

普及组模拟题第七套试题及答案

1. 以下属于系统软件的是：(A)。

A. C++编译器 B. 腾讯 QQ C. CAD D. 游戏软件

【解析】腾讯 QQ，CAD，游戏软件 这些都是应用软件。C++编译器，属于系统软件，所有的编译器都属于系统软件。

2. 年__月__日在国际电信标准组织 3GPP RAN 第 78 次全体会议上，5G NR 首发版本正式发布，这是全球第一个可商用部署的 5G 标准。(D)

A. 2017 年 8 月 18 日 B. 2018 年 1 月 1 日
C. 2017 年 12 月 25 日 D. 2017 年 12 月 21 日

【解析】常识题。

3. 如果用一个字节来表示整数，最高位用作符号位，其他位表示数值。例如：00000001 表示 +1, 10000001 表示 -1，试问这样表示法的整数 A 的范围应该是 (A)。

A. $-127 \leq A \leq 127$
B. $-128 \leq A \leq 128$
C. $-128 \leq A < 128$
D. $-127 < A \leq 128$

【解析】

最大 $01111111(2) = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 64 + 32 + 16 + 8 + 4 + 2 + 1 = 127$ ，最小 $11111111(2) = -127$ 。

4. 下列属于网络模型的名称是 (B)。

A. LAN B. TCP/IP C. FTP D. SMTP

【解析】TCP/IP 四层参考模型

网络接口层：网络接口层包括用于协作 IP 数据在已有网络介质上传输的协议。

可以理解为：确定网络数据包的形式。

网络层：网络层对应于 OSI 七层参考模型的网络层，本层包含 IP 协议、RIP 协议，负责数据的包装、寻址和路由。可以理解为：该层能确定计算机的位置。

传输层：传输层对应于 OSI 七层参考模型的传输层，它提供两种端到端的通信服务。TCP：三次握手、四次挥手（面向连接，可靠）；UDP：面向无连接

应用层：应用层对应于 OSI 七层参考模型的应用层和表达层会话层；
其他都是应用层。

5. 在 C++ 中， $(-7)\%(-5)$ 等于 (B)。

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

【解析】c++ 是向下取整， $x\%y = x - (y * (x/y))$ ，用这个公式计算就可以。

6. 学号为 1 到 30 的小朋友顺时针排成一圈，从 1 号小朋友开始顺时针报数，从数字 1 开始数下

去, 1, 2, 3, . . . , 28, 29, 30, 31, 32, . . . , 一圈又一圈, 问当数到数字 n , 所在的小朋友的学号为多少? (B)。

- A. $(n-1)\%30$ B. $1+(n-1)\%30$ C. $(n+1)\%30-1$ D. $(n+1)\%30$

【解析】 $0 \sim (n-1)$ 的数 $\%n$, 编号转圈报数, 例如 1 到 5 个小朋友,

$0\%5=0$ $1\%5=1$ $2\%5=2$ $3\%5=3$ $4\%5=4$ $5\%5=0$ $6\%5=1$ $7\%5=2$ $8\%5=3$ $9\%5=4$ $10\%5=0$

如果是 30 个小朋友想知道原来的编号, $(n-1)\%30$ 但是编号都减一了, 因此再加一个 1, 结果就是 $1+(n-1)\%30$ 。

7. 一棵完全二叉树的结点总数为 41, 其叶结点数为 (D)。

- A. 18 个 B. 19 个 C. 20 个 D. 21 个

【解析】完全二叉树: 二叉树中除去最后一层结点为满二叉树, 且最后一层的结点依次从左到右分布。完全二叉树的特点: 叶结点只能出现在最下层和次下层, 且最下层的叶结点集中在树的左部。结点总数是 41, $2^5-1=31$, $2^6-1=63$, 可以确认这是一棵高度为 6 的二叉树, $63-41=22$, 对比满二叉树缺少了 22 个结点, 第六层的节点总数 $2^{(6-1)}=32$, $32-22=10$, 第六层有 10 个叶节点, 第五层还有叶节点, 继续计算, 第六层有 10 个叶节点, 证明第 5 层有 5 个分支结点, 第五层总数 $2^{(5-1)}=16$, 节点总数-分支节点=第 5 层叶节点, $16-5=11$, 6 层叶节点+5 层叶节点=全部叶节点数, 即 $11+10=21$ 。

8. 给出 3 种排序: 插入排序、冒泡排序、选择排序。这 3 种排序的时间代价分别是 (D)。

- A. $O(n)$ 、 $O(n^2)$ 、 $O(\log_2 n)$ B. $O(\log_2 n)$ 、 $O(n)$ 、 $O(n^2)$
C. $O(n^2)$ 、 $O(n)$ 、 $O(n)$ D. $O(n^2)$ 、 $O(n^2)$ 、 $O(n^2)$

【解析】三种排序方法时间复杂度都是 $O(n^2)$

9. 请给以下四个事件发生的时间排序 (B)

1. 举办第一次 NOIP
2. 举办第一次 NOI 网络同步赛
3. NOIP 提高组由四题改为三题
4. 举办第一次 APIO

- A. 1234 B. 1432 C. 2134 D. 2143

【解析】noip 肯定在 NOI 之前, 提高组改为 3 题, 是近几年的事, 肯定晚于 APIO。

10. 在 OSI 模型中最底层是 (D)

- A. 会话层 B. 网络层 C. 数据链路层 D. 物理层

解析: 背下来七层模型

11. 以下关于图的不正确说法是 (B)。

- A. 所有顶点的度数之和等于边数的 2 倍
B. 所有顶点的度数之和不一定等于边数的 2 倍
C. 任意一个图一定有偶数个奇点

D. 在有向图中顶点的入度之和等于出度之和

【解析】每条边都会给两个顶点度数+1，所以一条件对度数和的贡献是 2。

12. 6 个人分乘两辆不同的汽车，每辆车最多坐 4 人，则不同的乘车方法数为（ B ）

A. 40 B. 50 C. 60 D. 70

【解析】

只有 2 辆车，所以不在 1 号车就在 2 号车。

考虑 1 号车，合法情况有坐 2 人、3 人、4 人。

坐 2 人的方案数是 $C(6, 2)$ 。

坐 3 人的方案数是 $C(6, 3)$ 。

坐 4 人的方案数是 $C(6, 4)$ 。

13. 为了实现两数交换，代码如下：

```
void swapAB(int &a, int &b)
{
    ;
    b=a-b;
    a=a-b;
}
```

则空格内要填入的语句是（ A ）

A. $a=a+b$ B. $a=a*b$; C. $a=a-b$; D. $a=a\&b$;

【解析】选 A，设原数为 x, y 。于是从上到下运行结果依次是：

$a=x+y, b=y$

$a=x+y, b=(x+y)-y=x$

$a=(x+y)-x=y, b=x$

14. 某数列有 1000 个各不相同的数，由低到高按序排列，现要对该数列进行二分法检索，在最坏的情况下，需要检索（ B ）个数据。

A. 1000 B. 10 C. 100 D. 500

【解析】二分复杂度 $O(\log n)$ ， $2^{10}=1024$ 。

15. 以下简称和全称不对应的是（ C ）

A. NAT(Network Address Translation)

B. TCP(Transmission Control Protocol)

C. ARP(Adobe Resolution Protocol)

D. ICMP(Internet Control Message Protocol)

【解析】ARP(Adobe Resolution Protocol)应该是 ARP(Address Resolution Protocol)

阅读程序

(1)

```
1  #include <bits/stdc++.h>
2  using namespace std;
```

```

3   int main()
4   {
5       string s1,s2;
6       cin>>s1;cin>>s2;
7       int cnt=0;
8       for(int i=0;i<s1.size();i++)
9       {
10          for(int j=0;j<s2.size();j++)
11             if(s1[i]==s2[j]) cnt++;
12      }
13      cout<<cnt;
14      return 0;
15  }

```

(1) 输入必须全要是字母，否则无法被识别。()

答案 ×

【解析】除空格之外的其他字符也可以。

(2) 将 10 行的 j 全部换成 i 是有问题的。()

答案 ✓

【解析】双重循环，j 换成 i 有问题。

(3) 本程序的功能是统计两个字符串的最长公共子序列长度。()

答案 ×

【解析】统计的是字符串 1 的字符出现在字符串 2 中的总数量。

(4) 本程序的时间复杂度为 $O(n^2)$ 。

答案 ✓

【解析】双重循环时间复杂度 $O(n^2)$ 。

(5) 若输入的两个字符串长度均为 12，那么输出最大为 (B)

A. 0 B. 144 C. 12 D. 24

【解析】输出最大为 $12 \times 12 = 144$ 。

(6) 若 s1 长度为 4，输出为 6，则 s2 的长度至少为 (B)

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

【解析】输出最少为 2 个。s1 字符串也有相同，例如 aaab，s2 为 aa。

(2)

```

1   #include <bits/stdc++.h>
2   using namespace std;
3   const int MAXN=1e5+7;
4   int a[MAXN],b[MAXN];
5   int main() {
6       int n,m,x,y;
7       cin>>n>>m;
8       for(int i=1;i<=n;i++)
9       {
10          cin>>x>>y;

```

```

11         a[x]++;
12         a[y+1]--;
13     }
14     int cnt=0,ans=0;
15     for(int i=0;i<=m;i++)
16     {
17         cnt+=a[i];
18         ans+=cnt;
19     }
20     cout<<ans;
21     return 0;
22}

```

注：输入流中 $1 \leq x \leq y \leq m$

(1) 输入的 x 和 y 可以是全体整数。()

答案 ×

【解析】 $1 \leq x \leq y \leq m$ ，这个范围内的整数。

(2) 将 14 行的清零过程除去没有问题。()

答案 ×

【解析】清零除去，结果不正确，因为主函数内定义的变量值是随机值。

(3) 将 17 行与 18 行交换位置不会影响最终结果。()

答案 ×

【解析】结果不一样，少加一次。

(4) 将 11 行的 x 改成 $x-1$ 并把 12 行的 $y+1$ 改成 y 不会影响最终结果。()

答案 ×

【解析】会影响最终结果。

(3)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int a[6];
4  int change(int a){a++;}
5  int changel(int &a){a++;}
6  int main(){
7      int c=1;for(int i=1;i<=5;i++) a[i]=i*3;
8      int *b=&a[1];
9      change(*b);cout<<*b<<endl;cout<<a[1]<<endl;
10     *b++;cout<<*b<<endl;cout<<a[1]<<endl;
11     changel(*b);cout<<*b<<endl;cout<<a[1]<<endl;
12     *b=c;
13     change(c);cout<<*b<<endl;cout<<c<<endl;
14     changel(c);cout<<*b<<endl;cout<<c<<endl;
15     return 0;
16 }

```

(1) 将第 7 行中 `int` 换为 `long long` 后程序依然能通过编译。()

答案 ×

【解析】不能通过编译，因为第 13，14 行 change 和 changel 函数是 int。

(2) change 与 changel 两个函数等价。()

答案 ×

【解析】change 不改变主函数的值，changel 改变主调函数的值。

(3) 将第 12 行换为 b=&c; 输出值不变。()

答案 ×

【解析】输出值有变化。

(4) 将第 8 行换为 int *b=a+1; 输出值不变。()

答案 ✓

【解析】输出的值不变。两种写法等同。

(5) 输出结果的最大值是 (C)

A. 6 B. 4 C. 7 D. 5

【解析】

模拟输出结果

3
3
6
3
7
3
1
1
1
2

(6) 输出结果的乘积是 (A)

A. 6804 B. 5760 C. 11520 D. 13608

【解析】

结果是 6804

完善程序

1. 给出 N 个整数，要统计每个数前面有多少比它大的数字。比如有 5 个数的数列：2 5 1 3 4，则第 1 个数 2 之前有 0 个数比它大；第 2 个数 5 之前有 0 个数比它大；第 3 个数 1 之前有 2 个数比它大；第 4 个数 3 之前有 1 个数比它大；第 5 个数 4 之前有 1 个数比它大。数据范围：每个数范围是 $[0 \dots 200]$ ， $N \leq \lfloor 10 \rfloor^5$

```
1  #include <iostream>
2  using namespace std;
3  int d[100002];
4  int c[1300];
5  int main() {
6      int n, ans, x;
7      cin >> n;
```

```

8      for(int i=0;i<n;i++)
9          ①;
10     for (int i=0;i<n;i++){
11         ②;
12         for(int j= ③ ;j<=200;j++)
13             ④;
14         cout<<ans<<" ";
15         ⑤;
16     }
17     cout<<endl;
18     return 0;
19 }

```

(1) ①处应该填 (B)

A. cin>>c[i] B. cin>>d[i] C. read(c[i]) D. read(d[i])

【解析】没有快读的函数，因此 C, D, 不对，另外根据 $N \leq \lfloor 10 \rfloor^5$ 可知输入的是 d 数组。

(2) ②处应该填 (C)

A. ans++ B. c[i]=d[i] C. ans=0 D. c[i]++

【解析】

ans 初始化为 0。

(3) ③处应该填 (C)

A. d[i] B. c[i]+1 C. d[i]+1 D. c[i]

【解析】要统计比当前数值大的。

(4) ④处应该填 (D)

A. c[j]+=d[i] B. ans+=(c[j]==1) C. ans++ D. ans+=c[j]

【解析】

ans 统计答案，统计所有下标比 d[i] 大的 c 数组数值。

(5) ⑤处应该填 (A)

A. c[d[i]]++ B. c[i]++ C. ans=c[i] D. d[c[i]]++

【解析】

计入 c 中使后面数字能够统计。

给定一个 $n \times n$ 的棋盘，棋盘有一些位置不能放皇后。现在要向棋盘放入 n 个黑皇后和 n 个白皇后，使任意的两个黑皇后都不在同一行、同一列或同一条对角线上，任意的两个白皇后都不在同一行、同一列或同一条对角线上。问总共有多少种放法？

输入描述

输入的第一行为一个整数 n，表示棋盘的大小。

接下来 n 行，每行 n 个 0 或 1 的整数，如果一个整数为 1，表示对应的位置可以放皇后，如果一个整数为 0，表示对应的位置不可以放皇后。

输出描述

输出一个整数，表示总共有多少种放法。

```
1  #include <iostream>
2  using namespace std;
3  int b[20],b1[20],c[20],c1[20],d[20],d1[20];
4  int n;
5  bool a[20][20],a1[20][20];
6  int ans=0;
7  void dfs(int t,int p){
8      if(t>n&& p==1){
9          ①
10         return;
11     }
12     if(t>n&& p==2){
13         ans++;
14         return;
15     }
16     for(int j=1;j<=n;j++){
17         if(p==1&&!b[j]&&!d[t-j+n]&&!c[t+j]&&a[t][j]){
18             a[t][j]=a1[t][j]=0;
19             ②
20             dfs(t+1,p);
21             b[j]=c[t+j]=d[t-j+n]=0;
22             a[t][j]=a1[t][j]=1;
23         }
24         if( ③ &&!b1[j]&&!d1[t-j+n]&&!c1[t+j]&&a1[t][j]){
25             a1[t][j]=0;
26             b1[j]=c1[t+j]=d1[t-j+n]=1;
27             dfs(t+1,p);
28             b1[j]=c1[t+j]=d1[t-j+n]=0;
29             ④
30         }
31     }
32 }
33 int main(){
34     cin>>n;
35     for(int i=1;i<=n;i++){
36         for(int j=1;j<=n;j++){
37             cin>>a[i][j];
38             a1[i][j]=a[i][j];
39         }
40         ⑤
41         cout<<ans;
42         return 0;
43     }
```

(1) ①处填写 (B)。

A. dfs(1,1); B. dfs(1,2); C. dfs(1,0); D. dfs(0,0);

解析：本题是 2n 皇后问题，if(t>n&&p==1)搜索完白皇后的位置后，应该开始搜索黑皇后的摆放位置，所以应该是 dfs(1,2)。p 的值为 2 开始搜索下一个皇后的位置。

(2) ②处填写 (D)。

- A. $b[j]=c[t+j]=d[t-j+n]=0;$
B. $b[j]==c[t+j]==d[t-j+n]==1;$
C. $b[j]==c[t+j]==d[t-j+n]=0;$
D. $b[j]=c[t+j]=d[t-j+n]=1;$

解析：这个点可以摆放换后，它的列，左对角线，右对角线全部标记为 1。

(3) ③处填写 (B)。

- A. $p==0$ B. $p==2$ C. $p==1$ D. $p=0;$

解析：2n 皇后问题，根据上下文（假设白黑两种皇后），假设 $p==1$ 的时候代表白皇后， $p==2$ 的时候代表黑皇后，第 17 行 $p==1$ 判断的是白皇后，那么③出应该是 $p==2$ 判断黑皇后的摆放。

(4) ④处填写 (A)。

- A. $a1[t][j]=1;$ B. $a[t][j]=a1[t][j]=1;$ C. $a1[t][j]=1;$ D. $a[t][j]=a1[t][j]=0$

解析：根据第 25 行，此处应该回溯，所以选择 A $a1[t][j]=1;$

(5) ⑤处填写 (A)。

- A. $dfs(1,1);$ B. $dfs(1,2);$ C. $dfs(0,0)$ D. $dfs(1,0);$

解析：从第一行，第一类颜色的皇后摆放开始搜索。所以选择 A