1、将一个递归算法改为对应的非递归算法时，通常需要使用（）  
A、优先队列   B、队列     C、循环队列     D、栈  
  
2、爸爸、妈妈、妹妹、小强，至少两个人同一生肖的概率是多少？  
A、41/96      B、55/96     C、72/128      D、90/128  
  
3、已知\*p=NULL， \*q=new char[100]，sizeof(p) 和 sizeof(q)的值各为多少？  
A、4和100      B、4和4      C、不确定  
  
4、求以下程序的输出结果（）

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/hackbuteer1/article/details/11263443)

1. **int** foo(**int** x , **int** y)
2. {
3. if(x <= 0 || y <= 0)
4. return 1;
5. return 4\*foo(x - 1 , y/2);
6. }
7. cout<<foo(3,5)<<endl;

int foo(int x , int y)

{

if(x <= 0 || y <= 0)

return 1;

return 4\*foo(x - 1 , y/2);

}

cout<<foo(3,5)<<endl;

A、256      B、64    C、16      D、4        E、1  
  
5、已知一张员工数据表A的表结构如图，请用一条SQL语句列出所有的工作岗位（JOB字段）的平均工资，并将其按照平均工资用升序排列。  
A {  
     ENAME VARCHAR(20)  
     JOB   VARCHAR(20)  
     SALARY NUMBER(5)  
}  
  
  
6、描述在浏览器中敲入一个网址并按下回车后所发生的事情（尽量详细）  
答：浏览器输入网址之后，首先  
步骤1：需要查找域名的IP地址，DNS查找过程如下：  
（1）浏览器缓存 – 浏览器的缓存DNS记录一段时间。 有趣的是，操作系统没有告诉浏览器储存DNS记录的时间，这样不同浏览器会储存各自固定的一个时间（2分钟到30分钟不等）。  
（2）系统缓存 – 如果在浏览器缓存里没有找到需要的记录，浏览器会做一个系统调用（windows里是gethostbyname）。这样便可获得系统缓存中的记录。  
（3）路由器缓存 – 接着，前面的查询请求发向路由器，它一般会有自己的DNS缓存。  
（4）ISP DNS 缓存 – 接下来要check的就是ISP缓存DNS的服务器。在这一般都能找到相应的缓存记录。  
（5）递归搜索 – 你的ISP的DNS服务器从跟域名服务器开始进行递归搜索，从.com顶级域名服务器到Facebook的域名服务器。一般DNS服务器的缓存中会有.com域名服务器中的域名，所以到顶级服务器的匹配过程不是那么必要了。  
步骤2：浏览器给web服务器发送一个HTTP请求。请求中也包含浏览器存储的该域名的cookies。可能你已经知道，在不同页面请求当中，cookies是与跟踪一个网站状态相匹配的键值。这样cookies会存储登录用户名，服务器分配的密码和一些用户设置等。Cookies会以文本文档形式存储在客户机里，每次请求时发送给服务器。  
步骤3：服务的永久重定向响应  
步骤4：浏览器跟踪重定向地址  
步骤5：服务器“处理”请求  
步骤6：服务器发回一个HTML响应  
步骤7：浏览器开始显示HTML  
步骤8：浏览器发送获取嵌入在HTML中的对象  
  
7、有一个在给定字符串中查找子串的函数strstr，该函数从给定的字符串src中查找substr并返回一个整数，指明substr第一次出现的位置（从0开始计数），如果找不到则返回-1。  
要求：  
1、实现该函数。  
2、为该函数设计与实现单元测试。  
说明：  
1、代码中不允许使用系统已有的库函数，所有用到的库函数都需要自己实现  
2、允许使用任何编程语言，函数原型自行给定。参考的C语言函数原型为 int strstr(char\* src , char\* substr)