# 吉林大学 2001 年 C 语言程序设计试题（含答案）

**报考专业：计算机系统结构　计算机软件与理论　计算机应用技术**
**研究方向：以上专业的方向**
**考试科目：C 语言程序设计**

注意：
1　答案一律书写在答题纸上
2　题签随答题纸交回
【一】（2 5 分）
(1)试说明下面函数 m 的功能是什么。

```
typedef int valtype[100]
int m(valtype a,int n)
{
   if (n == 0) return(a[0]);
   else if(m(a,n-1)>a[n]) return(m(a,n-1));
            else return(a[n]);
}
```

(2)试计算 f(109)和 f(97)的值。其中函数 f 的定义如下：

```
int f(int n)
{
   if (n>100) retrun (n-10);
   else return(f(f(n+11)));
}
```

(3)试计算 g(1,10)的值，其中函数 g 的定义如下：

```
float g(int x,int y)
{
   if (x>=y) return ((x+y)/2);
   else return(g(g(x+2,y-1),g(x+1,y-2)));
}
```

[参考答案]
第一题
(1).返回数组 a 前 n+1 个元素中最大的元素。
(2).f(109)=99,
　　f(97)=91.
(3). g(1,10)=4.
【二】（2 5 分）
(1)设有下面语句，其中 E 表示表达式，Si 表示 C 语言的非 break 语句．试把这个语句改写成另外一种形式，使得其中 E 和每个 Si 只出现一次，并且没有 goto 语句。

```
if (E == 1) {S1;S2;S3}
else if (E == 2) {S2;S3;}
else if (E == 3) S3;
else if (E == 4) S4;
```

(2)下面是 Ackrman 函数定义，试写出实现这个函数的非递归的 C 语言函数.

Ack(0,n) = n + 1;

Ack(m,0) = Ack(m-1,1);

Ack(m,n) = Ack(m-1,Ack(m,n-1));当 m≠0,n≠0

[参考答案]

```
/*==========================================================================
=======*/
/*          函          数          名          称          :          2001_2.c
*/
/*  程  序  目  的  :  将  Ackrman  函  数  改  写  为  非  递  归  函  数
*/
/*Writen          by          Apechn          ,Soft          Lab          of          JLU
*/
/*==========================================================================
=======*/
//[解题思想]：用一个堆栈来实现递归过程。
long ack(int m,int n)
{
   int a[2000],top=-1; //a[2000]用来模拟堆栈，top 为栈顶指针

   top++; //把 m 和 n 压入堆栈
   a[top]=m;
   top++;
   a[top]=n;

   while(top!=0) //直到堆栈中只剩下一个元素为止
   {
      n=a[top];
      top--;
      m=a[top];
      top--;

      if(m==0)
      {
         top++;
         a[top]=n+1;
         continue;
      }

      if(n==0)
      {
         top++;
         a[top]=m-1;
```

```
        top++;
        a[top]=1;
        continue;
      }

      top++;
      a[top]=m-1;
      top++;
      a[top]=m;
      top++;
      a[top]=n-1;
   }

   return a[top]; //返回堆栈中最后一个数
}
```

【三】（2 5 分）

假设给定两个表 L1 和 L2，则我们可以定义一个新表 LL（称为对偶表）：

LL ＝｛(a,b)|a∈L1,b∈L2}∪{(b,a)|a∈L1,b∈L2}

为了确定起见，考虑字符表，并且约定用数组来表示字符表 L1 和 L2，用链表来表示对偶表 LL．试写出一个对于任给字符表 L1 和 L2 求其对偶表 LL 的 C 语言函数。

[参考答案]

```
*=============================================================================
======*/
/*          函          数          名          称          ：          2001_3.c
*/
/*   程   序   目   的   ：   求   给   定   字   符   表   的   对   偶   表
*/
/*Writen          by          Apechn          ,Soft          Lab          of          JLU
*/
/*=============================================================================
=======*/
```

[解题思想]：把 L1 和 L2 两个表分别扫描一遍，把扫描得到的数对放到对偶表中。最后再把对偶表中的相同元素删除即可。

```
struct node //对偶表的结点结构
{
char x;
char y;
node* next;
}
node* head; //对偶表的头指针
void del() //删除函数，删除相同元素
{
```

```
node *p=head,*q,*r;
whiel(p!=NULL)
{
q=p->next;
r=p;
while(q!=NULL)
{
if(p->x==q->x&&p->y==q->y)
{
r->next=q->next;
delete q;
q=r->next;
break;
}
else
{
r=q;
q=q->next;
}
}
p=p->next;
}
}
void fun(char L1[],char L2[]) //题目所求的函数
{
int i,j;
node* p;
for(i=0;ifor(j=0;j{
p=new node; //从表头插入结点
p->x=L1;
p->y=L2[j];
p->next=head;
head=p;
p=new node; //从表头插入结点
p->x=L2[j];
p->y=L1;
p->next=head;
head=p;
}
del(); //删除相同的元素
}
```

【四】（2 5 分）

二维数组的元素能按不同的方式进行排序，如下面图 1 表示按行式排序，而图 2 则表示

螺旋式排序.由此我们可以称图 4 所示数组是图 3 所示数组的螺旋数组。试编一 C 语言函数，使得只要给出二维数组名 A 和 B，以及上界 n(下界为 0),则将在 B 中产生 A 的螺旋数组。我们假设数组元素是字符。

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

1 2 3 4
12 13 14 5
11 16 15 6
10 9 8 7

a b c
d e f
g h i

a b c
h i d
g f e

图 1 图 2 图 3 图 4

[参考答案]

```
/*==========================================================================
=======*/
/*          函          数          名          称          :          2001_4.c
*/
/*  程  序  目  的  :  写  出  给  定  数  组  的  螺  旋  数  组
*/
/*Writen          by          Apechn          ,Soft          Lab          of          JLU
*/
/*==========================================================================
=======*/
//[解题思想]：设一个方向变量 orient，取值 0、1、2、3 分别代表往右、往下、往左、往上。
当遇到边界或当前//位置已经有东西时，调整方向。
//否则一直顺着当前方向走下去。走到一个位置，把 a 数组的相应位置的字符写到 b 上。
const int n=4;
void fun(char a[][n],char b[][n])
{
   int i=0,j=0,p=0,orient=0;

   while(p    {
      if(orient==0) //如果当前方向为向右
      {
         if(j==n||b[j]!=32) //如果已经到达边界或当前位置已经有字符
         {
            i++; //调整方向
```

```
            j--;
            orient=(orient+1)%4;
            continue;
        }
        else
        {
            b[j]=a[p/n][p%n]; //否则把 a 的相同字符填到 b 中
            p++; //a 和 b 都指向下一个位置
            j++;
        }
    }

    if(orient==1) //如果当前方向为向下
    {
        if(i==n||b[j]!=32)
        {
            i--;
            j--;
            orient=(orient+1)%4;
            continue;
        }
        else
        {
            b[j]=a[p/n][p%n];
            p++;
            i++;
        }
    }

    if(orient==2) //如果当前方向为向左
    {
        if(j==-1||b[j]!=32)
        {
            i--;
            j++;
            orient=(orient+1)%4;
            continue;
        }
        else
        {
            b[j]=a[p/n][p%n];
            p++;
            i--;
        }
```

```
    }

    if(orient==3) //如果当前方向为向上
    {
      if(i==-1||b[j]!=32)
      {
        i++;
        j++;
        orient=(orient+1)%4;
        continue;
      }
      else
      {
        b[j]=a[p/n][p%n];
        p++;
        i--;
      }
    }
}
```