}
void p(int n) {
    int a;

```

```

    a=find(n);
    if(first) {
        printf("%4d",a);
        first=false;
    } else
        ___(4)___;
    if(a<n) ___(5)___;
}
int main() {
    cin>>n;
    first=true;
    printf("%5d = ",n);
    p(n);
    cout<<endl;
    return 0;
}

```

选择题

1) ①处应填 ()

A.a=c;b=a B.a=b;b=c C.a==c;b==a D.n==b;b==c

2) ②处应填 ()

A.b==n B.b<n C.a==n D.a<n

3) ③处应填 ()

A.return c B.return b C.return a+b D.return a

4) ④处应填 ()

A.printf(" %4d" ,a) B.printf("+ %4d" ,a) C.printf(" %4d",b) D.printf(" + %4d",b)

5) ⑤处应填 ()

A.p(a) B.p(b) C.p(n-a) D.p(n-b)

2、现在政府计划在某个区域的城市之间建立高速公路，以使得其中任意两个城市之间都有直接或间接的高速公路相连。费用为每千米为一个单位价格，求最小费用。

输入:n($n \leq 100$ ，表示城市数目)。

接下来 n 行，每行两个数 $x_i.y_i$ 表示第个城市的坐标，(单位:千米)。

输出:最小费用(保留 2 位小数)。

程序如下:

```

#include<iostream>
#include<cstdio>
using namespace std;
const int maxn=101;
struct tcity {
    float x,y;
};
tcity c[maxn];
float d[maxn][maxn],a,minf;
int p[maxn],n,i,j,k;

```

```

int main() {
    cin>>n;
    for(i=1; i<=n; i++)
        cin>>c[i].x>>c[i].y;
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
            d[i][j]=__(1)__;
    p[1]=0;
    for(i=2; i<=n; i++) __(2)__;
    for(i=1; i<=n-1; i++) {
        minf=1E10;
        for(j=1; j<=n; j++) {
            if(__(3)__) {
                minf = d[p[j]][j];
                __(4)__;
            }
        }
        a=a+d[p[k]][k];
        p[k]=0;
        for(j=1; j<=n; j++)
            if (__(5)__) p[j]=k;
    }
    printf("%0.2F",a);
    return 0;
}

```

选择题

1) ①处应填()

- A. $\sqrt{(c[i].x-c[i].y)*(c[i].x-c[i].y)+(c[j].x-c[j].y)*(c[i].x-c[j].y)}$;
- B. $\sqrt{(c[i].x-c[j].x)*(c[j].x-c[i].x)+(c[i].y-c[j].y)*(c[j].y-c[i].y)}$;
- C. $\sqrt{(c[i].x-c[j].x)*(c[i].x-c[j].x)+(c[i].y-c[j].y)*(c[i].y-c[j].y)}$;
- D. $\sqrt{(c[i].x-c[i].y)*(c[j].x-c[j].y)+(c[i].x-c[i].y)*(c[j].x-c[j].y)}$

2) ②处应填()

- A. $p[i]=1$; B. $p[i]=0$; C. $p[n-i]=0$; D. $p[n-i]=0$;

3) ③处应填()

- A. $p[i]==0 \&\& d[p[j]][j] < minf$
- B. $p[j]!=0 \&\& d[p[j]][j] < minf$
- C. $p[j]==0 \parallel d[p[j]][j] < minf$
- D. $p[i]!=0 \parallel d[p[j]][j] < minf$

4) ④处应填()

- A. $k=minf$; B. $k=0$; C. $k=i$; D. $k=j$;

5) ⑤处应填()

- A. $d[p[j]][j] > d[k][j]$ B. $d[p[j]][j] < d[k][j]$ C. $d[p[i]][j] > d[k][j]$ D. $d[p[i]][j] < d[k][j]$