

## 普及组模拟题第六套试题及答案

1. 在计算机内部用来传送、存贮、加工处理的数据或指令都是以（ A ）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

【解析】计算机的基本运算就是二进制运算。

2. 计算机的软件系统通常分为（ C ）。

- 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

【解析】

因为题里问的是计算机的软件系统，所以 A 肯定错，不包括硬件系统。

分为系统软件和应用软件，常识题。系统软件是指控制和协调计算机及外部设备，支持应用软件开发和运行的系统。

应用软件是和系统软件相对应的。应用软件包是利用计算机解决某类问题而设计的程序的集合，多供用户使用。应用软件是为满足用户不同领域、不同问题的应用需求而提供的那部分软件。它可以拓宽计算机系统的应用领域，放大硬件的功能。

3. 关于软盘读写孔，正确的说法是（ C ）。

- A. 从该孔读信息    B. 从该孔写信息  
C. 当该孔处于关状态时，不能删除盘中文件    D. 该孔没有作用

【解析】

软盘读写孔，实际上是控制全盘读写的，类似于锁、从该孔读信息或从该孔写信息，肯定不对，没有作用显然也不对。

4. 一棵二叉树的中序遍历为 DGBAECHF，后序遍历为 GDBEHFCA，则前序遍历是（ B ）。

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

【解析】模拟一下过程

后序遍历最后一个根节点，于是知道根节点是 A。

根据中序遍历即可确定 A 下面的左右两棵子树分别为 DGB 和 ECHF。

我们在后序遍历中找到 A 的左右两棵子树分别为 GDB 和 EHFC。

后序遍历最后一个根节点，于是知道 A 下的左子树根为 B，右子树根为 C。

根据中序遍历即可确定 B 下面的左子树为 DG，没有右子树。

我们在后序遍历中找到 B 的左子树为 GD，没有右子树。

后序遍历最后一个根节点，于是知道 B 下的左子树根为 D。

根据中序遍历即可确定 D 下面没有左子树，右子树为 G。

根据中序遍历即可确定 C 下面的左子树为 E，右子树为 HF。

根据中序遍历即可确定 E 为叶节点。

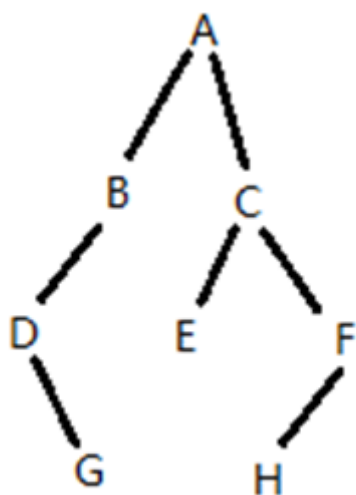
我们在后序遍历中找到 C 的右子树为 HF。

后序遍历最后一个为根节点，于是知道 C 下的右子树根为 F。

根据中序遍历即可确定 F 下面的左子树为 H，没有右子树。

根据中序遍历即可确定 H 为叶节点。

确定二叉树后序遍历：ABDGCEFH



5. 下列叙述中错误的是（ B ）

- 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. 微机应避免频繁开关，以延长其使用寿命
- D. 微机应经常使用，不宜长期闲置不用

【解析】

- A. 计算机附近的强磁场对显示器、磁记录设备等都有影响。
- B. 计算机连续使用几个小时是正常的，没有必要让计算机关机休息一会儿。
- C. 频繁的开关计算机会造成强电压脉冲瞬间冲击，损害计算机设备。
- D. 计算机长期闲置不用，不利于发挥作用，也会像所有电器那样容易受潮锈蚀，电路板落尘短路。

6. 计算机网络最主要的优点是（ B ）

- A. 运算速度快
- B. 共享资源
- C. 精度高
- D. 存储容量大

【解析】计算机网络最主要的优点其实就是网络的优点，显然是 B. 共享资源。

7. 下列 4 个不同进制表示的数中，最大的一个数是（ C ）。

- A. (220.1)<sub>10</sub>
- B. (11011011.1)<sub>2</sub>
- C. (334.1)<sub>8</sub>
- D. (DC.1)<sub>16</sub>

【解析】

都转换成十进制，第一个数 220.1，第二个数 219.5，第三个数 220.125，第四个数 220.0625。

8. 为了区分汉字与 ASCII 码，计算机中汉字编码的最高位为（ A ）。

- A. 1
- B. 0
- C. -1
- D. 2

【解析】ASCII 码一个字符占一个字节，汉字交换码一个汉字占两个字节。ASCII 码字节最高位为 0，汉字交换码字节最高位为 1。

9. 一般来说，TCP / IP 的 IP 提供的服务是（ D ）。

A. 运输层服务    B. 会话层服务    C. 表示层服务    D. 网络层服务

【解析】IP 肯定是网络层。

10. 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（ C ）。

A. D/A    B. A/D    C. Modem    D. Codec

【解析】实现模拟信号与数字信号之间转换功能的是调制解调器，称为 Modem

11. 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（ C ）。

A. 54312    B. 24135    C. 21543    D. 12534

【解析】

A 1 入栈 2 入栈 3 入栈 4 入栈 5 入栈 5 出栈 4 出栈 3 出栈 1 不能出栈，所以 A 不对。

B 1 入栈 2 入栈 2 出栈 3 入栈 4 入栈 4 出栈 1 不能出栈，B 也不对

C 1 入栈 2 入栈 2 出栈 1 出栈 3 入栈 4 入栈 5 入栈 5 出栈 4 出栈 3 出栈 正确

12. 不属于 Internet 的功能是（ D ）。

A. 聊天    B. 远程教育    C. 查询资料    D. 传送能量

【解析】常识，传送能量是不可能的。

13. 下列描述计算机病毒的特性中，（ C ）不是正确的。

A. 潜伏性    B. 传染性    C. 高速性    D. 危害性

【解析】计算机病毒具有潜伏性、传染性、危害性，不具有高速性

14. 下面不属于网络操作系统的是（ D ）。

A. UNIX    B. WINDOWS NT    C. NETWARE    D. DOS

【解析】DOS=磁盘操作系统，不是网络操作系统。

15. 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 ）。

A. 不论何人、何地及发表的软件    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

【解析】常识题，记住即可。

阅读程序

(1)

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  const int Maxn = 10005;
4  int n,b[Maxn];
5  inline void mergesort(int *a,int l,int r){
6      if(l==r) return;
```

```

7      int mid=l+r>>1;
8      mergesort(a,l,mid),mergesort(a,mid+1,r);
9      int i=1,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=1;i<=r;i++) a[i]=b[i-1+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}

```

(1) 该算法中“int \*a”没有传值。( )

答案 ×

【解析】mergesort(a,1,n);函数指针参数也是传值调用，除非&a把地址传过去。

(2) 该算法会换行。( )

答案 ✓

【解析】输出最后一个数的时候  $i=n$ ，所以  $(i==n?' \n' : ' ')$ ；这个三目运算符会输出 ' \n' 。

(3) 该算法中 mergesort 函数时间复杂度为  $O(n\log n)$ 。( )

答案 ✓

【解析】归并排序时间复杂度是  $O(n\log n)$

(4) 如果输入为“5 4 3 9 7 8”则输出为 3 4 7 8 9 \n。( )

答案 ✓

【解析】输入 5 个数，升序排序输出 3 4 7 8 9

(5) 下面哪句与“ $i==n ? ' \n' : ' '$ ”相同 ( C )

A.  $i!=1 ? ' \n' : ' '$     B.  $"\n "[i==n]$     C.  $"\n "[i!=n]$     D. ' '

【解析】这段代码表示  $i==n$  时输出 ' \n'， $i!=n$  时输出 ' '。

A. 会在  $i!=1$  时，输出 \n；在  $i==1$  时，输出 ' '。

B. 会在  $i!=n$  时，输出 ' '；在  $i==n$  时，输出 \n。当  $[i==n]$  为真，“ $"\n "[1]$ ”为 ' '；反之为“ $"\n "[0]$ ”，即 ' \n'。这样，在数组最后一个元素输出后，输出的就不是空格，而是换行。

C. 会在  $i!=n$  时，输出 ' '；在  $i==n$  时，输出 \n。

D. 会在任何时刻输出 ' '。

(6) 该算法的最劣复杂度与哪个排序算法相同 ( D )

A. 快速排序    B. 选择排序    C. 计数排序    D. 堆排序

【解析】该代码最劣复杂度为  $O(n \log n)$ ，快速排序为  $O(n^2)$ ，选择排序为  $O(n^2)$ ，计数排序为  $O(n + \max ai)$  堆排序  $O(n \log n)$ 。

(2)

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int i, j, k, n, m, f[10010], p1, p2, p3;
4  int find(int k) {
5      if(f[k]==k) return k;
6      return f[k]=find(f[k]);
7  }
8  int main()
9  {
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     }
23 }
```

(1) 该算法中 p1 的作用是确定操作类型。( )

答案 ✓

【解析】if(p1==1)

```
15     f[find(p2)]=find(p3);
```

```
16 if(p1==2) {
```

根据上面三行代码可以确定 p1 就是确定操作类型。

(2) 去掉 for(i=1;i<=n;i++) f[i]=i 对该算法没有影响。( )

答案 ✕

并查集初始化，每个元素都是一个集合。不初始化肯定有影响。

(3) 输入 2 2 1 1 2 2 1 2 输出为 Y。( )

答案 ✓

【解析】n=2 m=2

p1=1 p2=1 p3=2

f[find(p2)]=find(p3); 合并 1 和 2 有相同的祖先，并进行了状态压缩。所以再输入 p1=2 p2=1

p3=2 的时候，判断 1 和 2 是否有相同的祖先，有相同的祖先所以输出 Y。

(4) 输入 2 1 2 1 2 输出为 N。( )

答案 ✓

【解析】 $n=2$   $m=1$

$p1=2$   $p2=1$   $p3=2$ ;

因为  $p1==2$ , 所以直接判断是否有相同的祖先, 不是相同的祖先输出 N

(5) 该算法时间复杂度为 ( A )

A.  $O(m \log n)$     B.  $O(nm)$     C.  $O(n+m)$     D.  $O(n \cdot m^2)$

【解析】没有按秩合并的并查集复杂度是  $O(\log n)$ , 再做  $m$  次, 复杂度就为  $O(m \log n)$ 。

(6) 把 `return f[k]=find(f[k]);` 改成 `return find(f[k]);`; 最差时间复杂度为 ( B )

A.  $O(m \log n)$     B.  $O(nm)$     C.  $O(n+m)$     D.  $O(n \cdot m^2)$

【解析】没有路径压缩的并查集复杂度是  $O(n)$ , 再做  $m$  次, 复杂度就为  $O(nm)$ 。

(3)

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int t,x[100],a[100];
4  void work(int d,int i,int n){
5      int k;
6      if(n==1)
7          {
8              for(k=0;k<d;k++)
9                  printf("%3d",a[k]);
10             printf("\n");
11         }
12     else
13         for(k=i;k<t;k++)
14             if(n%x[k]==0)
15                 {
16                     a[d]=x[k];
17                     work(d+1,k,n/x[k]);
18                 }
19 }
20 int main(){
21     int i,k,n;
22     cin>>n;
23     for(i=n;i>1;i--) if(n%i==0) x[t++]=i;
24     work(0,0,n);
25     return 0;
26 }
```

(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。( )

答案 ✕

【解析】不是求因数, 因为  $n$  的因数包括 1, 而这个循环没有枚举到 1。

(2) 该程序的作用是对  $n$  进行质因数分解。( )

答案 ✕

【解析】不是求质因数分解, 两个数相乘等于  $n$  的所有方案。

(3) printf("%3d",a[k]);中 去掉 3 对程序没有影响。( )

答案 ×

【解析】3 是表示位宽。对输出的格式会有影响。

(4) 去掉 if(n%x[k]==0)对程序有影响。( )

答案 ✓

【解析】这里 n 是被分解过了，所以不一定能整除 x[k]

(5) 如果输入为 2，那么输出为 ( A )

A. 2     B. 2 1     C. 1 2     D. 2 2

【解析】2 仅有 2 这 1 种分解方案（因为没有 1）。

(6) 如果输入为 72，那么输出的非回车字符有多少行 ( C )

A. 14     B. 15     C. 16     D. 17

【解析】16 行，枚举方案。别漏了 72 这种方案。

```
72
36 2
24 3
18 4
18 2 2
12 6
12 3 2
9 8
9 4 2
9 2 2 2
8 3 3
6 6 2
6 4 3
6 3 2 2
4 3 3 2
3 3 2 2 2
```

完善程序

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）

```
1  #include <iostream>
2  using namespace std;
3  const int SIZE=100;
4  int x[SIZE],y[SIZE],f[SIZE];
5  int n,i,j,max_f,ans;
6  int main()
7  {
8      cin>>n;
9      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 )

A. 0     B. 1     C. i     D. INT\_MAX

【解析】f[i]是记录有多少个点在它左下方的，显然应该初值为0。

(2) ②处应该填 ( B )

A. y[j]<=y[i]     B. y[j]<y[i]     C. y[j]>y[i]     D. y[j]>=y[i]

【解析】左下方的点是 x、y 坐标都比它小，所以是小于。

(3) ③处应该填 ( B )

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

【解析】f[i]是计数的，所以应该 f[i]++

(4) ④处应该填 ( B )

A. (f[i]>max\_f)     B. (f[i]>=max\_f)     C. (f[i]<max\_f)     D. (f[i]<=max\_f)

【解析】因为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），所以创造的最好成绩的点，显然标号比 i 小，所以是大于等于。

(5) ⑤处应该填 ( C )

A. ans++     B. ans+=max\_f     C. ans=i     D. ans=max\_f

【解析】ans 是存标号的。

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





```
49     }  
50     return 0;  
51 }
```

(1) ①处应该填 ( A )

A. false    B. true    C. 1    D. -1

【解析】结合上下文去看，flag 应该初值为 false，之后才会再变为 true。

(2) ②处应该填 ( D )

A. used[i]=true    B. data[i]=i

C. used[data[i]]=true    D. used[data[i]]=false

【解析】先清空，标记成没用过，后面用的时候再设为用过。

(3) ③处应该填 ( A )

A. j    B. i    C. true    D. false

【解析】枚举能变哪个数字 j，如果可以变就变。

(4) ④处应该填 ( A )

A. n    B. m    C. i    D. j

【解析】数字  $1 \sim n$ 。

(5) ⑤处应该填 ( D )

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

【解析】排除法：AB 不对，找到答案至少要输出来，C 进入下一次枚举也不对。