《C语言程序设计》

试题集及答案

目录

（按住 CTRL键点击超链）

《C语言程序设计》精品课件制作小组

2009 年 4 月

一、单项选择题

导读：单项选择题要求从给出的四个备选答案中，选出一个最符合题意的答案。本类习题主要检查对 **C** 语言基本概念的掌握情况，　　读者可根据学习进度选做部分习题。　　在完成习题的过程中，不但要选出正确的答案，而且要清楚不正确的选项错在何处，以加深对概念的理解。对于掌握不准的问题，应该通过上机实验来检验。

【1.1】以下不正确的　**C** 语言标识符是　_____。

A) int B) a_1_2 C) ab1exe D) _x

【1.2】以下是正确的　**C** 语言标识符是　_____。

A) #define B) _123 C) %d D) \n

【1.3】下列四组字符串中都可以用作Ｃ语言程序标识符的一组是　　。

？？？ **A) print　B) i\am　C) Pxq　D) str_l**

？？？ **_3d　one_half　My->book　Cpp**

？？？ **oodbs　tart$it　line#　pow**

？？？ **aBc　3pai　His.age　while**

【1.4】下面各选项组中，均是　**C** 语言关键字的组是　。

A) auto，enum，include B) switch，typedef，continue

C) signed，union，scanf D) if，struct，type

【1.5】下列不属于　**C** 语言关键字的是　。 **A) default B) register C) enum D) external**

【1.6】C 语言程序从　**main()** 函数开始执行，所以这个函数要写在　　_____。

A) 程序文件的开始　　**B)** 程序文件的最后

C) 它所调用的函数的前面　　**D)** 程序文件的任何位置

【1.7】下列关于　**C** 语言的叙述错误的是　_____

A) 大写字母和小写字母的意义相同

B) 不同类型的变量可以在一个表达式中

C) 在赋值表达式中等号　**(=)** 左边的变量和右边的值可以是不同类型

D) 同一个运算符号在不同的场合可以有不同的含义

【1.8】在　**C** 语言中，错误的　**int** 类型的常数是　。

A) 32768 B) 0 C) 037 D) 0xAF

【1.9】执行语句　**printf("%x",-1)**　；屏幕显示　_____。

A) -1 B) 1 C) -ffff D) ffff

【1.10】已知　**long i=32768**；执行语句　**printf("%d",i)**　；屏幕显示　_____。

A) -1 B) -32768 C) 1 D) 32768

【1.11】已知　**long i=65539**；

执行语句 **printf("%d",i)** ；屏幕显示 ____。

A) 65539 B) -3 C) 3 D) 程序不能执行

【1.12】在C语言中，整数 **-8** 在内存中的存储形式是 。

A) 1111 1111 1111 1000 B) 1000 0000 0000 1000

C) 0000 0000 0000 1000 D) 1111 1111 1111 0111

【1.13】C语言中字符型 **(char)** 数据在内存中的存储形式是 ____。

A) 原码 **B)** 补码 **C)** 反码 **D) ASCII** 码

【1.14】将字符 **g** 赋给字符变量 **c**，正确的表达式是 。

A) c=\147 B) c="\147" C) c='\147' D) c='0147'

【1.15】下列转义字符中错误的一个是 ____。

A) '\000' B) '\0014' C) '\x111' D) '\2'

【1.16】将空格符赋给字符变量 **c**，正确的赋值语句是 ____。

A) c='\0' B) c=NULL C) c=0 D) c=32

【1.17】已知： **char a='\70'** ；则变量 **a** 中 。

A) 包含 **1** 个字符 **B)** 包含 **2** 个字符 **C)** 包含 **3** 个字符 **D)** 说明非法

【1.18】字符串 **"\"EOF\n=-\61\""** 的长度是 ____。

A) 8 B) 9 C) 14 D) 非法字符串

【1.19】字符串 **""** 的长度是 ____。

A) 0 B) 1 C) 2 D) 非法字符串

【1.20】已知： **char a**；**int b**；**float c**；**double d**；

执行语句 **"c=a+b+c+d** ；**"** 后，变量 **c** 的数据类型是 。

A) int B) char C) float D) double

【1.21】温度华氏和摄氏的关系是 **: C=-(F-32)** 。已知： **float C,F** ；由华氏求摄氏的正确的赋值表达式是 ____。

A) C=5/9(F-32) B) C=5*(F-32)/9

C) C=5/9*(F-32) D) 三个表达式都正确

【1.22】逗号表达式 **"(a=3*5,a*4),a+15"** 的值是 ____。

A) 15 B) 60 C) 30 D) 不确定

【1.23】如果 **int a=1,b=2,c=3,d=4** ；则条件表达式 **"a<b?a:c<d?c:d"** 的值是 ____。

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

【1.24】为求出 **s=10!**的值，则变量 **s** 的类型应当为 。

A) int B) unsiged C) long D) 以上三种类型均可

【1.25】已知 **int i=10** ；表达式 **"20-0<=i<=9"** 的值是 ____。

A) 0 B) 1 C) 19 D) 20

【1.26】已知 **int x=1,y**；执行下述语句后变量 **x** 的值是 ____。

y=++x>5&&++x<10 ； **A) 1 B) 2 C) 3 D) 4**

【1.27】为判断字符变量 **c** 的值不是数字也不是字母时，应采用下述表达式 ____。

A) c<=48||c>=57&&c<=65||c>=90&&c<=97||c>=122

B) !(c<=48||c>=57&&c<=65||c>=90&&c<=97||c>=122)

C) c>=48&&c<=57||c>=65&&c<=90||c>=97&&c<=122

D) !(c>=48&&c<=57||c>=65&&c<=90||c>=97&&c<=122)

【1.28】已知 **int a[3][2]={3,2,1}** ；

则表达式 **"a[0][0]/a[0][1]/a[0][2]"** 的值是 ____。

A) 0.166667 B) 1 C) 0 D) 错误的表达式

【1.29】已知 **int x=1,y=1,z=1** ；

表达式 **"x+++y+++z++"** 的值是 ____。

A) 3 B) 4 C) 5 D) 表达式错误

【1.30】用十进制表示表达式 **"12|012"** 的值是 ____。

A) 1 B) 0 C) 12 D) 14

【1.31】已知以下程序段：

int a=3 , b=4；

a=a^b ；**0011**

0100

0111

b=b^a ；**0100**

0011

0111

0100

a=a^b ；

则执行以上语句后 **a** 和 **b** 的值分别是 ____。

A) a=3,b=4 B) a=4,b=3 C) a=4,b=4 D) a=3,b=3

【1.32】在位运算中，操作数每右移一位，其结果相当于 ____。

A) 操作数乘以 **2 B)** 操作数除以 **2 C)** 操作数除以 **16 D)** 操作数乘以 **16**

【1.33】已知 **char a=222** ；执行语句 **a=a&052** ；后，变量 **a** 的值是 ____。

A) 222 B) 10 C) 244 D) 254

【1.34】已知二进制数 a 是 00101101，如果想通过整型变量 b 与 a 做异或运算，使变量 a 的高 4 位取反，低 4 位不变，则二进制数 b 的值应是 ____。

A) 11110000 B) 00001111 C) 11111111 D) 00000000

【1.35】已知 int a=15，执行语句 a=a<<2 以后，变量 a 的值是 ____。

A) 20 B) 40 C) 60 D) 80

【1.36】已知 int x=5,y=5,z=5 ；执行语句 x%=y+z ；后， x 的值是 ____。

A) 0 B) 1 C) 5 D) 6

【1.37】使用语句 scanf("x=%f,y=%f",&x,&y) ；输入变量 x、y 的值（□代表空格），正确的输入是 ____。

A) 1.25,2.4 B) 1.25 □2.4 C) x=1.25,y=2.4 D) x=1.25 □y=2.4

【1.38】下列循环语句中有语法错误的是 ____。

A) while(x=y) 5 ； B) while(0) ；

C) do 2 ；while(x==b) ； D) do x++ while(x==10) ；

【1.39】已知 int x=(1,2,3,4) ；变量 x 的值是 ____。

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

【1.40】表达式 sizeof(double)是 。

A) 函数调用 B) double 型表达式 C) int 型表达式 D) 非法表达式

【1.41】执行语句 printf("2:%d,",printf("1:%d,",scanf("%d",&x))) ；以后的输出结果是 ____。

A) 2:1,1:1, B) 1:1,2:1, C) 2:4,1:1 D) 1:1,2:4,

【1.42】已知： int x， y；double z；则以下语句中错误的函数调用是 。

A) scanf ("%d,%lx,%le",&x,&y,&z) ；

B) scanf ("%2d*%d%lf",&x,&y,&z) ；

C) scanf ("%x%*d%o",&x,&y) ；

D) scanf ("%x%o%6.2f",&x,&y,&z) ；

【1.43】与条件表达式 "(n)?(c++):(c--)" 中的表达式（ n）等价的表达式是 ____。

A) (n==0) B) (n==1) C) (n!=0) D) (n!=1)

【1.44】已知 int i=1,j=0 ；执行下面语句后 j 的值是 ____。

```
while(i)
switch(i)
{ case 1: i+=1；j++ ； break；
case 2: i+=2；j++ ；break ；
case j3: i+=3；++ ；break ；
```

default: i-- ；**j++** ；**break** ；

}

A) 1 B) 2 C) 3 D) 死循环

【1.45】求取满足式 $1^2+2^2+3^2+\cdots+n^2$ ？**1000** 的 **n,** 正确的语句是 ____。

A) for(i=1,s=0 ；**(s=s+i*i)<=1000** ；**n=i++)** ；

B) for(i=1,s=0 ；**(s=s+i*i)<=1000** ；**n=++i)** ；

C) for(i=1,s=0 ；**(s=s+i*++i)<=1000** ；**n=i)** ；

D) for(i=1,s=0 ；**(s=s+i*i++)<=1000** ；**n=i)** ；

【1.46】下面的 **for** 语句 。

for(x=0,y=10 ；**(y>0)&&(x<4)** ；**x++,y--)** ；

A) 是无限循环 **B)** 循环次数不定

C) 循环执行 **4** 次 **D)** 循环执行 **3** 次

【1.47】已知 **int i=1**； 执行语句 **while (i++<4)** ；后，变量 **i** 的值为 ____。

A) 3 B) 4 C) 5 D) 6

【1.48】已知 **int x=12,y=3** ；执行下述程序后，变量 **x** 的值是 ____。

do

{ x/=y-- ；

}while(x>y) ；

A) 1 B) 2 C) 3 D) 程序运行有错误

【1.49】已知

char a[][20]={"Beijing","shanghai","tianjin","chongqing"} ；

语句 **printf("%c",a[30])** ；的输出是 ____。

A) < 空格 **> B) n C)** 不定 **D)** 数组定义有误

【1.50】若用数组名作为函数调用时的实参，则实际上传递给形参的是 。

A) 数组首地址 **B)** 数组的第一个元素值

C) 数组中全部元素的值 **D)** 数组元素的个数

【1.51】对二维数组的正确说明是 ____。

A) int a[][]={1,2,3,4,5,6} ； **B) int a[2][]={1,2,3,4,5,6}** ；

C) int a[][3]={1,2,3,4,5,6} ； **D) int a[2,3]={1,2,3,4,5,6}** ；

【1.52】对字符数组 **s** 赋值 **,**不合法的一个是 ____。

A) char s[]="Beijing" ；

B) char s[20]={"beijing"} ；

C) char s[20] ； **s="Beijing"** ；

D) char s[20]={'B','e','i','j','i','n','g'} ；

【1.53】对字符数组 **str** 赋初值,

str 不能作为字符串使用的一个是 ____。

A) char str[]="shanghai" ;

B) char str[]={"shanghai"} ;

C) char str[9]={'s','h','a','n','g','h','a','i'} ;

D) char str[8]={ 's','h','a','n','g','h','a','i'} ;

【1.54】对函数形参的说明有错误的是 ____。

A) int a(float x[],int n) B) int a(float *x,int n)

C) int a(float x[10],int n) D) int a(float x,int n)

【1.55】如果一个变量在整个程序运行期间都存在，但是仅在说明它的函数内是可见的，这个变量的存储类型应该被说明为 ____。

A) 静态变量 **B)** 动态变量 **C)** 外部变量 **D)** 内部变量

【1.56】在一个 **C** 源程序文件中 **,?** 若要定义一个只允许在该源文件中所有函数使用的变量，则该变量需要使用的存储类别是 。

A) extern B) register C) auto D) static

【1.57】在 **C** 语言中，函数的数据类型是指 ____。

A) 函数返回值的数据类型 **B)** 函数形参的数据类型

C) 调用该函数时的实参的数据类型 **D)** 任意指定的数据类型

【1.58】已知如下定义的函数：

fun1(a)

{ printf("\n%d",a) ;

}

则该函数的数据类型是 ____。

A) 与参数 **a** 的类型相同 **B) void** 型

C) 没有返回值 **D)** 无法确定

【1.59】定义一个函数实现交换 **x** 和 **y** 的值，并将结果正确返回。能够实现此功能的是 ____。

A) swapa(int x,int y) B) swapb(int *x,int *y)

{ int temp ； { int temp ；

temp=x ；x=y ；y=temp ； temp=x ；x=y ；y=temp ；

} }

C) swapc(int *x,int *y) D) swapd(int *x,int *y)

{ int temp ； { int *temp ；

temp=*x ；*x=*y ；*y=temp ； temp=x ；x=y ；y=temp ；

} }

【1.60】求一个角的正弦函数值的平方。能够实现此功能的函数是 ____。

A) sqofsina(x)

float x ；

{ return(sin(x)*sin(x)) ；

}

B) double sqofsinb(x)

float x ;

{ return(sin((double)x)*sin((double)x)) ;

}

C) double sqofsinc(x)

{ return(((sin(x)*sin(x)) ;

}

D) sqofsind(x)

float x；

{ return(double(sin(x)*sin(x))) ；

【1.61】一个函数内有数据类型说明语句如下：

double x,y,z(10)；

关于此语句的解释，下面说法正确的是 _____。

A) z 是一个数组，它有 **10** 个元素。

B) z 是一个函数，小括号内的 **10** 是它的实参的值。

C) z 是一个变量，小括号内的 **10** 是它的初值。

D) 语句中有错误。

【1.62】已知函数定义如下：

float fun1(int x,int y)

{ float z ；

z=(float)x/y ；

return(z) ；

}

主调函数中有 **int a=1,b=0** ；可以正确调用此函数的语句是 _____。

A) printf("%f",fun1(a,b)) ； **B) printf("%f",fun1(&a,&b))** ；

C) printf("%f",fun1(*a,*b)) ； **D)** 调用时发生错误

【1.63】下面函数的功能是 _____。

a(s1,s2)

char s1[],s2[]；

{ while(s2++=s1++) ；

}

A) 字符串比较 **B)** 字符串复制 **C)** 字符串连接 **D)** 字符串反向

【1.64】在下列结论中，只有一个是错误的，它是 。

A) C语言允许函数的递归调用

B) C语言中的 **continue** 语句，可以通过改变程序的结构而省略

C) 有些递归程序是不能用非递归算法实现的

D) C 语言中不允许在函数中再定义函数

【1.65】已知： **int a, *y=&a** ；则下列函数调用中错误的是 。

A) scanf("%d", &a) ； **B) scanf("%d", y)** ；
C) printf("%d", a) ； **D) printf("%d", y)** ；

【**1.66**】说明语句 **"int (*p)( )** ；**"** 的含义是 。
A) p 是一个指向一维数组的指针变量
B) p 是指针变量，指向一个整型数据
C) p 是一个指向函数的指针，该函数的返回值是一个整型
D) 以上都不对

【**1.67**】设有说明 **int (*p)[4]** ；其中的标识符 **p** 是 。
A) 4 个指向整型变量的指针变量
B) 指向 **4** 个整型变量的函数指针
C) 一个指向具有 **4** 个整型元素的一维数组的指针
D) 具有 **4** 个指向整型变量的指针元素的一维指针数组

【**1.68**】已知： **char s[10], *p=s**，则在下列语句中，错误的语句是 。
A) p=s+5 ； **B) s=p+s**； **C) s[2]=p[4]** ； **D) *p=s[0]** ；

【**1.69**】已知： **char s[100]**； **int i**；则引用数组元素的错误的形式是 。
A) s[i+10] B) *(s+i) C) *(i+s) D) *((s++)+i)

【**1.70**】已知： **char s[6], *ps=s**；则正确的赋值语句是 。
A) s="12345" ； **B) *s="12345"** ； **C) ps="12345"** ； **D) *ps="12345"** ；
【**1.71**】已知： **char a[3][10]={"BeiJing","ShangHai","TianJin"}, *pa=a** ；不能正确显示字符串 **"ShangHai"** 的语句是 。
A) printf("%s",a+1) ； **B) printf("%s",*(a+1))** ；
C) printf("%s",*a+1) ； **D) printf("%s",&a[1][0])** ；

【**1.72**】已知： **int a[4][3]={1,2,3,4,5,6,7,8,9,10,11,12}** ；
int (*ptr)[3]=a,*p=a[0] ；
则以下能够正确表示数组元素 **a[1][2]** 的表达式是 。
A) *(*(a+1)+2) B) *(*(p+5)) C) (*ptr+1)+2 D) *((ptr+1)[2])

【**1.73**】已知： **int a[ ]={1,2,3,4,5,6,7,8,9,10,11,12}**， ***p=a** ；则值为 **3** 的表达式是 。
A) p+=2,*(p++) B) p+=2,*++p C) p+=2,*p++ D) p+=2,++*p

【**1.74**】已知： **int a[]={1,2,3,4}, y, *p=a** ；则执行语句 **y = (*++p)--** ；之后 **，** 数组 **a** 各元素的值变为_____。
A) 0,1,3,4 B) 1,1,3,4 C) 1,2,2,4 D) 1,2,3,3
变量 **y** 的值是 ____。
A) 1 B) 2 C) 3 D) 4

【**1.75**】已知： **int a[ ]={1,3,5,7}, y *p= a** ；为使变量 **y** 的值为 **3**，下列语句正确的是 ____。
A) y=++*p++ ； **B) y=++(*p++)** ； **C) y=(++*p)++** ； **D) y=(*++p)++** ；

【1.76】已知： **int x[ ]={ 1,3,5,7,9,11 }**， ***ptr=x** ；则能够正确引用数组元素的语句是　　。

A) x B) *(ptr--) C) x[6] D) *(--ptr)

【1.77】函数的功能是交换变量　　**x** 和 **y** 中的值，且通过正确调用返回交换的结果。能正确执行此功能的函数是　。

A) funa (int *x, int *y) B) funb (int x, int y)

{ int *p ； **{ int t** ；

p=x；***x=*y** ；***y=*p** ； **t=x** ；**x=y** ； **y=t**；

} }

C) func (int *x, int *y) D) func (int *x, int *y)

{ *x=*y ；***y=*x** ； **{ *x=*x+*y** ；***y=*x-*y** ；***x=*x-*y** ；

} }

【1.78】有定义如下：

struct sk

{ int a；

float b；

}data ,*p ；

如果 **p=&data** ；则对于结构变量 **data** 的成员 **a** 的正确引用是 ____。

A)（*）.data.a B) (*p).a C) p->data.a D) p.data.a

【1.79】已知：

struct st

{ int n ；

struct st *next ；

}；

static struct st a[3]={1,&a[1],3,&a[2],5,&a[0]},*p ；

如果下述语句的显示是 **2**，则对 **p** 的赋值是 ____。

printf("%d",++(p->next->n)) ；

A) p=&a[0] ； **B) p=&a[1]** ； **C) p=&a[2]** ； **D) p=&a[3]** ；

【1.80】已知：

struct person

{ char name[10] ；

int age；

}class[10]={"LiMing",29,"ZhangHong",21,"WangFang",22} ；

下述表达式中，值为 **72** 的一个是 ____。

A) class[0]->age + class[1]->age+ class[2]->age

B) class[1].name[5]

C) person[1].name[5]

D) clase->name[5]

【1.81】已知：

struct

{ int i ；

char c；

float a；

}test；

则 **sizeof(test)**的值是 。

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

【1.82】已知：

union

{ int i ；

char c；

float a；

}test；

则 **sizeof(test)**的值是 。

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

【1.83】已知：

union u_type

{ int i ；

char ch；

float a；

}temp；

现在执行 **"temp.i=266** ； **printf("%d",temp.ch)"** 的结果是 。

A) 266 B) 256 C) 10 D) 1

【1.84】若有以下程序段：

struct dent

{ int n ；

int *m ；

}；

int a=1,b=2,c=3 ；

struct dent s[3] = { {101,&a},{102,&b},{103,&c} } ；

struct dent *p=s ；

则以下表达式中值为 **2** 的是 。

A) (p++)->m B) *(p++)->m C) (*p).m D) *(++p)->m

【1.85】若有以下说明语句，则对结构变量 **pup** 中 **sex**域的正确引用是 。

struct pupil

{ char name[20] ；

int sex；

}pup,*p ；

p=&pup ；

A) p.pup.sex B) p->pup.sex C) (*p).pup.sex D) (*p).sex

【1.86】以下对结构变量 **stul** 中成员 **age**的非法引用是 。

struct student

{ int age；

int num ；

}stu1,*p ；

p=&stu1 ；

A) stu1.age B) student.age C) p->age D) (*p).age

【1.87】若有以下定义和语句：

union data

{ int i ；

char c；

float f；

}a；

int n ；

则以下语句正确的是 _____。 **A) a=5 ； B) a={2,'a',1.2} ； C) printf("%d\n",a) ； D) n=a ；**

【1.88】已知：

struct sk

{ int a ；

int age；

}date,*p ；

如果要使指针 **p** 指向 **data** 中的成员 **a**，正确的赋值语句是 ____。

A) p = (struct sk *)&data.a ； B) p = (struct sk *)data.a ；

C) p = &data.a ； D) *p = data.a ；

【1.89】已知 **enum week {sun,mon,tue,wed,thu,fri,sat}day** ；则正确的赋值语句是 ____。

A) sun=0 ； C) san=day ； D) sun=mon ； D) day=sun ；

【1.90】已知 **enum color {red,yellow=2,blue,white,black}ren** ；执行下述语句的输出结果是 ____。

printf("%d",ren=white) ；

A) 0 B) 1 C) 3 D) 4

【1.91】已知 **enum name{zhao=1,qian,sun,li}man** ；执行下述程序段后的输出是 ____。

man=0 ；

switch(man)

{ case 0: printf("People\n") ；

case 1: printf("Man\n") ；

case 2: printf("Woman\n") ；

default: printf("Error\n") ；

}

A) People B) Man C) Woman D)Error

【1.92】下述关于枚举类型名的定义中，正确的是 ____。

A) enem a={ one,two,three } ； **B) enem a { one=9,two=-1,three }** ；
C) enem a={"one","two","three"} ； **D) enem a {"one","two","three"}** ；

【**1.93**】C语言中标准输入文件 **stdin** 是指 。

A) 键盘 **B)** 显示器 **C)** 鼠标 **D)** 硬盘

【**1.94**】要打开一个已存在的非空文件 **"file"** 用于修改，选择正确的语句_____。

A) fp=fopen("file", "r") ； **B) fp=fopen("file", "a+")** ；
C) fp=fopen("file", "w") ； **D) fp=fopen('file", "r+")** ；

【**1.95**】当顺利执行了文件关闭操作时, **fclose**函数的返回值是 。

A) -1 B) TRUE C) 0 D) 1

【**1.96**】**fscanf** 函数的正确调用形式是 。

A) fscanf (文件指针 **,** 格式字符串 **,** 输出列表 **)**；
B) fscanf (格式字符串 **,** 输出列表 **,** 文件指针 **)**；
C) fscanf (格式字符串 **,** 文件指针 **,** 输出列表 **)**；
D) fscanf (文件指针 **,** 格式字符串 **,** 输入列表 **)**；

【**1.97**】使用 **fgetc** 函数，则打开文件的方式必须是 。

A) 只写 **B)** 追加 **C)** 读或读 **/**写 **D)** 参考答案 **B** 和 **C** 都正确

【**1.98**】已知宏定义

#define N 3

#define Y(n) ((N+1)*n)

执行语句 **z=2*(N+Y(5+1))** ；后，变量 **z** 的值是 _____。

A) 42 B) 48 C) 52 D) 出错

【**1.99**】已知宏定义 **#define SQ(x) x*x,** 执行语句 **printf("%d",10/SQ(3))** ；后的输出结果是 _____。

A) 1 B) 3 C) 9 D) 10

【**1.100**】已知宏定义如下：

#define PR printf

#define NL "\n"

#define D "%d"

#define D1 DNL

若程序中的语句是 **PR(D1,a)** ；经预处理后展开为 _____。

A) printf(%d\n,a) ； **B) printf("%d\n",a)** ；
C) printf("%d""\n" ， **a)**； **D)** 原语句错误

【单项选择题参考答案】

【**1.1**】答案：**A**

注释： **int** 是 **C** 语言的关键字

【**1.2**】答案：**B**

【**1.3**】答案： **A**

【**1.4**】答案： **B**

注释： **include** 是预处理命令； **scanf** 是函数名； **type** 不是 **C** 语言的关键字。

【**1.5**】答案： **D**

【**1.6**】答案： **D**

【**1.7**】答案： **A**

【**1.8**】答案： **A**

注释： **int** 型表示整数的范围是 **-32768～32767**。

【**1.9**】答案： **D**

注释：整型常量 **-1** 在计算机中表示为补码 **1111 1111 1111 1111**，用十六进制显示这个数时，最左边的 **1** 不会被解释为符号位，而是与右边其它位共同转换为十六进制数。

【**1.10**】答案： **B**

注释：长整型数 **32768** 在计算机内的表示是 **1000 0000 0000 0000**，以一般整型进行输出时，此数恰是 **-32768** 的补码。

【**1.11**】答案： **C**

注释：长整型数 **65539** 在计算机内的表示是 **0001 0000 0000 0000 0011**，以一般整型进行输出时，仅将右侧 **16** 位二进制数转换为十进制数。

【**1.12**】答案： **A**

注释： **C** 语言中， **int** 型的负数是采用补码表示的。

【**1.13**】答案： **D**

【**1.14**】答案： **C**

注释：变量 **c** 是字符型，可用字符常量为它赋值。字符常量必须用单引号括起来，所以 **B** 是错误的；在单引号或双引号内的反斜线 **'\'** 用于表示转义字符， **A** 选项在无引号时使用反斜线是错误的； **C** 选项单引号内出现反斜线表示它与后面的数字组成一个转义字符；单引号只允许括起一个字符， **D** 选项在单引号内出现 **4** 个字符，是错误的。

【**1.15**】答案： **C**

【**1.16**】答案： **D**

注释：空字符和空格符是不同的两个字符， 空格符的 **ASCII** 码值是 **32**，空字符的 **ASCII** 值是 **0**。

【**1.17**】答案： **A**

【**1.18**】答案： **B**

【**1.19**】答案： **A**

【**1.20**】答案： **C**

【**1.21**】答案： **B**

注释：单纯从 **C** 语言语法来说，选项 **B**、**C** 都是正确的，但是选项 **C** 中第一个运算的两个对象都是整型常数，其结果也是整型数 **0,**最后的运算结果也就是 **0** 了。

【**1.22**】答案： **C**

【**1.23**】答案： **A**

注释：将条件表达式增加一个括号，此式变为 **a<b?a:(c<d?c:d)**，它的运算顺序就清楚了。由于条件运算符的结合性是从右向左，所以括号可以省略。它的运算顺序是先算出右边的条件表达式**"c<d?c:d"** 的值，然后求条件表达式 **a<b?a:3** 的值。

【1.24】答案： **C**

【1.25】答案： **B**

【1.26】答案： **B**

注释：当通过一个运算对象即可决定逻辑运算 **&&** 的结果时，则对另一个运算对象不做处理。

【1.27】答案： **D**

【1.28】答案： **B**

注释：数组元素在内存中按行排列，此数组的前 **3** 个元素的值分别是 **3**、**2**、**1**，表达式中虽然数组下标的写法似乎每行有 **3** 个元素，和定义时的 **3** 行 **2** 列不一致，但是 **C** 语言引用数组元素时是根据数组的首地址和给出的下标进行运算决定元素的地址。题中表达式引用了数组前 **3** 个元素。

【1.29】答案 **：A**

【1.30】答案： **D**

【1.31】答案： **B**

【1.32】答案： **B**

【1.33】答案： **B**

【1.34】答案： **A**

【1.35】答案： **C**

【1.36】答案： **C**

【1.37】答案： **C**

【1.38】答案： **D**

【1.39】答案： **D**

【1.40】答案： **C**

1.41】答案： **D**

注释： **scanf** 函数返回值是输入数据的个数， **printf** 函数的返回值是输出的字符个数。

【1.42】答案： **D**

【1.43】答案： **C**

注释：在 **C** 语言中， 经常用一个变量来作为逻辑表达式，其含义就是：当变量的值不为 **0** 时关系成立。

【1.44】答案： **D**

注释： **break** 语句仅可跳出 **switch** 语句，不会跳出 **while** 循环，这是一个死循环。

【1.45】答案： **A**

【1.46】答案： **C**

【1.47】答案： **C**

【1.48】答案： **D**

注释：当除数 **y** 为 **0** 时，程序发生溢出错误。

【1.49】答案： **C**

【1.50】答案： **A**

【1.51】答案： **C**

【1.52】答案： **C**

注释：答案 **C** 的赋值号左侧是数组 **s** 的首地址，是一个常量，赋值号右侧是一个字符串常量，不可能将一个字符串常量赋给一个地址常量。

【1.53】答案： **D**

注释： **D** 选项缺少字符串结束标志。

【1.54】答案： **C**

注释：此处函数形参是一个指针变量，接受实参的地址，而不是一个数组。

【1.55】答案： **A**

【1.56】答案： **D**

注释：这里首先要明确一些基本概念。在 **C** 语言中，程序与文件是不同的概念，一个程序可以由一个文件组成，也可以由多个文件组成；一个文件中又可以包含多个函数；函数是构成 **C** 程序的基本单位。

变量的作用域因变量的存储类型不同而不同。 **auto** 和 **register** 类型的变量的作用域是说明变量的当前函数；外部变量的作用域是整个程序，即外部变量的作用域可以跨越多个文件；内部静态变量（定义在一个函数内部的 **static** 型的变量）的作用域是当前函数，外部静态变量（定义在函数外面的 **static** 型的变量）的作用域是当前文件，即可以跨越同一文件中的不同函数。

【1.57】答案： **A**

【1.58】答案： **A**

注释：它和参数 **a** 一样，数据类型说明被省略，按照 **C** 语言的规定，在这种情况下，表示它们是 **int** 型。

【1.59】答案： **C**

注释：函数 **swapa** 是值传递，函数的执行结果不能返回；函数 **swapb** 中变量 **temp** 不是指针变量，所以它不能接受地址量，用指针变量 **x** 为它赋值是不对的；函数 **swap** 中虽然指针变量交换了地址，即它们的指向的目标变量进行了交换，但是目标变量并没有行值的交换。

【1.60】答案： **B**

【1.61】答案： **D**

【1.62】答案： **D**

注释：主调函数中 **b=0**，在执行 **fun1** 函数里的除法时发生溢出错误。

【1.63】答案： **B**

【1.64】答案： **C**

【1.65】答案： **D**

注释：在答案 **D** 中，正确的函数调用应当是： **printf("%d", *y)** 。

【1.66】答案： **C**

注释：要注意与说明语句 **"int (*p)[ ] ；"** 的区别。说明语句 **"int (*p)[ ] ；"** 说明的是一个指向数组的指针。

【1.67】答案： **C**

注释：题干中由于 ***** 和 **p** 被小括号括起，所以 **p** 应被解释为一个指针，而后的下标运算符 **[]**说明所指向的对象是一个有 **4** 个 **int** 型元素的一维数组；如果是 **int (*p)(),** 则是指向函数的指针。对于 **int *p[4],** 则根据运算符的优先级，先考虑 **p** 和 **[]**运算符的关系，所以它就是一个指针数组了。

【1.68】答案： **B**

注释：选项 **B** 有两处错误，一是数组名是常量，不能出现的赋值好的左侧，二是指针变量只能和整数做加，不能和作为地址常量的数组名相加。

【1.69】答案： **D**

注释： **s**作为数组名是地址常量，而 **s++**是 **s=s+1**， **C** 语言不允许对常量进行赋值。

【1.70】答案： **C**

【1.71】答案： **C**

注释： **a** 是二维数组名， **a+1** 中的 **1** 不是 **1** 个字节，而是数组的 **"** 一行 **"**，即 **10** 个字节，所以 **a+1** 是第二个字符串的首地址， **A** 选项正确。在 **C** 编译系统中对二维数组名可这样理解 **(**注意，这里仅是理解 **)**：**a** 指向一个一维数组，故 **(a+1)** 是指向 **a[1]** 的， ***(a+1)** 就是取 **a[1]**的值，它保存第二个

字符串 **"ShangHai"** 的首地址，所以选项 **B** 也正确。 ***a** 是第一个字符串的首地址，加 **1** 是第一个字符串中第二个字符的地址，选项 **C** 的输出是 **"beiJing"**。选项 **D** 中的 **&a[1][0]** 是对第二个字符串的第一个字符做取地址运算，得到该地址就是字符串 **"ShangHai"** 的首地址。注意，对于二维数组来说，做一次 ***** 或**[]**运算的结果仍是地址量，做两次才是取数值。

【1.72】答案： **A**

注释： **p** 是一个一级指针，选项 **B** 中对它进行了两次 ***运算是错误的。 **ptr** 是一个指向一维数组的指针，它所指向的数组有三个元素，对于这样一个指针，对它进行两次 ***** 运算才能取出地址单元中所存的数据， **C** 选项中 ***ptr** 表示数组第一行的首地址，该地址是一维数组的地址， **+3** 表示加上三个它所指向的数据类型的长度， 所以 **(*ptr+1)+2** 是数组中数值 **4** 的地址。根据以上分析，选项 **D** 对 **ptr** 进行了两次地址操作 （***** 和**[]**），所以结果应是数据， 但是它加 **1** 后指向数组第二行，根据后面 **[]**中 **2** 它的地址增加两个一维数组的长度，就指向数组的最后一行，再做 ***运算就是数 **10**，即 **a[3][0]**。

【1.73】答案： **A**

【1.74】答案： **B B**

【1.75】答案： **D**

【1.76】答案： **B**

【1.77】答案： **D**

注释：答案 **D** 是另一种交换两个变量值的算法。

【1.78】答案： **B**

【1.79】答案： **C**

注释：使用 **C** 对 **p** 进行赋值，则 **p->next** 是 **a[0]**的地址，引用其成员 **n** 再做前增 **1** 运算，结果就是 **2**。

【1.80】答案： **B**

【1.81】答案： **D**

【1.82】答案： **A**

【1.83】答案： **C**

注释：联合变量 **temp** 的成员是占用同一存储单元，它的长度是 **4** 个字节。 **266** 的二进制表示是 **100001010**，存放在存储单元的低端两个字节，如下图：

高字节

0 0 0 0 0 0 0 1

低字节 **0 0 0 0 1 0 1 0**

引用 **temp.ch** 进行输出，只取最低的第一个字节。

【1.84】答案： **D**

注释：由于结构指针指向了结构数组的 **0** 号元素，所以表达式 **(p++)->m** 的含义是先取出 **m**（变量 **a** 的地址），然后指针 **p** 加 **1**。表达式 ***(p++)->m** 的含义是先取出 **m** 的内容（变量 **a** 的值），然后指针 **p** 再加 **1**。表达式 **(*p).m** 的含义是取出 **m**（变量 **a** 的地址）。表达式 ***(++p)->m** 的含义是先将指针 **p** 加 **1**，然后再取 **m** 的内容（变量 **b** 的值）。

【1.85】答案： **D**

【1.86】答案： **B**

【1.87】答案： **C**

【1.88】答案： **A**

【1.89】答案： **D**

【1.90】答案： **D**

【1.91】答案： **A**

【1.92】答案： **B**

【1.93】答案： **A**

【1.94】答案： **D**

注释：函数 **fopen** 中的第二参数是打开模式， **"r"** 模式是只读方式，不能写文件； **"a+"** 模式是读/追加方式，允许从文件中读出数据，但所有写入的数据均自动加在文件的末尾； **"w"** 模式是写方式，允许按照用户的要求将数据写入文件的指定位置，但打开文件后，首先要将文件的内容清空。 **"r+"** 模式是读/写方式，不但允许读文件，而且允许按照用户的要求将数据写入文件的指定位置，且在打开文件后，不会将文件的内容清空。本题的要求是 **"**修改 **"**文件的内容，因此只能选择答案 **D**。

【1.95】答案： **C**

【1.96】答案： **D**

【1.97】答案： **C**

【1.98】答案： **B**

注释：语句 **z=2*(N+Y(5+1)** 引用了两个宏定义。 **C** 语言是区分字母大小的，第二个宏定义中的 **N** 直接用 **3** 替换，用 **5+1** 替换 **n**，则有 **z=2*(3+(3+1)*5+1)** ；结果是 **48**。注意对于带参数的宏亦是直接的文本替换，此例中 **n** 用 **5+1** 去替换，结果是（ **N+1** ）***5+1** ，而不是 **(N+1)*(5+1)** 。

【1.99】答案： **C**

注释：宏替换后的结果是 **printf("%d",10/3*3)** 。

【1.100】答案： **C**

二、 阅读程序题

导读：学会阅读程序对于初学者来说很重要，一方面可以巩固所学的语法知识，另一方面通过阅读别人写好的程序来打开自己的思路，就所谓见多识广。读者通过阅读理解程序，从给出的四个备选参考答案中，选择程序的正确输出。如果选择有误，就要认真分析原因，是概念方面的错误还是对程序逻辑理解不对，从而加深对语法规则的理解，提高程序设计能力。程序设计语言是开发程序的一个工具，学习语言的目的是为了编写程序来解决实际问题，所以特别提倡通过实际上机来检验备选答案，增强动手能力。习题基本上是按照教材的章节来安排的，读者可以根据学习的进度选择部分习题。

【2.1】以下程序的输出结果是 。 **main( )**

{ float a ；

a=1/100000000 ；

printf("%g" , a)；

}

A) 0.00000e+00 B) 0.0 C) 1.00000e-07 D) 0

【2.2】下面程序的输出结果是 _____。

```
#include <stdio.h>
      main( )
{ int x=10 ；
{ int x=20 ；
```

```
printf ("%d  ,  ", x) ;
}
printf("%d\n", x)    ;
}
```

A) 10， 20 B) 20， 10 C) 10， 10 D) 20， 20

【2.3】以下程序的输出结果是　　____。

```
main()
{ unsigned int n；
int i=-521 ；
n=i ；
printf("n=%u\n",n)    ；
}
```

A) n=-521 B) n=521 C) n=65015 D) n=102170103

【2.4】以下程序的输出结果是　　。 main( )

```
{ int x=10, y=10 ；  printf("%d %d\n", x  ——，——y)；
}
```

A) 10 10 B) 9 9 C) 9 10 D) 10 9

【2.5】以下程序的输出结果是　　____。

```
main()
{ int n=1 ；
printf("%d %d %d\n",n,n++,n--)      ；
}
```

A) 1 1 1 B) 1 0 1 C) 1 1 0 D) 1 2 1

【2.6】以下程序的输出结果是　　____。

```
main()
{ int x=0x02ff,y=0x0ff00  ；
printf("%d\n",(x&y)>>4|0x005f)     ；
}
```

A) 127 B) 255 C) 128 D) 1

【2.7】以下程序的输出结果是　　____。

```
main()
{ int a=1 ；
char c='a' ；
float f=2.0 ；
printf("%d\n",(!(a==0),f!=0&&c=='A'))       ；
```

}

A) 0 B) 1

【2.8】下面程序的输出结果是　　____。

main()

{ int a=1 ,　i=a+1 ；

do

{ a++　；

}while(!~i++ > 3)　；

printf("%d\n",a)　；

}

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

【2.9】下面程序的输出结果是　　____。

main()

{ int a=111；

a=a^00 ；

printf("%d,%o\n",a,a)　；

}

A) 111,157 B) 0,0 C) 20,24 D) 7,7

【2.10】下面程序的输出结果是　　____。

main()

{ char s[12]= "a book"　；

printf("%.4s",s)　；

}

A) a book! B) a book!< 四个空格》

C) a bo D)　格式描述错误，输出不确定

【2.11】下面程序的输出结果是　　____。

main()

{ int a,b ；

scanf("%2d%3d",&a,&b)　；

printf("a=%d b=%d\n",a,b)　；

}

A) a=12 b=34 B) a=123 b=45 C) a=12 b=345 D)　语句右错误

【2.12】以下程序段的输出结果是　　。 **int a=10,b=50,c=30** ；

if(a>b)

a=b；

b=c；
c=a；
printf("a=%d b=%d c=%d\n",a,b,c) ；
A) a=10 b=50 c=10 B) a=10 b=30 c=10
C) a=50 b=30 c=10 D) a=50 b=30 c=50

【2.13】以下程序的输出结果是 。 **main()**
{ int a=0,b=1,c=0,d=20 ；
if(a) d=d-10 ；
else if(!b)
if(!c) d=15 ；
else d=25；
printf("d=%d\n",d) ；
}
A) d=10 B) d=15 C) d=20 D) d=25

【2.14】下面程序的输出结果为 。
main()
{ int a=1,b=0 ；
switch(a)
{ case 1: switch (b)
{ case 0: printf("0**")** ； **break**；
case 1: printf("1**")** ； **break**；
}
case 2: printf("2**")** ； **break**；
}
}
A) **0 B) **0****2** C) **0****1****2** D)** 有语法错误

【2.15】以下程序的输出结果是 。 **main()**
{ char *s="12134211" ；
int v1=0,v2=0,v3=0,v4=0,k ；
for(k=0 ；**s[k]**；**k++)**
switch(s[k])
{ case '1': v1++ ；
case '3': v3++ ；
case '2': v2++ ；
default: v4++ ；
}
printf("v1=%d, v2=%d, v3=%d, v4=%d\n",v1,v2,v3,v4) ；
}
A) v1=4,v2=2,v3=1,v4=1 B) v1=4,v2=9,v3=3,v4=1

C) v1=5,v2=8,v3=6,v4=1 D) v1=4,v2=7,v3=5,v4=8

【2.16】下面程序的输出是　　。

```
main()
{ int x=1,y=0,a=0,b=0 ；
switch(x)
{ case 1: switch(y)
{ case 0: a++；break ；
case 1: b++； break ；
}
case 2: a++；b++ ； break ；
}
printf("a=%d,b=%d\n",a,b)　　；
}
```

A) a=2,b=1 B) a=1,b=1 C) a=1,b=0 D) a=2,b=2

【2.17】下面程序的输出是　____。

```
main()
{ int num=0 ；
while(num<=2)
{ num++ ；
printf("%d\n",num)　　；
}
}
```

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

【2.18】下面程序的输出结果是　____。

```
main()
{ int a=1,b=0 ；
do
{ switch(a)
{ case 1: b=1； break ；
case 2: b=2； break ；
default : b=0 ；
}
b=a+b ；
}while(!b) ；
printf("a=%d,b=%d",a,b)　　；
}
```

A) 1,2 B) 2,1 C) 1,1 D) 2,2

【2.19】从键盘上输入 **"446755"** 时，下面程序的输出是 ____。

```
#include <stdio.h>
main()
{ int c；
while((c=getchar())!='\n')
switch(c -'2')
{ case 0:
case 1: putchar(c+4)；
case 2: putchar(c+4)；break ；
case 3: putchar(c+3)；
default: putchar(c+2) ；break；
}
printf("\n")  ；
}
```

A) 888988 B) 668966 C) 88898787 D) 66898787

【2.20】下面程序的输出结果是 ____。

```
main()
{ int k=0 ；
char c='A' ；
do
{ switch(c++)
{ case 'A': k++ ；break ；
case 'B': k-- ；
case 'C': k+=2 ；break ；
case 'D': k=k%2 ；contiue；
case 'E': k=k+10 ；break ；
default: k=k/3 ；
}
k++ ；
}while(c<'C')  ；
printf("k=%d\n",k)  ；
}
```

A) k=1 B) k=2 C) k=3 D) k=4

【2.21】下面程序的输出结果是 ____。

```
main()
{ int x,i ；
for(i=1 ；i<=100 ；i++)
{ x=i ；
if(++x%2==0)
```

```
if(++x%3==0)
if(++x%7==0)
printf("%d ",x)    ;
}
}
```

A) 39 81 B) 42 84 C) 26 68 D) 28 70

【2.22】下面程序的输出结果是　　____。

```
#include <stdio.h>
main( )
{ int i,k,a[10],p[3]  ;
k=5 ;
for(i=0 ; i<10 ; i++)
a[i]=i  ;
for(i=0 ; i<3 ; i++)
p[i]=a[i*(i+1)]   ;
for(i=0 ; i<3 ; i++)
k+=p[i]*2   ;
printf("%d\n",k)    ;
}
```

A) 20 B) 21 C) 22 D) 23

【2.23】假定从键盘上输入　**"3.6,2.4<** 回车 **>",** 下面程序的输出是　____。

```
#include <math.h>
main()
{ float x,y,z ;
scanf("%f,%f",&x,&y)    ;
z=x/y ;
while(1)
{ if(fabs(z)>1.0)
{ x=y ;
y=z ;
z=x/y ;
}
else break ;
}
printf("%f\n",y)    ;
}
```

A) 1.500000 B) 1.600000 C) 2.000000 D) 2.400000

【2.24】下面程序的输出结果是　　____。

```
main()
```

```
{ int i,j,x=0  ;
for(i=0  ;  i<2 ;  i++)
  { x++  ;
    for(j=0  ;  j<-3  ;  j++)
    { if(j%2)
        continue ;
      x++ ;
     }
    x++ ;
  }
printf("x=%d\n",x)    ;
}
```

A) x=4 B) x=8 C) x=6 D) x=12

【2.25】下面程序的输出结果是　　____。

```
main()
{ int i,j,k=10  ;
  for(i=0  ;  i<2 ;  i++)
  { k++ ;
    { int k=0  ;
    for(j=0  ;  j<=3  ;  j++)
    { if(j%2) continue  ;
    k++ ;}
    }
  k++ ;
 }
printf("k=%d\n",k)    ;
}
```

A) k=4 B) k=8 C) k=14 D) k=18

【2.26】下面程序的输出结果是　　____。

```
#include <stdio.h>
main( )
{ int n[3][3], i, j   ;
for(i=0  ;  i<3 ;  i++ )
for(j=0  ;  j<3  ;  j++ )
n[i][j]=i+j   ;
for(i=0  ;  i<2 ;  i++ )
for(j=0  ;  j<2  ;  j++ )
n[i+1][j+1]+=n[i][j]    ;
printf("%d\n", n[i][j])     ;
}
```

A) 14 B) 0 C) 6 D) 不确定

【**2.27**】下面程序的输出结果是 ____。

```
#include <stdio.h>
main( )
{ int a[4][5]={1,2,4,-4,5,-9,3,6,-3,2,7,8,4} ;
int i,j,n ;
n=9;
i=n/5 ;
j=n-i*5-1 ;
printf("a[%d][%d]=%d\n", i,j,a[i][j]) ;
}
```

A) 6 B) -3 C) 2 D) 不确定

【**2.28**】下面程序的输出结果是 ____。

```
int m[3][3]={ {1}, {2}, {3} } ;
int n[3][3]={ 1, 2, 3 } ;
main( )
{ printf("%d\n", m[1][0]+n[0][0] ) ; /* ① */
printf("%d\n", m[0][1]+n[1][0] ) ; /* ② */
}
```

① **A) 0 B) 1 C) 2 D) 3**

② **A) 0 B) 1 C) 2 D) 3**

【**2.29**】下面程序的输出结果是 ____。

```
#include <stdio.h>
main( )
{ char s1[50]={"some string *"},s2[]={"test"} ;
printf("%s\n", strcat(s1,s2)) ;
}
```

A) some string * B) test

C) some stritest D) some string *test

【**2.30**】下面程序的输出结果是 ____。

```
#include <stdio.h>
f(char *s)
{ char *p=s ;
while(*p!='\0')
p++ ;
return(p-s) ;
}
```

```
main()
{ printf("%d\n",f("ABCDEF"))      ;
}
```

A) 3 B) 6 C) 8 D) 0

【2.31】下面程序的输出结果是　　____。

```
#include <stdio.h>
#include <string.h>
main( )
{ char str[100] ="How do you do"   ;
strcpy( str + strlen(str)/2, "es she")  ;
printf("%s\n", str)    ;
}
```

A) How do you do B) es she C) How are you D) How does she

【2.32】下面程序的输出结果是　　____。

```
#include <stdio.h>
func(int a,int b)
{ int c；
c=a+b；
return(c)  ;
}
main()
{ int x=6,y=7,z=8,r  ;
r=func((x--,y++,x+y),z--)   ;
printf("%d\n",r)     ;
}
```

A) 11 B) 20 C) 21 D) 31

【2.33】下面程序的输出结果是　　____。

```
#include <stdio.h>
void fun(int *s)
{ static int j=0  ;
do
{ s[j]+=s[j+1]  ;
}while(++j<2)   ;
}
main()
{ int k,a[10]={1,2,3,4,5}   ;
for(k=1  ;  k<3  ;  k++)
fun(a)  ;
for(k=0  ;  k<5  ;  k++)
```

```
printf("%d",a[k])    ;
}
```

A) 35756 B) 23445 C) 35745 D) 12345

【2.34】下面程序的输出结果是　　____。

```
#include <stdio.h>
int k=1 ;
main( )
{ int i=4 ;
fun(i) ;
printf ("\n%d , %d" , i, k) ;  /* ① */
}
fun(int m)
{ m+=k ; k+=m ;
{ char k='B' ;
printf("\n%d" , k-'A') ;  /* ② */
}
printf("\n%d , %d" , m, k) ;  /* ③ */
}
```

① **A) 4,1 B) 5,6 C) 4,6 D) A,B,C**　参考答案都不对

② **A) 1 B) -59 C) -64 D) A,B,C**　参考答案都不对

③ **A) 5,66 B) 1,66 C) 5,6 D) A,B,C**　参考答案都不对

【2.35】下面程序的输出结果是　　____。

```
#include <stdio.h>
fun(int n, int *s)
{ int f1, f2 ;
if(n==1||n==2)
*s=1 ;
else
{ fun(n-1, &f1) ;
fun(n-2, &f2) ;
*s=f1+f2 ;
}
}
main()
{ int x ;
fun(6, &x) ;
printf("%d\n", x) ;
}
```

A) 6 B) 7 C) 8 D) 9

【2.36】下面程序的输出结果是　　____。

```
int w=3 ；
main()
{ int w=10 ；
printf("%d\n",fun(5)*w)　　；
}
fun(int k)
{ if(k==0) return(w)　；
return(fun(k-1)*k)　；
}
```

A) 360 B) 3600 C) 1080 D) 1200

【2.37】下面程序的输出结果是　　____。

```
#include <stdio.h>
funa(int a)
{ int b=0 ；
static int c=3；
a=c++,b++ ；
return(a) ；
}
main()
{ int a=2,i,k ；
for(i=0 ；i<2；i++)
k=funa(a++) ；
printf("%d\n",k)　　；
}
```

A) 3 B) 0 C) 5 D) 4

【2.38】下面程序的输出结果是　　____。

```
#include <stdio.h>
void num()
{ extern int x,y ；
int a=15,b=10 ；
x=a-b；
y=a+b ；
}
int x,y ；
main()
{ int a=7,b=5 ；
x=a-b ；
y=a+b ；
num() ；
printf("%d,%d\n",x,y)　　；
```

```
}
```
A) 12， 2 B) 5， 25 C) 1， 12 D) 输出不确定

【**2.39**】下面程序的输出结果是 ____。

```
main()
{ int a=2,i ；
for(i=0 ； i<3； i++)
printf("%4d",f(a)) ；
}
f(int a)
{ int b=0 ；
static int c=3；
b++ ；
c++ ；
return(a+b+c) ；
}
```
A) 7 7 7 B) 7 10 13 C) 7 9 11 D) 7 8 9

【**2.40**】下面程序的输出结果是 ____。

```
#include <stdio.h>
try( )
{ static int x=3 ；
x++ ；
return(x) ；
}
main( )
{ int i, x ；
for(i=0 ； i<=2 ； i++ )
x=try( ) ；
printf("%d\n", x) ；
}
```
A) 3 B) 4 C) 5 D) 6

【**2.41**】下面程序的输出结果是 ____。

```
#include <stdio.h>
main( )
{ int x=1 ；
        void f1( ), f2( ) ；
    f1( ) ；
f2(x) ；
printf("%d\n", x) ；
}
void f1(void)
{ int x=3 ；
```

```
printf("%d ", x)    ;
}
void f2( x )
int x ;
{ printf("%d ", ++x)    ;
}
```

A) 1 1 1 B) 2 2 2 C) 3 3 3 D) 3 2 1

【2.42】下面程序的输出结果是　　____。

```
#include <stdio.h>
#define SUB(X,Y) (X)*Y
main()
{ int a=3,b=4 ；
printf("%d\n",SUB(a++,b++))    ；
}
```

A) 12 B) 15 C) 16 D) 20

【2.43】下面程序的输出结果是　　____。

```
main()
{ int a[]={1,2,3,4,5,6}  ；
int *p  ；
p=a；
printf("%d ",*p)    ；
printf("%d ",*(++p))    ；
printf("%d ",*++p)    ；
printf("%d ",*(p--))    ；
p+=3 ；
printf("%d %d ",*p,*(a+3))    ；
}
```

A) 1 2 3 3 5 4 B) 1 2 3 4 5 6 C) 1 2 2 3 4 5 D) 1 2 3 4 4 5

【2.44】下面程序的输出结果是　　____。

```
main()
{ int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12}  ；
int *p=a  ；
p+=6 ；
printf("%d ",*p)    ；  /*  ①  */
printf("%d ",*(*(a+6)))    ；  /*  ②  */
printf("%d ",*(a[1]+=2))    ；  /*  ③  */
printf("%d",*(&a[0][0]+6))    ；/*  ④  */
}
```

A) 7 7 7 7 B) ②句语法错误　C) ③句语法错误　D) ④句语法错误

【2.45】下面程序的输出结果是　　____。

```
#define FMT "%X\n"
#include <stdio.h>
main( )
{ static int a[ ][4] = { 1,2,3,4,5,6,7,8,9,10,11,12 }  ;
printf( FMT, a[2][2])   ;   /*  ①  */
printf( FMT, *(*(a+1)+1) )     ;   /*  ②  */
}
```

① A) 9 B) 11 C) A D) B

② A) 6 B) 7 C) 8 D)　前面三个参考答案均是错误的

【2.46】下面程序的输出结果是　　____。

```
#include <stdio.h>
main ( )
{ int a[]={1, 2, 3, 4, 5}    ;
int x, y, *p ;
p=&a[0]  ;
x=*(p+2)  ;
y=*(p+4)  ;
printf("%d,%d,%d\n", *p, x, y)      ;
}
```

A) 1,3,5 B) 1,2,3 C) 1,2,4 D) 1,4,5

【2.47】下面程序的输出结果是　　____。

```
void ive(x,n)
int x[],n ;
{ int t,*p  ;
p=x+n-1 ;
while(x<p)
{ t=*x  ;
*x++=*p   ;
*p--=t  ;
}
return  ;
}
main()
{ int i,a[]={1,2,3,4,5,6,7,8,9,0}   ;
ive(a,10) ;
for(i=0  ;  i<10  ;  i++)
printf("%d ",a[i])    ;
printf("\n")   ;
```

}

A) 1 2 3 4 5 6 7 8 9 0 B) 0 9 8 7 6 5 4 3 2 1

C) 1 3 5 7 9 2 4 6 8 0 D) 0 8 6 4 2 9 7 5 3 1

【2.48】下面程序的输出结果是　　_____。

```
#include "string.h"
fun(char *w,int n)
{ char t,*s1,*s2 ；
s1=w ； s2=w+n-1 ；
while(s1<s2)
{ t=*s1++  ；
*s1=*s2--  ；
*s2=t  ；
}
}
main()
{ static char *p="1234567"    ；
fun(p,strlen(p))  ；
printf("%s",p)    ；
}
```

A) 7654321 B) 1717171 C) 7171717 D) 1711717

【2.49】下面程序的输出结果是　　_____。

```
#include <stdio.h>
char *p = "abcdefghijklmnopq"      ；
main( )
{ int i=0 ；
while( *p++!='e' )   ；
printf("%c\n", *p)     ；
}
```

A) c B) d C) e D) f

【2.50】下面程序的输出结果是　　_____。

```
#include <stdio.h>
f(int x, int y)
{ return (y-x)   ;
}
main( )
{ int a=5, b=6, c  ;
int f(), (*g)()=f    ;
printf("%d\n", (*g)(a,b) )      ;
}
```

A) 1 B) 2 C) 3 D) 前面三个参考答案均是错误的

【**2.51**】下面程序的输出结果是 _____。

```
#include <stdio.h>
main( )
{ int a=1,*p,**pp ；
pp=&p ；
p=&a ；
a++；
printf ("%d,%d,%d\n", a,*p, **pp) ；
}
```

A) 2,1,1 B) 2,1,2 C) 2,2,2 D) 程序有错误

【**2.52**】下面程序的输出结果是 _____。

```
main()
{ char *alpha[7]={"ABCD","EFGH","IJKL","MNOP","QRST","UVWX","YZ"} ；
char **p ；
int i；
p=alpha ；
for(i=0 ；i<4；i++)
printf("%c",*(p[i])) ；
printf("\n") ；
}
```

A) AEIM B) BFJN C) ABCD D) DHLP

【**2.53**】下面程序的输出结果是 _____。

```
#include <stdio.h>
char *pp[2][3]={ "abc", "defgh", "ijkl", "mnopqr", "stuvw", "xyz"} ；
main ( )
{ printf("%c\n",***(pp+1)) ； /* ① */
printf("%c\n",**pp[0]) ； /* ② */
printf("%c\n",(*(*(pp+1)+1))[4]) ； /* ③ */
printf("%c\n",*(pp[1][2]+2)) ； /* ④ */
printf("%s\n",**(pp+1)) ； /* ⑤ */
}
```

① **A) a B) d C) i D) m**

② **A) a B) d C) i D) m**

③ **A) h B) l C) q D) w**

④ **A) k B) o C) u D) z**

⑤ **A) ijkl B) mnopqr C) stuvw D) xyz**

【2.54】下面程序的输出结果是　　____。

```
#include "stdio.h"
struct str1
{ char c[5]；
char *s ；
}；
main( )
{ struct str1 s1[2]={ {"ABCD""EFGH"}　，{"IJK" ，"LMN"} }　；
struct str2
{ struct str1 sr ；
int d ；
}s2={"OPQ" ，"RST" ，32767} ；
struct str1 *p[2] ；
p[0]=&s1[0] ；
p[1]=&s1[1] ；
printf("%s" ，++p[1]->s) ； /* ① */
printf("%c" ，s2.sr.c[2]) ； /* ② */
}
```

① A) LMN B) MN C) N D) IJK

② A) O B) P C) Q D) R

【2.55】以下程序的输出结果是　　____。

```
struct st
{ int x,*y ；
}*p ；
int s[]={10,20,30,40};
struct st a[]={1,&s[0],2,&s[1],3,&s[2],4,&s[3]};
main()
{ p=a ；
printf("%d\n",++(*(++p)->y))　　；
}
```

A) 10 B) 11 C) 20 D) 21

【2.56】以下程序的输出结果是　　____。

```
#include <stdio.h>
main()
{ union EXAMPLE
{ struct
{ int x,y ；
}in ；
int a,b ；
}e ；
e.a=1 ； e.b=2 ；
```

```
e.in.x=e.a*e.b ；
e.in.y=e.a+e.b；
printf("%d,%d\n",e.in.x,e.in.y)     ；
}
```

A) 2,3 B) 4,4 C) 4,8 D) 8,8

【2.57】下面程序的输出结果是　　____。

```
#include <stdio.h>
main()
{ union
{ int i[2]  ;
long k ;
char c[4] ;
}r,*s=&r   ;
s->i[0]=0x39  ;
s->i[1]=0x38  ;
printf("%c\n",s->c[0])    ;
}
```

A) 39 B) 9 C) 38 D) 8

【2.58】下面程序的输出是　　。

```
main ( )
{ printf("%d\n", EOF)     ;
}
```

A) -1 B) 0 C) 1 D)　程序是错误的

【阅读程序题参考答案】

【2.1】参考答案：　**D**

注释：程序中除法运算的两个操作数均是整型，运算结果也是整型。

【2.2】参考答案：　**B**

注释：　**C** 语言允许在程序块　**(**分程序　**)**中说明变量。

【2.3】参考答案：　**C**

注释：变量　**i** 中的负号传送给变量　　**n** 后，因　**n** 是无符号数，已不作为负号处理。

【2.4】参考答案：　**D**

注释：对变量　**x** 的——操作是后缀形式，　　变量　**x** 的减　**1** 操作要在执行完　　**printf**　函数之后才进行，所以变量　**x** 的值在输出的时候仍然保持原值　　**10**。

【2.5】参考答案：　**B**

注释：　**C** 语言在执行　**printf()** 时，对函数中的表达式表列的处理顺序是从后向前，即先处理　　**n- -**，再处理　**n++**，最后处理　**n**，而且每一个表达式作为一个处理单元，也就是说在不同的表达式中自增自减运算是单独考虑的。

【2.6】参考答案：　**A**

注释：变量　**x** 和变量　**y** 做按位与，结果为　**0x0200**，右移　**4** 位为　**0x0020**，再与　**0x005f** 做按位或，最后结果为　**0x007f**。

【2.7】参考答案： A

注释：逗号表达式的结果是用逗号分开的最后一个表达式的值，此题由于 c=='A' 的值是 0,所以逗号表达式的值为 0。

【2.8】参考答案 ：B

【2.9】参考答案 ：A

【2.10】参考答案 ：C

注释：在输出格式描述 "%m.ns" 中， m 是输出总长度， n 是实际字符的个数，这里 m 没有给出，则输出总长度就是实际输出字符的个数。

【2.11】参考答案 ：C

【2.12】参考答案： B

【2.13】参考答案： C

【2.14】参考答案： B

【2.15】参考答案： D

【2.16】参考答案： A

【2.17】参考答案 ：C

【2.18】参考答案： A

【2.19】参考答案 ：C

注释：在 switch 语句中， case 本身仅起到语句标号的作用，不会改变语句的流程，执行 break 语句才能退出当前的 switch 语句。

【2.20】参考答案 ：D

注释： siwtch 语句的表达式中，变量 c 是后缀的增一运算，第一次执行 do-while 循环时，执行 case 'A'后面的语句。

【2.21】参考答案 ：D

【2.22】参考答案 ：B

【2.23】参考答案 ：B

注释： fabs()是浮点数绝对值函数。

【2.24】参考答案 ：A

【2.25】参考答案 ：C

注释： C 语言允许在程序块（分程序）内说明变量，如果在程序块内说明的变量和程序块外的变量同名，在块外说明的变量在块内是不可见的。可将此题和【 2.11】进行比较，加深理解。

【2.26】参考答案 ：C

【2.27】参考答案 ：B

【2.28】参考答案： ① D ② A

【2.29】参考答案： D

【2.30】参考答案 ：B

注释：输出结果为字符串长度。

【2.31】参考答案 ：D

注释：字符串拷贝函数 strcpy()要求的两个参数都是字符串首地址。本题中第二个参数是字符串常量，接受这个字符串的第一个参量不是直接给出字符数组名，而是进行了地址运算后的结果。由于 str 字符串的长度是 13，除 2 取整后是 6，第一个参数给出的地址是字符数组 str 的首地址加 6，也就是原来字符串中第二个空格的位置，把"es she"从该处放入， 字符串 str 变为 "How does she"。

【2.32】参考答案 ：C

注释： **main** 函数调用 **func** 函数时，第一个实参使用的是逗号表达式的值，也就是 **x+y** 的结果。由于对变量 **x**、**y**、**z** 进行的是后缀运算，所以函数 **func** 的参数值是 **13** 和 **8**。

【2.33】参考答案 **：C**

【2.34】参考答案 **：** ① **C** ② **A** ③ **C**

【2.35】参考答案 **：C**

【2.36】参考答案 **：B**

注释：函数 **fun** 进行了递归调用，实际进行的运算是 **5**×**4**×**3**×**2**×**1**×**3**×**10**。主函数内说明的局部变量 **w** 屏蔽了外部变量 **w**，所以在主函数中外部变量 **w** 是不可见的，在调用 **printf** 函数时表达式 **"fun(5)*w"** 中 **w** 的值是 **10**。

【2.37】参考答案 **：D**

注释： **main** 函数三次调用了函数 **funa**，在 **funa** 函数中的静态变量 **c** 仅在第一次调用时进行了初始化，再次调用时不再对静态变量赋初值。

【2.38】参考答案 **：B**

注释： **main** 函数和 **num** 函数中都说明了变量 **a** 和 **b**，由于它们是内部变量，所以它们分别在说明它们的函数内有效。 外部变量 **x** 和 **y** 在函数 **num** 之后被说明， 而在 **num** 函数中又要引用它们，所以在 **num** 函数中用关键字 **"extern"** 说明变量 **x** 和 **y** 是一个外部变量，也就是通知计算机这两个变量在 **fun** 函数以外被说明，此处不是定义两个 **int** 型变量。

【2.39】参考答案 **：D**

注释：函数 **f** 中的变量 **c** 是静态变量，仅在第一次调用函数 **f** 时它被初始化为 **3,**第二次调用函数 **f** 时 **c** 的值是 **4**，第三次调用函数 **f** 时 **c** 的值是 **5**。

【2.40】参考答案 **：D**

【2.41】参考答案 **：D**

注释： 程序中有三个 **"x"** 分别在三个不同的函数中， 这三个 **"x"** 都是自动变量， 所以三个 **"x"** 分别局部于三不同的函数，在三个函数中对 **"x"** 的操作互不影响。

【2.42】参考答案 **：A**

【2.43】参考答案 **：A**

注释： ***(++p)** 和 ***++p** 都是指针变量值前加 **1**，第一次指向 **a[1]**，第二次指向 **a[2]**；**a+3** 是 **a[3]** 的地址。

【2.44】参考答案 **：C**

注释：②句没有语法错误，但是 **a+6** 指向数组之外，因为 **a** 是 **a[0]**的地址， **a+1** 是 **a[1]**的地址，**a+2** 是 **a[2]**的地址，显然数组 **a** 没有 **a[6]**分量。③句错误，因为 **a[1]**是地址常量，它是 **a[1][0]** 的地址，对于地址常量是不可以进行赋值运算的。

【2.45】参考答案 **：** ① **D** ② **A**

注释：如果 **FMT** 定义为 **"%x\n"** ，则输出的 **16** 进制数据用小写字母表示。

【2.46】参考答案 **：A**

注释：语句 **"p=&a[0]"** 表示将数组 **a** 中元素 **a[0]**的地址赋给指针变量 **p**，则 **p** 就是指向数组首元素 **a[0]**的指针变量， **"&a[0]"** 是取数组首元素的地址。对于指向数组首址的指针， **p+i**（或 **a+i**）是数组元素 **a[i]**的地址， ***(p+i)(** 或 ***(a+i))** 就是 **a[i]**的值。

【2.47】参考答案 **：B**

【2.48】参考答案 **：D**

【2.49】参考答案 **：D**

【2.50】参考答案 **：A**

注释：变量 **g** 是指向函数的指针， **(*g)(a,b)** 是调用指针 **g** 所指向的函数。

【2.51】参考答案 **：C**

注释： **p** 是指针， **pp** 是指向指针的指针。

【2.52】参考答案 **：A**

注释：对于指向数组的指针变量可以做下标运算， **p[i]**和 **alpha[i]** 都是指向字符串的首地址， ***p[i]** 取出字符串的第一个字符。

【2.53】参考答案 **：** ① **D** ② **A** ③ **D** ④ **D** ⑤ **B**

注释：**pp** 是一个二维指针数组， **pp+1** 指向数组的第二维， ***(pp+1)** 是第二维的起始地址， ****(pp+1)** 是第二维第一个元素的地址， ***(pp+1)** 是第二维第一个元素的内容，所以，①的参考答案应选 **D**。***(pp+1)+1** 是第二维第二个元素的地址， ***(*(pp+1)+1)** 是第二维第二个元素， **(*(*(pp+1)+1))[4]** 则是第二维第二个元素所指字符串下标为 **4** 的元素，即是字符 **w**，故③应当选 **D**。

【2.54】参考答案 **：** ① **B** ② **C**

【2.55】参考答案 **:D**

【2.56】参考答案 **:C**

注释：联合体成员的取值是最后一次给成员赋的值。

【2.57】参考答案 **: B**

注释：整型数组 **i** 和字符数组 **c** 共用存储空间，给 **i** 赋值也等于给 **c** 赋值，所以 **s->c[0]=0x39**，所以输出 **9**。

【2.58】参考答案： **A**

注释：基本概念。 **EOF** 是由 **C** 语言在头文件 **stdio.h** 中定义的，用户可以直接使用。

三、程序填空题

导读：在程序填空题中，已经给出了程序的主干，读者首先要理解程序的思路，再选择正确的内容填入空白处，使程序完成既定的功能。这类习题的设计就是要引导读者逐步掌握编程的方法。本节习题的难度适中，可能有些典型的程序在课堂上已经有所接触，读者一定要独立完成它，这样就可以逐步提高自己的编程能力。在程序设计语言学习的中期，读者对程序设计已经有了初步的了解，而自己编写程序又不知从何处入手，此时解答此类题目可以避免盲目性，从而提高学习的效率。

【3.1】下面程序的功能是不用第三个变量，实现两个数的对调操作。

```
#include <stdio.h>
main()
{ int a,b  ;
scanf("%d%d",&a,&b)     ;
printf("a=%d,b=%d\n",a,b)      ;
a=  ①    ;
b=  ②    ;
a=  ③    ;
printf("a=%d,b=%d\n",a,b)      ;
}
```

【3.2】下面程序的功能是根据近似公式： π 2/6≈ 1/12+1/22+1/32+ …… +1/n2,求 π 值。

```
#include <math.h>
double pi(long n)
{ double s=0.0；
long i；
for(i=1 ； i<=n ； i++)
s=s+ ①  ；
return( ②  )；
}
```

【3.3】下面的程序的功能是求一维数组中的最小元素。

```
findmin(int *s,int t,int *k)
{ int p ；
for(p=0,*k=p  ； p<t ； p++)
if(s[p]<s[*k])
①  ；
}
main()
{ int a[10],i,*k=&i  ；
for(i=0 ； i<10 ； i++)
scanf("%d",&a[i])  ；
findmin(a,10,k) ；
printf("%d,%d\n",*k,a[*k])  ；
}
```

【3.4】下面程序的功能是计算 1-3+5-7+ …… -99+101 的值。

```
main()
{ int i,t=1,s=0 ；
for(i=1 ； i<=101 ； i+=2)
{ ①  ；
s=s+t；
②  ；
}
printf("%d\n",s)  ；
}
```

【3.5】有以下程序段：

```
s=1.0；
for(k=1 ； k<=n ； k++)
s=s+1.0/(k*(k+1)) ；
```

printf("%f\n",s)　；

填空完成下述程序，使之与上述程序的功能完全相同。

s=0.0；

①　；

k=0；

do

{ s=s+d；

②　；

d=1.0/(k*(k+1))　；

}while(　③　)；

printf("%f\n",s)　；

【3.6】下面程序的功能是从键盘上输入若干学生的学习成绩，　　统计并输出最高成绩和最低成绩，当输入为负数时结束输入。

main()

{ float x,amax,amin　；

scanf("%f",&x)　；

amax=x；

amin=x；

while(　①　)

{ if(x>amax) amax=x　；

if(　②　) amin=x；

scanf("%f",&x)　；

}

printf("\namax=%f\namin=%f\n",amax,amin)　　；

}

【3.7】下面程序的功能是将形参　　**x** 的值转换为二进制数，所得的二进制数放在一个一维数组中返回，二进制数的最低位放在下标为　　**0** 的元素中。

fun(int x,int b[])

{ int k=0,r　；

do

{ r=x%　①　；

b[k++]=r　；

x/=　②　；

}while(x)；

}

【3.8】下面程序的功能是输出　　**1** 到 **100** 之间每位数的乘积大于每位数的和的数。例如数字　　**26**，数位上数字的乘积　**12** 大于数字之和　**8**。

main()

{ int n,k=1,s=0,m　；

```
for(n=1 ； n<=100 ； n++)
{ k=1 ；
s=0 ；
①  ；
while( ② )
{ k*=m%10  ；
s+=m%10 ；
③  ；
}
if(k>s)
printf("%d",n)  ；
}
}
```

【3.9】下面程序的功能是统计用 0 至 9 之间的不同的数字组成的三位数的个数。

```
main()
{ int i,j,k,count=0 ；
for(i=1 ； i<=9 ； i++)
for(j=0 ； j<=9 ； j++)
if( ① ) continue ；
else for(k=0 ； k<=9 ； k++)
if( ② ) count++ ；
printf("%d",count)  ；
}
```

【3.10】下面程序的功能是输出 100 以内的个位数为 6、且能被 3 整除的所有数。

```
main()
{ int i,j ；
for(i=0 ； ①  ； i++)
{ j=i*10+6 ；
if( ② ) countinue ；
printf("%d",j)  ；
}
}
```

【3.11】下面程序的功能是用辗转相除法求两个正整数 m 和 n 的最大公约数。

```
hcf(int m,int n)
{ int r ；
if(m<n)
{ r=m ；
①  ；
n=r ；
```

```
}
r=m%n ;
while( r!=0 )
{ m=n ;
n=r ;
r=m%n ;
}
return n
;
}
```

【3.12】下面程序的功能是使用冒泡法对输入的 10 个浮点数从小到大进行排序。 排好序的 10 个数分两行输出。程序如下：

```
#include <stdio.h>
main()
{ ① ;
int i,j ;
printf("Input 10 numbers please\n") ;
for(i=0 ; ② ; i++ )
scanf("%f", &a[i]) ;
printf("\n") ;
for(i=2 ; ③ ; i++ )
for(j=0 ; ④ ; j++ )
if( ⑤ )
{ x=a[j] ;
⑥ ;
a[j+1]=x ;
}
printf("The sorted 10 numbers ; \n") ;
for(i=0 ; ⑦ ; i++ )
{ if( ⑧ )
printf("\n") ;
printf("%f\t",a[i]) ;
}
printf("\n") ;
}
```

【3.13】下面程序的功能是读入 20 个整数，统计非负数个数，并计算非负数之和。

```
#include "stdio.h"
main()
{ int i,a[20],s,count；
s=count=0；
for(i=0 ；i<20 ；i++ )
scanf("%d", ① )；
```

```
for(i=0 ；i<20；i++)
{ if(a[i]<0)
②  ;
s+=a[i]；
count++；
}
printf("s=%d\t count=%d\n",s,count)    ;
}
```

【3.14】下面程序的功能是删除字符串 s中的空格。

```
#include <stdio.h>
main()
{ char *s="Beijing ligong daxue"  ；
int i,j ；
for(i=j=0 ；s[i]!='\0' ；i++)
if(s[i]!= ' ')  ①  ;
else ②  ；s[j++]=s[i]
②s[j]=s[i]
s[j]= '\0' ；printf("%s",s)  ;
}
```

【3.15】下面程序的功能是将字符串 s中所有的字符 'c'删除。请选择填空。

```
#include<stdio.h>
main( )
{ char s[80]；
int i,j ；
gets(s)；
for(i=j=0 ；s[i]!= '\0' ；i++ )
if(s[i]!= 'c')  ①  ;
s[j]= '\0' ；
puts(s)；
}
```

【3.16】下面程序的功能是输出两个字符串中对应相等的字符。请选择填空。

```
#include <stdio.h>
char x[]="programming"  ;
char y[]="Fortran"  ;
main()
{ int i=0 ;
while(x[i]!= '\0' && y[i]!= '\0')
if(x[i]==y[i])
printf("%c",  ①x[i++]  ) ;
else
```

```
i++ ；
}
```

【3.17】下面程序的功能是将字符串 s 中的每个字符按升序的规则插到数组 a 中，字符串 a 已排好序。

```
#include <string.h>
main()
{ char a[20]="cehiknqtw" ；
 char s[]="fbla" ；
 int i,k,j ；
 for(k=0 ；s[k]!= '\0' ；k++ )
 { j=0 ；
 while(s[k]>=a[j] && a[j]!= '\0' )
 j++ ；
for( ① )
② ；
 a[j]=s[k] ；
 }
 puts(a)；
 }
```

【3.18】下面程序的功能是对键盘输入的两个字符串进行比较，然后输出两个字符串中第一个不相同字符的 ASCII 码之差。例如：输入的两个字符串分别为 "abcdefg" 和 "abceef"，则输出为 -1。

```
#include <stdio.h>
main()
{ char str1[100],str2[100],c ；
int i,s；
printf("Enter string 1: ") ； gets(str1)；
printf("Enter string 2: ") ； gets(str2)；
i=0 ；
while((str1[i] == str2[i] && str1[i]!= ① ))
i++ ；
s= ② ；
printf("%d\n", s) ；
}
```

【3.19】下面的函数 expand 在将字符串 s 复制到字符串 t 时，将其中的换行符和制表符转换为可见的转义字符表示，即用 '\n' 表示换行符，用 '\t' 表示制表符。

```
expand(char s[],char t[])
{ int i,j ；
for(i=j=0 ；s[i]!= '\0' ；i++ )
switch (s[i])
```

```
{ case '\n': t[ ① ] = ② ;
t[j++] = 'n' ;
break ;
case '\t': t[ ③ ] = ④ ;
t[j++] = 't' ;
break ;
default: t[ ⑤ ] = s[i] ;
break ;
}
t[j] = ⑥ ;
}
```

【3.20】下面的函数 index(char s[], char t[]) 检查字符串 s 中是否包含字符串 t，若包含，则返回 t 在 s 中的开始位置（下标值），否则送回 -1。

```
index(char s[], char t[])
{ int i,j,k ;
for(i=0 ; s[i]!= '\0' ; i++ )
{ for(j=i,k=0 ; ① && s[j]==t[k] ; j++,k++) ;
if( ② )
return (i) ;
}
return(-1) ;
}
```

【3.21】下面程序的功能是计算 S= k! 。

```
k=0
long fun(int n)
{ int i ;
long s;
for(i=1 ; i ① ; i++)
s*=i ;
return( ② );
}
main()
{ int k,n ;
long s;
scanf("%d",&n) ;
s= ③ ;
for(k=0 ; k<=n ; k++)
s+= ④ ;
printf("%ld\n",s) ;
}
```

【3.22】下面程序的功能是显示具有 n 个元素的数组 s 中的最大元素。

```
#define N 20
main()
{ int i,a[N] ；
for(i=0 ； i<N ； i++)
scanf("%d",&a[i]) ；
printf("%d\n", ① )；
}
fmax(int s[],int n)
{ int k,p ；
for(p=0,k=p ； p<n ； p++)
if(s[p]>s[k]) ② ；
return(k) ；
}
```

【3.23】下面程序的功能是由键盘输入 n，求满足下述条件的 x、y：

nx 和 ny 的末 3 位数字相同，且 x≠ y，x、y、n 均为自然数，并使 x+y 为最小。

```
#include <stdio.h>
pow3(int n,int x)
{ int i, last ；
for(last=1,i=1 ； i<=x ； i++ )
last= ① ；
return(last) ；
}
main()
{ int x,n,min,flag=1 ；
scanf("%d", &n) ；
for(min=2 ； flag ； min++)
for(x=1 ； x<min && flag ； x++ )
if( ② && pow3(n,x)==pow3(n,min-x))
{ printf("x=%d,y=%d\n", x, min-x ) ；
③ ；
}
```

【3.24】下面的程序是用递归算法求 a 的平方根。求平方根的迭代公式如下：

```
#include <math.h>
double mysqrt( double a, double x0 )
{ double x1, y；
x1 = ① ；
```

```
if( fabs(x1-x0)>0.00001 )
y = mysqrt(  ②  ) ;
else y = x1 ;
return( y )  ;
}
main()
{ double x ;
printf("Enter x: ")    ;
scanf("%lf", &x)    ;
printf("The sqrt of %lf=%lf\n", x, mysqrt( x, 1.0) )      ;
}
```

【3.25】以下程序是计算学生的年龄。已知第一位最小的学生年龄为 10 岁，其余学生的年龄一个比一个大 2 岁，求第 5 个学生的年龄。

```
#include <stdio.h>
age( int n )
{ int c ;
if( n==1 ) c=10  ;
else c= ①   ;
return(c)  ;
}
main()
{ int n=5  ;
printf("age:%d\n",    ②  )  ;
}
```

【3.26】下面的函数 sum(int n)完成计算 1～n 的累加和。

```
        sum(int n)
{ if(n<=0)
printf("data error\n")    ;
if(n==1)  ①   ;
else ②   ;
}
```

【3.27】下面的函数是一个求阶乘的递归调用函数。

```
facto(int n)
{ if( n == 1 )  ①   ;
else return(  ②  ) ;
}
```

【3.28】组合问题，由组合的基本性质可知：

(1) C(m,n)=C(n-m,n)

(2) C(m,n+1)=C(m,n)+C(m-1,n)

公式 **(2)**是一个递归公式，一直到满足 **C(1,n)=n** 为止。当 **n<2*m** 时，可先用公式 **(1)** 进行简化，填写程序中的空白，使程序可以正确运行。

```
#include"stdio.h"
main()
{ int m,n ；
printf("Input m,n=")    ；
scanf("%d%d", &m, &n)     ；
printf("The combination numbeers is %d\n", combin(m,n))    ；
}
combin( int m, int n)
{ int com ；
if( n<2*m ) m=n-m   ；
if( m==0 ) com=1  ；
else if(m==1)  ①   ；
else ②   ；
return(com) ；
}
```

【3.29】下列函数是求一个字符串 **str** 的长度。

```
? ? ? ? ? ?  int strlen( char *str )
? ? ? ? ? ? { if(  ①  ) return (0) ；
? ?  ? ? ? ?  else return (  ②  )；
}
```

【3.30】用递归实现将输入小于 **32768** 的整数按逆序输出。如输入 **12345**，则输出 **54321**。

```
#include"stdio.h"
main()
{ int n ；
printf("Input n : ")    ；
scanf("%d",   ①  )；
r(n) ；
printf("\n")   ；
}
r( int m )
{ printf("%d",    ②  ) ；
m =  ③   ；
if(  ④  )
⑤   ；
}
```

【3.31】输入 n 值，输出高度为 n 的等边三角形。例如当 n=4 时的图形如下：

```
*
***
*****
*******
#include <stdio.h>
void prt( char c, int n )
{ if( n>0 )
{ printf( "%c", c )  ;
①  ;
}
}
main()
{ int i, n  ;
scanf("%d", &n)  ;
for( i=1  ;  i<=n  ;  i++ )
{  ②  ;
③  ;
printf("\n")  ;
}
}
```

【3.32】下面的函数实现 N 层嵌套平方根的计算。

```
double y(double x, int n)
{ if( n==0 )
return(0) ；
else return ( sqrt(x+(  ①  )) )；
}
```

【3.33】函数 revstr(s)将字符串 s 置逆， 如输入的实参 s 为字符串 "abcde"， 则返回时 s 为字符串"edcba"。递归程序如下：

```
revstr( char *s )
{ char *p=s, c ；
while(*p) p++  ;
①  ;
if(s<p)
{ c=*s ;
*s=*p  ;
②* p = \ 0  ;
revstr(s+1) ;
③* p = c  ;
}
```

}

如下是由非递归实现的 **revstr(s)**函数：

revstr (s)
char *s ；
{ char *p=s, c ；
while(*p) p++ ；
④ ；
while(s<p)
{ c=*s ；
⑤ **= *p** ；
***p-- = c** ；
}
}

【**3.34**】下面函数用递归调用的方法，将 **str** 中存放的长度为 **n** 的字符串反转过来，例如原来是 **"ABCDE"** ，反序为 **"EDCBA"** 。

void invent(char *str ， int n)
{ char t ；
t=*str ； ***str=*(str+n-1)** ； ***(str+n-1)=t** ； **if(n>2) invent (** ① ， **n-2)** ；
else ② ；
}

【**3.35**】从键盘上输入 **10** 个整数，程序按降序完成从大到小的排序。

#include <stdio.h>
int array[10] ；
sort(int *p, int *q)
{ int *max, *s ；
if(① **)**
return ；
max=p ； **for(s=p+1** ； **s<=q** ； **s++)**
if(*s > *max)
② ； **swap(** ③ **)** ；
sort(④ **)** ； **}**
swap(int *x, int *y) { int temp ；
temp=*x ；
***x=*y** ；
***y=temp** ；
}
main()
{ int i ； **printf("Enter data :\n")** ； **for(i=0** ； **i<10** ； **i++)**
scanf("%d", &array[i]) ； **sort(** ⑤ **)** ；
printf("Output:") ；
for(i=0 ； **i<10** ； **i++)**

```
printf("%d ", array[i])      ;
}
```

【3.36】下面函数的功能是将一个整数存放到一个数组中。存放时按逆序存放。例如：　　**483** 存放成**"384"** 。

```
#include <stdio.h>
void convert(char *a, int n)
{ int i ;
if((i=n/10) !=0 )
convert(  ①   , i ) ;
*a =   ②    ;
}
char str[10]= " "    ;
main()
{ int number  ;
scanf("%d", &number)     ;
convert( str, number )  ;
puts(str) ;
}
```

【3.37】下面程序的功能是实现数组元素中值的逆转。

```
#include <string.h>
main()
{ int i,n=10,a[10]={1,2,3,4,5,6,7,8,9,10}    ;
invert(a,n-1)  ;
for(i=0  ;  i<10  ;  i++)
printf("%4d",a[i])     ;
printf("\n")    ;
}
invert(int *s,int num)
{ int *t,k   ;
t=s+num  ;
while(   ①   )
{ k=*s  ;
*s=*t   ;
*t=k  ;
②    ;
③    ;
}
}
```

【3.38】下面程序通过指向整型的指针将数组　　**a[3][4]**　的内容按３行×４列的格式输出，请给

printf( ) 填入适当的参数，使之通过指针 **p** 将数组元素按要求输出。

```
#include <stdio.h>
int a[3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}}, *p=a ;
main()
{ int i,j ;
for(i=0 ; i<3 ; i++ )
{ for(j=0 ; j<4 ; j++ )
printf("%4d ", ① );
}
```

【3.39】下面程序的功能是： 从键盘上输入一行字符， 存入一个字符数组中， 然后输出该字符串。

```
#include <stdio.h>
main ( )
{ char str[81], *sptr ;
int i ;
for(i=0 ; i<80 ; i++ )
{ str[i]=getchar( ) ;
if(str[i]== '\n') break ;
}
str[i]= ① ;
sptr=str ;
while( *sptr )
putchar( *sptr ② );
}
```

【3.40】下面函数的功能是将字符变量的值插入已经按 **ASCII** 码值从小到大排好序的字符串中。

```
void fun(char *w,char x,int *n)
{ int i,p=0 ;
while(x>w[p]) ① ;
for(i=*n ; i>=p ; i--) ② ;
w[p]=x ;
++*n ;
}
```

【3.41】下面程序的功能是从键盘上输入两个字符串， 对两个字符串分别排序； 然后将它们合并，合并后的字符串按 **ASCII** 码值从小到大排序，并删去相同的字符。

```
#include <stdio.h>
strmerge(a, b, c) /* 将已排好序的字符串 a、b 合并到 c */
char *a , *b , *c ;
{ char t , *w ;
w=c ;
```

```
while( *a!= '\0'    ①  *b!='\0' )
{ t=  ②  ?*a++:*b<*a ? *b++ : (     ③  ) ;   /*  将 *a、 *b 的小者存入   t */
if( *w   ④  '\0' ) *w=t  ;
else if( t  ⑤  *w) *++w=t   ;   /*  将与 *w  不相同的   t 存入  w */
}
while( *a != '\0' ) /*     以下将   a 或  b 中剩下的字符存入     w */
if( *a != *w ) *++w=*a++      ;
else a++ ;
while( *b != '\0')
if( *b != *w ) *++w=*b++      ;
else b++ ;
*++w =   ⑥   ;
}
strsort( char *s ) /*   将字符串   s 中的字符排序     */
{ int i ,  j,  n ;
char t,  *w  ;
⑦   ;
for( n=0  ; *w != '\0'  ;   ⑧  )
w++  ;
for( i=0  ; i<n-1  ; i++ )
for( j=i+1  ; j<n  ; j++ )
if( s[i]>s[j] )
{  ⑨  }
}
main( )
{ char s1[100],  s2[100],  s3[200] ;
printf("\nPlease Input First String:")    ;
scanf("%s" ,  s1) ;
printf("\nPlease Input Second String:")   ;
scanf("%s" ,  s2) ;
strsort(s1) ;
strsort(s2) ;
⑩   = '\0'  ;
strmerge(s1,  s2,  s3) ;
printf("\nResult:%s"   ,  s3) ;
}
```

【3.42】已知某数列前两项为 **2** 和 **3**，其后继项根据前面最后两项的乘积，按下列规则生成：

① 若乘积为一位数，则该乘积即为数列的后继项；

② 若乘积为二位数，则该乘积的十位上的数字和个位上的数字依次作为数列的两个后继项。

下面的程序输出该数列的前N项及它们的和，其中，函数 **sum(n,pa)** 返回数列的前 **N** 项和，并将生成的前 **N** 项存入首指针为 **pa** 的数组中， 程序中规定输入的 **N** 值必须大于 **2**，且不超过给定的常数值 **MAXNUM** 。

例如：若输入N的值为 **10**，则程序输出如下内容：

```
sum(10)=44
2 3 6 1 8 8 6 4 2 4
#include "stdio.h"
#define MAXNUM 100
int sum(n, pa)
int n, *pa ;
{ int count, total, temp ;
*pa = 2 ;
①  =3 ;
total=5 ;
count=2 ;
while( count++<n )
{ temp = *(pa-1) * *pa  ;
if( temp<10 )
{ total += temp ;
*(++pa) = temp  ;
}
else
{ ②  = temp/10 ;
total += *pa  ;
if( count<n )
{ count ++ ;  pa++ ;
③  = temp%10  ;
total += *pa  ;
}
}
}
④  ;
}
main()
{ int n, *p, *q, num[MAXNUM]  ;
do
{ printf("Input N=? (2<N<%d):", MAXNUM+1)  ;
scanf("%d", &n)  ;
}while( ⑤ ) ;
printf("\nsum(%d)=%d\n", n, sum(n, num))  ;
for( p=num, q = ⑥ ;  p<q ;  p++ )
printf("%4d", *p)  ;
printf("\n")  ;
}
```

【3.43】下面程序的功能是输入学生的姓名和成绩，然后输出。

```
#include <stdio.h>
struct stuinf
```

```
{ char name[20] ；  /*  学生姓名   */
int score；  /*  学生成绩   */
} stu, *p ；
main ( )
{ p=&stu  ；
printf("Enter name:")   ；
gets( ①  ) ；
printf("Enter score: ")   ；
scanf("%d" ，   ②  ) ；
printf("Output: %s, %d\n"   ，   ③ ，   ④  ) ；
}
```

【3.44】下面程序的功能是按学生的姓名查询其成绩排名和平均成绩。查询时可连续进行，直到输入 0 时才结束。

```
？？ #include <stdio.h>
#include <string.h>
#define NUM 4
？ struct student
？ { int rank  ；
char *name  ；
float score；
？？？？？？？？ } ；
？？？？？？？？ ①  stu[ ]={ 3， "liming" ， 89.3，
？？？？？？？？ 4， "zhanghua" ， 78.2，
？？？？？？？？ 1， "anli" ， 95.1，
？？？？？？？？ 2， "wangqi" ， 90.6
} ；
？？？？？？？？ main()
？？？？？？？？ { char str[10]  ；
？？？？？？？？ int i ；
？？？？？？？？ do
{ printf("Enter a name")   ；
？？？？？？？？ scanf("%s" ，  str)；
？？？？？？？？ for( i=0 ；i<NUM  ；i++ )
？？？？？？？？ if(  ②  )
？？？？？？？？ { printf("Name :%8s\n"   ，  stu[i].name) ；
？？？？？？？？ printf("Rank :%3d\n"   ，  stu[i].rank)  ；
？？？？？？？？ printf("Average :%5.1f\n"   ，  stu[i].score)；
？？？？？？？？ ③  ；
？？？？？？？？ }
？？？？？？？？ if( i>=NUM ) printf("Not found\n")      ；
？？？？？？？？ }while( strcmp(str ，  "0")!=0 )  ；
？？？？？？？？ }
```

【3.45】下面程序的功能是从终端上输入５个人的年龄、性别和姓名，然后输出。

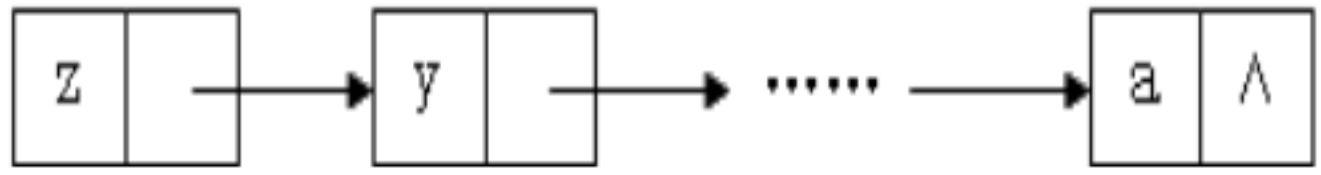
```
#include "stdio.h"
struct man
{ char name[20] ；
unsigned age；
char sex[7]；
}；
main ( )
{ struct man person[5] ；
data_in(person,5) ；
data_out(person,5) ；
}
data_in(struct man *p, int n )
{ struct man *q =  ①  ；
for(  ；p<q；p++ )
{ printf( "age:sex:name" )  ；
scanf("%u%s", &p->age, p->sex)  ；
②  ；
}
}
data_out( struct man *p, int n )
{ struct man *q = __ ③__；
for(  ；p<q；p++ )
printf("%s  ；%u ；%s\n", p->name, p->age, p->sex)  ；
}
```

【3.46】输入 N 个整数，储存输入的数及对应的序号，并将输入的数按从小到大的顺序进行排列。要求：当两个整数相等时，整数的排列顺序由输入的先后次序决定。例如：输入的第 3 个整数为 5，第 7 个整数也为 5，则将先输入的整数 5 排在后输入的整数 5 的前面。程序如下：

```
#include "stdio.h"
#define N 10
struct
{ int no ；
int num ；
} array[N] ；
main( )
{ int i ， j， num ；
for( i=0 ；i<N ；i++ )
{ printf("enter No. %d:"  ，  i) ；
scanf("%d" ，  &num)  ；
for(  ①  ；j>=0&&array[j].num  ②  num ；  ③  )
array[j+1]=array[j]  ；
array[  ④  ].num=num ；
array[  ⑤  ].no=i ；
```

```
}
for( i=0 ；i<N ；i++ )
printf("%d=%d  ，  %d\n" ，  i，  array[i].num ，  array[i].no)  ；
}
```

【3.47】以下程序的功能是：读入一行字符 (如： a、 ...y、 z)，按输入时的逆序建立一个链接式的结点序列， 即先输入的位于链表尾 （如下图），然后再按输入的相反顺序输出， 并释放全部结点。




```
#include <stdio.h>
main( )
{ struct node
{ char info ；
struct node *link ；
} *top ，  *p ；
char c；
top=NULL  ；
while((c= getchar( ))  ①  )
{ p=(struct node *)malloc(sizeof(struct node))  ；
p->info=c ；
p->link=top  ；
top=p；
}
while( top )
{ ②  ；
top=top->link  ；
putchar(p->info)  ；
free(p)；
}
}
```

【3.48】下面函数将指针 p2 所指向的线性链表，串接到 p1 所指向的链表的末端。假定 p1 所指向的链表非空。

```
#define NULL 0
struct link
{ float a；
struct link *next  ；
}；
concatenate ( p1,  p2 )
struct list *p1 ，  *p2 ；
```

```
{ if( p1->next==NULL )
p1->next=p2 ；
else
concatenate( ①  ，  p2)；
}
```

【3.49】下面程序的功能是从键盘输入一个字符串，然后反序输出输入的字符串。

```
#include <stdio.h>
struct node
{ char data ；
struct node *link ；
}*head ；
main()
{ char ch ；
struct node *p ；
head = NULL  ；
while(( ch=getchar())!='\n' )
{ p = (struct node *)malloc(sizeof(struct node)) ；
p->data = ch ；
p->link =  ①  ；
head =  ②  ；
}
③  ；
while( p!=NULL )
{ printf("%c ", p->data)  ；
p = p->link ；
}
}
```

【3.50】下面程序的功能是从键盘上顺序输入整数，直到输入的整数小于 0 时才停止输入。然后反序输出这些整数。

```
#include <stdio.h>
struct data
{ int x ；
struct data *link  ；
}*p ；
input()
{ int num ；
struct data *q ；
printf("Enter data:")  ；
scanf("%d", &num)  ；
if( num<0 )
①  ；
```

```
q = ②  ;
q->x = num  ;
q->link = p  ;
p=q ;
③  ;
}
main()
{ printf("Enter data until data<0:\n")     ;
p=NULL  ;
input() ;
printf("Output:")   ;
while( ④ )
{ printf("%d\n", p->x)    ;
⑤  ;
}
}
```

【3.51】下面函数的功能是创建一个带有头结点的链表，将头结点返回给主调函数。链表用于储存学生的学号和成绩。新产生的结点总是位于链表的尾部。

```
struct student
{ long num;
int score ;
struct student *next  ;
} ;
struct student *creat()
{ struct student *head=NULL,*tail    ;
long num ;  int a ;
tail=  ①  malloc(LEN)  ;
do
{ scanf("%ld,%d",&num,&a)     ;
if(num!=0)
{ if(head==NULL) head=tail    ;
else ②   ;
tail->num=num  ;  tail->score=a ;
tail->next=(struct student *)malloc(LEN)    ;
}
else tail->next=NULL  ;
}while(num!=0)  ;
return( ③ ) ;
}
```

【3.52】下面 **create** 函数的功能是建立一个带头结点的单向链表，新产生的结点总是插入在链表的末尾。单向链表的头指针作为函数值返回。

```
#include <stdio.h>
#define LEN sizeof(struct student)
struct student
{ long num；
int score；
struct student *next ；
}；
struct student *creat()
{ struct student *head=NULL,*tail    ；
long num；
int a；
tail=(  ①  )malloc(LEN)  ；
do
{ scanf("%ld,%d",&num,&a)     ；
if(num!=0)
{ if(head==NULL) head=tail    ；
else tail=tail->next ；
tail->num=num  ；
tail->score=a；
tail->next=(  ②  )malloc(LEN)  ；
}
else tail->next=NULL   ；
}while(num!=0)  ；
③   ；
}
```

【3.53】下面程序的功能是统计文件中的字符的个数。

```
#include <stdio.h>
main()
{ long num=0 ；
①  *fp ；
if((fp=fopen("fname.dat", "r"))==NULL)
{ printf("Can't open the file! ")    ；
exit(0)；
}
while(  ②  )
{ fgetc(fp) ；
num++ ；
}
printf("num=%d\n",num)     ；
fclose(fp)；
}
```

【3.54】下面程序的功能是把从键盘输入的文件（用　　@　作为文件结束标志）复制到一个名为 second.txt 的新文件中。

```
#include <stdio.h>
FILE *fp  ;
main()
{ char ch ;
if((fp=fopen(  ①  ))==NULL)
exit(0) ;
while((ch=getchar())!='@')
fputc(ch,fp) ;
②  ;
}
```

【3.55】下面程序的功能是将磁盘上的一个文件复制到另一个文件中,　　两个文件名在命令行中给出（假定给定的文件名无误）　。

```
#include <stdio.h>
main(int argc,char *argv[])
{ FILE &f1,*f2   ;
if(argc<  ①  )
{ printf("The command line error! ")   ;
exit(0) ;
}
f1=fopen(argv[1], "r")   ;
f2=fopen(arhv[2], "w")   ;
while(  ②  )
fputs(fgetc(f1),  ③  ) ;
④  ;
⑤  ;
}
```

【3.56】下面程序的功能是根据命令行参数分别实现一个正整数的累加或阶乘。例如：如果可执行文件的文件名是　sm, 则执行该程序时输入：　"sm + 10" , 可以实现　10 的累加； 输入："sm - 10" , 可以实现求　10 的阶乘。

```
#include <stdio.h>
#include <stdlib.h>
main (int argc,char *argv[])
{ int n  ;
void sum(),mult()  ;
void (*funcp)()  ;
n=atoi(argv[2])  ;
if(argc!=3 || n<=0)
dispform( ) ;
switch (  ①  )
```

```
{ case '+': funcp=sum ；
break ；
case '-': funcp=mult ；
break ；
default: dispform( ) ；
}
②  ；
}
void sum(int m)
{ int i,s=0 ；
for(i=1 ； i<m ； i++ )
③  ；
printf("sum=%d\n",s)  ；
}
void mult(int m)
{ long int i, s=1 ；
for(i=1 ； i<=m ； i++ )
s *= i ；
printf("mult= %ld\n"  ； s) ；
}
dispform( )
{ printf ("usage:sm n(+/!) (n>0)\n")  ；
exit (0) ；
}
```

【3.57】下面程序的功能是键盘上输入一个字符串，把该字符串中的小写字母转换为大写字母，输出到文件 test.txt 中，然后从该文件读出字符串并显示出来。

```
#include <stdio.h>
main()
{ char str[100] ；
int i=0 ；
FILE *fp  ；
if((fp=fopen("test.txt",  ①  ))==NULL)
{ printf("Can't open the file.\n")  ；
exit(0) ；
}
printf("Input a string:\n")  ；
gets(str) ；
while(str[i])
{ if(str[i]>= 'a'&&str[i]<= 'z')
str[i]=  ②  ；
fputc(str[i],fp)  ；
i++ ；
}
```

```
fclose(fp)；
fp=fopen("test.txt",  ③  )；
fgets(str,strlen(str)+1,fp) ；
printf("%s\n",str)   ；
fclose(fp)；
}
```

【3.58】下面程序的功能是将从终端上读入的 10 个整数以二进制方式写入名为 "bi.dat" 的新文件中。

```
#include <stdio.h>
FILE *fp   ；
main()
{ int i, j  ；
if(( fp=fopen(  ①  , "wb" )) == NULL )
exit (0) ；
for( i=0 ； i<10 ； i++ )
{ scanf("%d", &j )   ；
fwrite(  ②  , sizeof(int), 1,  ③  ) ；
}
fclose( fp) ；
}
```

【3.59】以字符流形式读入一个文件，从文件中检索出六种C语言的关键字，并统计、输出每种关键字在文件中出现的次数。本程序中规定：单词是一个以空格或 '\t'、 '\n' 结束的字符串。

```
#include <stdio.h>
#include <string.h>
FILE *cp   ；
char fname[20], buf[100]  ；
int num ；
struct key
{ char word[10]  ；
int count ；
}keyword[]={ "if", 0, "char", 0, "int", 0,
"else", 0, "while", 0, "return", 0}    ；
char *getword (FILE *fp)
{ int i=0  ；
char c ；
while((c=getc(fp)) != EOF && (c==' '||c=='\t'||c=='\n'))       ；
if( c==EOF ) return (NULL)       ；
else buf[i++]=c ；
while((c =  ①  && c!= ' ' && c!= '\t' && c!= '\n' )
buf[i++] = c  ；
buf[i]= '\0'   ；
```

```
return(buf)  ;
}
lookup(char *p)
{ int i  ;
char *q, *s  ;
for(i=0  ;  i<num  ;  i++)
{ q =  ②   ;
s=p ;
while( *s && (*s==*q) )
{  ③
}
if(  ④  )
{ keyword[i].count++  ;
break  ;
}
}
return  ;
}
main()
{ int i  ;
char *word  ;
printf("Input file name:")    ;
scanf("%s", fname)  ;
if((cp=fopen(fname, "r")) ==NULL )
{ printf("File open error: %s\n", fname)    ;
exit(0)  ;
}
num = sizeof(keyword) / sizeof(struct key)  ;
while(  ⑤  )
lookup(word)  ;
fclose(cp) ;
for(i=0  ;  i<num  ;  i++)
printf("keyword:%-20scount=%d\n",keyword[i].word,keyword[i].count)       ;
}
```

【3.60】下面程序的功能是从键盘接受姓名（例如：输入 **"ZHANG SAN"**），在文件 **"try.dat"** 中查找，若文件中已经存入了刚输入的姓名，则显示提示信息；若文件中没有刚输入的姓名，则将该姓名存入文件。要求： ?若磁盘文件 **"try.dat"**，已存在，则要保留文件中原来的信息；若文件 **"try.dat"** 不存在，则在磁盘上建立一个新文件； ?当输入的姓名为空时（长度为 **0**），结束程序。

```
#include <stdio.h>
main()
{ FILE *fp  ;
int flag ;
```

```
char name[30], data[30]  ;
if((fp=fopen("try.dat",        ①     ))==NULL )
{ printf("Open file error\n")     ;
exit(0) ;
}
do
{ printf("Enter name:")     ;
gets(name) ;
if( strlen(name)==0 )
            break ;
strcat(name, "\n")   ;
   ②      ;
flag=1 ;
while( flag && (fgets(data, 30, fp)       ③     ) )
if( strcmp(data, name) == 0 )
④   ;
if( flag )
fputs(name, fp) ;
else
printf("\tData enter error !\n")     ;
} while(       ⑤       ) ;
fclose(fp) ;
}
```

【程序填空题参考答案】

【3.1】答案：① **a+b** ② **a-b** ③ **a-b**

【3.2】答案：① **1.0/(float)(i*i)** ② **sqrt(6*s)**

【3.3】答案：① ***k=p**

【3.4】答案：① **t=t*i** ② **t=t>0?-1:1**

【3.5】答案：① **d=1** ② **k++** ③ **k<=n**

【3.6】答案：① **x>=0** ② **x<amin**

【3.7】答案：① **2** ② **2**

【3.8】答案：① **m=n** ② **m>0** ③ **m=m/10**

【3.9】答案：① **i==j** ② **k!=i&&k!=j**

【3.10】答案：① **i<=9** ② **j%3!=0**

【3.11】答案：① **m=n** ② **r!=0** ③ **return(n)**

【3.12】答案：① **float a[10],x** ② **i<=9** ③ **i<=8** ④ **j<=9-i**
⑤ **a[j]>a[j+1]** ⑥ **a[j]=a[j+1]** ⑦ **i<=9** ⑧ **i%5==0**

【3.13】答案：① **&a[i]** ② **continue**

注释：①是基本概念，使用 **scanf** 函数输入数组元素的值。当输入的元素值小于 **0** 时，应当跳过后面的语句，取下一个数，所以②要填入 **continue**。

【3.14】答案：① **s[j++]=s[i]** ② **s[j]=s[i]**

【3.15】答案：① **s[j++]=s[i]**

【3.16】答案：① **x[i++]**

【3.17】答案：① **i=strlen(a)** ； **i>=j** ； **i--** ② **a[i+1]=a[i]**

【3.18】答案：① **'\0'** ② **str1[i]-str2[i]**

【3.19】答案：① **j++** ② **'\\'** ③ **j++** ④ **'\\'** ⑤ **j++** ⑥ **'\0'**

【3.20】答案：① **t[k]!= '\0'** ② **t[k]== '\0'**

【3.21】答案：① **<=n** ② **s** ③ **0** ④ **fun(k)**

【3.22】答案：① **a[fmax(a,N)]** ② **k=p**

【3.23】答案：① **last*n%1000** ② **x!=min-x** ③ **flag=0**

【3.24】答案：① **(x0+a/x0)/2** ② **a,x1**

注释：根据迭代公式，①处应当是计算迭代值 **x1=(x0+a/x0)/2** 。按照求平方根的要求，当迭代的精度不能满足 **"(fabs(x1-x0)>0.00001)"** 时，则要继续迭代，因此②处应当填写 **"a,x1"** 。程序中调用了求绝对值的库函数 **fabs( )**。

【3.25】答案：① **2+age(n-1)** ② **age(5)**

注释：由于程序是递归算法，因此首先要建立问题的递归数学模型。根据原题的描述可以写出如下递归公式：

age(n) = 10 (n=1)

age(n) = 2+age(n-1) (n>1)

对照程序和递归公式可以看出： **n** 的含义是第 **n** 位学生。很显然，要求第 **5** 位学生的年龄，②处应当是调用函数 **age**，实参的值应当是 **5**。在①处应该是函数的递归调用，根据递归公式，应当填写： **2+age(n-1)**。

【3.26】答案：① **return(1)** ② **return (sum(n-1)+n)**

注释：按照常规的编程方法，此问题可采用一个循环语句实现。阅读程序，没有发现循环语句，这时，应当认为原来的编程者使用的是非常规的算法。对于这样常规算法需要用循环实现而没有使用循环的程序，就可以肯定地认为，一定是使用了递归算法。

将问题 **"**求１～ **n** 的累加和 **"**的公式写成递归定义，可以是如下形式；

sum(n)=1 当 **n=1** 时

sum(n)=sun(n-1)+n 当 **n>1** 时

根据此递归定义，可以很容易完成程序。

【3.27】答案：① **return(1)** ② **n*facto(n-1)**

注释：我们熟悉的求 **n!**的算法一般是采用循环语句实现，但在此程序中根本没有循环语句。这时我们应该想到：是采用递归算法实现的。首先写出求 **n**！的递归公式；

n!=1 当 **n=1** 时

n!=n*(n-1) 当 **n>1** 时

根据此递归定义，可以很容易完成程序。

【3.28】答案：① **com=n** ② **com=combin(n-1,m-1) + combin(n-1,m)**

注释：题目的说明中已经给出组合问题的递归定义，不需要读者自己寻找递归表达式。程序中的语句 **"if (n<2*m) m=n-m ；"**完成了题目中叙述的 **"**用公式 **(1)**进行简化 **"**的工作。

【3.29】答案：① ***str=='\0'** ② **1+strlen(str+1)**

注释：求串长算法的关键是确定串结束标记 **'\0'** 的位置。根据求串长的方法，可以得到如下递归算法：指针 **str** 指向字符串的首字符

如果 当前字符（ ***str** ）**==** 串结束标记 **'\0'**

则 串长 **=0**

否则 串长 **= 1**＋除第一个字符之外的剩余字符串的串长

因此，在①的位置上应当填写 **"*str=='\0'"** ，以判断当前字符（ ***str** ）是否是串结束标记 **'\0'**。在②的位置应当是根据上面的递归算法进行递归调用，因此应当填写 **"1+strlen(str+1)"** 。

【3.30】答案：① **&n** ② **m%10** ③ **m/10** ④ **m>0** ⑤ **r(m)**

【3.31】答案：① **prt(c, n-1)** ② **prt(' ' , n-i)** ③ **prt('*', i)**

注释：函数 **prt** 的功能是输出 **n** 个字符 **c**。

【3.32】答案：① **y(x, n-1)**

注释：这显然是一个递归问题，首先要对原来的数学函数定义形式进行变形，推导出原来函数的等价递归定义。可以推导出原来函数的递归定义如下。

y(x,n)=x 当 **n=0** 时

y(x,n)=sqrt(x+y(x,n-1)) 当 **n>0** 时

【3.33】答案：① **p--** ② ***p='\0'** ③ ***p=c** ④ **p--** ⑤ ***s++**

注释：在递归算法中，指针 **s** 指向字符串首部要反向的字符，即要将指针 **s** 所指向的字符与指针 **p** 所指向的字符串尾的字符（**'\0'**）进行交换，在交换过程中，将尚没有交换的字符串的中间部分作为一个整体，进行递归处理。程序中首先执行 **"c=*s"**，将首字符存入临时变量；然后执行 **"*s=*p"**，将尾字符存入串首；执行 **"revstr(s+1)"** 是递归处理串的中间部分，这时，在②处应当填入 **"*p='\0'"**，即存入串结束标记。这是这一程序中的关键所在。在③处要完成将存在临时变量 **c** 中的字符存入串尾的工作，应当填写 **"*p=c"**。

【3.34】答案：① **str+1** ② **return** ①改为 **n-2**

【3.35】答案：① **p>=q** ② **max=s** ③ **p,max**
④ **p+1,q** ⑤ **&array[0], &array[9]**

注释：本程序中的排序部分采用的是递归算法。函数 **sort** 的两个形参的含义是：对指针 **p** 和指针 **q** 之间的数据进行排序。由语句 **"for(s=p+1 ； s<=q； s++)"** 中指针 **p** 和指针 **q** 之间的关系可以得出：指针 **p** 不应在指针 **q** 之后，因此①处应填 **"p>=q"**、⑤处应填 **"&array[0],&array[9]"**。

由于变量 **max** 是指向当前最大值的指针，则当找到新的最大值时，**max** 中保存的应该是新的最大值的指针，因此②处应填 **"max=s"**。

当调用函数 **swap** 交换两个变量值的时候，要求实参是变量的地址，因此，③处应填 **"p,max"** 将最大值存入指针 **p** 所指的单元。

由于问题的要求是 **"** 从大到小 **"** 排序，通过执行一次函数 **sort** 使最大值已经放到了指针 **p** 所指的单元中，因此，下一遍排序的时候，只要对指针 **p** 之后的元素进行即可，所以④处应填 **"p+1,q"**。

【3.36】答案：① **a+1** ② **n%10+'0'**

【3.37】答案：① **s<t** ② **s++** ③ **t--**

【3.38】答案；① ***(p+4*i+j)**

注释： **p** 是一个一级指针，赋值后保存二维数组 **a** 的首地址 **,**做加法运算加 **1** 时，实际地址增加一个它所指向的数据类型的长度。在 **C** 语言中，多维数组在计算机中是按行存储的，所以在本题中要通过指针访问二维数组中的数据，必须将二维下标转换为一维下标。

【3.39】答案：① **'\0'** 或 **0** ② **++**

注释：在 **C** 语言中，进行字符串处理时，必须注意串结束标记 **'\0'**，它是在进行串处理时的最基本的要求，所以①中要填入 **'\0'**。为了使用 **putchar** 输出一个字符串，则必须有改变指针的运算，这里只能使用 **++**运算。

【3.40】答案：① **p++** ② **w[i+1]=w[i]**

【3.41】答案：① **&&** ② ***a<*b** ③ ***a++, *b++** ④ **==**
⑤ **!=** ⑥ **'\0'** ⑦ **w=s** ⑧ **n++**
⑨ **t=s[i]； s[i]=s[j]； s[j]=t；** ⑩ **s3[0]**

【3.42】答案：① ***++pa** ② ***++pa** ③ ***pa**
④ **return(total)** ⑤ **n<=2 || n>=MAXNUM+1** ⑥ **num+n**

【3.43】答案：① **stu.name** ② **&stu.score** ③ **p->name** ④ **p->score**

注释：这是结构中的最基本概念。

【3.44】答案：① **struct student** ② **strcmp(stu[i].name， str)==0** ③ **break**

注释：程序的主体是一个二重循环，内层 **for** 循环完成查找学生的工作。①处是进行结构数组说明并初始化，按照结构变量说明的格式规定，应该填写：**strcut student**。②处为 **if** 语句的逻辑条件，应当是当查找到指定的学生后输出学生的情况，因此应当填写：**strcmp(stu[i].name,str)==0** 。③处应当将控制退出内层的 **for** 循环，只能选择 **break** 语句。

【3.45】答案：① **p+n** ② **gets(p->name)** ③ **p+n**

注释：本程序是通过函数完成对于结构数组的输入和输出操作。函数 **data_in** 和 **data_out** 十分相似，都是通过结构指针 **p** 和结构指针 **q** 来操作结构数组的元素。由于指针 **q** 在两个函数中的作用相同，所以①和③填写的内容也应该是相同的；由 **for** 语句中的循环终止条件 **"p<q"** 可以看出，**q** 应该指在数组的最后一个元素之后，所以①和③应当填入 **p+n**。②应当完成姓名的输入工作，应当为 **gets(p->name)**。

【3.46】答案：① **j=i-1** ② **>** ③ **j--** ④ **j+1** ⑤ **j+1**

注释：程序的基本思想是：对于输入的第 **i** 个整数 **num**，从数组 **array** 中已有的元素中倒序开始查找。若数组 **array** 中的第 **j** 个元素的值大于 **num**，则将数组中的元素 **j** 向后移动一个位置；否则，就应将 **num** 插入到当前位置作为元素 **j**。因此，程序的基本设计思想就是插入排序。

程序中内层的 **for** 循环完成查找插入位置的工作，因此答案①、②和③有密切的关系，要统一考虑。同样，程序中的答案④和⑤也有密切的关系，要统一考虑。

【3.47】答案：① **!= '\n'** ② **p=top**

【3.48】答案：① **p1->next**

【3.49】答案：① **head** ② **p** ③ **p=head**

注释：程序在从键盘接受字符的同时就在建立起链表，所建立的链表本身就已经是反序排列的，因此在反序输出字符串的时候实际只需沿着链表的第一个结点开始，顺序操作即可。

【3.50】答案：① **return** ② **(struct data *) malloc(sizeof(struct data))** ③ **input( )** ④ **p!=NULL** ⑤ **p=p->next**

【3.51】答案：① **(struct student *)** ② **tail=tail->next** ③ **head**

注释：① **malloc** 函数的作用是在内存开辟指定字节数的存储空间，并将此存储空间的地址返回赋给尾指针 **tail**，但是此地址为 **void** 型，应将其强制转换为所要求的结构指针类型。②新开辟的结点的内存地址存于 **tail** 所指向的已建立的链表的尾结点的结构成员 **next**，新结点连入链表以后，尾指针 **tail** 应指向新的结点。

【3.52】答案：① **(struct student *)** ② **(struct list *)** ③ **return(head)**

【3.53】答案：① **FILE** ② **!feof(fp)**

注释：**FILE** 是文件结构类型名。**feof()**是测试文件结束标志的函数。

【3.54】答案：① **"second.txt"** ② **fclose(fp)**

【3.55】答案：① **3** ② **!feof(f1)**或 **feof(f1)==0** ③ **f2** ④ **fclose(f2)** ⑤ **fclose(f1)**

注释：程序中使用了带参数的 **main** 函数，其中整型参数 **argc** 为命令行中字符串的个数，此程序运行时输入的字符串有可运行程序名、文件 **1** 和文件 **2**, 故 **argc** 不应小于 **3**。字符串指针 **argv[0]** 指向可运行程序名、字符串指针 **argv[1]**指向输入文件名、字符串指针 **argv[2]** 指向输出文件名，由上所述②处给出循环条件是输入文件是否结束，③处需要填出输出文件名。最后两处是关闭两个文件，原则上关闭文件没有顺序要求，但习惯上是后打开的文件先关闭。

【3.56】答案：① ***argv[1]** ② **(*funcp)(n)** ③ **s+=i**

注释：程序执行时输入的命令及参数的个数（操作系统规定用空格表示字符串的分隔）由系统赋给主函数的形数 **argc**，输入的命令和参数以字符串的格式保存，字符串的首地址分别赋给指针数组 **argv** 的各个元素，其中 **argv[1]**是 **'+'** 或 **'-'**，分别表示累加或阶乘。程序根据 **argv[1]** 所指向的字符串的内容给指向函数的指针变量 **funcp** 赋值。②处要求的语句是根据指向函数的指针变量的内容对相应的函数实现调用，所以选择 **A** 或 **B** 是错误的；据 **funcp** 是被调函数的地址，***funcp** 实现了对函数的调用，根运算符的结合性，**(*funcp)** 表示取 **funcp** 的目标，而 ***funcp(n)**

则 **funcp** 先和 **(n)**结合， **funcp** 就被解释为函数名，显然是错误的。

【3.57】答案：① **"w"** ② **-32** ③ **"r"**

【3.58】答案：① **"bi.dat"** ② **&j** ③ **fp**

【3.59】答案：① **fgetc(fp))!=EOF** ② **&keyword[i].word[0]**
③ **s++**； **q++**； ④ ***s==*q** ⑤ **(word=getword(cp))!=NULL**

【3.60】答案：① **"a+"** ② **rewind(fp)** ③ **!=NULL** ④ **flag=0** ⑤ **ferror(fp)==0**

四、编写程序题及参考答案

导读：虽然题目基本按照教材章节顺序排列的，但是把同类题目尽量排在一起，便于读者学习掌握编程方法和思路，提高自己的编程能力。

【4.1】已知银行整存整取存款不同期限的月息利率分别为：

0.315% 期限一年

0.330% 期限二年

月息利率 = **0.345%** 期限三年

0.375% 期限五年

0.420% 期限八年

要求输入存钱的本金和期限，求到期时能从银行得到的利息与本金的合计。

【4.2】输入年份 **year** 和月 **month**，求该月有多少天。判断是否为闰年，可用如下C语言表达式：**year%4==0 && year%100!=0 || year%400==0** 。若表达式成立（即表达式值为 **1**），则 **year** 为闰年；否则，表达式不成立（即值为 **0**）， **year** 为平年。

【4.3】编写一个简单计算器程序，输入格式为： **data1 op data2**。其中 **data1** 和 **data2** 是参加运算的两个数， **op** 为运算符，它的取值只能是 **+**、**-**、*、*/*。

【4.4】输入 **n** 值，输出如图所示矩形。

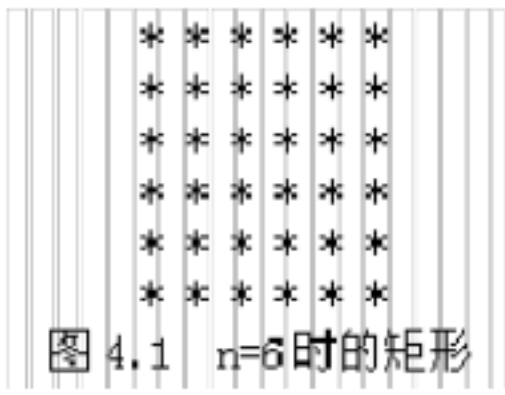


图 4.1 n=6 时的矩形

【4.5】输入 **n** 值，输出如图所示平行四边形。

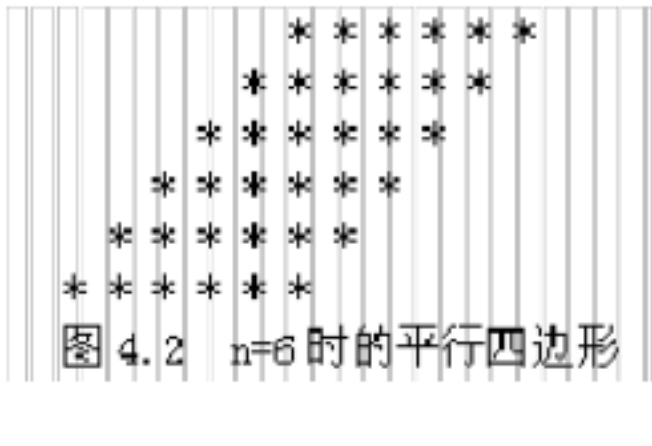
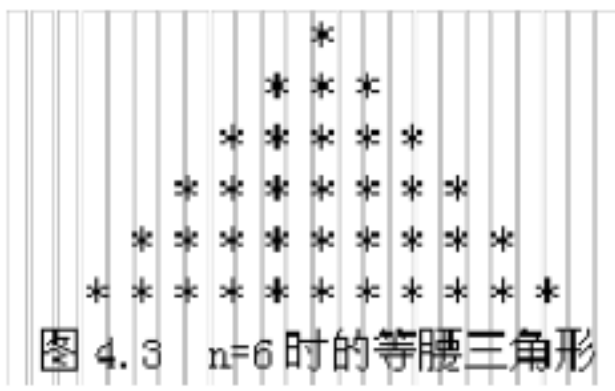
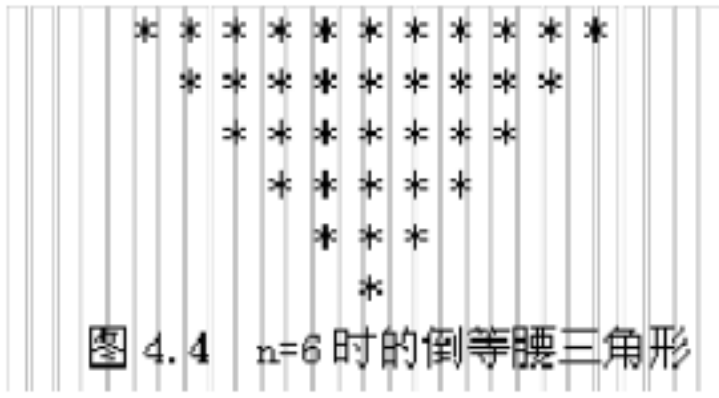


图 4.2 n=6 时的平行四边形

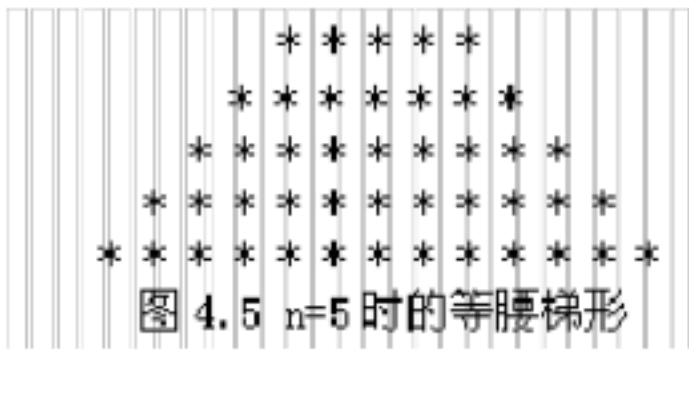
【4.6】输入 **n** 值，输出如图所示高为 **n** 的等腰三角形。



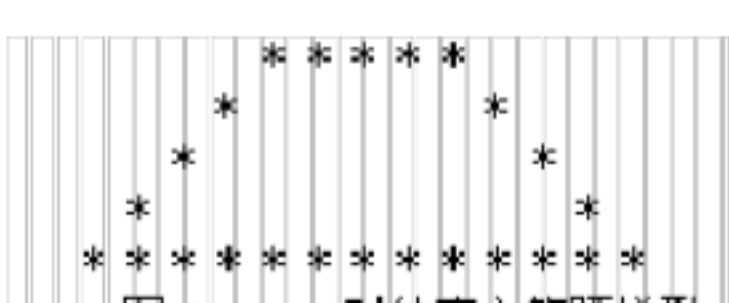
【4.7】输入 **n** 值，输出如图所示高为 **n** 的等腰三角形。



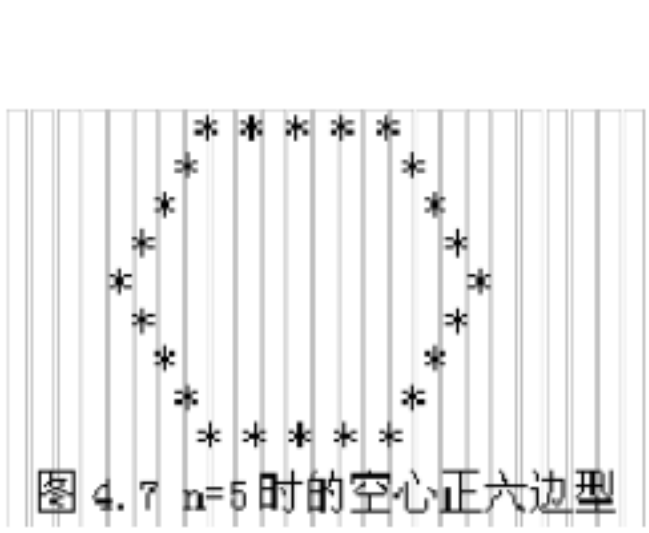
【4.8】输入 **n** 值，输出如图所示高和上底均为 **n** 的等腰梯形。



【4.9】输入 **n** 值，输出如图所示高和上底均为 **n** 的等腰空心梯形。



【4.10】输入 **n** 值，输出如图所示边长为 **n** 的空心正六边型。



【4.11】输入 **n** 值，输出如图所示图形。

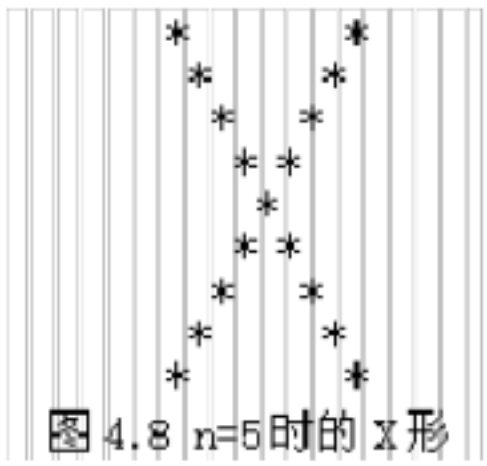


图 4.8 n=5 时的 X 形

【4.12】输入 **n** 值，输出如图所示图形。

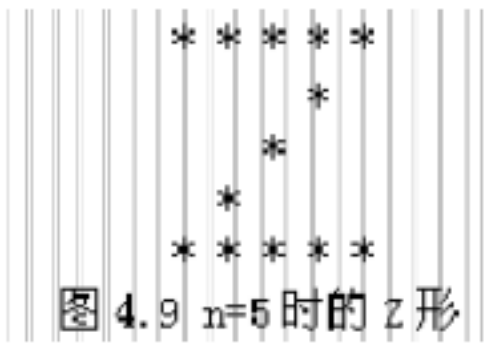


图 4.9 n=5 时的 Z 形

【4.13】输入 **n** 值，输出如图所示图形。

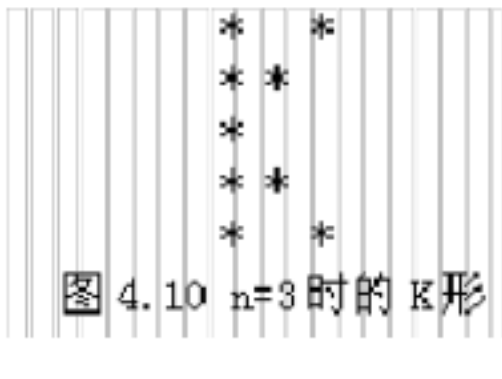


图 4.10 n=3 时的 K 形

【4.14】输入 **n** 值，输出如图所示图形。

图 4.12 n=5 时的 N 形

【4.15】输入 **n** 值，输出如图所示图形。

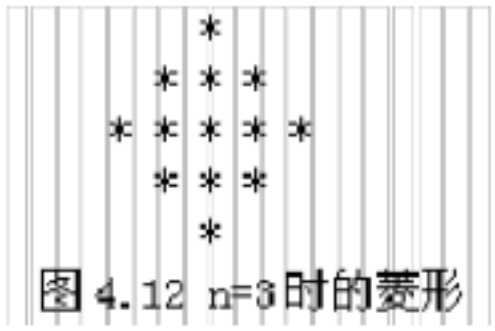
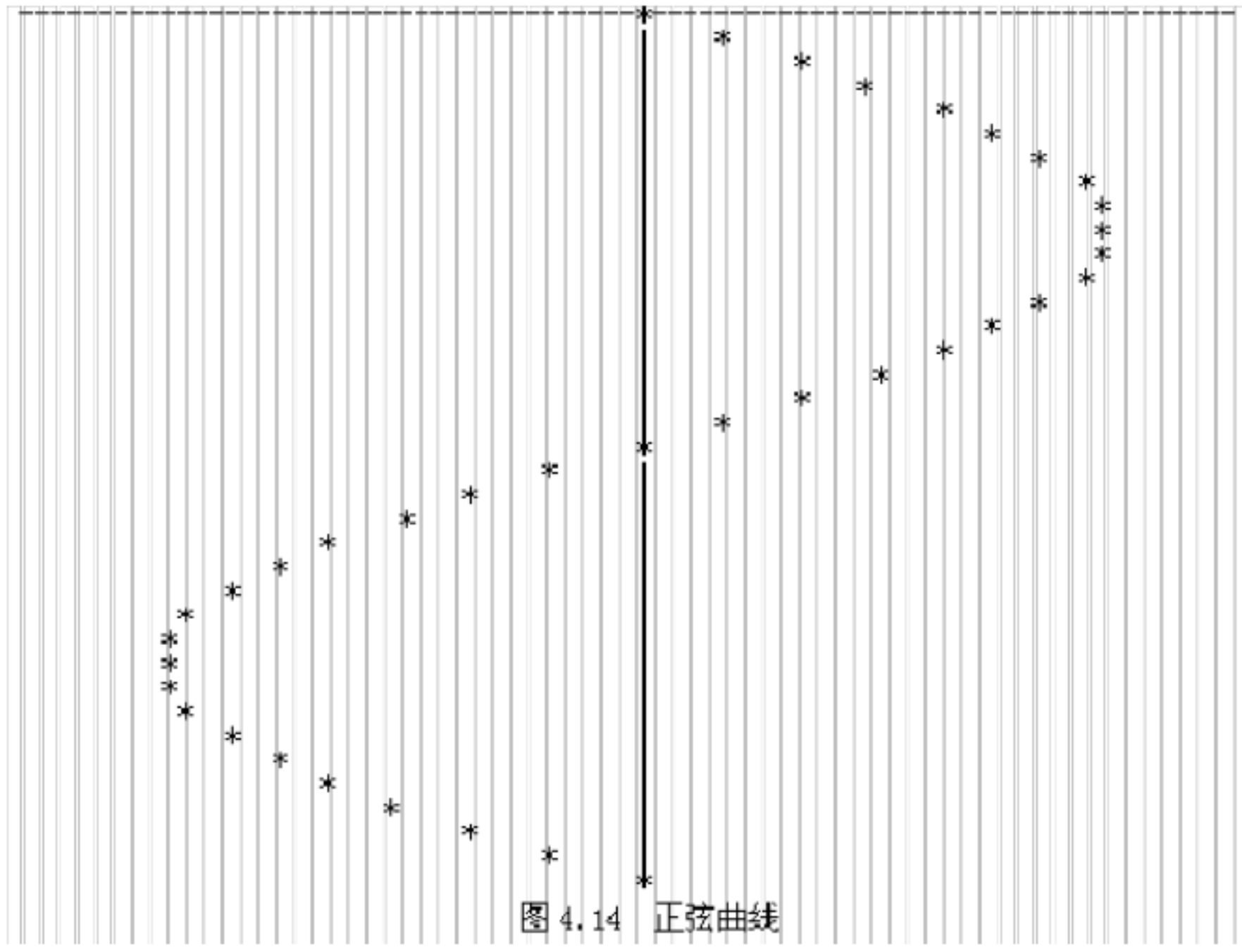


图 4.12 n=3 时的菱形

【4.16】输入 **n** 值，输出如图所示图形。 （例为 **n=6** 时）

【**4.17**】编写程序，输出如图所示 **sin(x)** 函数 **0** 到 **2**π 的图形。



【**4.18**】编写程序，在屏幕上输出一个由 *****号围成的空心圆。

【**4.19**】编写程序， 在屏幕上绘制如图余弦曲线和直线。 若屏幕的横向为 **x** 轴，纵向为 **y** 轴， 在屏幕上显示 **0**～**360** 度的 **cos(x)**曲线与直线 **x=f(y)=45*(y-1)+31** 的迭加图形。 其中 **cos**图形用 **"*"** 表示， **f(y)**用**"+"** 表示， 在两个图形的交点处则用 **f(y)** 图形的符号。

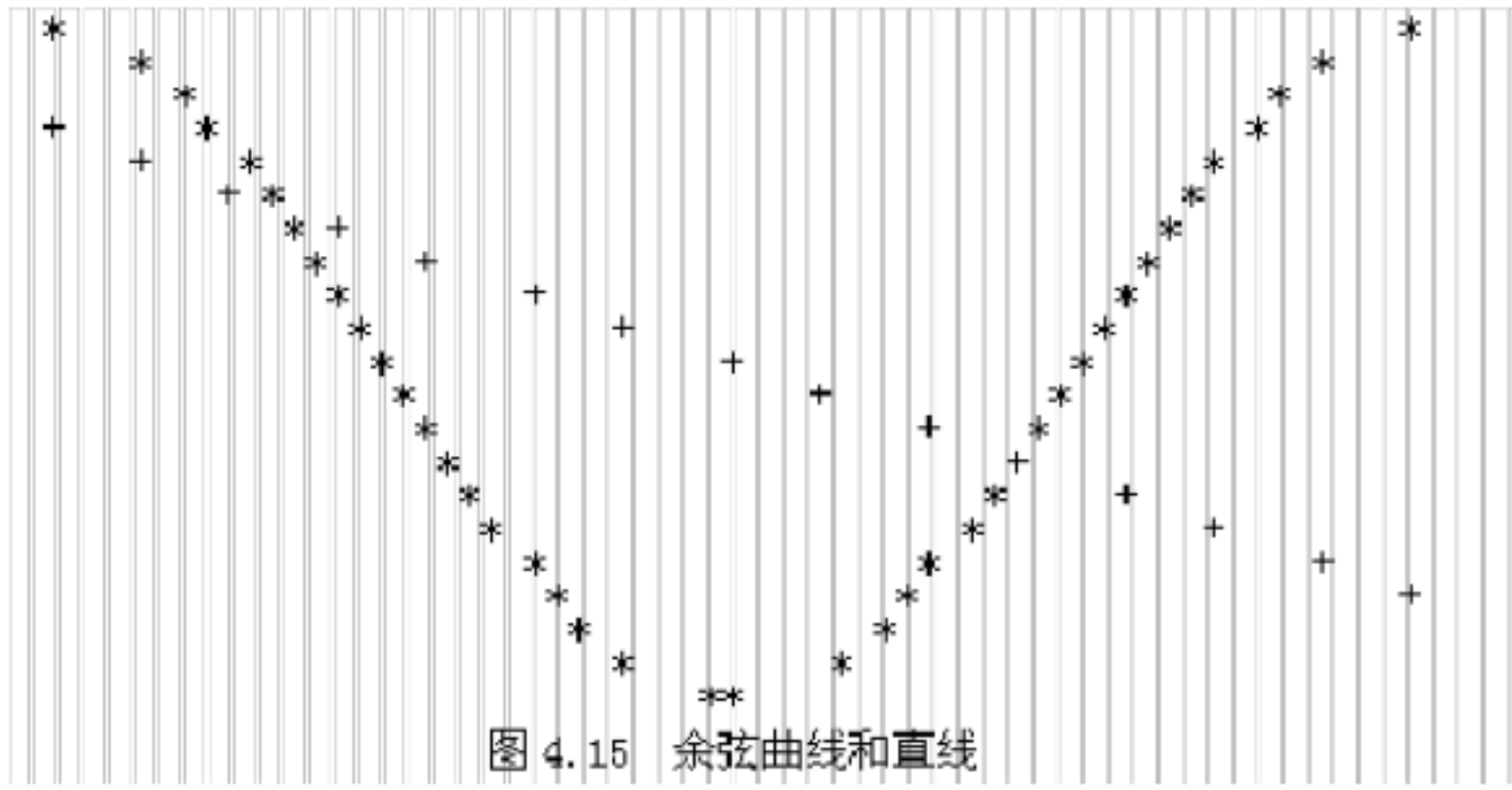


图 4.15　余弦曲线和直线

【4.20】编写程序，输出如图所示高度为　　**n** 的图形。

```
 1   2   3   4   5   6
 7   8   9  10  11  12
13  14  15  16  17  18
19  20  21  22  23  24
25  26  27  28  29  30
31  32  33  34  35  36
```

图 4.16　n=6 时的数字正方形

【4.21】编写程序，输出如图所示高度为　　**n** 的图形。

```
        1   3   6  10  15  21
        2   5   9  14  20
        4   8  13  19
        7  12  18
       11  17
       16
```

图 4.17　n=6 时的数字倒三角

【4.22】输入　**n** 值，输出如图所示图形。

```
1  2  3  4  5
1  1  2  3  4
1  1  1  2  3
1  1  1  1  2
1  1  1  1  1
```

图 4.18 n=5 时的数字矩形

【4.23】输入 **n** 值，输出如图所示的 **n×n(n<10)**阶螺旋方阵。

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

图 **4.19 n=5** 时的螺旋方阵

【4.24】输入 **n** 值，输出如图所示回型方阵。

当 n=5 时：

```
3  2  3  2  3
3  2  2  2  3
3  2  1  2  3
3  3  2  3  3
3  3  3  3  3
```

当 n=6 时：

```
3  3  3  3  3  3
3  2  2  2  2  3
3  2  1  1  2  3
3  2  1  1  2  3
3  2  2  2  2  3
3  3  3  3  3  3
```

图 4.20 回形方阵

【4.25】输出如图所示的数字金字塔

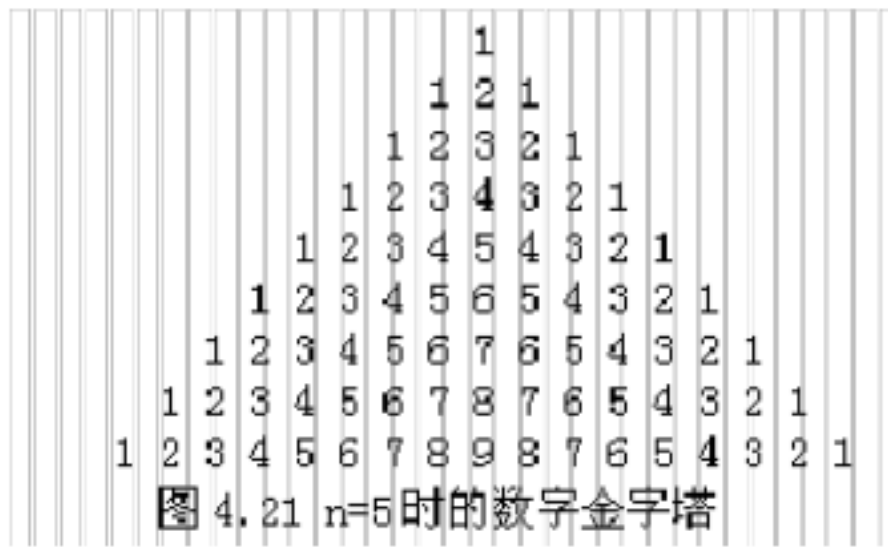


图 4.21 n=5时的数字金字塔

【4.26】输入 n 值，输出如图所示图形。

图 4.22 n=5时的空心菱形

【4.27】输入顶行字符和图形的高，输出如图所示图形。

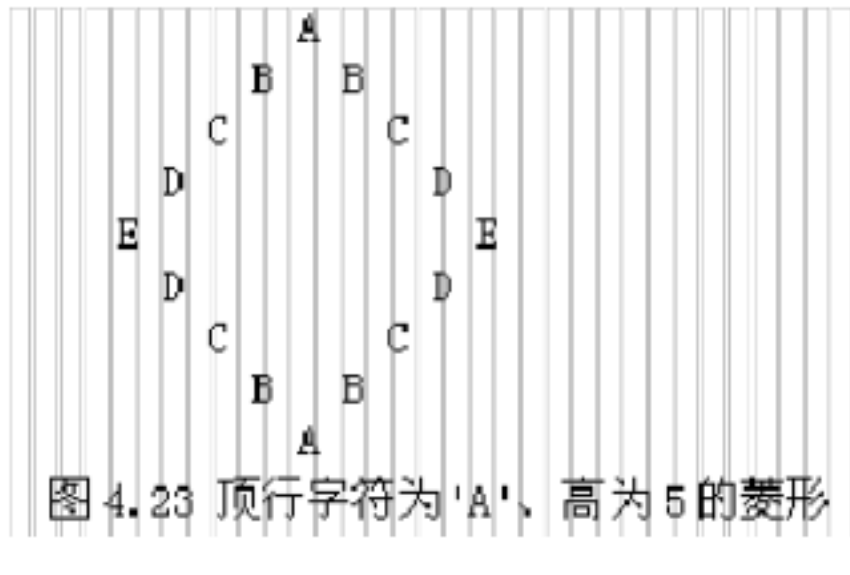


图 4.23 顶行字符为'A'、高为5的菱形

【4.28】输入首字符和高后，输出如图所示回型方阵。

A A A A A

A B B B A

A B C B A

A B B B A

A A A A A

图 **4.24** 首字符为 **'A'**、高为 **5** 的方阵

【4.29】输入中心字符和高后，输出如图所示回型方阵。

X X X X X

X Y Y Y X

X Y Z Y X

X Y Y Y Y

X X X X X

图 **4.25** 中心字符为 **'Z'**、高为 **5** 的方阵

【4.30】编写程序，输出如图所示上三角形式的乘法九九表。

```
1  2  3  4  5  6  7  8  9
-----------------------
1  2  3  4  5  6  7  8  9
   4  6  8 10 12 14 16 18
      9 12 15 18 21 24 27
        16 20 24 28 32 36
           25 30 35 40 45
              36 42 48 54
                 49 56 63
                    64 72
                       81
```

图 4.26 上三角乘法九九表

【4.31】编写程序，输出如图所示下三角乘法九九表。

1	2	3	4	5	6	7	8	9
								81
							64	72
						49	56	63
					36	42	48	54
				25	30	35	40	45
			16	20	24	28	32	36
		9	12	15	18	21	24	27
	4	6	8	10	12	14	16	18
1	2	3	4	5	6	7	8	9

图 4.27 下三角乘法九九表

【4.32】编写程序，输入三角型的三条边长，求其面积。注意：对于不合理的边长输入要输出数据错误的提示信息。

【4.33】编写程序求出 **555555** 的约数中最大的三位数是多少。

【4.34】编写程序计算下列算式的值：

$$C = 1 + \frac{1}{x^1} + \frac{1}{x^2} + \frac{1}{x^3} + \frac{1}{x^4} \cdots\cdots \quad (x>1)$$

直到某一项 **A<=0.000001** 时为止。输出最后C的值。

【4.35】从键盘输入任意的字符，按下列规则进行分类计数。

第一类 **'0'**，**'1'**，**'2'**，**'3'**，**'4'**，**'5'**，**'6'**，**'7'**，**'8'**，**'9'**

第二类 **'+'**，**'-'**，**'*'**，**'/'**，**'%'**，**'='**

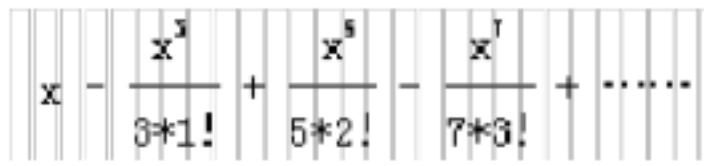
第三类 其它字符

当输入字符 **'\'** 时先计数，然后停止接收输入，打印计数的结果。

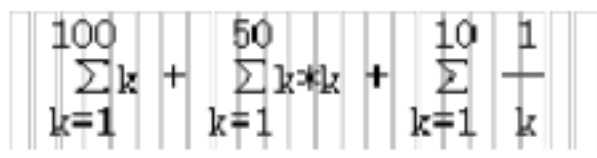
【4.36】对从键盘上输入的行、单词和字符进行计数。我们将单词的定义进行化简，认为单词是不包含空格、制表符 **(\t)** 及换行符的字符序列。例如： **"a+b+c"**，认为是 1 个单词，它由 5 个字符组成。又如： **"xy abc"**，为 2

个单词，６个字符。一般用 **[CTRL+D]** 作为文件结束标记，其字符码值为 **-1**，当输入 **[CTRL+D]** 时表示文件输入结束，停止计数。

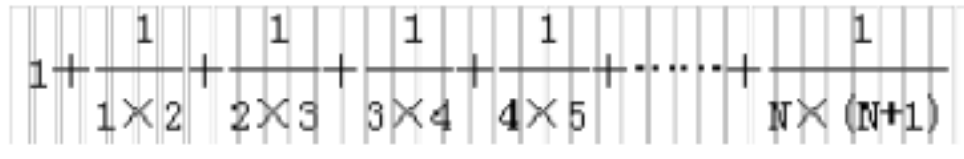
【4.37】编写程序计算当 **x=0.5** 时下述级数和的近似值，使其误差小于某一指定的值 **epsilon**（例如：**epsilon=0.000001**）：



【4.38】编写程序计算下式的值：

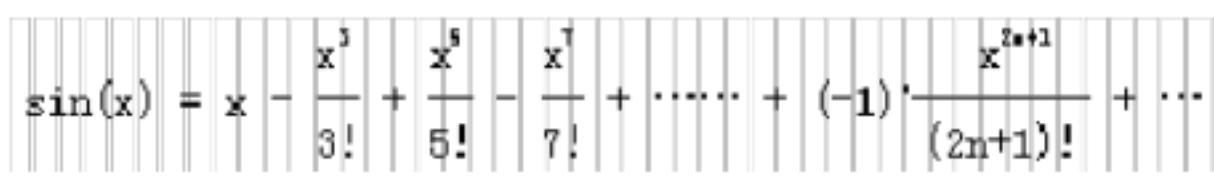


【4.39】编写程序计算下列序列的值：



要求最后一项小于 **0.001** 时、或者当 **N=20** 时尚未达到精度要求，则停止计算。

【4.40】已知求正弦 **sin(x)**的近似值的多项式公式为：



编写程序，要求输入 **x** 和 ε，按上述公式计算 **sin(x)**的近似值，要求计算的误差小于给定的 ε。

【4.41】从键盘输入十个整数，用插入法对输入的数据按照从小到大的顺序进行排序，将排序后的结果输出。

【4.42】输入一个正整数，要求以相反的顺序输出该数。例如输入 **12345**，输出位 **54321**。

【4.43】编写程序，读入一个整数N；若N为非负数，则计算N到 **2**×N之间的整数和；若N为一个负数，则求2×N到N之间的整数和。分别利用 **for** 和 **while** 写出两个程序。

【4.44】求解爱因斯坦数学题。有一条长阶梯，若每步跨 **2** 阶，则最后剩余 **1** 阶，若每步跨 **3** 阶，则最后剩 **2** 阶，若每步跨 **5** 阶，则最后剩 **4** 阶，若每步跨 **6** 阶则最后剩 **5** 阶，若每步跨 **7** 阶，最后才正好一阶不剩。请问，这条阶梯共有多少阶 **?**

【4.45】一个自然数被 **8** 除余 **1**，所得的商被 **8** 除也余 **1**，再将第二次的商被 **8** 除后余 **7**，最后得到一个商为a。又知这个自然数被 **17** 除余 **4**，？所得的商被 **17** 除余 **15**，最后得到一个商是a的 **2** 倍。编写程序求这个自然数。

【4.46】编写程序，用二分法求一元二次方程 **2x3-4x2+3x-6=0** 在（**10**，**10**）区间的根。

【4.47】中国古代科学家祖冲之采用正多边形逼近的割圆法求出了 π 的值。请编写一程序，采用割圆法求出 π 的值，要求精确到小数点之后的第十位。

【4.48】A、**B**、**C**、**D**、**E** 五人在某天夜里合伙去捕鱼，到第二天凌晨时都疲惫不堪，于是各自找地方睡觉。日上三竿，**A** 第一个醒来，他将鱼分为五份，把多余的一条鱼扔掉，拿走自己的一份。**B** 第二个醒来，也将鱼分为五份，把多余的一条鱼扔掉，拿走自己的一份。**C**、**D**、**E** 依次醒来，也按同样的方法拿鱼。编写程序求出他们合伙至少捕了多少条鱼。

【4.49】一辆卡车违犯交通规则，撞人逃跑。现场三人目击事件，但都没记住车号，只记下车号的一些特征。甲说：牌照的前两位数字是相同的；乙说：牌照的后两位数字是相同的；丙是位数学家，他说：四位的车号刚好是一个整数的平方。请根据以上线索求出车号。

【4.50】若一个口袋中放有 **12** 个球，其中有 **3** 个红的， **3** 个白的和 **6** 个黑的，每次从中任取 **8** 个球，编写程序求出共有多少种不同的颜色搭配。

【4.51】100 匹马驮 **100** 担货，大马一匹驮 3 担，中马一匹驮 2 担，小马两匹驮 1 担。试编写程序计算大、中、小马的数目。

【4.52】编写程序，输出用一元人民币兑换成 **1** 分、 **2** 分和 **5** 分硬币的不同兑换方法。

【4.53】显示 **200** 以内的完全平方数和它们的个数。 （完全平方数： **A2+B2=C2,** 求 **A**、 **B**、 **C**）

【4.54】设 **N** 是一个四位数，它的 **9** 倍恰好是其反序数（例如： **123** 的反序数是 **321**），求 **N** 的值。

【4.55】将一个数的数码倒过来所得到的新数叫原数的反序数。如果一个数等于它的反序数，则称它为对称数。求不超过 **1993** 的最大的二进制的对称数。

【4.56】编写程序求解下式中各字母所代表的数字。

```
  P E A R
－ A R A
---------
  P E A
```

【4.57】一个自然数的七进制表达式是一个三位数， 而这个自然数的九进制表示也是一个三位数，且这两个三位数的数码顺序正好相反，求这个三位数。

【4.58】请验证 **2000** 以内的哥德巴赫猜想， 对于任何大于 **4** 的偶数均可以分解为两个素数之和。

【4.59】如果一个正整数等于其各个数字的立方和， 则称该数为阿姆斯特朗数 （亦称为自恋性数） 。如 **407=43+03+73** 就是一个阿姆斯特朗数。编写程序求 **1000** 以内的所有阿姆斯特朗数。

【4.60】任意输入一个偶数，请将它分解为两个素数之和。

【4.61】如果整数A的全部因子（包括1，不包括A本身）之和等于B；且整数B的全部因子(包括1，不包括B本身)之和等于A，则将整数A和B称为亲密数。求 3000 以内的全部亲密数。

【4.62】猜数游戏。由计算机 "想"一个数请人猜，如果人猜对了，则结束游戏，否则计算机给出提示，告诉人所猜的数是太大还是太小，直到人猜对为止。计算机记录人猜的次数，以此可以反映出猜数者 "猜"的水平。

【4.63】编写程序求出 1000!后有多少个零。

【4.64】求矩阵 A[2*3] 的转置矩阵 B[3*2]。设矩阵 A 为：

```
┌ 1 2 3 ┐   ┌ 1 4 ┐
A = │     │   B = │ 2 5 │
└ 4 5 6 ┘   └ 3 6 ┘
```

【4.65】十个小孩围成一圈分糖果，老师分给第一个小孩 10 块，第二个小孩 2 块，第三个小孩 8 块，第四个小孩 22 块，第五个小孩 16 块，第六个小孩 4 块，第七个小孩 10 块，第八个小孩 6 块，第九个小孩 14 块，第十个小孩 20 块。然后所有的小孩同时将自己手中的糖分一半给右边的小孩；糖块数为奇数的人可向老师要一块。问经过这样几次调整后大家手中的糖的块数都一样？每人各有多少块糖？

【4.66】输入 5×5 的数组，编写程序实现：

(1)求出对角线上各元素的和；

(2)求出对角线上行、列下标均为偶数的各元素的积；

(3)找出对角线上其值最大的元素和它在数组中的位置。

【4.67】编写程序，以字符形式输入一个十六进制数，将其变换为一个十进制整数后输出。

【4.68】编写程序，输入一个十进制整数，将其变换为二进制后储存在一个字符数组中。

【4.69】编写程序，输出 1000 以内的所有完数及其因子。所谓完数是指一个整数的值等于它的因子之和，例如 6 的因子是 1、2、3，而 6=1+2+3，故 6 是一个完数。

【4.70】对数组 A 中的 N（0＜N＜100＝个整数从小到大进行连续编号， 输出各个元素的编号。要求不能改变数组A中元素的顺序，且相同的整数要具有相同的编号。例如数组是：

A=(5,3,4,7,3,5,6) 则输出为： (3,1,2,5,1,3,4)

【4.71】现将不超过 2000 的所有素数从小到大排成第一行， 第二行上的每个数都等于它 " 右肩 " 上的素数与 " 左肩 " 上的素数之差。 请编程求出： 第二行数中是否存在这样的若干个连续的整数，它们的和恰好是 1898？假如存在的话，又有几种这样的情况？

第一行： 2 3 5 7 11 13 17 ..... 1979 1987 1993 第二行： 1 2 2 4 2 4 ..... 8 6

【4.72】将 1、2、3、4、5、6、7、8、9 九个数字分成三组，每个数字只能用一次，即每组三个数不许有重复数字，也不许同其它组的三个数字重复，要求将每组中的三位数组成一个完全平方数。

【4.73】一个自然数的七进制表达式是一个三位数， 而这个自然数的九进制表示也是一个三位数，且这两个三位数的数码顺序正好相反，求这个三位数。

【4.74】使用数组精确计算 M/N(0<M<N<=100) 的各小数位的值。如果 M/N 是无限循环小数，则计算并输出它的第一循环节，同时要求输出循环节的起止位置（小数的序号）。

为了实现高精度计算结果，可将商M存放在有N（N＞1）个元素的一维数组中，数组的每个元素存放一位十进制数，即商的第一位存放在第一个元素中，商的第二位存放在第二个元素中……，依次类推。这样可使用数组来表示计算的结果。

【4.75】使用数组完成两个超长（长度小于１００）正整数的加法。

为了实现高精度的加法，可将正整数M存放在有N（N＞1）个元素的一维数组中，数组的每个元素存放一位十进制数，即个位存放在第一个元素中，十位存放在第二个元素中……，依次类推。这样通过对数组中每个元素的按位加法就可实现对超长正整数的加法。

【4.76】使用数组完成两个超长（长度小于１００）正整数的加法。

为了实现高精度的加法，可将正整数M存放在有N（N＞1）个元素的一维数组中，数组的每个元素存放一位十进制数，即个位存放在第一个元素中，十位存放在第二个元素中……，依次类推。这样通过对数组中每个元素的按位加法就可实现对超长正整数的加法。

【4.77】使用数组完成两个超长（长度小于１００）正整数的乘法。

【4.78】马步遍历问题：已知国际象棋棋盘有 8*8 共 64 个格子。设计一个程序，使棋子从某位置开始跳马，能够把棋盘上的格子走遍。每个格子只允许走一次。

【4.79】八皇后问题：

在一个 8 × 8 的国际象棋盘，有八个皇后，每个皇后占一格；要求棋盘上放上八个皇后时不会出现相互 "攻击"的现象，即不能有量个皇后在同一行、列或对角线上。问共有多少种不同的方法。

【4.80】编制一个计算函数 y=f(x)的值程序，其中：

-x + 2.5 0<= x <2

y= 2 - 1.5(x-3)*(x-3) 2<= x <4

x/2 - 1.5 4<= x <6

【4.81】编写程序，实现比较两个分数的大小。

【4.82】求这样一个三位数，该三位数等于其每位数字的阶乘之和。

即： **abc = a! + b! + c!**

【4.83】已知两个平方三位数 abc 和 xyz，其中数码 a、b、c、x、y、z 未必是不同的；而 ax、by、cz 是三个平方二位数。编写程序，求三位数 abc 和 xyz。任取两个平方三位数 n 和 n1，将 n 从高向低分解为 a、b、c，将 n1 从高到低分解为 x、y、z。判断 ax、by、cz 是否均为完全平方数。

【4.84】找出一个二维数组中的鞍点，即该位置上的元素是该行上的最大值，是该列上的最小值。二维数组也可能没有鞍点。

【4.85】将数字 1、2、3、4、5、6 填入一个 2 行 3 列的表格中，要使得每一列右边的数字比左边的数字大，每一行下面的数字比上面的数字大。编写程序求出按此要求可有几种填写方法？

【4.86】编写一个函数实现将字符串 str1 和字符串 str2 合并，合并后的字符串按其 ASCII 码值从小到大进行排序，相同的字符在新字符串中只出现一次。

【4.87】已知计算 x 的 n 阶勒让德多项式值的公式如下：

1 (n=0)

Pn(x) = x (n=1)

((2n-1)*x*Pn-1(x)-(n-1)*Pn-2(x))/n (n>1)

请编写递归程序实现。

【4.88】编写函数，采用递归方法实现将输入的字符串按反序输出。

【4.89】编写函数，采用递归方法在屏幕上显示如下杨辉三角形：

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

…… …… …… …… ……

【4.90】编写函数，采用递归方法将任一整数转换为二进制形式。

【4.91】设有字母 a、b、c，请编程用递归的方法产生由这些字母组成的，且长度为 n 的所有可能的字符串。例如，输入 n=2，则输出：

aa ab ac ba bb bc ca cb cc

【4.92】将一个数的数码倒过来所得到的新数，叫作原数的反序数，如果一个数等于它的反序数，则称它为对称数。编写程序，采用递归算法求不超过 1993 的最大的二进制的对称数。

【4.93】从 1 到 n(n<1000)个自然数中选出 r 个数进行组合，并按指定的格式输出组合的结果。

例如： n=5， r=3 时，共有 10 种组合，运行程序，要按下面的格式输出：

1 2 3

4

5

3 4

5

4 5

2 3 4

5

4 5

3 4 5

请用递归算法实现。

【4.94】从键盘输入十个整数，　用合并排序法对输入的数据按照从小到大的顺序进行排序，　　将排序后的结果输出。

【4.95】编写程序，读入一个以符号　**"."** 结束的长度小于　**20** 字节的英文句子，检查其是否为回文（即正读和反读都是一样的，不考虑空格和标点符号）。例如：

读入句子：　**MADAM I'M ADAM.**　　它是回文，所以输出：　**YES**

读入句子：　**ABCDBA).**　它不是回文，所以输出：　**NO**

【4.96】编写程序，其中包括一个函数，此函数的功能是：对一个长度为　　**N** 的字符串从其第 **K** 个字符起，删去　**M** 个字符，组成长度为　**N-M** 的新字符串 **(**其中 **N**、**M<=80,K<=N)** 。例如输入字符串 **"We are poor students."**，利用此函数进行删除　**"poor"** 的处理，输出处理后的字符串是**"We are students."** 。

【4.97】编写函数，通过指针将一个字符串反向。

【4.98】编写一个函数　**insert(s1,s2,ch)** 实现在字符串　**s1** 中的指定字符　**ch** 位置处插入字符串 **s2**。

【4.99】编写程序将输入的两行字符串连接后，将串中全部空格移到串首后输出。

【4.100】编写程序，输入字符串，分别统计字符串中所包含的各个不同的字符及其各自字符的数量。如：输入字符串：　　abcedabcdcd 则输出： a=2 b=2 c=3 d=3 e=1　。

【4.101】利用结构： struct complx

{ int real ；
int im ；
}；
编写求两个复数之积的函数 **cmult**，并利用该函数求下列复数之积：
？ **(3+4i)** × **(5+6i)** ？ **(10+20i)** × **(30+40i**

【4.102】编写成绩排序程序。按学生的序号输入学生的成绩，按照分数由高到低的顺序输出学生的名次、该名次的分数、相同名次的人数和学号；同名次的学号输出在同一行中，一行最多输出 **10** 个学号。

【4.103】编写程序，实现输入的时间屏幕显示一秒后的时间。显示格式为 **HH:MM:SS** 。程序需要处理以下三种特殊情况：
？ 若秒数加 **1** 后为 **60**，则秒数恢复到 **0**，分钟数增加 **1**；
？ 若分钟数加 **1** 后为 **60**，则分钟数恢复到 **0**，小时数增加 **1**；
？ 若小时数加 **1** 后为 **24**，则小时数恢复到 **0**。

【4.104】编写程序，从键盘输入 **3** 个学生的数据，将它们存入文件 **student**；然后再从文件中读出数据，显示在屏幕上。

【4.105】编写程序，从键盘输入一行字符串，将其中的小写字母全部转换成大写字母，然后输出到一个磁盘文件 **"test"** 中保存。

【4.106】编写程序，读入磁盘上Ｃ语言源程序文件 **"test8.c"**，删去程序中的注释后显示。

【编写程序题参考答案】

【4.1】参考答案：

```
#include <stdio.h>
main( )
{ int year ；
float money,rate,total ； /* money: 本金 rate:月利率 total:本利合计 */
printf("Input money and year =?") ；
scanf("%f%d", &money, &year) ； /* 输入本金和存款年限 */
if(year==1) rate=0.00315 ； /* 根据年限确定利率 */
else if(year==2) rate=0.00330 ；
else if(year==3) rate=0.00345 ；
else if(year==5) rate=0.00375 ；
else if(year==8) rate=0.00420 ；
else rate=0.0；
total=money + money * rate * 12 * year ； /* 计算到期的本利合计 */
printf(" Total = %.2f\n", total) ；
}
```

【4.2】参考答案：

```
#include <stdio.h>
main( )
{ int year, month, days；
printf("Enter year and month:") ；
scanf("%d%d", &year, &month) ；
switch (month)
{ case 1: case 3: case 5: case 7: case 8: case 10: case 12:
days=31； break； /* 处理"大"月 */
case 4: case 6: case 9: case 11:
days=30； break； /* 处理"小"月 */
case 2: if(year%4==0&&year%100!=0 || year%400==0)
days=29； /* 处理闰年平月 */
else days=28； /* 处理不是闰年平月 */
break；
default: printf("Input error!\n") ； /* 月份错误 */
days=0；
}
if( days!=0 )
printf("%d, %d is %d days\n", year, month, days) ；
}
```

【4.3】参考答案：

```
#include <stdio.h>
main ( )
{ float data1, data2 ； /* 定义两个操作数变量 */
char op； /* 操作符 */
printf("Enter your expression:") ；
scanf("%f%c%f", &data1, &op, &data2) ； /* 输入表达式 */
```

```
switch(op) /*  根据操作符分别进行处理      */
{ case '+' : /*  处理加法   */
printf("%.2f+%.2f=%.2f\n", data1, data2, data1+data2)       ;   break；
case '-' : /*  处理减法   */
printf("%.2f-%.2f=%.2f\n", data1, data2, data1-data2)       ;   break；
case '*' : /*  处理乘法   */
printf("%.2f*%.2f=%.2f\n", data1, data2, data1*data2)       ;   break；
case '/' : /*  处理除法   */
if( data2==0 ) /*   若除数为  0 */
printf("Division by zero.\n")   ；
else
printf("%.2f/%.2f=%.2f\n", data1, data2, data1/data2)       ；
break；
default: /*   输入了其它运算符     */
printf("Unknown operater.\n")   ；
}
}
```

【4.4】分析：打印此图形用两重循环实现。

图形要重复 n 行，故采用循环结构实现循环 n 次，循环体内部打印一行 '*' 号,把上述思路表示为：

```
for(i=1 ；i<=n ；i++)
```

打印一行 '*' 号；

每行有 n 个'*' 号，再采用循环结构实现 n 次循环，循环内部用格式输出语句打印一个 '*' 号，即：

```
for(j=1 ；j<=n ；j++)
printf("*")   ；
```

按照上述思路，实现打印矩形。

参考答案：

```
main()
{ int i,j,n ；
printf("\nPlease Enter n:")   ；
scanf("%d",&n)   ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n ；j++)
printf("*")   ；
printf("\n")   ；
}
}
```

【4.5】分析：此图形和上题的区别在于在每一行先要打印空格，然后再打印 n 个 '*' 号,在上题第一层循环体内打印 '*' 号的循环前面增加一个循环打印空格。每行空格的个数是逐行减少的，由于第一层循环的控制变量 i 是逐行增 1，所以用一个固定值的数减去 i 就可实现对空格个数的控制，在此题中固定值可使用变量 n。

参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n:")   ；
scanf("%d",&n)   ；
```

```
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n-i ；j++)
printf(" ")  ；
for(j=1 ；j<=n ；j++)
printf("*")  ；
printf("\n")  ；
}
}
```

【4.6】分析：此题和上题的区别在于每行 '*' 的数量逐行减少，可以使用上题控制空格个数的思路来控制 '*' 号的个数，请注意每行 '*' 的个数都是奇数。

参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n:")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n-i ；j++)
printf(" ")  ；
for(j=1 ；j<=2*i-1 ；j++)
printf("*")  ；
printf("\n")  ；
}
```

【4.7】分析：此题图形是第 3 题图形的垂直反转，在编程上我们可以变换一个思路。对于图形中的第 i 行（1? i? n），共需要输出 2n-i 个字符，其中前面的 i-1 个字符为空格，后面的字符为 '*' 号。按照这一思路可以编写出如下程序。

参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n:")  ；
scanf("%d", &n)  ；
for( i=1 ；i<=n ；i++ ) /*  重复输出图形的  n 行  */
{ for( j=1 ；j<=2*n-i ；j++ ) /*  重复输出图形一行中的每个字符  */
if(j<=i-1) printf(" ")  ；  /*  输出前面的空格  */
else printf("*")  ；  /*  输出后面的 *号  */
printf("\n")  ；
}
}
```

【4.8】分析：此题和第 3 题的区别仅是每行的 '*' 个数增加 n-1 个。

参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n:")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++)
```

```
{ for(j=1 ；j<=n-i ；j++)
printf(" ")  ；
for(j=1 ；j<=2*i-1+(n-1)  ；j++)
printf("*")  ；
printf("\n")  ；
}
}
```

【4.9】分析：对于空心图形，我们可以在上题的基础上，对于打印 '*' 号的循环进行修改，仅在循环开始值 (j=1) 和循环结束值 (j=2*(i-1)+n) 时打印 '*' 号，其它位置都打印空格。另一种思路是将每行打印的空格和 '*' 的两个循环合为一体考虑，在判断出需要打印 '*' 的两个位置及第一行和最后一行相应位置外，其余位置都打印空格。

参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n:")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=2*n+i-3  ；j++)
if(j==n-i+1 || j>n-i+1 && (i==1||i==n)) printf("*")  ；
else printf(" ")  ；
printf("*\n")  ；
}
}
```

【4.10】分析：此图形可以理解为两个空心梯形反向连接而成，因此可以利用上题的思路进行输出。

参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n:")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++) /*  输出图形的上半部分  (含中心行 ) */
{ for(j=1 ；j<=2*n-i-1  ；j++)
if(j==i) printf("*")  ；
else printf(" ")  ；
printf("*\n")  ；
}
for(i=1 ；i<n ；i++) /*  输出图形的下半部分（不含中心行）  */
{ for(j=1 ；j<=n+i  ；j++)
if(j==n-i) printf("*")  ；
else printf(" ")  ；
printf("*\n")  ；
}
}
```

【4.11】分析： 此题与上题的区别在于打印 '*' 号的位置不同， 编程时要找出应打印 '*' 号的位置和

两个循环变量　**i、j** 以及行数　**n** 的关系。

参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n:")  ；
scanf ("%d", &n)  ；
for(i=1 ；i<=n ；i++) /*  输出图形的上半部分  (含中心行 ) */
{ for(j=1 ；j<=2*n-i  ；j++)
if(j==n-i+1 || j>n-i+1 && i==1) printf("*")       ；
else printf(" ")  ；
printf("*\n")  ；
}
for(i=1 ；i<n ；i++) /*  输出图形的下半部分（不含中心行）       */
{ for(j=1 ；j<=3*(n-1)-i  ；j++)
if(j==i+1 || j>i+1 && i==n-1) printf("*")       ；
else printf(" ")  ；
printf("*\n")  ；
}
}
```

【4.12】参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n:")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n ；j++)
if(j==n-i+1 || i==1 || i==n) printf("*")     ；
else printf(" ")  ；
printf("\n")  ；
}
}
```

【4.13】参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n: ")  ；
scanf("%d", &n)  ；
for(i=1 ；i<=n ；i++) /*  输出图形的上半部分  (含中心行 ) */
{ for(j=1 ；j<=n-i  ；j++)
if(j==1 || j==n-i+1) printf("* ")     ；
else printf(" ")  ；
printf("\n")  ；
}
for(i=1 ；i<n ；i++) /*  输出图形的下半部分（不含中心行）       */
{ for(j=1 ；j<=i+1  ；j++)
if(j==1 || j==i+1) printf("* ")     ；
```

```
else printf(" ") ；
printf("\n")  ；
}
}
```

【4.14】参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n: ")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n ；j++)
if(j==1 || j==i || j==n) printf("*")    ；
else printf(" ") ；
printf("\n")  ；
}
}
```

【4.15】参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n: ")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n+i-1 ；j++)
if(j>n-i) printf("*")    ；
else printf(" ") ；
printf("\n")  ；
}
for(i=1 ；i<n ；i++)
{ for(j=1 ；j<=2*n-i-1 ；j++)
if(j>i) printf("*")    ；
else printf(" ") ；
printf("\n")  ；
}
}
```

【4.16】参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n: ")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n+i-2 ；j++)
if(j==n-i+1) printf("*")    ；
else printf(" ") ；
printf("*\n")  ；
}
```

}

【4.17】分析：首先对图形进行设计，坐标的 **X** 轴和 **Y** 轴分别对应屏幕的列和行，一个正弦函数的周期为 **0～360** 度，我们把一个步长定义为 **10** 度，打印时每换一行等于函数的自变量增加 **10** 度；屏幕的列宽为 **80**，函数值为 **0** 对应屏幕的第 **40** 列，**sin(x)**的值在 **-1～1**，变换成列数为以 **0** 为中心的 **-30～30**，对应屏幕上第 **10～70** 列。设计程序时，控制换行的自变量 **i** 乘以 **10** 得到正弦函数的 **X** 值，调用库函数 **sin()**求出函数值再乘以 **30** 输出的列宽，因为我们以屏幕的第 **40** 列为 **0** 点，故再加上 **40** 得到应在屏幕上显示的点。

参考答案：

```
#define PAI 3.14159
#include <math.h>
main( )
{ double x；
int y,i,yy；
for(i=1 ；i<80 ；i++) /*  打印图形的第一行  */
if(i==40) printf("*")  ；  /* i 控制打印的列位置  */ else printf("-")  ；
printf("\n")  ；
for(x=10.0 ；x<=360.0 ；x+=10.) /*  从 10 度到 360 度  */
{ y = 40+30*sin(x*PAI/180.0)  ；  /*  计算对应的列  */
yy = 40>y ? 40 : y ；  /*  下一行要打印的字符总数  */
for (i=1 ；i<=yy ；i++) /*  控制输出图形中的一行  */
{ if(i==y) printf("*")  ；  /* i 控制打印的列位置  */
else if(i==40) printf("|")  ；  /*  打印中心的竖线  */
else printf(" ")  ；
}
printf("\n")  ；
}
}
```

【4.18】分析：首先设计屏幕图形，如果预计圆形在屏幕上打印 **20** 行，所以定义圆的直径就是 **20**，半径为 **10**，圆的方程是 **X2** × **Y2=R2,** 因为图形不是从中心开始打印而是从边沿开始，所以 **Y** 从 **10** 变化到 **-10**，根据方程求出 **X,** 对求得的 **X** 值再根据屏幕行宽进行必要的调整得到应打印的屏幕位置。

参考答案：

```
#include <math.h>
main( )
{ double y；
int x,m ；
for(y=10 ； y>=-10 ； y--) /*  圆的半径为  10 */
{ m = 2.5 * sqrt(100-y*y)  ；  /*  计算行 y 对应的列坐标  m */
for(x=1 ；x<30-m ；x++)
printf(" ")  ；  /*  输出圆左侧的空白  */
printf("*")  ；  /*  输出圆的左侧  */
for( ；x<30+m ；x++)
printf(" ")  ；  /*  输出圆的空心部分  */
printf("*\n")  ；  /*  输出圆的右侧  */
}
```

```
}
```

【4.19】参考答案：

```
#include <stdio.h>
#include <math.h>
main( )
{ double y；
int x, m, n, yy ；
for( yy=0 ；yy<=20 ； yy++)
{ y = 0.1*yy ；
m = acos(1-y)*10 ；
n = 45 * (y-1)+31 ；
for( x=0 ； x<=62 ； x++ )
if( x==m && x==n ) printf("+")      ；
else if(x==n) printf("+")    ；
else if(x==m || x==62-m) printf("*")     ；
else printf(" ") ；
printf("\n")   ；
}
}
```

【4.20】分析：编程的关键为两点，　一是使用控制输出的行和列，　这方面的内容在前面已经叙述，另一点是输出的数字和所在行、列关系。此题第一行输出的数字恰好是列数，从第二行起每行的数字均比上一行增　n。

参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n: ")   ；
scanf("%d",&n)   ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n ；j++)
printf("%4d",(i-1)*n+j)      ；
printf("\n")   ；
}
}
```

【4.21】分析：此题的关键是找到输出数字和行、列数的关系。审查图形中每行中数字的关系发现，右边数字和前面数字之差逐次增　1；同列数字依然是这样的关系，编程的关键转换为找到每一行左方的第一个数字，然后利用行和列的循环变量进行运算就可得到每个位置的数字。用　ai,j 此表示第 i 行第 j 列的数字，则 a11=1 ;由第 i 行第一列的数字推出第　i+1 行第一列的数字是　ai+1,1 = ai,1+i ；同样由第 j 列推出第 j+1 列的数字是 ai,j+1 = ai,j+i+j 。另外只有当 j<i 时才输出数字。

参考答案：

```
main( )
{ int i,j,m,n,k=1 ；  /* k 是第一列元素的值    */
printf("Please enter m=" ")   ；
scanf("%d",&m)   ；
```

```
for(i=1 ；i<=m ；i++)
{ n=k ；  /* n 第 i 行中第 1 个元素的值    */
for(j=1 ；j<=m-i+1 ；j++)
{ printf("%3d",n)   ；
n = n+i+j ；  /*  计算同行下一个元素的值      */
}
printf("\n")  ；
k=k+i ；  /*  计算下一行中第    1 个元素   */
}
```

【4.22】参考答案：

```
main( )
{ int i,j,n ；
printf("\nPlease Enter n: ")   ；
scanf("%d",&n)   ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n ；j++)
if(j<=i) printf(" 1")    ；
else printf("%3d",j-i+1)    ；
printf("\n")   ；
}
}
```

【4.23】分析：可用不同的方案解决此问题，为了开阔读者的思路，这里给出了两个参考答案，其中第二个答案是使用了递归方法。

方案一：

首先寻找数字输出数字和行列的关系。

每圈有四个边， 把每边的最后一个数字算为下边的开始， 最外圈每边数字个数是 **n-1** 个，以后每边比外边一边少两个数字。

因为数字是一行一行输出的，再分析每行数字的规律。实际没有的数字有三种规律：位于对角线之间的数字是上半图增一，下半图减一。对角线左侧的各列，右侧比左侧增加了一圈数字，例如数字 **39** 和它左侧的 **22** 比较，数字 **39** 所在的圈每边 **4** 个数字，左侧 **22** 加上一圈 **16** 个数字在加 **1** 就是 **39**。同理，对角线右侧的各列，则减少一圈的数字个数。

根据以上分析，用两个对角线将图形分为四个区域，如下图所示，图中黑斜体字为对角线上的数字。

1 2 3 4 5 6 7

24 25 26 27 28 29 8

23 40 41 42 43 30 9

22 39 48 49 44 31 10

21 38 47 46 45 32 11

20 37 36 35 34 33 12

19 18 17 16 15 14 13

为叙述方便我们称四个区域为上、下、左、右区。设 **i**、**j** 为行列号， **n** 为图形的总行数，则满足各区的范围是， 上区：**j>=i** 且 **j<=n-i+1** ；下区：**j<=i** 且 **j>=n-i+1** ；左区：**j<i** 且 **j<n-i+1** ；右区：**j>i** 且 **j>n-i+1** 。

现在问题是，如果知道一行在不同区域开始第一个位置的数字，然后该区后续的数字就可利用

前面分析的规律得到。

对于右区开始各行第一个数字最易求出，为 **4*(n-1)-i+1** 。后续一个和同行前一个数字之差是 **4*[n-1-(j-1)*2]+1,** 其中方括号内是每边的数字个数。

对角线上的数字是分区点，对角线上相临数字仍然相差一圈数字个数，读者自行分析得到计算公式。

右区开始的第一个数字可以从上区结束时的数字按规律求出。

下述程序用变量 **s**保存分区对角线上的数字。

参考答案一：

```
main()
{ int i,j,k,n,s,m,t ；
printf("Please enter n:") ；
scanf("%d",&n) ；
for(i=1 ；i<=n ；i++)
{ s=(i<=(n+1)/2)? 1:3*(n-(n-i)*2-1)+1 ；
m=(i<=(n+1)/2)? i:n-i+1 ； /* m-1 是外层圈数 */
for(k=1 ；k<m ；k++) s+=4*(n-2*k+1) ；
for(j=1 ；j<=n ；j++)
{ if(j>=n-i+1 && j<=i) /* 下区 */
t=s-(j-(n-i))+1 ；
if(j>=i && j<=n-i+1) /* 上区 */
t=s+j-i ；
if(j>i && j>n-i+1) /* 右区 */
t-=4*(n-2*(n-j+1))+1 ；
if(j<i && j<n-i+1) /* 左区 */
{ if(j==1) t=4*(n-1)-i+2 ；
else t+=4*(n-2*j+1)+1 ；
}
printf("%4d",t) ；
}
printf("\n") ；
}
}
```

方案二：

根据本题图形的特点，我们可以构造一个递归算法。我们可以将边长为 **N** 的图形分为两部分：第一部分最外层的框架，第二部分为中间的边长为 **N-2** 的图形。

对于边长为 **N** 的正方型，若其中每个元素的行号为 **i**（**1**？ **i**？ **N**），列号为 **j**（**1**？ **j**？ **N**），第 **1** 行第 **1** 列元素表示为 **a1,1**（**a11=1**），则有：

对于最外层的框架可以用以下数学模型描述：

上边： **a1,j=a1,1+j-1 (j ≠ 1)**

右边： **ai,N=a1,1+N+i-2 (i ≠ 1)**

下边： **ai,1=a1,1+4N-i-3 (i ≠ 1)**

左边： **aN,j=a1,1+3N-2-j (j ≠ 1)**

对于内层的边长为 **N-2** 的图形可以用以下数学模型描述：

左上角元素： **ai,i=ai-1,i-1+4(N-2i-1) (i ＞1)**

若令： **ai,j=fun(ai-1,i-1+4(N-2i-1)** ，当： **i＜(N+1)/2** 且 **j＜(N+1)/2** 时， **min=MIN(i,j)** ，则有：

a2,2 = fun(a1,1, min, min, n)

ai,j=fun(a2,2, i-min+1, j-min+1, n-2*(min-1))

我们可以根据上述原理，分别推导出 **i** 和 **j** 为其它取值范围时的 **min** 取值。根据上述递归公式，可以得到以下参考程序。

参考答案二 **：**

```
#include <stdio.h>
#define MIN(x,y) (x>y) ? (y) : (x)
fun ( int a11, int i, int j, int n)
{ int min, a22;
if( i==j && i<=1 ) return(a11);
else if( i==j && i<=(n+1)/2) return( fun(a11,i-1,i-1,n)+4*(n-2*i+3));
else if( i==1 && j!=1) return( a11+j-1 );
else if( i!=1 && j==n) return( a11+n+i-2 );
else if( i!=1 && j==1 ) return ( a11+4*n-3-i );
else if( i==n && j!=1 ) return ( a11+3*n-2-j );
else
{ if(i>=(n+1)/2 && j>=(n+1)/2) min = MIN(n-i+1,n-j+1);
else if(i<(n+1)/2 && j>=(n+1)/2) min = MIN(i,n-j+1);
else if(i>=(n+1)/2 && j<(n+1)/2) min = MIN(n-i+1,j);
else min = MIN(i,j);
a22 = fun(a11,min,min,n);
return(fun(a22, i-min+1, j-min+1, n-2*(min-1)));
}
}
main()
{ int a11=1, i, j, n;
printf("Enter n=");
scanf("%d", &n);
for(i=1; i<=n; i++)
{ for(j=1; j<=n; j++)
printf("%4d", fun(a11,i,j,n) );
printf("\n");
}
}
```

【4.24】分析：此题的关键还是要找到输出数字 **aij** 和行列数 **i、j** 的关系。为此将图形分为四个区域如下图：

3 3 3 3 3

3 2 2 2 3

3 2 1 2 3

3 2 2 2 3

3 3 3 3 3 (此图 **n** 为 **5)**

在左上区域，即 **i<=(n+1)/2** 、**j<=(n+1)/2** 时，输出数字为 **(n+1)/2-i+1** 和 **(n+1)/2-j+1** 中的大者，记为 **max{(n+1)/2-i+1,(n+1)/2-j+1}** ；在右上区，即 **i<=(n+1)/2** 、**j>(n+1)/2** 时，输出数字为 **max{(n+1)/2-i+1,j-n/2}** ；在左下区，即 **i>(n+1)/2** 、**j<=(n+1)/2** 时，输出数字为

max{i-n/2,(n+1)/2-j+1} ；在右下区，即 **i>(n+1)/2** 、**j>(n+1)/2** 时，输出数字为 **max{i-n/2,j-n/2}** 。

参考答案：

```
#define max(x,y) ((x)>(y)?(x):(y))
main( )
{ int i,j,n ；
printf("\nPlease Enter n:")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n ；j++)
if(i<=(n+1)/2)
if(j<=(n+1)/2)
printf("%4d",max((n+1)/2-i+1,(n+1)/2-j+1))      ；
else
printf("%4d",max((n+1)/2-i+1,j-n/2))      ；
else if(j<=(n+1)/2)
printf("%4d",max(i-n/2,(n+1)/2-j+1))      ；
else
printf("%4d",max(i-n/2,j-n/2))      ；
printf("\n")  ；
}
}
```

【**4.25**】分析：前面我们已经见到过上下对称的图形，这是一个左右对称的图形，垂直中心线上的数字恰好是行号，在每行位于图形垂直中心线左方的数字是逐渐增加的，而右方是逐渐减小的。 **j==i** 是分区的标志，左方输出数字就是列数 **j**，而右方的数字从 **i** 开始逐步减小 **1**。

参考答案：

```
main()
{ int i,j ；
for(i=1 ；i<=9 ；i++)
{ for(j=1 ；j<=9-i ；j++) printf(" ")  ；
for(j=1 ；j<=i ；j++) printf("%2d",j)    ；
for(j=i-1 ；j>=1 ；j--) printf("%2d",j)    ；
printf("\n")  ；
}
}
```

【**4.26**】分析：这类输出字符的图形和输出数字的图形考虑是近似的，因为字符的 **ASCII** 码就是一个整数。在字符码值的变化过程中，应该注意应该判断码值是否超出字符的范围，进行必要的处理，为了保持程序的简洁，本题没有考虑这个问题，在下题里对这个问题进行了处理。

参考答案：

```
main( )
{ char c='Z' ；
int i,j,n ；
printf("\nPlease Enter n:")  ；
scanf("%d",&n)  ；
for(i=1 ；i<=n ；i++)
{ for(j=1 ；j<=n+i-2 ；j++)
```

```
if(j==n-i+1) printf("%c",c--)      ;
else printf(" ")  ;
printf("%c\n",c--)    ;
}
for(i=1  ;  i<n  ;  i++)
{ for(j=1  ;  j<=2*(n-1)-i   ;  j++)
if(j==i+1) printf("%c",c--)      ;
else printf(" ")  ;
printf("%c\n",c--)    ;
}
}
```

【4.27】分析：此题与上题相近，区别在于输出时字符的 ASCII 码值的变化在图形的中间一行为最大，同时一行的两个字符是相同的。程序考虑在输入字符时设计了一个循环，保证输入的是英文字母。字符变化后进行了处理，程序中使用条件运算。在字符码值增加的过程中，首先判断是大写还是小写字符，然后判断字符码值是否超出英文字母 z(或 Z)，如果超出则重新赋为 a(或 A)；在输出图象下半部分时，ASCII 码值减少用同样的思路进行判断。在判断字符大小写（条件语句的第一个判断）时，用的是两个不同的值，请读者自行思考为什么，用同一个值是否可以？

参考答案：

```
main( )
{ char c；
int i,j,n ；
do
{ printf("\nPlease Enter n,char:")    ;
scanf("%d,%c",&n,&c)     ;
}while(c<'A'||c>'Z'&&c<'a'||c>'z')      ;
for(i=1  ;  i<=n  ;  i++)
{ for(j=1  ;  j<=n+i-2  ;  j++)
if(j==n-i+1) printf("%c",c)      ;
else printf(" ")  ;
printf("%c\n",c++)    ;
c=c<'a'?(c>'Z'?'A':c):(c>'z'?'a':c)    ;
}
c-=2 ;
c=c<'Z'?(c<'A'?'Z':c):(c<'a'? 'z':c)     ;
for(i=1  ;  i<n  ;  i++)
{ for(j=1  ；j<=2*(n-1)-i   ；j++)
if(j==i+1) printf("%c",c)     ;
else printf(" ")  ;
printf("%c\n",c--)    ;
c=c<'Z'?(c<'A'?'Z':c):(c<'a'? 'z':c)     ;
}
}
```

【4.28】参考答案：

```
#define max(x,y) ((x)>(y)?(x):(y))
```

```
main( )
{ char c ;
int i,j,n  ;
do
{ printf("\nPlease Enter n  ,  char:")  ;
scanf("%d,%c",&n,&c)     ;
}while(c<'A'||c>'Z'&&c<'a'||c>'z')      ;
for(i=1  ;  i<=n  ;  i++)
{ for(j=1  ;  j<=n  ;  j++)
if(i<=(n+1)/2)
if(j<=(n+1)/2)
printf(" %c",c-max((n+1)/2-i+1,(n+1)/2-j+1)+(n+1)/2)        ;
else
printf(" %c",c-max((n+1)/2-i+1,j-n/2)+(n+1)/2)       ;
else
if(j<=(n+1)/2)
printf(" %c",c-max(i-n/2,(n+1)/2-j+1)+(n+1)/2)       ;
else
printf(" %c",c-max(i-n/2,j-n/2)+(n+1)/2)       ;
printf("\n")    ;
}
}
```

【4.29】参考答案：

```
#define max(x,y) ((x)>(y)?(x):(y))
main( )
{ char c ;
int i,j,n  ;
do
{ printf("\nPlease Enter n  ,  char:")  ;
scanf("%d,%c",&n,&c)     ;
}while(c<'A'||c>'Z'&&c<'a'||c>'z')      ;
for(i=1  ;  i<=n  ;  i++)
{ for(j=1  ;  j<=n  ;  j++)
if(i<=(n+1)/2)
if(j<=(n+1)/2)
printf(" %c",c-max((n+1)/2-i+1,(n+1)/2-j+1)+1)        ;
else
printf(" %c",c-max((n+1)/2-i+1,j-n/2)+1)       ;
else
if(j<=(n+1)/2)
printf(" %c",c-max(i-n/2,(n+1)/2-j+1)+1)       ;
else
printf(" %c",c-max(i-n/2,j-n/2)+1)      ;
printf("\n")    ;
}
```

```
}
```

【4.30】参考答案：

```
#include <stdio.h>
main()
{ int i, j ；
for(i=1 ；i<10 ；i++)
printf("%4d",i)    ；
printf("\n----------------------------------\n")        ；
for(i=1 ；i<10 ；i++)
{ for(j=1 ；j<10 ；j++)
if(j<i) printf(" ")    ；
else printf( "%4d" , i*j)    ；
printf("\n")   ；
}
}
```

【4.31】参考答案：

```
#include <stdio.h>
main( )
{ int i,j ；
for(i=1 ；i<10 ；i++)
printf("%4d",i)    ；
printf("\n----------------------------------\n")        ；
for(i=1 ；i<10 ；i++)
{ for(j=1 ；j<10 ；j++)
if(j<10-i) printf(" ")    ；
else printf( "%4d" , (10-i)*j)    ；
printf("\n")   ；
}
}
```

【4.32】参考答案：

```
#include "math.h"
main()
{ int flag=0 ；
float a,b,c,s；
do
{ printf("Please enter a b c:")   ；
scanf("%f%f%f",&a,&b,&c)       ；
if(a>b+c || b>a+c || c>a+b)
flag=1；
}while(flag) ；
s=(a+b+c)/2 ；
printf("S=%f",s=sqrt((s-a)*(s-b)*(s-c)))    ；
}
```

【4.33】参考答案：

```
#include <stdio.h>
main( )
{ int j ；
long n；  /*  使用长整型变量，以免超出整数的表示范围          */
printf("Please input number:")  ；
scanf("%ld", &n)  ；
for(j=999 ；j>=100 ；j--)/*  可能取值范围在  999 到 100 之间，  j 从大到小  */
if(n%j == 0 ) /*  若能够整除  j，则 j 是约数，输出结果  */
{ printf("The max factor with 3 digits in %ld is: %d.\n",n,j)  ；
break；  /*  控制退出循环  */
}
}
```

【4.34】参考答案：

```
#define E 0.000001
main()
{ float x,y=1,s=0 ；
printf("Please enter x=")  ；
scanf("%f",&x)  ；
while(1/y>E)
{ s=s+1/y；
y=y*x ；
}
printf("S=%f\n",s)  ；
}
```

【4.35】参考答案：

```
#include <stdio.h>
main( )
{ int class1, class2, class3;
char ch；
class1=class2=class3=0；  /*  初始化分类计数器  */
do
{ ch=getch( )；
switch(ch)
{ case '0': case '1': case '2': case '3': case '4':
case '5': case '6': case '7': case '8': case '9':
class1++；  break ；  /*  对分类 1 计数  */
case '+': case '-': case '*': case '/': case '%': case '=':
class2++；  break ；  /*  对分类 2 计数  */
default: class3++；  break ；  /*  对分类 3 计数  */
}
}while (ch!= '\\')  ；  /*  字符 '\' 在 C 程序中要使用转义符  '\\' */
printf("class1=%d, class2=%d, class3=%d\n", class1, class2, class3) ；
}
```

【4.36】分析：程序的关键是怎样判断一个单词。由单词的定义已知它是用空格、制表符或换行符分隔开的，两个字符之间没有空格、制表符或换行符，则认为是一个单词中的两个字符。

参考答案：

```
#define EOF -1
#define YES 1
#define NO 0
#include <stdio.h>
main( ) /*  对输入的行、字符和单词进行计数统计          */
{ int c, nl, nc, nw, inword ；
inword=NO ；  /* inword=NO   已处理的最后一个字符是空格、      \t 或 \n */
/* inword=YES   已处理的最后一个字符不是空格、      \t 或 \n */
nl=nc=nw=0 ；  /*  行、字符、字计数器置     0 */
while((c=getchar())!= EOF)
{ ++nc ；  /*  进行字符计数    */
if(c=='\n' )
++nl ；  /*  进行行计数    */
if(c=='\t'||c=='\n'||c==' ')
inword=NO ；/*  如果读入的字符是空格、     \t 或\n，则置  inword 为 NO */
else /*  读入的字符不是空格、     \t 或\n */
if(inword==NO) /*    如果前一个字符是空格、     \t 或\n */
{ inword=YES  ；  /*  则读入的字符为一个单词的第一个字符      */
++nw ；  /* 置 inword 为 YES，进行单词计数    */
}
}
printf("Lines=%d\nWords=%d\nChars=%d\n",nl,nw,nc)      ；  /*  输出结果    */
}
```

【4.37】参考答案：

```
#define E 0.000001
#include "math.h"
main()
{ int i,k=1 ；
float x,y,t=1,s,r=1 ；
printf("Please enter x=")  ；
scanf("%f",&x)   ；
for(s=x,y=x,i=2 ；fabs(r)>E ；i++)
{ t=t*(i-1)  ；
y=y*x*x  ；
k=k*(-1)  ；
r=k*y/t/(2*i-1)   ；
s=s+r；
}
printf("S=%f\n",s)   ；
}
```

【4.38】参考答案：

```
main()
{ int i ；
float s=0；
```

```
for(i=1 ；i<=100 ；i++)
s=s+i；
for(i=1 ；i<=50 ；i++)
s=s+i*i ；
for(i=1 ；i<=10 ；i++)
s=s+1.0/i；
printf("Result=%f\n",s)    ；
}
```

【4.39】参考答案：

```
main()
{ int i ；
float s=1；
for(i=1 ；i<=20 && 1.0/i/(i+1)>0.001    ；i++)
s=s+1.0/i/(i+1) ；
printf("Result=%f i=%d\n",s,i)     ；
}
```

【4.40】参考答案：

```
#include <math.h>
main()
{ float x,eps,s,y=0,y0,t；
int n,j ；
printf("Enter x & eps:")    ；
scanf("%f%f", &x, &eps)    ；
n=t=j=1 ；
s=x；
do
{ y0=y ；
if(n%2==0) y=y-s/t   ；
else y=y+s/t；
s *= x*x ；  /*  求 x 的乘方   */
t *= (j+1)*(j+2)    ；  /*  求 n! */
j += 2 ；
n++ ；
}while( fabs(y0-y) > eps )；  /*  控制误差    */
printf("sin(%f)=%f\n",x,sin(x))    ；  /*  输出标准  sin(x)的值  */
printf("%d,sin(%f)=%f\n",n,x,y)      ；  /*  输出计算的近似值     */
}
```

【4.41】参考答案：

```
main( )
{ int i,j,num,a[10]  ；
for(i=0 ；i<10 ；i++)
{ printf("Enter No. %d:", i+1)     ；
scanf("%d",&num)   ；
for(j=i-1 ；j>=0&&a[j]>num    ；j--)
```

```
a[j+1]=a[j] ；
a[j+1]=num ；
}
for(i=0 ；i<10；i++)
printf ("No.%d=%d\n", i+1, a[i])      ；
}
```

【4.42】参考答案：

```
main()
{ int n ；
printf("Please enter n:")  ；
scanf("%d",&n)   ；
while(n>0)
{ printf("%d",n%10)     ；
n=n/10 ；
}
}
```

【4.43】参考答案：

```
main()
{ int i,n ；
long s1=0,s2=0；
printf("Please enter N:")  ；
scanf("%d",&n)   ；
if(n>=0)
for(i=n ；i<=2*n ；i++)
s1=s1+i；
else
for(i=n ；i>=2*n ；i--)
s1=s1+i；
i=n ；
if(i>=0)
while(i<=2*n)
s2=s2+i++ ；
else
while(i>=2*n)
s2=s2+i-- ；
printf("Result1=%ld result2=%ld\n",s1,s2)     ；
}
```

【4.44】分析：据题意，阶梯数满足下面一组同余式：

$x \equiv 1 \pmod 2$

$x \equiv 2 \pmod 3$

$x \equiv 4 \pmod 5$

$x \equiv 5 \pmod 6$

$x \equiv 0 \pmod 7$

参考答案：

```
#include <stdio.h>
```

```
main()
{ int i=1 ；  /* i 为所设的阶梯数    */
while( !((i%2==1)&&(i%3==2)&&(i%5==4)&&(i%6==5)&&(i%7==0)) )
++i ；  /*  满足一组同余式的判别      */
printf("Staris_number=%d\n", i )    ；
}
```

【4.45】参考答案：

```
main( )
{ int i,n,a ；
for(i=0  ；   ；i++)
{ if(i%8==1)
{ n=i/8 ；
if(n%8==1)
{ n=n/8 ；
if(n%8==7) a=n/8   ；
}
}
if(i%17==4)
{ n=i/17 ；
if(n%17==15) n=n/17   ；
}
if(2*a==n)
{ printf("result=%d\n",i)    ；
break；
}
}
}
```

【4.46】分析：二分法的基本原理是，若函数有实根，则函数的曲线应当在根这一点上与ｘ轴有一个交点，在根附近的左右区间内，函数值的符号应当相反。利用这一原理，逐步缩小区间的范围，保持在区间的两个端点处的函数值符号相反，就可以逐步逼近函数的根。

参考答案：

```
#include "stdio.h"
#include "math.h"
main()
{ float x0, x1, x2, fx0, fx1, fx2 ；
do
{ printf("Enter x1,x2:")   ；
scanf("%f,%f", &x1, &x2)    ；
fx1=2*x1*x1*x1-4*x1*x1+3*x1-6    ；  /*  求出 x1 点的函数值   fx1 */
fx2=2*x2*x2*x2-4*x2*x2+3*x2-6    ；  /*  求出 x2 点的函数值   fx2 */
}while (fx1*fx2>0)  ；  /*  保证在指定的范围内有根，即      fx 的符号相反    */
do
{ x0=(x1+x2)/2 ；  /*  取 x1 和 x2 的中点   */
fx0=2*x0*x0*x0-4*x0*x0+3*x0-6    ；  /*  求出中点的函数值    fx0 */
if((fx0*fx1)<0) /*    若 fx0 和 fx1 符号相反   */
```

```
{ x2=x0 ；  /*  则用  x0 点替代  x2 点  */
fx2=fx0 ；
}
else
{ x1=x0 ；  /*  否则用  x0 点替代  x1 点  */
fx1=fx0 ；
}
}while(fabs((double)fx0)>=1e-5)  ；  /*  判断  x0 点的函数与  x 轴的  距离  */
printf("x=%6.2f\n", x0)    ；
}
```

【4.47】分析：做圆的内接 4 边形，从圆心和 4 边形顶点连接形成 4 个三角形，可以求出每个三角形的面积 （r2/2）现在我们知道三角形的面积和两个边长 （均为半径 a=r、b=r），可以用公式：S=s(s-a)(s-b)(s-c)求出第三边 c。我们将内接 4 边形换为内接 8 边形，原来的三角形被一分为二，故 c/2 就是每个三角形的高，面积又是可以求出的。再将三角形一分为二，……。当三角形的面积求出时，内接多边形的面积就可求出。

参考答案：

```
main()
{ int n=4 ；
double r=10,s,cr,c,p；
s=r*r/2  ；
do
{ cr=n*s ；
p=16*r*r*r*r-64*s*s    ；
c=(4*r*r-sqrt(p))/2   ；
c=sqrt(c)；
s=c*r/4 ；
n=2*n ；
}while(n*s-cr>1.0e-10)  ；
printf("PAI=%lf\n",cr/r/r)     ；
}
```

【4.48】分析：根据题意，总计将所有的鱼进行了五次平均分配，每次分配时的策略是相同的，即扔掉一条后剩下的鱼正好分为五份， 然后拿走自己的一份， 余下其它四份。 假定鱼的总数为 x，则 x 可以按照题目的要求进行五次分配： x-1 后可被 5 整除，余下的鱼为 4×（x-1）÷ 5。若 x 满足上述要求，则 x 就是题目的解。

参考答案：

```
main( )
{ int n,i,x,flag=1 ；  /* flag: 控制标记  */
for(n=6 ；flag ；n++) /*  采用试探的方法，令试探值  n 逐步加大  */
{ for(x=n,i=1 ；flag && i<=5  ；i++) /*  判断是否可按要  */
if((x-1)%5 == 0) x=4*(x-1)/5  ；  /*  求进行  5 次分配  */
else flag=0；  /*  若不能分配则置标记  flag=0 退出分配过程  */
if(flag) break ；  /*  若分配过程正常 ,找到结果 ,退出试探的过程  */
else flag=1；  /*  否则继续试探下一个数  */
}
printf("Total number of fish catched = %d\n", n)  ；  /*  输出结果  */
```

```
}
```

【4.49】分析：按照题目的要求造出一个前两位数相同、后两位数相同且相互间又不同的整数，然后判断该整数是否是另一个整数的平方。

参考答案：

```
#include "math.h"
main()
{ int i,j,k,c ；
for(i=1 ； i<=9 ； i++) /* i: 车号前二位的取值     */
for(j=0 ； j<=9 ； j++) /* j: 车号后二位的取值     */
if( i!=j ) /*   判断两位数字是否相异      */
{ k=i*1000+i*100+j*10+j    ；  /*  计算出可能的整数     */
for( c=31 ； c*c<k ； c++)；  /*  判断该数是否为另一整数的平方       */
if(c*c==k)
printf("Lorry_No. is %d .\n", k)    ；  /*  若是，打印结果     */
}
}
```

【4.50】分析：用穷举法解决此类问题。设任取红球的个数为 **i**，白球的个数为 **j**，则取黑球的个数为 **8-i-j**，据题意红球和白球个数的取值范围是 **0～3**，在红球和白球个数确定的条件下，黑球的个数取值应为 **8-i-j<=6**。

参考答案：

```
main( )
{ int i,j,count=0 ；
printf(" RED BALL WHITE BALL BLACK BALL\n")          ;
printf("---------------------------------------\n")        ;
for(i=0 ； i<=3 ； i++) /*   循环控制变量   i 控制任取红球个数    0～3 */
for(j=0 ； j<=3 ； j++) /*   循环控制变量   j 控制任取白球个数    0～3 */
if((8-i-j)<=6)
printf("%2d: %d %d %d\n",++count, i,j,8-i-j)        ;
}
```

【4.51】分析：此题采用穷举法。

参考答案：

```
main()
{ int x,y,z,j=0 ；
for(x=0 ；  x<=33 ；  x++)
for(y=0 ；  y<=(100-3*x)/2  ；  y++)
{ z=100-x-y ；
if( z%2==0 && 3*x+2*y+z/2==100)
printf("%2d:l=%2d m=%2d s=%2d\n",++j,x,y,z)       ;
}
}
```

【4.52】分析：此题采用穷举法。

参考答案：

```
main( )
{ int f1,f2,f5,count=0 ；
```

```
for(f5=0 ；f5<=20 ；f5++)
for(f2=0 ；f2<=(100-f5*5)/2  ； f2++)
{ f1=100-f5*5-f2*2   ；
if(f5*5+f2*2+f1==100)
printf("No.%2d >> 5: %4d 2: %2d 1: %2d\n",++count,f5,f2,f1)        ；
}
}
```

【4.53】分析：此题采用穷举法。

参考答案：

```
main( )
{ long int i,j,k,count=0 ；
for(i=1 ；i*i<=200 ；i++)
for(j=1 ；j*j<=200 ；j++)
for(k=1 ；k*k<=200 ；k++)
if(i*i==(j*j+k*k))
{ printf("\nA^2==B^2+C^2: %4ld%4ld%4ld",i,j,k)        ；
count++；
}
printf("\ncount=%ld",count)    ；
}
```

【4.54】分析：此题采用穷举法。可设整数 N 的千、百、十、个位为 i、 j、 k、 m，其取值均为 0～9，则满足关系式： (i*103+j*102+10k+m)*9=(m*103+k*102+10j+i) 的 i、j、k、m 即构成 N。

参考答案：

```
#include <stdio.h>
main( )
{ int i ；
for(i=1002 ； i<1111； i++) /*   穷举四位数可能的值      */
if(i%10*1000+i/10%10*100+i/100%10*10+i/1000==i*9 )
printf("The number satisfied states condition is: %d\n", i)    ；
/*  判断反序数是否是原整数的      9 倍若是则输出     */
}
```

【4.55】分析：此题采用穷举法。

参考答案：

```
main()
{ int i,j,n,k,a[16]={0}   ；
for(i=1 ；i<=1993 ；i++)
{ n=i ； k=0；
while(n>0) /*   将十进制数转变为二进制数       */
{ a[k++]=n%2   ；
n=n/2 ；
}
for(j=0 ；j<k ；j++)
if(a[j]!=a[k-j-1]) break    ；
if(j>=k)
{ printf(" %d: ",i)    ；
```

```
for(j=0 ；j<k ；j++)
printf("%2d",a[j])    ；
printf("\n")   ；
}
}
}
```

【4.56】分析：类似的问题从计算机算法的角度来说是比较简单的，可以采用最常见的穷举法解决。程序中采用循环穷举每个字母所可能代表的数字，然后将字母代表的数字转换为相应的整数，代入算式后验证算式是否成立即可解决问题。

参考答案：

```
#include <stdio.h>
main( )
{ int p,e,a,r ；
for(p=1 ；p<=9 ；p++) /*   从 1 到 9 穷举字母   p 的全部可能取值     */
for(e=0 ；e<=9；e++) /*   从 0 到 9 穷举字母   e 的全部可能取值     */
if(p!=e)
for(a=1 ；a<=9； a++) /*   从 0 到 9 穷举字母   a 的全部可能取值     */
if(a!=p && a!=e)
for(r=0 ；r<=9 ；r++) /*   从 0 到 9 穷举字母   r */
if(r!=p && r!=e && r!=a /*        四个字母互不相同      */
&& p*1000+e*100+a*10+r-(a*100+r*10+a)
== p*100+e*10+a )
{ printf(" PEAR %d%d%d%d\n", p, e, a, r)        ;
printf(" - ARA - %d%d%d\n", a, r, a)       ;
printf("-------- ---------\n")      ;
printf(" PEA %d%d%d\n", p, e, a)       ;
}
}
```

【4.57】参考答案：

```
main()
{ int i,n,k,a[3],b[3]  ；
for(i=248 ；i<=343 ；i++)
{ for(n=i,k=0  ； n>0； n/=7)
a[k++]=n%7   ；
for(n=i,k=0  ； n>0； n/=9)
b[k++]=n%9   ；
if(k==3)
for(n=0 ；n<k ；n++)
if(a[n]!=b[k-n-1])
break ；
if(n==k)
printf("%d\n",i)    ；
}
}
```

【4.58】参考答案：

```
main()
{ int i,j,k,m,error  ；
for(i=6 ；i<=2000 ；i+=2)
{ error=1 ；
for(j=2 ；j<i ；j++) /*  穷举法分解  i 为两个素数  j 和 m 之和  */
{ for(k=2 ；k<j ； k++) /*  检验 j 是否素数  */
if(j%k==0) /* j  能够被小于它的一个数整除就不是素数  */
break ；
if(k>=j) /* j  是素数  */
{ m=i-j ；
for(k=2 ；k<m ；k++) /*  检验 m 是否素数  */
if(m%k==0)
break ；
if(k>=m) /* m  也是素数，输出结果  */
{ printf("%4d = %4d + %4d\n",i,j,m)  ；
error=0 ；
break ；  }
}
}
if(error)
printf("%4d error!")  ；
}
```

【4.59】分析：可采用穷举法，依次取 1000 以内的各数（设为 i），将 i 的各位数字分解后，据阿姆斯特朗数的性质进行计算和判断。

参考答案：

```
#include <stdio.h>
main()
{ int i,t,k,a[4]={0}  ；
printf ("There are following Armstrong number smaller than 1000:\n")  ；
for(i=2 ；i<1000 ；i++) /*  穷举要判定的数  i 的取值范围  1～1000 */
{ for(t=0,k=1000  ；k>=10 ；t++) /*  截取整数  i 的各位 (从高位向低位  ) */
{ a[t]=(i%k)/(k/10)  ；  /*  分别赋给  a[0]～a[3] */
k /= 10 ；
}
if(a[0]*a[0]*a[0]+a[1]*a[1]*a[1]+a[2]*a[2]*a[2]+a[3]*a[3]*a[3]==i)
printf(" %d ",i)  ；  /*  判断 i 是否为阿姆斯特朗数,  */
/*  若满足条件，则输出  */
}
}
```

【4.60】参考答案：

```
main( )
{ int j,k,n,m  ；
printf("Please enter n:")  ；
scanf("%d",&n)  ；
```

```
for(j=2 ；j<n ；j++) /*  穷举法分解  n 为两个素数  j 和 m 之和  */
{ for(k=2 ；k<j ； k++) /*  检验 j 是否素数  */
if(j%k==0) break  ；  /* j 能够被小于它的一个数整除就不是素数  */
if(k>=j) /* j  是素数  */
{ m=n-j ；
for(k=2 ；k<m ；k++) /*  检验 m 是否素数  */
if(m%k==0) break  ；
if(k>=m) /* m  也是素数，输出结果  */
{ printf("%4d = %4d + %4d\n",n,j,m)  ；
break ；
}
}
}
}
```

【4.61】分析：按照亲密数定义，要判断数 a 是否有亲密数，只要计算出 a 的全部因子的累加和为 b，再计算 b 的全部因子的累加和为 n，若 n 等于 a 则可判定 a 和 b 是亲密数。计算数 a 的各因子的算法：用 a 依次对 i(i=1～a/2) 进行模运算，若模运算结果等于 0，则 i 为 a 的一个因子；否则结束对 a 的因子的计算。

参考答案：

```
#include <stdio.h>
#include <stdio.h>
main( )
{ int a, i, m, n ；
printf("Friendly-numbers pair samller than 3000:\n")  ；
for(a=1 ；a<3000 ； a++) /*  穷举 3000 以内的全部整数  */
{ for(m=0,i=1 ；i<=a/2 ；i++ ) /*  计算数 a 的各因子 ,各因子之和存于 m */
if(!(a%i))
m+=i ；  /*  计算 m 的各因子，各因子之和存于 n */
for(n=0,i=1 ；i<=m/2 ；i++)
if(!(m%i))
n+=i ；
if(n==a && a<m) /*  若 n=a，则 a 和 m 是一对亲密数，输出  */
printf(" %4d ～%4d",a,m) ；
}
```

【4.62】参考答案：

```
#include <stdio.h>
#include <stdlib.h>
main( ) /*  猜数程序  */
{ int magic ；  /*  计算机 "想" 的数  */
int guess；  /*  人猜的数  */
int counter ；
magic=rand( ) ；  /*  通过调用随机函数任意  "想"一个数  */
guess=magic-1；  /*  初始化变量 guess的值  */
counter=0 ；  /*  计数器清零  */
```

```
while(magic != guess)
{ printf("guess the magic number:")  ;
scanf("%d", &guess)  ;                    /*  人输入所猜的数     */
counter++ ;
if(guess>magic)
printf("**** Wrong **** too hight\n")       ;
else if(guess<magic )
printf("**** Wrong **** too low\n")        ;
}
printf("**** Right ****\n")        ;
printf("guess counter is %d\n", counter)  ;
}
```

【4.63】分析：直接计算阶乘的结果显然超出整型数的范围。此题的关键是如何减少计算中数的规模，注意在计算过程中出现 0 后，我们可以先行统计 0 的个数，然后将 0 从结果中移去，另外，结果仅保存个位数即可，其它位的数不会对 0 的个数产生影响。

参考答案：

```
main()
{ int i,n=0 ;
long s=1;
for(i=1 ; i<=1000 ; i++)
{ s=s*i;
while(s%10==0)
{ s=s/10;
n++ ;
}
s=s%10;
}
printf("n=%d,s=%d\n",n,s)    ;
}
```

【4.64】参考答案：

```
main()
{ int i,j,b[3][2]  ;
int a[2][3]={{1,2,3},{4,5,6}}   ;
for(i=0 ; i<=1 ; i++)
for(j=0 ; j<=2 ; j++)
b[j][i]=a[i][j]   ;
for(i=0 ; i<=2 ; i++)
{ for(j=0 ; j<=1 ; j++)
printf("%d ",b[i][j])    ;
printf("\n")   ;
}
}
```

【4.65】参考答案：

```
main()
{ int i,count=0,a[11]={0,10,2,8,22,16,4,10,6,14,20}   ;
```

```
while(1)
{ for(i=1 ; i<=10 ; i++)
a[i-1]=a[i-1]/2+a[i]/2   ;
a[10]=a[10]/2+a[0]  ;
for(i=1 ; i<=10 ; i++)
if(a[i]%2==1) a[i]++    ;
for(i=1 ; i<10 ; i++)
if(a[i]!=a[i+1]) break   ;
if(i==10) break  ;
else
{ a[0]=0 ;
count++ ;
}
}
printf("count=%d number=%d\n",count,a[1])      ;
}
```

【4.66】参考答案：

```
main()
{ int i,j,s1=0,s2=1,a[5][5]  ;
for(i=0 ; i<5 ; i++)
for(j=0 ; j<5 ; j++)
{ printf("%d %d: ",i,j)      ;
scanf("%d",&a[i][j])     ;
}
for(i=0 ; i<5 ; i++)
{ for(j=0 ; j<5 ; j++)
printf("%5d",a[i][j])     ;
printf("\n")   ;
}
j=0 ;
for(i=0 ; i<5 ; i++)
{ s1=s1+a[i][i] ;
if(i%2==0) s2=s2*a[i][i]    ;
if(a[i][i]>a[j][j]) j=i    ;
}
printf("SUN=%d\nACCOM=%d\na[%d]=%d\n",s1,s2,j,a[j][j])          ;
}
```

【4.67】参考答案：

```
#include "stdio.h"
main()
{ int i,n=0,a[4]={0}  ;
printf("Please enter a digit:")  ;
for(i=0 ; i<4 && (a[i]=getchar())!='\n'    ; i++)   ;
for(i=0 ; i<4 ; i++)
if(a[i]>=48&&a[i]<=57) a[i]=a[i]-48     ;
```

```
else if(a[i]>=65&&a[i]<=69) a[i]=a[i]-55     ;
else if(a[i]>=97&&a[i]<=102) a[i]=a[i]-87     ;
else printf("input Error!")    ;
for(i=0 ；i<4；i++)
n=n*16+a[i]  ;
printf("%d",n)    ;
}
```

【4.68】参考答案：

```
main()
{ int i,n,k=16,a[16]={0}  ;
printf("Please enter a digit:")  ;
scanf("%d",&n)   ;
while(n>0) /*   将十进制数转变为二进制数       */
{ a[--k]=n%2   ;
n=n/2；
}
for(i=0 ；i<16；i++)
printf("%2d",a[i])    ;
}
```

【4.69】参考答案：

```
#include <stdio.h>
main()
{ int i,j,m,s,k,a[100] ；
for(i=1 ；i<=100 ；i++ ) /*   寻找 1000 以内的完数    */ { m=i  ；s=0；k=0 ；
while(m>0) /*   寻找 i 的因子   */
{ for(j=1 ；j<m ； j++)
if(m%j==0)
{ s=s+j；
m=m/j ；
a[k++]=j  ；
}
if(j>=m) break  ；
}
if(s!=0&&i==s+m)
{ a[k++]=m  ；
for(j=0 ；j<k ；j++)
printf("%4d",a[j])    ;
printf("==%4d\n",i)    ;
}
}
}
```

【4.70】参考答案：

```
main()
{ int i,j,k,n,m=1,r=1,a[2][100]={0}    ;   printf("Please enter n:")  ;
scanf("%d",&n)   ;
```

```
for(i=0 ；i<n；i++)
{ printf("a[%d]= ",i)    ；
scanf("%d",&a[0][i])    ；
}
while(m<=n) /* m   记录已经登记过的数的个数       */
{ for(i=0 ；i<n ；i++) /*   记录未登记过的数的大小      */
{ if(a[1][i]!=0) /*    已登记过的数空过      */
continue；
k=i ；
for(j=i ；j<n；j++) /*   在未登记过数中找最小数      */
if(a[1][j]==0 && a[0][j]<a[0][k]) k=j      ；
a[1][k]=r++  ；  /*  记录名次,    r 为名次   */
m++；  /*  登记过的数增   1 */
for(j=0 ；j<n；j++) /*   记录同名次    */
if(a[1][j]==0 && a[0][j]==a[0][k])
{ a[1][j]=a[1][k]  ；
m++；
}
break；
}
}
for(i=0 ；i<n；i++)
printf("a[%d]=%d, %d\n",i,a[0][i],a[1][i])       ；
}
```

【4.71】参考答案：

```
#include <stdio.h>
main()
{ int i,j,k=0,m=2,s,r=0,a[500]   ；   printf("%4d ",m)    ；
for(i=3 ；i<=2000 ；i++ )
{ for(j=2 ；j<=i-1 ；j++)
if(i%j==0) break   ；
if(j==i)
{ printf("%4d ", i )    ；
a[k++]=i-m  ；
m=i ；
}
}
for(i=0 ；i<k ；i++)
{ s=0；
for(j=i ；j<k ；j++)
{ s=s+a[j] ；
if(s>=1898) break  ；
}
if(s==1898)
```

```
r++ ；
}
printf("\nresult=%d\n",r)    ；
}
```

【4.72】分析：本问题的思路很多，我们介绍一种简单快速的算法。

首先求出三位数中不包含 0 且是某个整数平方的三位数，这样的三位数是不多的。然后将满足条件的三位数进行组合，使得所选出的三个三位数的九个数字没有重复。程序中可以将寻找满足条件三位数的过程和对该三位数进行数字分解的过程结合起来。

参考答案：

```
#include <stdio.h>
main( )
{ int a[20],num[20][3],b[10] ； /* a: 存放满足条件的三位数 */
/* num ：满足条件的三位数分解后得到的数字, b: 临时工作 */
int i,j,k,m,n,t,flag ；
printf("The 3 squares with 3 different digits each are:\n") ；
for(j=0,i=11 ；i<=31 ；i++) /* 求出是平方数的三位数 */
if(i%10 != 0) /* 若不是 10 的倍数，则分解三位数 */
{ k=i*i ； /* 分解该三位数中的每一个数字 */
num[j+1][0]=k/100 ； /* 百位 */
num[j+1][1]=k/10%10 ； /* 十位 */
num[j+1][2]=k%10 ； /* 个位 */
if(!(num[j+1][0]==num[j+1][1] || num[j+1][0]==num[j+1][2]
|| num[j+1][1]==num[j+1][2]) )
/* 若分解的三位数字均不相等 */
a[++j]=k ； /* j: 计数器，统计已找到的满足要求的三位数 */
}
for(i=1 ；i<=j-2 ；++i ) /* 从满足条件的三位数中选出三个进行组合 */
{ b[1]=num[i][0] ； /* 取第 i 个数的三位数字 */
b[2]=num[i][1] ；
b[3]=num[i][2] ；
for(t=i+1 ；t<=j-1 ；++t)
{ b[4]=num[t][0] ； /* 取第 t 个数的三位数字 */
b[5]=num[t][1] ；
b[6]=num[t][2] ；
for(flag=0, m=1 ；!flag&&m<=3 ；m++) /* flag: 出现数字重复的标记 */
for(n=4 ；!flag&&n<=6 ；n++) /* 判断前两个数的数字是否有重复 */
if(b[m]==b[n]) flag=1 ； /* flag=1: 数字有重复 */
if(!flag)
for(k=t+1 ；k<=j ；++k)
{ b[7]=num[k][0] ； /* 取第 k 个数的三位数字 */
b[8]=num[k][1] ；
b[9]=num[k][2] ；
/* 判断前两个数的数字是否与第三个数的数字重复 */
for(flag=0,m=1 ；!flag&&m<=6 ；m++)
for(n=7 ；!flag&&n<=9 ；n++)
```

```
if(b[m]==b[n]) flag=1  ;
if(!flag) /*   若均不重复则打印结果      */
printf("%d, %d, %d\n",a[i],a[t],a[k])       ;
}
}
}
}
```

【4.73】参考答案：

```
main()
{ int i,n,k,a[3],b[3]  ;
for(i=248  ;  i<=343  ;  i++)
{ for(n=i,k=0  ;  n>0 ;  n/=7)
a[k++]=n%7  ;
for(n=i,k=0  ;  n>0 ;  n/=9)
b[k++]=n%9  ;
if(k==3)
for(n=0  ;  n<k  ;  n++)
if(a[n]!=b[k-n-1])
break ;
if(n==k)
printf("%d\n",i)    ;
}
}
```

【4.74】参考答案：

```
#include <stdio.h>
int pos[101],div[101] ;
main ()
{ int m, n, i, j  ;
printf("Please input m/n(<0<m<n<=100):")    ;
scanf("%d%d", &m,&n)     ;
printf("%d/%d=0.", m, n)     ;
for(i=1  ;  i<=100  ;  i++)
{ pos[m]=i ;
m*=10  ;
div[i]=m/n  ;
m=m%n  ;
if(m==0)
{ for( j=1  ;  j<=i  ;  j++) printf("%d",div[j])     ;
break ;
}
if(pos[m]!=0)
{ for( j=1  ;  j<=i  ;  j++) printf("%d",div[j])     ;
printf("\nloop: start=%d, end=%d",pos[m], i)     ;
break ;
}
```

```
}
printf("\n")   ;
}
```

【4.75】参考答案：

```
#include "stdio.h"
int a[20],b[20]  ;
main()
{ int t=0,*m,*n,*k,*j,z,i=0      ;
printf("Input number 1:")     ;
do
{ a[++t]=getchar()-'0'    ;
}while(a[t]!=-38)   ;
printf("Input number 2:")     ;
do
{ b[++i]=getchar()-'0'    ;
}while(b[i]!=-38)   ;
if(t>i)
{ m=a+t  ;  n=b+i  ;  j=a  ;  k=b  ;  z=i  ;
}
else
{ m=b+i  ;  n=a+t  ;  j=b  ;  k=a  ;  z=t  ;
}
while(m!=j)
{ (*(--n-1))+=(*(--m)+*n)/10      ;
*m=(*m+*n)%10     ;
if (n==k+1 && *k!=1 ) break      ;
if (n==k+1 && *k)
{ n+=19  ;  *(n-1)=1   ;
}
if (n>k+z && *(n-1)!=1) break      ;
}
while (*(j++)!=-38) printf("%d",*(j-1))        ;
printf("\n")   ;
}
```

【4.76】参考答案：

```
#include "stdio.h"
int a[20],b[20]  ;
main()
{ int t=0,*m,*n,*k,*j,z,i=0      ;
printf("Input number 1:")     ;
do
{ a[++t]=getchar()-'0'    ;
}while(a[t]!=-38)   ;
printf("Input number 2:")     ;
do
```

```
{ b[++i]=getchar()-'0'    ;
}while(b[i]!=-38)   ;
if(t>i)
{ m=a+t  ;  n=b+i  ;  j=a  ;  k=b  ;  z=i  ;
}
else
{ m=b+i  ;  n=a+t  ;  j=b  ;  k=a  ;  z=t  ;
}
while(m!=j)
{ (*(--n-1))+=(*(--m)+*n)/10      ;
*m=(*m+*n)%10     ;
if(n==k+1 && *k!=1 ) break      ;
if(n==k+1 && *k)
{ n+=19  ;  *(n-1)=1   ;
}
if(n>k+z && *(n-1)!=1) break      ;
}
while(*(j++)!=-38) printf("%d",*(j-1))        ;
printf("\n")    ;
}
```

【4.77】参考答案：

```
#include "stdio.h"
int a[20],b[20],c[40]   ;
main()
{ int t=0,*m,*n,*k,f,e=0,*j,i=0      ;
printf("Input number 1:")     ;
do
{ a[++t]=getchar()-'0'    ;
}while(a[t]!=-38)   ;
printf("Input number 2:")     ;
do
{ b[++i]=getchar()-'0'    ;
}while(b[i]!=-38)   ;
j=c  ;
for(m=a+t-1   ;  m>=a+1  ;  m--,e++)
{ j=c+e  ;
for(n=b+i-1   ;  n>=b+1  ;  n--)
{ f=*j+*m * *n     ;
*(j++)=f%10    ;
*j+=f/10   ;
}
}
while(j>=c) printf("%d",*(j--))        ;
printf("\n")    ;
}
```

【4.78】这是一个使用数组解决较复杂问题的典型题目。

棋盘如左图所示，图中箭头表示一个棋子从 [1、 1]点跳到 [2、 3]点。为了下面叙述的方便，将 I、J 表示棋子起跳点的行、列号， X、Y 表示落子点的行、列号。

首先我们讨论如何从起跳点坐标求出可能的落子点的坐标。

从某点起跳，棋子最多可能有八个落子点，例如从 I=3、J=5 点起跳，八个可能的落子点的坐标是[4、 5]、[5、 4]、[5、 2]、[4、 1]、[2、 1]、[1、 2]、[1、 4]、[2、 5]。将起落点的行列坐标分开考虑，则由起点的行坐标分别与下列八个数相加，就可得到可能的八个落子点的行坐标： 1、 2、2、 1、 -1、 -2、 -2、 -1，将这八个数存入数组 b，即：

b[]={1,2,2,1,-1,-2,-2,-1},

落子点的行坐标 X 和起跳点行坐标有如下关系：

X=b[k]+I 1 ？ k？ 8

如果由上式计算得到的落子点 X 的坐标值小于 0 或大于 8，则表示落在了棋盘之外，应予舍弃。

同理得到起落点之间的列坐标关系数组是：

d[]={2,1,-1,-2,-2,-1,1,2} 。

我们再讨论落子点的度数问题。对于棋盘中的某一点来说，周围最多有 8 个方向的棋子在这个点落子，把可能的落子数称为度数，棋盘上各点的度数如下图所示。

根据题意，一个点只能落子一次，所以落过子的点的度数应记为 0，可跳向度数为 0 点的度数相应要减 1。

根据上述数组，从一个起跳点出发，可能求出数个可以落子点的坐标，跳棋时到底确定落在这些点中的哪一个呢？我们确定一个原则是落在度数最少的点。如果可能落子点中有两个点的度数一样且都为是度数最少时，取后求出的点为落子点。因此，如果改变数组 b、d 中数的存放顺序，遇到两个度数最少点的先后顺序就要改变，整个跳棋路径就可改变。

2 3 4 4 4 4 3 2

3 4 6 6 6 6 4 3

4 6 8 8 8 8 6 4

4 6 8 8 8 8 6 4

4 6 8 8 8 8 6 4

4 6 8 8 8 8 6 4

3 4 6 6 6 6 4 3

2 3 4 4 4 4 3 2

在下面的程序中，将起落子行列关系的两个一维数组合并为一个二维数组，为了提高程序的可读性，不使用下标为 0 的数组元素。

参考程序：

```
int base[9][3]={0, 0, 0, /*  从起跳点求落脚点的基础系数数组      */
0, 1, 2,
0, 2, 1,
0, 2,-1,
0, 1,-2,
0,-1,-2,
0,-2,-1,
0,-2, 1,
0,-1, 2 } ;
main()
{ int a[9][9],object[9][9]  ;
int i,j,k,p,x,y,m,n,cont  ;
int min,rm1,rm2,rm0=1   ;
```

```
for(cont=1 ； cont>0； )
{ for(i=0 ； i<=8 ； i++) /*  保存个点度数的数组      清零  */
for(j=0 ； j<=8 ； j++)
a[i][j]=0 ；
rm1=base[1][1] ；  /*  改变基础数组元素排列顺序       */
rm2=base[1][2] ；
base[1][1]=base[rm0][1] ；
base[1][2]=base[rm0][2] ；
base[rm0][1]= rm1 ；
base[rm0][2]= rm2 ；
for(i=1 ； i<=8 ； i++)
{ for(j=1 ； j<=8 ； j++) /*  计算各点度数存入数组      a */
{ for(p=1 ； p<=8 ； p++)
{ x=i+base[p][1] ；
y=j+base[p][2] ；
if(x>=1&&x<=8&&y>=1&&y<=8)
a[x][y]++ ；
}
printf(" %d",a[i][j])    ；  /*  输出度数表    */
}
printf("\n")  ；
}
printf("Please Input start position:line,colume=?\n")  ；
scanf("%d,%d",&i,&j)    ；  /*  输入起跳点坐标     */
for(k=1 ； k<=63 ； k++) /*  求棋盘上  63 个落步点   */
{ object[i][j]=k ；  /*  跳步路径存入数组    object */
min=10 ；
for(p=1 ； p<=8 ； p++) /*  求从当前起跳点出发的     8 个可能落点    */
{ x=i+base[p][1] ；
y=j+base[p][2] ；
if(x>=1&&x<=8&&y>=1&&y<=8) /*       求出的可能落点在棋盘内       */
if(a[x][y]!=0) /*  此点没有落过棋子     */
{ a[x][y]--  ；  /*  由于 [i、 j]点落过棋子，此点度数减     1 */
if(min>a[x][y]) /*  判断当前可能点度数是否最小       */
{ min=a[x][y] ；  /*  保存可能最小度数点的度数       */
m=x ；  /*  保存可能最小度数点的坐标       */
n=y；
}
}
}
}
a[i][j]=0 ；  /*  落过棋子的  [i、 j]点度数为零    */
i=m ；  /*  已求出的最小度数点为下次搜寻的起跳点         */
j=n ；
}
object[i][j] = 64   ；
```

```
for(i=1 ； i<=8 ； ++i) /*   输出跳步结果路径     */
{ for(j=1 ；j<=8 ；j++)
if(j==8) printf("%2d",object[i][j])      ；
else printf("%2d ",object[i][j])    ；
printf("\n")   ；
if(i!=8) printf(" \n")    ；  /*  每行输出  8 个数据   */
}
rm0%=8  ；  /*  放在基础数组第一位的元素循环变化        */
rm0++  ；  /*  基础数组下一元素放在第一位        */
printf("continue?(1 or 0)")   ；
scanf("%d",&cont)   ；
}
}
```

【4.79】分析：采用试探法求解。

如图所示，用 I、J 表示行、列坐标。

开始棋盘为空， 对于第 1 个皇后先占用第一行即 I=1，先试探它占用第一列 J=1 位置，则它所在的行、列和斜线方向都不可在放其它皇后了，用线将它们划掉。第 2 个皇后不能放在 J=1、2 的位置，试 J=3。第 2 个皇后占用 [2、3]后，它的行列和斜线方向也不可再放其它皇后。第 3 个皇后不能放在 J=1、2、3、4 的位置， 试 J=5。第 4 个皇后可以试位置 [4、2]，第 5 个皇后试位置 [5、4]。第 6 个皇后已经没有可放的位置（棋盘上所有格子都已占满） ，说明前面所放位置不对。

退回到前一个皇后 5，释放它原来占用的位置 [5、4]，改试空位置 [5、8]。然后再前进到第 6 个皇后，此时仍无位置可放，退回到第 5 个皇后，它已没有其它位置可选择。进一步退回到第 4 个皇后释放位置 [4、2]改试位置 [4、7]，再前进到第 5 个皇后进行试探，如此继续，直到所有 8 个皇后都选择一个合适的位置，即可打印一个方案。

然后从第 8 个皇后开始，改试其它空位置，若没有可改选的空位置，则退回到第 7 个皇后改试其它位置，若也没有空位置可改，继续退，直到有另外的空位置可选的皇后。将它原来占用的位置释放，改占其它新位置，然后前进到下一个皇后进行试探，直到所有 8 个皇后都找到合适位置，又求出一个解，打印输出新方案。按此方法可得到 92 个方案。

参考答案：

```
#define NUM 8
int a[NUM+1]  ；
main()
{ int number,i,k,flag,nonfinish=1,count=0  ；
i=1 ；
a[1]=1 ；
while(nonfinish)
{ while(nonfinish && i<=NUM)
{ for(flag=1,k=1  ； flag && k<i  ； k++)
if(a[k]==a[i]) flag=0   ;
for(k=1  ；flag && k<i   ；k++)
if((a[i]==a[k]-(k-i)) || (a[i]==a[k]+(k-i))) flag=0      ;
if(! flag)
{ if(a[i]==a[i-1])
{ i--  ;
if(i>1 && a[i]==NUM) a[i]=1      ;
```

```
else if(i==1 && a[i]==NUM) nonfinish=0      ;
else a[i]++ ;
}
else if(a[i]==NUM) a[i]=1     ;
else a[i]++ ;
}
else if( ++i<=NUM )
if(a[i-1]==NUM) a[i]=1      ;
else a[i]=a[i-1]+1  ;
}
if(nonfinish)
{ printf("\n%2d:",++count)     ;
for(k=1  ；k<=NUM   ；k++)
printf(" %d",a[k])     ;
if(a[NUM-1]<NUM) a[NUM-1]++        ;
else a[NUM-1]=1  ;
i=NUM-1   ;
}
}
}
```

【4.80】参考答案：

```
double findy(float x)
{ if(x>=0 && x<2)
retuen(2.5-x) ;
else if(x>=2 && x<4)
return(2-1.5*(x-3)*(x-3))     ;    else if(x>=4 && x<6)
return(x/2.0-1.5)   ;
}
main()
{ float x ;
printf("Please enter x:")   ;
scanf("%f",&x)    ;
if(x>=0 && x<6)
printf("f(x)=%f\n",findy(x))      ;
else
printf("x is out!\n")    ;
}
```

【4.81】注释：此程序采用模拟手工方式，对分数进行通分后比较分子的大小。

参考答案：

```
main( )
{ int i, j, k, l, m, n   ;
printf("Input two FENSHU :\n")      ;
scanf("%d/%d,%d/%d", &i, &j, &k, &l)          ;  /*  输入两个分数     */
m = zxgb(j,l)/j * i   ;   /*  求出第一个分数通分后的分子        */
```

```
n = zxgb(j,l)/l * k  ;   /*  求出第二个分数通分后的分子        */
if(m>n )
printf("%d/%d > %d/%d\n",i,j,k,l)        ;   /*  比较分子的大小      */
else if(m==n)
printf("%d/%d = %d/%d\n",i,j,k,l)        ;   /*  输出比较的结果      */
else printf("%d/%d < %d/%d\n",i,j,k,l)        ;
}
zxgb(a,b)
int a,b ;
{ long int c;
int d ;
if(a<b) c=a, a=b, b=c  ;   /*  若 a<b，则交换两变量的内容       */
for( c=a*b  ;   b!=0 ;   ) /*  用辗转相除法求    a 和  b 的最大公约数     */
{ d=b ;   b=a%b  ;   a=d ;
}
return((int) c/a)  ;   /*  返回最小公倍数     */
}
```

【4.82】参考答案：

```
main()
{ int a[5],i,t,k  ;
for (i=100  ; i<1000 ; i++)
{ for(t=0,k=1000  ; k>=10  ; t++)
{ a[t]=(i%k)/(k/10)    ;
k/=10 ;
}
if(f(a[0])+f(a[1])+f(a[2])==i)
printf("%d ",i)    ;
}
}
f(m)
int m ;
{ int i=0,t=1  ;
while(++i<=m) t*=i    ;
return(t)  ;
}
```

【4.83】分析：任取两个平方三位数 n 和 n1，将 n 从高向低分解为 a、 b、 c，将 n1 从高到低分解为 x、 y、 z。判断 ax、 by、 cz 是否均为完全平方数。

参考答案：

```
main( )
{ void f( ) ;
int i,t,a[3],b[3]  ;
printf("The possible perfect squares combinations are:\n")  ;
for(i=11 ; i<=31 ; i++) /*   穷举平方三位数的取值范围       */
for(t=11 ; t<=31 ; t++)
{ f(i*i,a)  ;   /*  分解平方三位数的各位，每位数字分别存入数组中          */
```

```
f(t*t,b) ；
if(sqrt(a[0]*10+b[0])==(int)sqrt(a[0]*10+b[0])
&& sqrt(a[1]*10+b[1])==(int)sqrt(a[1]*10+b[1])
&& sqrt(a[2]*10+b[2])==(int)sqrt(a[2]*10+b[2]))
/*  若三个新的数均是完全平方数        */
printf(" %d and %d\n",i*i,t*t)       ；  /*  则输出   */
}
}
void f(n,s) /*  分解三位数   n 的各位数字，将各个数字       */
int n, *s ；   /*  从高到低依次存入指针     s 所指向的数组中     */
{ int k ；
for(k=1000 ；k>=10 ；s++)
{ *s = (n%k)/(k/10)   ；
k /= 10 ；
}
}
```

【4.84】参考答案：

```
main()
{ int i,j,l,n,m,k,a[20][20]   ；
printf("Please enter n,m=")   ；
scanf("%d,%d",&n,&m)      ；
for(i=0 ；i<n ；i++)
for(j=0 ；j<m ；j++)
{ printf("a[%d][%d]=",i,j)      ；
scanf("%d",&a[i][j])     ；
}
for(i=0 ；i<n ；i++)
{ for(j=0 ；j<m ；j++)
printf("%6d",a[i][j])     ；
printf("\n")   ；
}
for(i=0 ；i<n ；i++)
{ for(j=0,k=0   ；j<m ；j++)
if(a[i][j]>a[i][k]) k=j    ；  /*  找出该行最大值     */
for(l=0 ；l<n ；l++) /*   判断  a[i][k] 是否为该列最小     */
if(a[l][k]<a[i][k]) break    ；  /*  该列有一个数比    a[i][k] 小   */
if(l>=n) /*    没有比  a[i][k] 小的数，循环变量     l 就超过最大值     */
printf("Point:a[%d][%d]==%d",i,k,a[i][k])       ；
}
}
```

【4.85】分析：按题目的要求进行分析，数字 1 一定是放在第一行第一列的格中，数字 6 一定是放在第二行第三列的格中。在实现时可用一个一维数组表示，前三个元素表示第一行，后三个元素表示第二行。先根据原题初始化数组，再根据题目中填写数字的要求进行试探。

参考答案：

```
#include <stdio.h>
```

```
int count；  /*  计数器  */
main( )
{ static int a[ ]={1,2,3,4,5,6} ；  /*  初始化数组  */
printf("The possible table satisfied above conditions are:\n") ；
for(a[1]=a[0]+1 ；a[1]<=5 ；++a[1]) /* a[1] 必须大于 a[0] */
for(a[2]=a[1]+1 ；a[2]<=5 ；++a[2]) /* a[2] 必须大于 a[1] */
for(a[3]=a[0]+1 ；a[3]<=5 ；++a[3]) /*  第二行的 a[3]必须大于 a[0] */
for(a[4]=a[1]>a[3]?a[1]+1:a[3]+1  ；a[4]<=5 ；++a[4])
/*  第二行的 a[4]必须大于左侧 a[3]和上边 a[1] */
if(jud1(a))
print(a) ；  /*  如果满足题意，打印结果  */
}
jud1(s) /*  判断数组中的数字是否有重复的  */
int s[ ]；
{ int i,l ；
for(l=1 ；l<4；l++)
for(i=l+1 ；i<5；++i)
if(s[l]==s[i])
return(0) ；  /*  若数组中的数字有重复的，返回０  */
return(1) ；  /*  若数组中的数字没有重复的，返回１  */
}
print(u)
int u[ ]；
{ int k ；
printf("\nNo.:%d", ++count)  ；
for(k=0 ；k<6 ；k++)
if(k%3==0) /*  输出数组的前三个元素作为第一行  */
printf("\n %d ",u[k])  ；
else /*  输出数组的后三个元素作为第二行  */
printf("%d ",u[k])  ；
}
```

【4.86】参考答案：

```
#include "string.h"
strcmbn(a,b,c) /*  数组合并函数：将数组 a、 b 合并到  */
char a[],b[],c[] ；
{ char tmp ；
int i,j,k,m,n ；
m=strlen(a) ；
n=strlen(b) ；
for(i=0 ；i<m-1 ；i++) /*  对数组 a 排序  */
{ for(j=i+1,k=i  ；j<m ；j++)
if(a[j]<a[k]) k=j  ；
tmp=a[i] ；  a[i]=a[k] ；  a[k]=tmp ；
}
for(i=0 ；i<n-1；i++) /*  对数组 b 排序  */
```

```
{ for(j=i+1,k=i  ；j<n ；j++)
if(b[j]<b[k]) k=j  ；
tmp=b[i] ；  b[i]=b[k]  ；  b[k]=tmp  ；
}
i=0 ；j=0 ；k=0；
while(i<m&&j<n) /*      合并   */
if(a[i]>b[j])
c[k++]=b[j++]  ；  /*  将 a[i]、 b[j]中的小者存入   c[k] */
else
{ c[k++]=a[i++]  ；
if(a[i-1]==b[j]) j++  ；  /*  如果 a、 b 当前元素相等，删掉一个      */
}
while(i<m) c[k++]=a[i++]  ；  /*  将 a 或 b 中剩余的数存入   c */
while(j<n) c[k++]=b[j++]  ；
c[k]='\0' ；
}
```

【4.87】参考答案：

```
pxn(x,n)
float x；
int n ；
{ if(n==0) return(1)  ；
else if(n==1) return(x)  ；
else return(((2*n-1)*x*pxn(x,n-1)-(n-1)*pxn(x,n-2))/2)      ；
}
```

【4.88】参考答案：

```
#include "stdio.h"
strout(s)
char *s ；
{ if(*s!='\0')
{ strout(s+1)；  /*  递归调用   strout 函数，字符串首地址前移一个字符        */
putch(*s) ；  /*  输出字符串首地址所指向的字符        */
}
else return；  /*  遇到字符串结束标志结束递归调用        */
}
```

【4.89】参考答案：杨辉三角形中的数，正是 (x+y)的 N 次方幂展开式中各项的系数。本题作为程序设计中具有代表性的题目，求解的方法很多（可以使用一维数组，也可以使用二维数组），前面我们给出用数组的答案，这里给出一种使用递归求解的方法。

从杨辉三角形的特点出发，可以总结出：

？ 第 N 行有 N+1 个值（设起始行为第 0 行）；

？ 对于第 N 行的第 J 个值： (N>=2)

当 J=1 或 J=N+1 时： 其值为 1

当 J!=1 且 J!=N+1 时： 其值为第 N-1 行的第 J-1 个值与第 N-1 行第 J 个值之和。

将这些特点提炼成数学公式可表示为：

c(x,y) = 1 x=1 或 x=N+1

c(x,y) = c(x-1,y-1) + c(x-1,y) 其它

下面给出的程序就是根据以上递归的数学表达式编制的。

参考答案：

```
#include <stdio.h>
main( )
{ int i,j,n=13 ；
printf("N=")  ；
while( n>12 )
scanf("%d", &n)  ；  /*  最大输入值不能大于  12 */
for(i=0 ；i<=n ；i++) /*  控制输出  N 行  */
{ for(j=0 ；j<12-i ；j++)
printf(" ")  ；  /*  控制输出第  i 行前面的空格  */
for(j=1 ；j<i+2 ；j++)
printf("%6d", c(i,j))  ；  /*  输出第  i 行的第  j 个值  */
printf("\n")  ；
}
}
int c(x,y) /*  求杨辉三角形中第  x 行第  y 列的值  */
int x, y ；
{ int z ；
if((y==1)||(y==x+1))
return(1) ；  /*  若为  x 行的第  1 或第  x+1 列，则输出  1 */
else /*  否则；其值为前一行中第  y-1 列与第  y 列值之和  */
z = c(x-1,y-1) + c(x-1,y)  ；
return(z)  ；
}
```

【4.90】分析：整型数在计算机中就是以二进制形式存储的，此题的目的仅是为了学习递归程序的编程。

参考答案：

```
turn(n,a,k)
int n,a[ ],k  ；
{ if(n>0)
{ a[k]=n%2  ；
turn(n/2,a,k-1)  ；
}
else return；
}
main()
{ int i,n,a[16]={0}  ;
printf("\nPlease enter n:")  ;
scanf("%d",&n)  ;
turn(n,a,15)  ;
for(i=0  ；i<16 ；i++)
printf("%d",a[i])  ;
}
```

【4.91】分析：分析题目，我们可以将题目进行抽象：在有放回的前提下，求全部从 m 个不同的元素中任取 n 个元素的排列。根据题目的含义，我们可以用整数 0～m-1 表示这 m 个不同的元素，将要生成的 n 个元素分为两部分：第一个元素和其它 n-1 个元素。如果 n=1，即要从 m 个元素中任取 1 种，这样有 m 种不同得取法，我们可以直接使用循环完成。若 n>1，则可以知道，第一个元素一定有 m 种不同的取法，可以针对第一个元素 m 种不同取法种的 1 种，对后面的 n-1 个元素进行同样的（递归）操作即可产生一种新的不同的排列。具体算法描述如下：

fun（指向第一个元素的指针，从 m 个元素中，取 n 个元素）
{ for (i=0 ； i<m； i++)
{ 确定第一个元素的选取方法： =i；
if （n>1） fun（指向下一个元素的指针，从 m 中，取 n-1 个）
else 完成一种排列
}
}
根据以上算法分析可以得出程序。
参考答案：

```
#include <stdio.h>
int a[10]；
fun( int *p, int m, int n ) /*  从 m 个元素中取 n 个存入数组 p 中 */
{ int i ； /*  用数 0～m-1 表示 m 个不同的元素 */
for( i=0 ；i<m ；i++ ) /*  依次从数 0 开始逐个作为第一个元素 */
{ *p = i ；
if( n > 1 ) fun( p+1, m, n-1) ；
else print(p)；
}
}
print( int *p )
{ int *q ；
for( q=a ；q<=p ；q++ ) /*  输出结果，将整数转换为字母 a 起始的序列 */
printf("%c", 'a'+ *q) ；
printf("\t") ；
}
main( )
{ int m, n ；
printf("\nEnter m n:") ；
scanf("%d%d", &m, &n) ；
fun( a, m, n) ；
}
```

【4.92】参考答案：

```
smmt ( char s[ ] ) /*  指针 s 指向字符串的第一个字符 */
{ char *p ；
p=s；
while(*p!='\0') p++ ；
```

```
p--；  /*  指针  p 指向字符串的最后一个字符        */
if(p==s) return(1)  ；  /*  两个指针指向同一个字符表示字符串对称          */
else
{ if(*s!=*p)
return(0)  ；  /*  两个指针指向字符不等表示字符串不对称          */
else
{ *p='\0'  ；
smmt(s+1)；  /*  取掉首尾比较过的字符继续比较        */
}
}
```

【4.93】参考答案 ：

```
#include <stdio.h>
int n, r, flag ；  /* flag ：标志,   =0：表示要另起一行      */
main( )
{ int s；
printf("Enter N,R:")    ；
scanf("%d%d", &n, &r)     ；
printf("combinations:\n")   ；
flag=1；
combination (1,r)  ；
}
combination ( s, j )
int s, j；  /*  从  s 开始选  j 个元素   */
{ int i,k  ；
for( i=s ；i<=n-j+1  ；i++ )
{ if( flag )
for( k=0  ；k<r-j  ；k++ ) printf(" ")    ；
printf("%3d ", i)    ；
flag=0；
if(j>1) combination ( i+1, j-1 )   ；
else { putchar('\n')  ；  flag=1；  }
}
}
```

【4.94】分析：此题给出的参考答案使用了指针和函数递归的概念。读者在学习完指针的概念后再研究此题。放于此处主要是便于和其它排序方法比较。

合并排序法排序的步骤是：第一次将数组中相邻的 2 个数两两排序，第二遍 4 个 4 个地排序，第三遍 8 个 8 个地排序 ..... 。程序中的合并排序函数 (mergesort)采用了递归调用。例如有一组数是： 4，3，1，81，45，8，0，4，-9，26，7，4，2，9，1，-1

采用合并排序法的过程如下：

未排序时 4 3 1 81 45 8 0 4 -9 26 7 4 2 9 1 -1

第一遍后 3 4 1 81 8 45 0 4 -9 26 4 7 2 9 -1 1

第二遍后 1 3 4 81 0 4 8 45 -9 4 7 26 -1 1 2 9

第三遍后 0 1 3 4 4 8 45 81 -9 -1 1 2 4 7 9 26

第四遍后 -9 -1 0 1 1 2 3 4 4 4 7 8 9 26 45 81

参考答案：

```
#define N 16
#include "stdio.h"
merge(a, b, c, m) /* 数组合并函数：将长度为 m 的*/
int a[], b[], c[], m； /* 数组 a、b 合并到 c */
{ int i=0 , j=0 , k=0 ；
while(i<m&&j<m)
if(a[i]>b[j])
c[k++]=b[j++] ； /* 将 a[i]、*b[j] 中的小 */
else c[k++]=a[i++] ； /* 者存入 c[k] */
while(i<m) c[k++]=a[i++] ； /* 将 a 或 b 中剩余的数 */
while(j<m) c[k++]=b[j++] ； /* 存入 c */
}
mergesort(w, n) /* 数组排序函数 : 对长度为 n */
int w[] , n； /* 的数组 w 排序 */
{ int i , t, ra[N] ；
for(i=1 ；i<n ；i*=2) ；
if(i==n)
{ if(n>2) /* 递归调用结束条件 */
{ mergesort (w , n/2)； /* 将数组 w 一分为二，递归调 */
mergesort (w+n/2 , n/2)； /* 用 mergesort */
merge( w, w+n/2 , ra, n/2 )； /* 将排序后的两数组重新合并 */
for(i=0 ; i<n ; i++)
w[i]=ra[i] ;
} else if(*w>*(w+1))
{ t=*w ; *w=*(w+1) ; *(w+1)=t ;
} }
else printf("Error:size of array is not a power of 2/n") ;
}
main( )
{ int i ;
static int key[N]={4,3,1,81,45,8,0,4,-9,26,7,4,2,9,1,-1} ;
mergesort(key, N) ;
for(i=0 ; i<N ; i++)
printf("%d " , key[i]) ;
printf("\n") ;
}
```

【4.95】参考答案：

```
#include "stdio.h"
main( )
{ char s[21],*p,*q ；
gets(s)；
p=s ；
q=s ；
while(*q!='\0') q++ ；
```

```
q-=2；
while(p<q) /*   指针  p 指向字符串首，指针     q 指向串末   */
if(*p++ != *q--) /*     指针  p、 q 同时向中间移动，比较对称的两个字符          */
{ printf("NO\n")   ；
break ；
}
if(p>=q)
printf("YES\n")   ；
}
```

【4.96】参考答案：

```
strcut(s,m,k)
char s[ ]；
int m,k ；
{ char *p ；
int i；
p=s+m ；  /*  指针  p 指向要被删除的字符      */
while((*p=*(p+k))!='\0') /* p+k      指向要前移的字符      */
p++ ；
}
```

【4.97】参考答案：

```
strchg(s)
char *s ；
{ char c,*p ；
p=s；
while(*p!='\0') p++   ；
p-- ；
while(s<p)
{ c=*s ；
*s++=*p ；
*p--=c ；
}
}
```

【4.98】参考答案：

```
insert(s1,s2,ch)
char s1[],s2[],ch；
{ char *p,*q  ；
p=s1 ;
while(*p++!=ch)    ;
while(*s2!='\0')
{ q=p ；
while(*q!='\0') q++   ;
while(q>=p)
*(q+1)=*q--   ;
*++q=*s2++   ;
p++ ；
```

```
}
}
```

【4.99】参考答案：

```
strcnb(s1,s2)
char s1[],s2[]；
{ char *p  ；
int i=1 ；
p=s1；
while(*p!='\0') p++    ；
while((*p++=*s2++)!='\0')      ；  /*  将 s2 接于  s1后面  */
p=s1；
while(*p!='\0') /*     扫描整个字符串      */
{ if(*p==' ') /*     当前字符是空格进行移位        */
{ while(*(p+i)==' ') i++    ；  /*  寻找当前字符后面的第一个非空格         */
if(*(p+i)!='\0')
{ *p=*(p+i)   ；  /*  将非空格移于当前字符处        */
*(p+i)=' '   ；  /*  被移字符处换为空格      */
}
else break；   /*  寻找非空格时到字符串尾，移位过程结束          */
}
p++ ；
}
```

【4.100】参考答案：

```
#include "stdio.h"
struct strnum
{ int i ；
char ch；
}
main( )
{ char c；
int i=0,k=0  ；
struct strnum s[100]={0,NULL}    ；
while((c=getchar())!='\n')
{ for(i=0 ；s[i].i!=0 ；i++)
{ if(c==s[i].ch)
{ s[i].i++ ；
break ；
}
}
if(s[i].i==0)
{ s[k].ch=c；
s[k++].i=1 ；
}
}
```

```
i=0 ；
while(s[i].i>0)
{ printf("%c=%d ",s[i].ch,s[i].i)    ；
i++ ；
}
}
```

【4.101】分析：程序中函数 **cmult** 的形式参数是结构类型，函数 **cmult** 的返回值也是结构类型。在运行时，实参 **za** 和 **zb** 为两个结构变量，实参与形参结合时，将实参结构的值传递给形参结构，在函数计算完毕之后，结果存在结构变量 **w** 中，**main** 函数中将 **cmult** 返回的结构变量 **w** 的值存入到结构变量 **z** 中。这样通过函数间结构变量的传递和函数返回结构型的计算结果完成了两个复数相乘的操作。

参考答案：

```
#include "stdio.h"
struct complx
{ int real ；  /* real 为复数的实部    */
int im ；  /* im 为复数的虚部    */
}；
main( )
{ static struct complx za = {3,4} /*  说明结构静态变量并初始化      */
static struct complx zb = {5,6} ；
struct complx x, y, z；
struct complx cmult() ；/*  说明函数 cmult 的返回值类型是结构    complx 型  */
void cpr( ) ；
z=cmult(za, zb) ；  /*  以结构变量调用    cmult 函数，返回值赋给结构变量      z */
cpr (za, zb, z) ；  /*  以结构变量调用    cpr 函数，输出计算结果      */
x.real = 10 ；  x.im = 20 ；
y.real = 30 ；  y.im = 40 ；  /*  下一组数据    */
z = cmult (x, y) ；
cpr (x, y, z)；
}
struct complx cmult(za, zb) /*  计算复数 za×zb，函数的返回值为结构类型      */
struct complx za, zb ；  /*  形式参数为结构类型      */
{ struct complx w ；
w.real = za.real * zb.real - za.im * zb.im   ；
w.im = za.real * zb.im + za.im * zb.real    ；
return (w) ；  /*  返回计算结果，返回值的类型为结构      */
}
void cpr (za,zb,z) /*  输出复数 za×zb=z */
struct complx za, zb, z；  /*  形式参数为结构类型      */
{ printf ("(%d+%di)*(%d+%di)=", za.real, za.im, zb.real, zb.im)        ；
printf ("(%d+%di)\n", z.real, z.im)     ；
}
```

【4.102】参考答案一 ：

```
#include "stdio.h"
```

```
struct student
{ int n ;
int mk ;
} ;
main()
{ int i,j,k,count=0,no  ;
struct student stu[100],*s[100],*p  ;
printf("\nPlease enter mark(if mark<0 is end)\n")     ;
for(i=0 ; i<100 ; i++)
{ printf("No.%04d==",i+1)     ;
scanf("%d",&stu[i].mk)    ;
s[i]=&stu[i]  ;
stu[i].n=i+1  ;
if(stu[i].mk<=0) break   ;
for(j=0  ; j<i  ; j++)
for(k=j+1   ; k<=i  ; k++)
if(s[j]->mk<s[k]->mk)
{ p=s[j] ;   s[j]=s[k] ; s[k]=p ;
}
}
for(no=1,count=1,j=0   ; j<i  ; j++)
{ if(s[j]->mk > s[j+1]->mk)
{ printf("\nNo.%3d==%4d%4d : ",no,s[j]->mk,count)       ;
for(k=j-count+1   ; k<=j  ; k++)
{ printf("%03d ",s[k]->n)     ;
if((k-(j-count))%10==0&&k!=j)
printf("\n ")   ;
}
count=1 ;
no++ ;
}
else count++ ;
}
}
```

参考答案二：

```
#include "stdio.h"
#define N 5
struct student
{ int number  ;
int score ;
int rank  ;
int no  ;
}stu[N]  ;
main( )
{ int i, j, k, count, rank, score  ;
```

```
struct student temp；
for( i=1 ； i<=N ； i++ )
{ printf("Enter N.o %d=", i )     ；
scanf("%d%d", &temp.number, &temp.score )     ；
for( j=i-1  ； j>0 ； j-- )
if( stu[j-1].score < temp.score )
stu[j]=stu[j-1]  ；
else break；
stu[j]=temp  ；
}
stu[0].rank=1  ；
count = 1；
k = 0  ；
for( i=0  ； i<N-1  ； i++ )
{ score = stu[i].score；
rank = stu[i].rank   ；
if( stu[i+1].score == stu[i].score )
{ stu[i+1].rank = stu[i].rank    ；
count++ ；
}
else
{ for( j=0  ； j<count ； j++ )
stu[k+j].no = count-j  ；
stu[i+1].rank = stu[i].rank+1    ；
count = 1；
k = i+1  ；
}
if( i==N-2 )
for( j=0  ； j<count ； j++ )
stu[k+j].no = count-j  ；
}
for( i=0  ； i<N  ； i++ )
{ rank = stu[i].rank   ；
count = stu[i].no ；
printf ( "\n%3d (%3d)-%d: ", rank, stu[i].score, count )       ；
for( k=1  ； k<=count ； k++ )
if( (k-1)%3 !=0 )
printf( "%d ", stu[i+k-1].number )     ；
else printf ( "\n %d ", stu[i+k-1].number )     ；
i+=count-1；
}
}
```

【4.103】参考答案：

```
#include "stdio.h"
struct time
```

```
{ int hour ;
int minute ;
int second ;
} ;
main()
{ struct time now ;
printf("Please enter now time(HH,MM,SS)=\n")      ;
scanf("%d,%d,%d",&now.hour,&now.minute,&now.second)      ;
now.second++ ;
if(now.second==60)
{ now.second=0 ;
now.minute++ ;
}
if(now.minute==60)
{ now.minute=0 ;
now.hour++ ;
}
if(now.hour==24)
now.hour=0 ;
printf("\nNow is %02d:%02d:%02d",now.hour,now.minute,now.second)      ;
}
```

【4.104】参考答案：

```
#include <stdio.h>
#define SIZE 3
struct student /*  定义结构   */
{ long num ;
char name[10] ;
int age ;
char address[10] ;
} stu[SIZE], out  ;
void fsave ( )
{ FILE *fp   ;
int i ;
if((fp=fopen("student", "wb"))== NULL) /*       以二进制写方式打开文件       */
{ printf("Cannot open file.\n")   ;   /*  打开文件的出错处理      */
exit(1) ;   /*  出错后返回，停止运行       */
}
for(i=0 ; i<SIZE  ; i++) /*   将学生的信息（结构）以数据块形式写入文件          */
if(fwrite(&stu[i],sizeof(struct student),1,fp) != 1 )
printf("File write error.\n")    ;   /*  写过程中的出错处理      */
fclose(fp) ;   /*  关闭文件   */
}
main()
{ FILE *fp   ;
int i ;
```

```
for(i=0 ；i<SIZE ；i++) /*  从键盘读入学生的信息  (结构 ) */
{ printf("Input student %d:",i+1)  ；
scanf("%ld%s%d%s",
&stu[i].num,stu[i].name,&stu[i].age,stu[i].address)  ；
}
fsave()；  /*  调用函数保存学生信息  */
fp = fopen("student","rb")  ；  /*  以二进制读方式打开数据文件  */
printf(" No. Name Age Address\n")  ；
while(fread(&out,sizeof(out),1,fp)) /*  以读数据块方式读入信息  */
printf ("%8ld %-10s %4d %-10s\n",
out.num,out.name,out.age,out.address)；
fclose(fp)；  /*  关闭文件  */
}
```

【4.105】参考答案：

```
#include <stdio.h>
main( )
{ FILE *fp  ；
char str[100], filename[15]  ；
int i；
if((fp=fopen("test", "w")) == NULL)
{ printf("Cannot open the file.\n")  ；
exit(0)；
}
printf("Input a string:")  ；
gets(str)；  /*  读入一行字符串  */
for(i=0 ；str[i]&&i<100  ；i++) /*  处理该行中的每一个字符  */
{ if(str[i] >= 'a' && str[i] <= 'z') /*  若是小写字母  */
str[i] -= 'a'-'A'  ；  /*  将小写字母转换为大写字母  */
fputc(str[i],fp)  ；  /*  将转换后的字符写入文件  */
}
fclose(fp)；  /*  关闭文件  */
fp=fopen( "test", "r")  ；  /*  以读方式打开文本文件  */
fgets(str,100,fp)；  /*  从文件中读入一行字符串  */
printf("%s\n", str)  ；
fclose(fp)；
}
```

【4.106】参考答案：

```
#include "stdio.h" FILE *fp  ；
main( )
{ int c, d ；
if((fp = fopen("d:\\tc\\test8.c","r")) == NULL)
exit(0)；
while((c=fgetc(fp)) != EOF)
if( c=='/' ) /*  如果是字符注释的起始字符  '/' */
if((d=fgetc(fp)) == '*') /*  则判断下一个字符是否为  '*' */
```

```
in_comment()；  /*  调用函数处理  (删除 )注释  */
else /*  否则原样输出读入的两个字符        */
{ putchar(c) ；
putchar(d) ；
}
else
if( c=='\'' || c=='\"') /*      判断是否是字符  '或" */ echo_quote(c) ；  /*  调用函数处理字符  '或" 包含的字符  */ else putchar(c) ；  } in_comment()
{ int c, d ；
c=fgetc(fp)；
d=fgetc(fp) ；
while( c!='*' || d!='/' )
{ /*  连续的两个字符不是      *  和  /  则继续处理注释      */
c = d ；
d = fgetc(fp) ；
}
}
echo_quote (c)
int c ；  /* c  中存放的是定界符  '或" */
{ int d ；
putchar(c) ；
while(( d=fgetc(fp))!=c) /*    读入下一个字符判断是否是定界符        c */
{ putchar(c) ；  /*  当不是定界符  c 时继续循环    */
if(d=='\\') /*    若出现转义字符  \ */
putchar( fgetc(fp)) ；  /*  则下一个字符不论是何均原样输出        */
}
putchar(d) ；
}
```