

普及组模拟题第三套试题及答案

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 (C)。

A. 个人计算机 B. 副机 C. 服务器 D. 主机

【解析】服务器指的是网络环境下为客户机提供某种服务的专用计算机

2. 下列抢占不同数制表示的数中, 最大的一个数是 (C)。

A. 十进制数 220.1 B. 二进制数 11011011.1

C. 八进制数 334.1 D. 十六进制数 DC.1

【解析】都统一为十进制, 二进制转十进制:

整数部分

$11011011.1 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 128 + 64 + 0 + 16 + 8 + 0 + 2 + 1.1 \times 2^(-1) = 219.5$

八进制转十进制

$334.1 = 3 \times 8^2 + 3 \times 8^1 + 4 \times 8^0.1 \times 8^(-1) = 192 + 24 + 4.125 = 220.125$

十六进制转十进制

$DC.1 = 13 \times 16^1 + 12 \times 16^0.1 \times 16^(-1) = 220.0625$

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 码是 65, 字母“E”的 ASCII 码是 (C)。

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

【解析】E 比 A 大 4, $65 + 4 = 69$

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

A. IP、四、网络地址、计算机地址、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

【解析】常识题, 排除法也可, 网络上每台计算机都有个 ip 地址, 最后一个是域名。

5. 在 TCP / IP 协议中, TCP 和 IP 分别提供什么服务 (A)。

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

【解析】TCP 传输控制协议, IP 网际协议即传输层和网络层。

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 (A) 个树叶。

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

【解析】设树 T 有 n 个结点、m 条边。; 边数为结点的度数之和, 即 $m = 2 \times 2 + 1 \times 3 + 3 \times 4 = 19$, $n = m + 1 = 20$ 。N 个结点中有 $1 + 2 + 3 = 6$ 个分支结点, 有叶结点 $20 - 6 = 14$ 个。

7. 在一个图中, 所有顶点的度数之和等于所有边数的 (C) 倍。

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

【解析】无向图中顶点的度是关联于该顶点的边的数目，每一个边关联于两个顶点，因而所有顶点的度数之和等于所有边数的两倍。

8. 下列 IP 地址中正确的是 (B)。

A. 202. 300. 12. 4 B. 192. 168. 0. 3 C. 100:128:35:91 D. 111-102-35-21

【解析】中间是. 其次每部分的值在 0~255 之间。

9. 设有 100 个顶点，利用二分法查找时，最大比较次数是 (D)。

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

【解析】二分查找 $\log_2 n$ 向上取整， $2^7=128$ 所以最大比较次数是 7 次。

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 (B)。

A. ABCDFGHE B. ABDGCEFH C. ACBGDHEF D. ACEFHBGD

【解析】通过中序和后序可以画出一棵二叉树，然后先序遍历即可。

1、后序遍历找到根节点 A

2、在中序找到 A，A 左边的都是 A 的左子树，A 右边的都是 A 的右子树。左子树：DGB 右子树：ECHF

3、DGB 再到后序遍历中找根节点，后序 GDB，A 的左子树根节点就是 B

4、在中序里面找到 B，B 的左子树是 DG，再到后序找 DG，是 GD，D 是 B 的左子树根节点，找到这就得到答案了，先序肯定是 ABD*****
选 B 即可。

11. 下面合法的电子邮件地址是 (C)。

A. http://ybw.online.sh.cn B. ftp://ybw.online.sh.cn

C. ybw@online.sh.cn D. http.ybw.online.sh.cn

【解析】电子邮件地址格式：用户名@域名。

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)。

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

【解析】Telnet 远程登陆 E_mail 电子邮件 TCP/IP 传输控制协议，网际协议 www (万维网) 全称 world wide web

13. 在数据结构中，链表是 (B)。

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

【解析】链表不是顺序存储，其次链表是线性表结构

14. C++ 程序运行时，是在哪种存储器中进行的？ (B)

A. 硬盘 B. RAM C. ROM D. CACHE

【解析】程序运行从硬盘读入到内存（RAM）中运行，ROM 只读存储器

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

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

【解析】逻辑或两边的表达式只要有一边为真，就为真，由 $A \geq B \& B \geq C$ 的取值为真可知，可推出 $A \geq C$ ，当三数相等时 $A == B == C$ ，否则 $A > C$ 。

阅读程序

(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(int 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 }
```

(1) (1 分) 把第 13 行与第 15 行互换位置，结果不会改变。（ ）

答案 ×

【解析】y 数组记录大于 x 数据对应下标数的个数，交换作用相反。

(2) (1 分) 第 14 行把 $\text{if}(x[i] < x[j])$ 删掉效果一样。（ ）

答案 ×

【解析】相等的情况加进来了，所以结果有可能不一样。

(3) 第 11 行把 $i+1$ 改成 1，数组 y 每个元素的值增加 1 倍。（ ）

答案 ✓

【解析】改成 1，存在重复计算。

(4) 数组 $y[i]$ 中存的是 $x[i]$ 在数列中从大到小的次序。（ ）

答案 ✓

【解析】y 数组记录大于 x 数据对应下标数的个数，因此可以看成是从大到小的次序。

(5) 此程序的时间复杂度是 (C)。

A. $O(\log n)$ B. $O(n \log n)$ C. $O(n^2)$ D. $O(n)$

【解析】双重循环，时间复杂度为 $O(n^2)$

(6) 此程序如果 n 输入 4，然后输入 2 4 1 3，输出结果是 (B)。

A. 1 2 3 4 B. 2 0 3 1 C. 4 3 2 1 D. 1 3 0 2

【解析】

查找比第 i 个数大的数的个数

比 2 大的 2 个，比 4 大的 0 个，比 1 大的 3 个比 3 大的 1 个

(2)

```
1#include <iostream>
2#include<cstdio>
3using namespace std;
4int n,i,x;
5int d[11];
6int 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}
```

(1) (1 分) 输入的 x 值应在 $[0, n-1]$ 的范围内。()

答案 ✕

【解析】d 数组定义 `int d[11];`，因为 d 数组定义的元素在 $[0, 10]$ 之间，所以输入的 d 的值也应该在 $[0, 10]$ 之间。

(2) (1 分) 把第 12 行改成 `"d[x]++"`，不影响程序运行结果。()

答案 ✓

【解析】正确的，`d[x]=d[x]+1;`和 `d[x]++;`等同。

(3) 对任意在 $[1, 9]$ 之间的数 i，都有 $d[i-1] < d[i] < d[i+1]$ 。()

答案 ✕

【解析】当某个数没输入时，d 数组对应下标的值与前一个下标的值相等。

(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。()

答案 ✕

【解析】d 数组相邻两数之间只有相等或小于的关系。因此把不等于改为大于效果一样。

(5) 输入 n=10 后，接着输入：1 3 4 3 5 2 4 3 5 2 后，程序输出结果为（ B ）。

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

【解析】前面的是数 i，后面是比数 i 小的数的个数加 1，很容易得出答案。

(6) 把程序第 19 行改成 “printf(“%3d”,i);” 后，输入 n=10 后，接着输入：1 3 4 1 5 4 4 3 5 1 后，程序输出结果为（ D ）。

A. 1 2 3 4 5 B. 3 2 3 2 C. 4 3 4 3 D. 1 3 4 5

【解析】按从小到大输出这些数（去重）。

(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
18         for(i=2;i<=5;i++)
19             for(j=i+1;j<=6;j++)
20                 if(a[i]<a[j]) m=0;
21     }while(m==0);
22     printf("%d",a[2]);
23     return 0;
24 }
```

(1) 程序结束时，a[2]的值一定是数组 a 中的最大值。（ ）

答案 ✓

【解析】根据第 20 行，可知只要有前面的数小于后面的数，m 的值为 0 可知，while 语句的退出条件是前面的数不小于后面的数。

(2) 第 21 行 “m==0” 成立时，数组 a[i] ($2 \leq i \leq 6$) 从大到小排序。（ ）

答案 ✓

【解析】根据第 20 行，可知只要有前面的数小于后面的数，m 的值为 0 可知，while 语句的退出条

件是前面的数不小于后面的数。

(3) 程序输出时, a 数组满足: 对任意的 $2 \leq i < 6$, 有 $a[i] > a[i+1]$ 。()

答案 ×

【解析】退出条件是前面的数不小于后面的数, 包含等于。

(4) 删除第 16 行代码 “m=1” 程序结果会发生改变。()

答案 ×

在前面 m 已经有一个非零值, 删除 “m=1” 不影响结果。

(5) 程序的输出结果为 (C)。

A. 58 B. 59 C. 61 D. 60

【解析】本题通过模拟可以得到结果, 另通过分析, $a[2]$ 每次加 2, 初值为 3, 因此 $a[2]$ 始终是奇数, 从而排除 A 和 D, 按照类似的方法分析可以快速得到答案。

(6) 此程序的时间复杂度是 (A)。

A. $O(n^3)$ B. $O(n \log n)$ C. $O(n^2)$ D. $O(n)$

【解析】

三重循环, 时间复杂度 $O(n^3)$ 。

完善程序

1. Fibonacci 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 Fibonacci 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 Fibonacci 数。

算法说明:

(1) 寻找小于等于 n 的最大 Fibonacci 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。

```
1  #include <iostream>
2  #include <cstdio>
3  using namespace std;
4  int n;
5  bool first;
6  int find(int n)
7  {
8      int a,b,c;
9      a=1;b=1;
10     do
11     {
12         c=a+b;
13         ( ① );
14     }while(b<n);
15     if( ② )
16         return b;
17     else
```

```

18         ( ③ );
19     }
20     void p(int n)
21     {
22         int a;
23         a=find(n);
24         if(first)
25         {
26             printf("%4d",a);
27             first=false;
28         }else
29             ( ④ );
30         if(a<n) ( ⑤ );
31     }
32     int main()
33     {
34         cin>>n;
35         first=true;
36         printf("%5d=",n);
37         p(n);
38         cout<<endl;
39         return 0;
40 }

```

(1) ①处应填 (B)。

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

【解析】斐波那契数列，第三个数等于前两个数之和，计算下一个数，需要提前更新前两个数的值。

(2) ②处应填 (A)。

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

【解析】当 b==n 的时候，任务完成，否则返回不超过 n 的最大值。为继续求解做准备。

(3) ③处应填 (D)。

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

【解析】否则返回不超过 n 的最大值。为继续求解做准备。

(4) ④处应填 (B)。

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

【解析】非第一个斐波那契数前面加上 “+” 号

(5) ⑤处应填 (C)。

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

【解析】继续递归，求余下的数(n-a)。

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<=100，表示城市数目)。

接下来 n 行，每行两个数 xi，yi，表示第 i 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：

```
1  #include <iostream>
2  #include <cstdio>
3  using namespace std;
4  const int maxn=101;
5  struct tcity
6  {
7      float x,y;
8  };
9  tcity c[maxn];
10 float d[maxn][maxn],a,minf;
11 int p[maxn],n,i,j,k;
12 int main()
13 {
14     cin>>n;
15     for(i=1;i<=n;i++)
16         cin>>c[i].x>>c[i].y;
17     for(i=1;i<=n;i++)
18         for(j=1;j<=n;j++)
19             d[i][j]=( ① )
20     p[1]=0;
21     for(i=2;i<=n;i++) ( ② )
22     for(i=1;i<=n-1;i++)
23     {
24         minf=1E10;
25         for(j=1;j<=n;j++)
26         {
27             if( ③ )
28             {
29                 minf=d[p[j]][j];
30                 ( ④ )
31             }
32         }
33         a=a+d[p[k]][k];
34         p[k]=0;
35         for(j=1;j<=n;j++)
36             if( ⑤ ) p[j]=k;
37     }
38     printf("%.2f",a);
39     return 0;
40 }
```

(1) ①处应填 (C)。

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

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

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

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));`

【解析】用两点间的距离公式求解，二维空间欧式距离公式

$$\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

(2) ②处应填 (A)。

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

【解析】`p[i]`初始化为 1，假设 `i` 点到起点距离最短。

(3) ③处应填 (B)。

A. `p[j]==0&&d[p[j]][j]<minf` B. `p[j]!=0&&d[p[j]][j]<minf` C. `p[j]==0||d[p[j]][j]<minf`
D. `p[j]!=0||d[p[j]][j]<minf`

【解析】查找没加入最小费用集合的点到最小费用集合里点的距离最小值的点。

(4) ④处应填 (D)。

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

【解析】查找没加入最小费用集合的点到最小费用集合里点的距离最小值的点，并记录该点。

(5) ⑤处应填 (A)。

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]`

【解析】找到集合外的点与刚加入的点是否有更优的值，并更新最优点。