

## C 语言课后习题答案 第三版 谭浩强

1.5 请参照本章例题，编写一个 C 程序，输出以下信息：

\*\*\*\*\*

Very Good!

\*\*\*\*\*

解：

```
mian()
{printf("*****");
printf("\n");
printf("Very Good!\n");
printf("\n");
printf("*****");
}
```

1.6 编写一个程序，输入 a、b、c 三个值，输出其中最大值。

解：

```
mian()
{int a,b,c,max;
printf("请输入三个数 a,b,c:\n");
scanf("%d,%d,%d",&a,&b,&c);
max=a;
if(max<b)
max=b;
if(max<c)
max=c;
printf("最大数为: %d",max);
}
```

## 第三章

3.6 写出以下程序运行的结果。

```
main()
{char c1='a',c2='b',c3='c',c4='\101',c5='\116';
printf("a%cb%c\tc%c\tabc\n",c1,c2,c3);
printf("\t\b%c %c",c4,c5);
}
```

解：

```
aa bb bb cc bb bb bb abc
A b N
```

3.7 要将"China"译成密码，译码规律是：用原来字母后面的第 4 个字母代替原来的字母。例如，字母"A"后面第 4 个字母是"E"。"E"代替"A"。因此，"China"应译为"Glmre"。请编一程序，用赋初值的方法使 c1、c2、c3、c4、c5 五个变量的值分别为，'C'、'h'、'i'、'n'、'a'，经过运算，使 c1、c2、c3、c4、c5 分别变为'G'、'l'、'm'、'r'、'e'，并输出。

解：

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



```

main()
{ char c1='C',c2='h',c3='i',c4='n',c5='a';
  c1+=4;
  c2+=4;
  c3+=4;
  c4+=4;
  c5+=4;
  printf("密码是%c%c%c%c%c\n",c1,c2,c3,c4,c5);
}

```

运行结果:

密码是 Glmre

3.9 求下面算术表达式的值。

- (1)  $x+a\%3*(int)(x+y)\%2/4$   
 设  $x=2.5, a=7, y=4.7$
- (2)  $(float)(a+b)/2+(int)x\%(int)y$   
 设  $a=2, b=3, x=3.5, y=2.5$
- (1) 2.5
- (2) 3.5

3.10 写出程序运行的结果。

```

main()
{int i,j,m,n;
 i=8;
 j=10;
 m=++i;
 n=j++;
 printf("%d,%d,%d,%d",i,j,m,n);
}

```

解:

9, 11, 9, 10

3.12 写出下面表达式运算后 a 的值, 设原来  $a=12$ 。设 a 和 n 都已定义为整型变量。

- (1)  $a+=a$     (2)  $a-=2$     (3)  $a*=2+3$     (4)  $a/=a+a$
- (5)  $a\%=(n\%=2)$ , n 的值等于 5
- (6)  $a+=a-=a*=a$

解:

(1) 24    (2) 10    (3) 60    (4) 0    (5) 0    (6) 0

#### 第四章

4. 4 若  $a=3, b=4, c=5, x=1.2, y=2.4, z=-3.6, u=51274, n=128765, c1='a', c2='b'$ 。想得到以下输出格式和结果, 请写出程序 (包括定义变量类型和设计输出)。

```

a=_3_ b=_4_ c=_5_
x=1.200000,y=2.400000,z=-3.600000
x+y=_3.600_ y+z=-1.20_ z+x=-2.40
c1='a' _or_97(ASCII)
c2='b' _or_98(ASCII)
main()

```

```

{int a=3,b=4,c=5;
long int u=51274,n=128765;
float x=1.2,y=2.4,z=3.6;
char c1='a',c2='b';
printf("a=%2d b=%2d c=%2d\n",a,b,c);
printf("x=%f,y=%f,z=%f\n",x,y,z);
printf("x+y=%5.2f y+z=%5.2f z+x=%5.2f\n",x+y,y+z,z+x);
printf("u=%6ld n=%9ld\n",u,n);
printf("%s %s %d%s\n","c1='a'", "or", c1, "(ASCII)");
printf("%s %s %d%s\n","c2='a'", "or", c2, "(ASCII)");
}

```

4. 7 用 scanf 下面的函数输入数据，使 a=3，b=7，x=8.5，y=71.82，c1= 'A'，c2= 'a'，问在键盘上如何输入？

```

main()
{
int a, b; float x, y; char c1c2;
scanf ("a=%d_b=%d", &a, &b);
scanf ("%f%e",&x, &y);
scanf ("%c%c", &c1, &c2);
}
a=3_b=7
_x=8.5_y=71.82
_c1=A_c2=a

```

4. 8 设圆半径 r=1.5，圆柱高 h=3，求圆周长、圆面积、圆球表面积、圆球体积、圆柱体积。用 scanf 输入数据，输出计算结果，输出时要求文字说明，取小数点后两位数字。请编程序。

```

main()
{float r,h,C1,Sa,Sb,Va,Vb;
scanf("%f,%f",&r,&h);
C1=2*3.14*r;
Sa=3.14*r*r;
Sb=4*Sa;
Va=4*3.14*r*r*r/3;
Vb=Sa*h;
printf("C1=%.2f\n",C1);
printf("Sa=%.2f\nSb=%.2f\nVa=%.2f\nVb=%.2f\n",Sa,Sb,Va,Vb);
}

```

4. 9 输入一个华氏温度，要求输出摄氏温度。公式为

$$c=5(F-32)/9$$

输出要求有文字说明，取位 2 小数。

```

main()
{float F,c;
scanf("%f",&F);
c=5*(F-32)/9;
}

```

```
printf("c=%.2f",c);
}
```

4. 10 编程序，用 `getchar` 函数读入两个字符给 `c1`、`c2`，然后分别用函数和函数输出这两个字符。并思考以下问题：（1）变量 `c1`、`c2` 应定义为字符型或整形？抑二者皆可？（2）要求输出 `c1` 和 `c2` 值的 ASCII 码，应如何处理？用 `putchar` 函数还是 `printf` 函数？（3）整形变量与字符变量是否在任何情况下都可以互相代替？如：

```
char c1, c2;
与
int c1, c2;
是否无条件的等价？
#include"stdio.h"
main()
{char c1,c2;
c1=getchar();c2=getchar();
putchar(c1);putchar('\n');putchar(c2);putchar('\n');
}
#include"stdio.h"
main()
{char c1,c2;
c1=getchar();c2=getchar();
printf("c1=%d c2=%d\n",c1,c2);
printf("c1=%c c2=%c\n",c1,c2);
}
```

## 第五章

5.1 什么是算术运算？什么是关系运算？什么是逻辑运算？

解：略。

5.2 C 语言中如何表示“真”和“假”？系统如何判断一个量的“真”和“假”？

解：设有一个逻辑表达式，若其结果为“真”，则以 **1** 表示；若其结果为“假”，则以 **0** 表示。但是判断一个逻辑量的值时，以 **0** 代表“真”，以非 **0** 代表“假”。例如 `3&&5` 的值为“真”，系统给出 `3&&5` 的值为 **1**。

5.3 写出下面各逻辑表达式的值。设 `a=3`，`b=4`，`c=5`。

- (1) `a+b>c&&b==c`
- (2) `a||b+c&&b-c`
- (3) `!(a>b)&&!c||1`
- (4) `!(x=a)&&(y=b)&&0`
- (5) `!(a+b)+c-1&&b+c/2`

解：

- (1) 0
- (2) 1
- (3) 1
- (4) 0
- (5) 1

5.4 有 3 个整数 `a`、`b`、`c`，由键盘输入，输出其中最大的数。

解：

方法一

```
#include <stdio.h>
main()
{ int a,b,c;
printf("请输入 3 个整数: ");
scanf("%d,%d,%d",&a,&b,&c);
if(a<b)
if(b<c) printf("max=%d\n",c);
else printf("max=%d\n",b);
else if(a<c) printf("max=%d\n",c);
else printf("max=%d\n",a);
}
```

方法二：使用条件表达式，可以使程序更加简明、清晰。

程序如下：

```
#include <stdio.h>
main()
{ int a,b,c,temp,max;
printf("请输入 3 个整数: ");
scanf("%d,%d,%d",&a,&b,&c);
temp=(a>b)?a:b;          /* 将 a 和 b 中的大者存入 temp 中 */
max=(temp>c)?temp:c;      /* 将 a 和 b 中的大者与 c 比较，取最大者 */
printf("3 个整数的最大数是%d\n",max);
}
```

5.5 有一函数：

写一程序，输入  $x$  值，输出  $y$  值。

解：

```
#include <stdio.h>
main()
{int x,y;
printf("输入 x: ");
scanf("%d",&x);
if(x<1)                                /* x<1 */
{ y=x;
printf("x=%3d, y=x=%d\n",x,y);
}
else if (x<10)                          /* 1≤x-10 */
{ y=2*x-1;
printf("x=%3d, y=2*x-1=%d\n",x,y);
}
else                                    /* x≥10 */
{ y=3*x-11;
printf("x=%3d, y=3*x-11=%d\n",x,y);
}
}
```

5.6 给一个百分制成绩，要求输出等级 'A'、'B'、'C'、'D'、'E'。90 分以上为 'A'，80~90 分为 'B'，70~79 分为 'C'，60 分以下为 'D'。

解：

程序如下：

```
#include <stdio.h>
main()
{ float score;
  char grade;
  printf("请输入学生成绩: ");
  scanf("%f",&score);
  while(score>100||(score<0)
  { printf("\n 输入有误,请重新输入: ");
    scanf("%f",&score);
  }
  switch((int)(score/10))
  { case 10:
    case 9: grade='A';break;
    case 8: grade='B';break;
    case 7: grade='C';break;
    case 6: grade='D';break;
    case 5:
    case 4:
    case 3:
    case 2:
    case 1:
    case 0: grade='E';
  }
  printf("成绩是%5.1f，相应的等级是%c。\\n",score,grade);
}
```

说明：对输入的数据进行检查，如小于 0 或大于 100，要求重新输入。(int)(score/10)的作用是将 (score/10) 的值进行强制类型转换，得到一个整型值。

5.7 给定一个不多于 5 位的正整数，要求：① 求它是几位数；② 分别打印出每一位数字；③ 按逆序打印出各位数字。例如原数为 321，应输出 123。

解：

```
#include <stdio.h>
main()
{ long int num;
  int indiv,ten,hundred,thousand,ten_thousand,place;
      /*分别代表个位、十位、百位、千位、万位和位数*/
  printf("请输入一个整数 (0~99999): ");
  scanf("%ld",&num);
  if (num>9999) place=5;
  else if(num>999) place=4;
  else if(num>99) place=3;
```

```

else if(num>9) place=2;
else place=1;
printf("place =%d\n", place);
ten_thousand=num/10000;
thousand=num/1000%10;
hundred=num/100%10;
ten=num%100/10;
indiv=num%10;
switch(place)
{
    case 5:
printf("%d,%d,%d,%d,%d",ten_thousand,thousand,hundred,ten,indiv);
    printf("\n 反序数字为: ");

printf("%d%d%d%d%d\n",indiv,ten,hundred,thousand,ten_thousand);
    break;
    case 4: printf("%d,%d,%d,%d",thousand,hundred,ten,indiv);
    printf("\n 反序数字为: ");
    printf("%d%d%d%d\n",indiv,ten,hundred,thousand);
    break;
    case 3: printf("%d,%d,%d",hundred,ten,indiv);
    printf("\n 反序数字为: ");
    printf("%d%d%d\n",indiv,ten,hundred);
    break;
    case 2: printf("%d,%d",ten,indiv);
    printf("\n 反序数字为: ");
    printf("%d%d\n",indiv,ten);
    break;
    case 1: printf("%d",indiv);
    printf("\n 反序数字为: ");
    printf("%d\n",indiv);
    break;
}
}

```

5.8 企业发放的奖金根据利润提成。利润  $I$  低于或等于 10 万元时，奖金可提成 10%；利润高于 10 万元，低于 20 万元 ( $100000 < I \leq 200000$ ) 时，其中 10 万元按 10% 提成，高于 10 万元的部分，可提成 7.5%； $200000 < I \leq 400000$  时，其中 20 万元仍按上述办法提成（下同），高于 20 万元的部分按 5% 提成； $400000 < I \leq 600000$  时，高于 40 万元的部分按 3% 提成； $600000 < I \leq 1000000$  时，高于 60 万的部分按 1.5% 提成； $I > 1000000$  时，超过 100 万元的部分按 1% 提成。从键盘输入当月利润  $I$ ，求应发放奖金总数。要求：(1) 用 if 语句编程序；(2) 用 switch 语句编程序。

解：计算利润时，要特别注意不同利润的不同提成比例。例如，利润为 15 万元，其中有 10 万元按 10% 的比例提成，另外 5 万元则按 7.5% 提成。

(1) 用 if 语句编程序。

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

```

main()
{ long i;
float bonus,bon1,bon2,bon4,bon6,bon10;
bon1=100000*0.1;          /*利润为 10 万元时的奖金*/
bon2=bon1+100000*0.075;   /*利润为 20 万元时的奖金*/
bon4=bon2+200000*0.05;    /*利润为 40 万元时的奖金*/
bon6=bon4+200000*0.03;    /*利润为 60 万元时的奖金*/
bon10=bon6+400000*0.015;  /*利润为 100 万元时的奖金*/
printf("请输入利润 i: ");
scanf("%ld",&i);
if(i<=100000)
bonus=i*0.1;              /*利润在 10 万元以内按 0.1 提成奖金*/
else if(i<=200000)
bonus=bon1+(i-100000)*0.075; /*利润在 10 万至 20 万元时的奖金*/
else if(i<=400000)
bonus=bon2+(i-200000)*0.05;  /*利润在 20 万至 40 万元时的奖金*/
else if(i<=600000)
bonus=bon4+(i-400000)*0.03;  /*利润在 40 万至 60 万元时的奖金*/
else if(i<=1000000)
bonus=bon6+(i-600000)*0.015; /*利润在 60 万至 100 万元时的奖金*/
else
bonus=bon10+(i-1000000)*0.01; /*利润在 100 万元以上时的奖金*/
printf(" 奖金是%10.2f\n",bonus);
}

```

此题的关键在于正确写出每一区间的奖金计算公式。例如利润在 10 万元至 20 万时，奖金应由两部分组成：①利润为 10 万元时应得的奖金。即  $100000 \times 0.1$ ；②10 万元以上部分应得的奖金。即  $(\text{num}-100000) \times 0.075$ 。同理，20 万~40 万这个区间的奖金也应由两部分组成：①利润为 20 万元时应得的奖金，即  $100000 \times 0.1 + 10 \text{ 万} \times 0.075$ ；②20 万元以上部分应得的奖金，即  $(\text{num}-200000) \times 0.05$ 。程序中先把 10 万、20 万、40 万、60 万、100 万各关键点的奖金计算出来，即 bon1、bon2、bon4、bon6、hon10；然后再加上各区间附加部分的奖金。

(2) 用 switch 语句编程序。

输入利润 i，确定相应的提成等级 branch

根据 branch 确定奖金值

```

0    奖金=i*0.1
1    奖金=bon1+(i-105)*0.075
2    奖金=bon2+(i-2*105)*0.05
3
4    奖金=bon4+(i-4*105)*0.03
5
6    奖金=bon6+(i-6*105)*0.015
7
8
9

```



10 奖金=bon10+(i-106)\*0.01

输出奖金

```
#include <stdio.h>
main()
{ long i;
float bonus, bon1, bon2, bon4, bon6, bon10;
int c;
bon1=100000*0.1;
bon2=bon1+100000*0.075;
bon4=bon2+200000*0.05;
bon6=bon4+200000*0.03;
bon10=bon6+400000*0.015;
printf("请输入利润 i: ");
scanf("%ld",&i);
c=i/100000;
if(c>10) c=10;
switch(c)
{ case 0: bonus=1*0.1;break;
case 1: bonus=bon1+(i-100000)*0.075;break;
case 2 :
case 3: bonus=bon2+(i-200000)*0.05; break;
case 4:
case 5: bonus=bon4+(i-400000)*0.03;break;
case 6:
case 7:
case 8:
case 9: bonus=bon6+(i-600000)*0.015;break;
case 10: bonus=bon10+(i-1000000)*0.01;
}
printf("奖金是%10.2f",bonus);
}
```

5.9 输入 4 个整数，要求按由大到小的顺序输出。

解：此题采用依次比较的方法排出其大小顺序。在学习了循环和数组以后，可以有更多的排序方法。

```
#include <stdio.h>
main()
{ int t,a,b,c,d;
printf("请输入 4 个整数: ");
scanf("%d,%d,%d,%d",&a,&b,&c,&d);
printf("\n a=%d,b=%d,c=%d,d=%d\n",a,b,c,d);
if(a>b) {t=a; a=b; b=t;}
if(a>c) {t=a; a=c; c=t;}
if(a>d) {t=a; a=d; d=t;}
if(b>c) {t=b; b=c; c=t;}
```

```

if(b>d) {t=b; b=d; d=t;}
if(c>d) {t=c; c=d; d=t;}
printf("排序结果如下: \n");
printf("%d, %d, %d, %d\n",a,b,c,d);
}

```

5.10 有 4 个圆塔，圆心分别为(2, 2)、(-2, 2)、(2, -2)、(-2, -2)，圆半径为 1。这 4 个塔的高度分别为 10m。塔以外无建筑物。今输入任一点的坐标，求该点的建筑高度(塔外的高度为零)。

程序如下：

```

#include <stdio.h>
main()
{ int h=10;
float x1=2,y1=2,x2=-2,y2=2,x3=-2,y3=-2,x4=-2,y4=-2,x,y,d1,d2,d3,d4;
printf("请输入一个点(x,y): ");
scanf("%f,%f",&x,&y);
d1=(x-x1)*(x-x1)+(y-y1)*(y-y1);          /*求该点到各中心点的距离*/
d2=(x-x2)*(x-x2)+(y-y2)*(y-y2);
d3=(x-x3)*(x-x3)+(y-y3)*(y-y3);
d4=(x-x4)*(x-x4)+(y-y4)*(y-y4);
if(d1>1&&d2>1&&d3>1&&d4>1) h=0;          /*判断该点是否在塔外*/
printf("该点高度为%d\n",h);
}

```

## 第六章 循环控制

6.1 输入两个正整数 m 和 n，求其最大公约数和最小公倍数。

```

main()
{long m,n,i=1,j,s;
scanf("%ld,%ld",&m,&n);
for(;j<=m&&i<=n;i++)
{if(m%i==0&&n%i==0) s=i;}
if(m>=n) j=m;
else j=n;
for(;!(j%m==0&&j%n==0);j++);
printf("s=%ld,j=%ld\n",s,j);
}

```

6.2 输入一行字符，分别统计出其中英文字母、空格、数字和其他字符的个数。

```

#include"stdio.h"
main()
{char c;int i=0,j=0,k=0,l=0;
while((c=getchar())!='\n')
{if(c>=65&&c<=90||c>=97&&c<=122) i++;
else if(c>=48&&c<=57) j++;
else if(c==32) k++;
else l++;}
printf("i=%d,j=%d,k=%d,l=%d\n",i,j,k,l);
}

```

```
}
```

6.3 求  $S_n = a + aa + aaa + \cdots + aa \cdots aaa$  (有  $n$  个  $a$ ) 之值, 其中  $a$  是一个数字。例如:  $2 + 22 + 222 + 2222 + 22222$  ( $n=5$ ),  $n$  由键盘输入。

```
#include "math.h"
main()
{int n,sum=0,i=1,s=2;
scanf("%d",&n);
while(i<=n)
{sum=sum+s;s=s+2*pow(10,i);
i++;}
printf("sum=%d\n",sum);
}
```

6.4 求  $1! + 2! + 3! + 4! + 5! + \cdots + 20!$

```
main()
{int n,i=1;long sum=0,s=1;
scanf("%d",&n);
while(i<=n) {s=s*i;sum=sum+s;i++;}
printf("sum=%ld\n",sum);
}
```

6.5 求

```
main()
{double i=1,j=1,k=1,s1=0,s2=0,s3=0,sum;
for(;i<=100;i++) s1=s1+i;
for(;j<=50;j++) s2=s2+j*j;
for(;k<=10;k++) s3=s3+1/k;
sum=s1+s2+s3;
printf("sum=%f\n",sum);
}
```

6.6 打印出所有"水仙花数", 所谓"水仙花数"是指一个三位数, 其各位数字立方和等于该本身。例如: 153 是一个水仙花数, 因为  $153 = 1^3 + 5^3 + 3^3$ 。

```
#include "math.h"
main()
{int x=100,a,b,c;
while(x>=100&& x<1000) {a=0.01*x;b=10*(0.01*x-a);c=x-100*a-10*b;
if(x==(pow(a,3)+pow(b,3)+pow(c,3))) printf("%5d",x);x++;}
}
```

6.7 一个数如果恰好等于它的因子之和, 这个数就称为"完数"。例如, 6 的因子为 1、2、3, 而  $6 = 1 + 2 + 3$ , 因此 6 是"完数"。编程序找出 1000 之内的所有完数, 并按下面格式输出其因子:

6 its factors are 1、2、3

```
main()
{int m,i,j,s;
for(m=6;m<10000;m++)
{s=1;
```

```

for(i=2;i<m;i++)
if(m%i==0) s=s+i;
if(m-s==0)
{printf("%5d its fastors are 1 ",m);for(j=2;j<m;j++) if(m%j==0)
printf("%d ",j);printf("\n");}
}
}

```

或

```

main()
{int m,i,j,s;
for(m=6;m<1000;m++)
{s=m-1;
for(i=2;i<m;i++)
if(m%i==0) s=s-i;
if(s==0)
{printf("%5d its fastors are 1 ",m);for(j=2;j<m;j++) if(m%j==0)
printf("%d ",j);printf("\n");}
}
}

```

6.8 有一分数序列:

求出这个数列的前 20 项之和。

```

main()
{int i=1,n;double t,x=1,y=2,s,sum=0;
scanf("%ld",&n);
while(i<=n) {s=y/x;sum=sum+s;t=y;y=y+x;x=t;i++;}
printf("%f\n",sum);
}

```

6.9 一球从 100 米高度自由下落，每次落地后返回原高度的一半，再落下。求它在第 10 次落地时共经过多少米？第 10 次反弹多高？

```

main()
{int i,n;double h=100,s=100;
scanf("%d",&n);
for(i=1;i<=n;i++)
{h*=0.5;if(i==1) continue;s=2*h+s;}
printf("h=%f,s=%f\n",h,s);
}

```

6.10 猴子吃桃问题。猴子第一天摘下若干个桃子，当即吃了一半，还不过瘾，又多吃了一个。第二天早上又将剩下的桃子吃掉一半，又多吃一个。以后每天早上都吃了前一天剩下的一半零一个。到第 10 天早上想再吃时，见只剩下一个桃子了。求第一天共摘多少桃子。

```

main()
{int i=1,sum=0;
for(;i<=10;sum=2*sum+1,i++);
printf("sum=%d\n",sum);
}

```

6.11 用迭代法求  $\sqrt{a}$ 。求平方根的迭代公式为：  
要求前后两次求出的得差的绝对值少于 0.00001。

```
#include "math.h"
main()
{float x0,x1,a;
scanf("%f",&a);
x1=a/2;
do
{x0=x1;x1=(x0+a/x0)/2;}
while(fabs(x0-x1)>=0.00001);
printf("%.3f\n",x1);
}
```

6.12 用牛顿迭代法求方程在 1.5 附近的根。

```
main()
{double x,y;x=1.5;
do{y=2*x*x*x-4*x*x+3*x-6;
x=x-y/(6*x*x-8*x+3);}
while(y!=0);
printf("x=%.3f\n",x);
}
```

6.13 用二分法求方程在(-10, 10)之间的根

```
main()
{double x1,x2,y1,y2;x1=-10;x2=10;
do{y1=2*x1*x1*x1-4*x1*x1+3*x1-6;
x1=x1-y1/(6*x1*x1-8*x1+3);}
while(y1!=0);
do
{y2=2*x2*x2*x2-4*x2*x2+3*x2-6;
x2=x2-y2/(6*x2*x2-8*x2+3);}
while(y2!=0);
printf("x1=%.3f,x2=%.3f\n",x1,x2);
}
```

6.14 打印以下图案

```
  *
 * * *
* * * * *
* * * * * *
 * * * * *
  * * *
   *
```

```
#include "math.h"
main()
{int i,j,k;
for(i=0;i<=3;i++)
```

```

{for(j=0;j<=2-i;j++)
    printf(" ");
    for(k=0;k<=2*i;k++)
        printf("*");
    printf("\n");
}
for(i=0;i<=2;i++)
{for(j=0;j<=i;j++)
    printf(" ");
    for(k=0;k<=4-2*i;k++)
        printf("*");
    printf("\n");
}
}

```

## 第七章 数组

### 7.1 用筛法求之内的素数。

```

main()
{ int i,j,a[100];
for(i=2;i<100;i++)
{ a[i]=i;
for(j=2;j<=i;j++)
{ if(j<i)
if(a[i]%j==0)
break;
if(a[i]-j==0)
printf("%5d",a[i]);
}
}
printf("\n");
}
或
#include"math.h"
main()
{static int i,j,k,a[98];
for(i=2;i<100;i++)
{ a[i]=i;k=sqrt(i);
for(j=2;j<=a[i];j++)
if(j<k) if(a[i]%j==0)
break;
if(j>=k+1)
printf("%5d",a[i]);
}
printf("\n");
}

```

7.2 用选择法对 10 个整数从小到大排序。

```
main()
{ int i,j,a[10],t;
  for(i=0;i<10;i++)
  scanf("%d",&a[i]);
  for(j=1;j<10;j++)
  for(i=0;i<=9-j;i++)
  if(a[i]>a[i+1])
  {t=a[i+1];a[i+1]=a[i];a[i]=t;}
  for(i=0;i<10;i++)
  printf("%5d",a[i]);
}
```

或

```
main()
{static int a[10],i,j,k,t;
  for(i=1;i<11;i++)
  scanf("%d",&a[i]);
  for(j=1;j<10;j++)
  for(i=1;i<=10-j;j++)
  if (a[i]>a[i+1])
  {t=a[i+1];a[i+1]=a[i];a[i]=t;}
  for(i=1;i<11;i++)
  printf("%d",a[i]);
  printf("\n");
}
```

7.3 求一个 3×3 矩阵对角线元素之和。

```
main()
{int i=0,j=0,a[3][3],s1,s2;
  for(i=0;i<3;i++)
  for(j=0;j<3;j++)
  scanf("%d",&a[i][j]);
  s1=a[0][0]+a[1][1]+a[2][2];
  s2=a[0][2]+a[1][1]+a[2][0];
  printf("s1=%d,s2=%d\n",s1,s2);
}
```

或

```
main()
{
  static int i,j,s1,s2,a[3][3];
  for(i=1;i<=3;i++)
  for(j=1;j<=3;j++)
  scanf("%d",&a[i][j]);
  s1=a[1][1]+a[2][2]+a[3][3];
  s2=a[1][3]+a[2][2]+a[3][1];
}
```

```
printf("%d,%d\n",s1,s2);
}
```

7.4 已有一个已排好的数组今输入一个数要求按原来排序的规律将它插入数组中。

```
main()
{ static int a[10]={1,7,8,17,23,24,59,62,101};int i,j,t;
scanf("%d",&a[9]);
for(i=9;i>0;i--)
if(a[i]<a[i-1])
{t=a[i-1];a[i-1]=a[i];a[i]=t;}
for(i=0;i<10;i++)
printf("%5d",a[i]);printf("\n");
}
```

或

```
main()
{
static int a[5]={1,4,5,6,7};
int i,t,b;
scanf("%d",&b);
for(i=0;i<5;i++)
{if(b<=a[i])
{t=a[i];a[i]=b;b=t;}
printf("%d ",a[i]);}
printf("%d",b);
}
```

7.5 将一个数组的值按逆序重新存放，例如，原来顺序为：8，6，5，4，1。要求改为：1，4，5，6，8。

```
main()
{ int i,b[10];
for(i=0;i<10;i++)
scanf("%d",&b[i]);
for(i=9;i>=0;i--)
printf("%5d",b[i]);
printf("\n");}
```

7.6 打印出以下杨辉三角形（要求打印出 10 行）。

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
:
:
```

```
main()
{ static int m,n,k,b[15][15];
```



```

b[0][1]=1;
for(m=1;m<15;m++)
{for(n=1;n<=m;n++)
{ b[m][n]=b[m-1][n-1]+b[m-1][n];
printf("%-5d",b[m][n]);}printf("\n");
}
}
}

```

或

```

main()
{ int i,j,n,k,a[10][10];
static a[][1]={ {1},{1},{1},{1},{1},{1},{1},{1},{1},{1}};
a[1][1]=1;
for(k=2,k<11;k++)
for(i=2;i<=k;i++)
for(j=2;j<=i;j++)
a[i][j]=a[i-1][j-1]+a[i-1][j];
for(k=1;k<11;k++)
for(i=1;i<=k;i++)
for(j=1;j<=i;j++)
printf("%d",a[i][j]);
}

```

7.7 打印“魔方阵”，所谓魔方阵是指这样的方阵，它的每一行、每一列和对角线之和均相等。例如，三阶魔方阵为

```

8   1   6
3   5   7
4   9   2

```

要求打印出由 1~n<sup>2</sup> 的自然数构成的魔方阵。

解：

```

#include <stdio.h>
main()
{ int a[16][16],i,i,k,p,m,n;
p=1;
while(p==1) /*要求阶数为 1~15 的商数*/
{ printf("Enter n(n=1~15): ");
scanf("%d",&n);
if((n!=0)&&(n<=15)&&(n%2!=0)) p=0;
}
for(i=1;i<=n;i++) /*初始化*/
for(j=1;j<=n;j++) a[i][j]=0;
j=n/2+1; /*建立魔方阵*/
a[1][j]=1;
for(k=2;k<=n*n;k++)
{ i=i-1;

```

```

        j=j+1;
        if((i<1)&&(j>n))
        { i=i+2;
          j=j-1;
        }
        else
        { if(i<1) i=n;
          if(j>n) j=1;
        }
        if(a[i][j]==0) a[i][j]=k;
        else
        { i=i+2;
          j=j-1;
          a[i][j]=k;
        }
    }
    for(i=1;i<=n;i++) /*输出魔方阵*/
    { for(j=1;j<=n;j++)
      printf("%4d",a[i][j]);
      printf("\n");
    }
}

```

7.8 找出一个二维数组中的鞍点，即该位置上的元素在该行上最大，在该列上最小，也可能没有鞍点。

```

main()
{int a[5][5],b[5],c[5],d[5][5],k=0,l=0;int i,j;
for(i=0;i<5;i++)
for(j=0;j<5;j++)
scanf("%d",&d[i][j]);
for(i=0;i<5;i++)
for(j=0;j<5;j++,a[i][j]=d[i][j]);
for(i=0,k=0;i<5;i++,k++)
for(j=0;j<4;j++)
{if(a[i][j]>=a[i][j+1])
b[k]=a[i][j+1]=a[i][j];
else
b[k]=a[i][j+1];
}
for(j=0,l=0;j<5;j++,l++)
for(i=0;i<4;i++)
{if(a[i][j]<=a[i+1][j])
c[l]=a[i+1][j]=a[i][j];
else
c[l]=a[i+1][j];
}
}

```

```

}
for(i=0,k=0;i<5;i++,k++)
for(j=0,l=0;j<5;j++,l++)
if(d[i][j]-b[k]==0)
{if(d[i][j]-c[l]==0)
printf("d[%d][%d]=%d\n",i,j,d[i][j]);
else
printf("d[%d][%d]=%d isnot andi\n",i,j,d[i][j]);
}
}
}

```

7.9 有个 15 数按由小到大顺序存放在一个数组中，输入一个数，要求用折半查找法找出该数组中第几个元素的值。如果该数不在数组中，则打印出"无此数"

```

#include"math.h"
main()
{static int i,j,m,a[15]={1,4,9,13,21,34,55,89,144,233,377,570,671,703,812};
scanf("%d",&m);
for(j=0;j<15;j++)
printf("%4d",a[j]);
printf("\n");
i=7;
while(fabs(i-7)<8)
{if(m<a[7])
{if(a[i]-m==0)
{printf("it is at (%d)\n",i+1);break;}i--;}
else if(m>a[7])
{if(a[i]-m==0)
{printf("it is at (%d)\n",i+1);break;}i++;}
else
printf("8\n");
}
if(fabs(i-7)-8==0)
printf("There is not\n");
}

```

7.10 有一篇文章，共有 3 行文字，每行有个 80 字符。要求分别统计出其中英文大写字母、小写字母、空格以及其它字符的个数。

```

main()
{int i,j=0,k=0,l=0,m=0,n=0;char str0[301],str1[100],str2[100],str3[100];
gets(str1);gets(str2);gets(str3);
strcat(str0,str1);strcat(str0,str2);strcat(str0,str3);
for(i=0;str0[i]!='\0';i++)
{if(str0[i]>=65&&str0[i]<=90) j++;
else if(str0[i]>=97&&str0[i]<=122) k++;
else if(str0[i]>=48&&str0[i]<=57) l++;
else if(str0[i]==32) m++;
}
}

```

```

else n++;}
printf("Daxie Xiaoxie Shuzi Kongge Qita\n");
printf("%5d %7d %5d %6d %4d\n",j,k,l,m,n);
}

```

7.11 打印以下图案

```

* * * * *
* * * * *
* * * * *
*  * * * *
* * * * *

```

```

main()
{int i,j,k;char a[5][5];
for(i=0;i<5;i++)
{for(j=0;j<5;j++)
{a[i][j]='*';printf("%c",a[i][j]);}
printf("\n");
for(k=1;k<=i+1;k++)
printf("\40");}
printf("\n");
}

```

7.12 有一行电文译文下面规律译成密码:

```

A->Z  a->z
B->Y  b->y
C->X  c->x
...

```

即第一个字母变成第 26 个字母，第 i 个字母变成第 (26-i+1) 个字母。非字母字符不变，要求编程序将密码回原文，并打印出密码和原文。

```

main()
{ int i;char str1[100],str2[100];
gets(str1);
for(i=0;str1[i]!='\0';i++)
if(str1[i]>=65&&str1[i]<=90)
str2[i]=155-str1[i];
else if(str1[i]>=97&&str1[i]<=122)
str2[i]=219-str1[i];
else
str2[i]=str1[i];
printf("%s\n%s\n",str1,str2);
}

```

7.13 编一程序，将两个字符串连接起来，不要 strcat 函数。

```

main()
{ int i,j;char str1[100],str2[100],str3[201];
gets(str1);
gets(str2);

```

```

for(i=0;str1[i]!='\0';i++)
str3[i]=str1[i];
for(j=0;str2[j]!='\0';j++)
str3[j+i]=str2[j];
printf("%s\n%s\n%s\n",str1,str2,str3);
}

```

7.14 编一个程序，将两个字符串 S1 和 S2 比较，如果 S1>S2，输出一个正数；S1=S2，输出 0；S1<S2，输出一个负数。不要用 strcpy 函数。两个字符串用 gets 函数读入。输出的正数或负数的绝对值应是相比较的两个字符串相对应字符的 ASCII 码的差值。例如，'A' 与 'C' 相比，由于 'A' < 'C'，应输出负数，由于 'A' 与 'C' 的码差值为 2，因此应输出"-2"。同理："And"和"Aid"比较，根据第 2 个字符比较结果，'n' 比 'i' 大 5，因此应输出"5"。

```

#include <stdio.h>
#include <string.h>
main()
{ int i,resu;
char s1[100],s2[100];
printf("\n input string1: ");
gets(s1);
printf("\n Input string2: ");
gets(s2);
i=0;
while(s1[i]==s2[i]&& s1[i]!='\0') i++;
if(s1[i]=='\0'&& s2[i]=='\0') resu=0;
else resu=s1[i]-s2[i];
printf("\n result: %d\n",resu);
}

```

7.15 编写一个程序,将字符数组 s2 中的全部字符拷贝到字符数组 s1 中,不用 strcpy 函数。拷贝时,' \0' 也要拷贝过去,' \0' 后面的字符不拷贝。

解:

```

#include "stdio.h"
main()
{ char s1[80],s2[80];
int i;
printf("Input s2: ");
scanf("%s",s2);
for(i=0;i<strlen(s2);i++)
s1[i]=s2[i];
printf("s1: %s\n",s1);
}

```

## 第八章 函数

1.1 写两个函数，分别求两个整数的最大公约数和最小公倍数，用主函数调用这两个函数，并输出结果两个整数由键盘输入。

maxyueshu(m,n)

```

int m,n;
{ int i=1,t;
for(;i<=m&&i<=n;i++)
{if(m%i==0&&n%i==0)
t=i;
}
return(t);
}
minbeishu(m,n)
int m,n;
{int j;
if(m>=n) j=m;
else j=n;
for(;(j%m==0&&j%n==0);j++);
return j;
}
main()
{int a,b,max,min;
printf("enter two number is: ");
scanf("%d,%d",&a,&b);
max=maxyueshu(a,b);
min=minbeishu(a,b);
printf("max=%d,min=%d\n",max,min);
}

```

8.2 求方程 的根，用三个函数分别求当  $b^2-4ac$  大于 0、等于 0、和小于 0 时的根，并输出结果。从主函数输入 a、b、c 的值。

```

#include"math.h"
float yishigen(m,n,k)
float m,n,k;
{float x1,x2;
x1=(-n+sqrt(k))/(2*m);
x2=(-n-sqrt(k))/(2*m);
printf("two shigen is x1=%.3f and x2=%.3f\n",x1,x2);
}
float denggen(m,n)
float m,n;
{float x;
x=-n/(2*m);
printf("denggen is x=%.3f\n",x);
}
float xugen(m,n,k)
float m,n,k;
{float x,y;
x=-n/(2*m);

```

```

y=sqrt(-k)/(2*m);
printf("two xugen is x1=%.3f+%.3fi and x2=%.3f-%.3fi\n",x,y,x,y);
}
main()
{float a,b,c,q;
printf("input a b c is ");
scanf("%f,%f,%f",&a,&b,&c);
printf("\n");
q=b*b-4*a*c;
if(q>0) yishigen(a,b,q);
else if(q==0) denggen(a,b);
else xugen(a,b,q);
}

```

8.2 写一个判断素数的函数，在主函数输入一个整数，输出是否是素数的消息。

```

psushu(m)
int m;
{int i=2,t;
for(;i<=m;i++)
if(m%i==0&&i<m) break;
if(m-i==0) t=1;
else t=0;
return m;
}
main()
{int a,s;
printf("enter sushu is \n");
scanf("%d",&a);
s=psushu(a);
if(s==1) printf("a is sushu\n");
else printf("s is not sushu\n");
}

```

8.4 写一个函数，使给定的一个二维数组（ $3 \times 3$ ）转置，即行列互换。

```

int zhuangzhi(b)
int b[3][3];
{int i,j,t;
for(i=0;i<3;i++)
for(j=0;j>=i&&j<3-i;j++)
{t=b[i][j];b[i][j]=b[j][i];b[j][i]=t;}
}
main()
{int a[3][3];int i,j;
for(i=0;i<3;i++)
for(j=0;j<3;j++)
scanf("%d",&a[i][j]);
}

```

```

for(i=0;i<3;i++)
{for(j=0;j<3;j++)
printf(" %d",a[i][j]);
printf("\n");}
zhuangzhi(a);
for(i=0;i<3;i++)
{for(j=0;j<3;j++)
printf(" %d",a[i][j]);
printf("\n");}
}

```

8.5 写一函数，使输入的一个字符串按反序存放，在主函数中输入输出字符串。

```

main()
{char str0[100];
gets(str0);
fanxu(str0);
puts(str0);
}
fanxu(str1)
char str1[100];
{int i,t,j;
char str2[100];strcpy(str2,str1);
t=strlen(str1);
for(i=0,j=t-1;j>-1;i++,j--)
str1[i]=str2[j];
}

```

8.6 写一函数，将两个字符串连接。

```

lianjie(a,b)
char a[100],b[100];
{strcat(a,b);
}
main()
{char str1[100],str2[100];
gets(str1);gets(str2);
lianjie(str1,str2);
puts(str1);
}

```

8.7 写一函数，将两个字符串中的元音字母复制到另一个字符串，然后输出。

```

fuzhi(a,b)
char a[100],b[100];
{int i,j=0;
for(i=0;a[i]!='\0';i++)
if(a[i]==97||a[i]==101||a[i]==105||a[i]==111||a[i]==117||a[i]==65||
a[i]==69||a[i]==73||a[i]==85) {b[j]=a[i];j++;}
}

```



```

main()
{char str1[100],str2[100];
gets(str1);
fuzhi(str1,str2);
puts(str2);
}

```

8.8 写一函数，输入一个四位数字，要求输出这四个数字字符，但每两个数字间空格。如输入 1990，应输出"1\_9\_9\_0"。

```

char f(b)
char b[4];
{int i=0;
for(;i<4;i++)
{printf(" ");
printf("%c",b[i]);}
printf("\n");
}
main()
{int a,u,v,w,t;char c[4];
scanf("%4d",&a);
u=a*0.001;v=0.01*(a-1000*u);w=(a-1000*u-100*v)*0.1;t=a-1000*u-100*v-10*w;
c[0]=u+48;
c[1]=v+48;
c[2]=w+48;
c[3]=t+48;
f(c);
}

```

8.9 编写一函数，由实参传来一个字符串，统计此字符串中字母、数字、空格和其它字符的个数，在主函数中输入字符串以及输出上述结果。

```

char tongji(str0,b)
char str0[100];
int b[4];
{int i;
for(i=0;str0[i]!='\0';i++)
{if(str0[i]>=65&&str0[i]<=90||str0[i]>=97&&str0[i]<=122) b[0]++;
else if(str0[i]>=48&&str0[i]<=57) b[1]++;
else if(str0[i]==32) b[2]++;
else b[3]++;}
}
main()
{char str1[100];static int i,a[4];
gets(str1);
tongji(str1,a);
printf("zimu Shuzi Kongge Qita\n");
}

```

```

for(i=0;i<4;i++)
printf("%-8d ",a[i]);printf("\n");
}

```

8.10 写一函数，输入一行字符，将此字符串中最长的单词输出。

```

cechang(str1,word0)
char str1[100],word0[15];
{int i=0,j=0,t=0;
static char word1[15];
for(;str1[i]!='\0';i++)
{if(!(str1[i]>=97&&str1[i]<=122||str1[i]>=65&&str1[i]<=90))
{t=j;j=0;continue;}
word1[j]=str1[i];j++;
if(j>=t) strcpy(word0,word1);}
}
main()
{char str0[100],longword[15];
gets(str0);
cechang(str0,longword);
puts(longword);
}

```

8.11 写一函数用起泡法对输入的几个字符按由小到大的顺序排列。

```

int paixu(x)
int x[];
{int i,j,t;
for(j=1;j<10;j++)
for(i=0;i<=9-j;i++)
if(x[i]>x[i+1]) {t=x[i+1];x[i+1]=x[i];x[i]=t;}
}
main()
{int y[10];int i;
for(i=0;i<10;i++)
scanf("%d",&y[i]);
paixu(y);
for(i=0;i<10;i++)
printf("%5d",y[i]);
printf("\n");
}

```

8.12 用牛顿迭代法求根。方程为：  $3x^3 + 2tx + u = 0$ ，系数  $a$ ， $b$ ， $c$ ， $d$  由主函数输入。求  $X$  在 1 附近的一个实根。求出后由主函数输出。

```

double qigen(s,t,u,v)
int s,t,u,v;
{double x,y;x=1;
do{y=s*x*x*x+t*x*x+u*x+v;
x=x-y/(3*s*x*x+2*t*x+u);}
}

```

```

while(y!=0);
return x;
}
main()
{int a,b,c,d;double x;
scanf("%d,%d,%d,%d",&a,&b,&c,&d);
x=qigen(a,b,c,d);
printf("x=%.3f\n",x);
}

```

8.13 用递归方法求  $n$  阶勒让德多项式的值递归公式为

```

float p(x0,n)
int n;float x0;
{float y;
if(n==0||n==1) if(n==1) y=x0;else y=1;
else y=((2*n-1)*x0*p(x0,n-1)-(n-1)*p(x0,n-2))/n;
return(y);
}
main()
{float x,y0;int a,i;
scanf("%f,%d",&x,&a);
y0=p(x,a);
printf("y0=%.3f\n",y0);
}

```

8.14 输入 10 个学生 5 门课的成绩，分别用函数求：①每个学生平均分；②每门课的平均分；③找出最高分所对应的学生和课程；④求平均分方差： $\delta = [SXi^2]/n - (SXi/n)^2$ ，为一学生的平均分

```

float x1[10],x2[5];
float pp(),cc(),find(),xx();
main()
{char name[10][20],class[5][20];float score[10][5],o,k=0,max[5];int a[5],i,j;
for(i=0;i<10;i++)
gets(name[i]);
for(j=0;j<5;j++) gets(class[j]);
for(i=0;i<10;i++)
for(j=0;j<5;j++)
scanf("%f",&score[i][j]);
pp(score);
cc(score);
find(score,max,a);
o=xx(k);
for(i=0;i<10;i++)
{puts(name[i]);
printf("%.3f\n",x1[i]);}
for(j=0;j<5;j++)

```

```

{puts(class[j]);printf("%.3f\n",x2[j]);}
for(j=0;j<5;j++) {printf("%.3f\n",max[j]);
puts(name[a[j]]);
puts(class[j]);}
printf("o=%.3f\n",o);
}
float pp(f)
float f[10][5];
{float sum=0;int i,j;
for(i=0,sum=0;i<10;i++)
{for(j=0;j<5;j++)
sum=sum+f[i][j];
x1[i]=sum/5;}
}
float cc(y)
float y[10][5];
{float sum=0;int i,j;
for(j=0;j<5;j++)
{for(i=0;i<10;i++)
sum=sum+y[i][j];
x1[j]=sum/10;}
}
float find(z,s,t)
float z[10][5],s[5];int t[5];
{int i,j;
for(j=0,s[j]=z[0][j];j<5;j++)
for(i=0;i<10;i++)
if(s[j]<z[i][j]) {s[j]=z[i][j];t[j]=i;}
}
float xx(q)
float q;
{float f=0,e=0;int i;
for(i=0;i<10;i++)
{e=x1[i]*x1[i]+e;
f=f+x1[i];}
q=e/10-(f/10)*(f/10);
return(q);
}

```

**8.15** 写几个函数：①输个职工的姓名和职工号；②按职工号由小到大顺序排序，姓名顺序也随之调整；③要求输入一个职工号，用折半法找出该职工的姓名，从主函数输入要查找的职工号，输出该职工姓名。

```

#define N 10
find(a,b)
int a[],b[];

```

```

{int i,j,s,t,c[N][2];
for(i=0;i<N;i++)
{c[i][1]=a[i];c[i][1]=i;}
for(i=0;i<N;i++)
for(j=0;j<N-i-1;j++)
if(c[i][0]>c[i+1][0])
{t=c[i][0];c[i][0]=c[i+1][0];c[i+1][0]=t;
s=c[i][1];c[i][1]=c[i+1][1];c[i+1][1]=s;}
for(i=0;i<N;i++)
b[i]=c[i][1];
return;
}
lookfor(h,k)
int h[],k;
{int i,j;
for(i=0;i<N;i++)
if(h[i]-k==0) j=i;
return j;
}
main()
{int number[N],x[N],i,j,u,p;char name[N][20];
for(i=0;i<N;i++)
{gets(name[i]);
scanf("%d",&number[i]);}
scanf("%d",&p);
find(number,x);
u=lookfor(number,p);
for(i=0;i<N;i++)
{printf("%d",number[i]);
puts(name[x[i]]);}
puts(name[x[u]]);
}

```

8.16 写一函数，输入一个十六进制数，输出相应的十进制数。

```

#include"math.h"
int x;
ff(shu)
char shu[];
{int i=strlen(shu)-1,sum=0;
for(;i>=0;i++)
{if (48<=shu[i]<=57) sum=sum+(shu[i]-48)*pow(16,(i-1));
else if(65<=shu[i]<=90) sum=sum+(shu[i]-55)*pow(16,(i-1));
else if(97<=shu[i]<=102) sum=sum+(shu[i]-87)*pow(16,(i-1));
x=1;}
else x=0;

```

```

return x;
}
main()
{char shufu[100];int s;
gets(shufu);s=ff(shufu);
if(x) printf("s(D)=%d\n",s);
else printf("The number is not ox\n");
}

```

8.17 用递归法将一个整数  $n$  转换成字符串。例如，输入 486，应输出字符串"486"。  $n$  的位数不确定，可以是任意位数的整数。

```

#include"math.h"
int x[10];
pf(m,n)
unsigned long m;int n;
{int y;
if(n==0) {y=(int)(m%10);x[0]=y;}
else {y=(unsigned long)((m-pf(m,n-1))/pow(10,n))%10;x[n]=y;}
return(y);
}

```

```

main()
{unsigned long a,b;int i,j,k;char c[11];
scanf("%ld",&a);
for(j=0,b=a;b>0.1;j++,b/=10);
pf(a,j-1);
for(i=0,k=j-1;i<j;i++,k--)
c[i]=x[k]+48;c[10]='\0';
puts(c);}

```

或

```

#include"math.h"
char x[11];
pf(m,o)
unsigned long m;int o;
{int j,i;
for(i=o-1,j=0;i>-1;i--,j++)
x[i]=(int)((unsigned long)(m/pow(10,j))%10)+48;
return;
}

```

```

main()
{unsigned long a,b;int j,i;
scanf("%ld",&a);
for(j=0,b=a;b>0.1;j++,b/=10);
pf(a,j);
puts(x);printf("%d\n",j);}

```

或

```

#include"math.h"
int x[10];unsigned long m;
pf(n)
int n;
{int y;
if(n==0) {y=m%10;x[0]=y;}
else {y=(unsigned long)((m-pf(n-1))/pow(10,n))%10;x[n]=y;}
return(y);
}
main()
{unsigned long a;int i,j,k;char c[11];
scanf("%ld",&m);
for(j=0,a=m;a>0.1;j++,a/=10);
pf(j-1);
for(i=0,k=j-1;i<j;i++,k--)
c[i]=x[k]+48;c[10]='\0';
puts(c);}

```

8.18 给出年、月、日，计算该日是该年的第几天。

```

int find(x,y,z)
int x,y,z;
{ int i,t,s,days=0;
if(x%4==0) t=1;
else t=0;
for(i=1;i<y;i++)
{if(i==2) s=2-t;
else s=0;
days=days+30+i%2-s;
}
days=days+z;
return(days);
}
main()
{int year,month,date,day;
scanf("%d %d %d",&year,&month,&date);
day=find(year,month,date);
printf("THE DATE IS THE %dth DAYS\n",day);
}

```

## 第九章 编译预处理

9.1 定义一个带参的宏，使两个参数的值互换，并写出程序，输入两个数作为使用宏时的实参。输出已交换后的两个值。

```

#define CHANGE(a,b,t) t=a;a=b;b=a
main()
{int c,d,s;
scanf("%d,%d",&c,&d);

```

```
CHANGE(c,d,s);
printf("c=%d,d=%d\n",c,d);
}
```

9.2 输入两个整数，求他们相除的余数。用带参的宏来实现，编程序。

```
#define Q(a,b) a%b
main()
{int c,d,t;
scanf("%d %d",&c,&d);
t=Q(c,d);
printf("t=%d\n",t);
}
```

9.3 三角形面积为：

其中  $S=(a+b+c)/2$ ， $a$ 、 $b$ 、 $c$  为三角形的三边。定义两个带参的宏  $S$ ，一个用来求  $area$ ，另一个宏用来求。写程序，在程序中用带实参的宏名来求面积  $area$ 。

```
#include"math.h"
#define SSS(m,n,k) (m+n+k)/2
#define AQRT(m,n,k)
sqrt(SSS(m,n,k)*(SSS(m,n,k)-m)*(SSS(m,n,k)-n)*(SSS(m,n,k)-k))
main()
{float a,b,c,s,area;
scanf("%f %f %f",&a,&b,&c);
s=SSS(a,b,c);
area=AQRT(a,b,c);
printf("s=%.3f area=%.3f\n",s,area);
}
```

或

```
#include"math.h"
#define SSS(m,n,k) (m+n+k)*0.5
#define AQRT(m,n,k)
sqrt(((m+n+k)/2)*((m+n+k)/2-m)*((m+n+k)/2-n)*((m+n+k)/2-k))
main()
{float a,b,c,t,area;
scanf("%f %f %f",&a,&b,&c);
t=SSS(a,b,c);
area=AQRT(a,b,c);
printf("t=%.3f;area=%.3f\n",t,area);
}
```

或

```
#include"math.h"
#define SSS(m,n,k) ((m+n+k)/2)
#define AQRT(m,n,k,m,n,k,m,n,k,m,n,k,m,n,k,m,n,k,m,n,k)
sqrt(SSS*(SSS-m)*(SSS-n)*(SSS-k))
main()
```



```

{float a,b,c,t,area;
scanf("%f %f %f",&a,&b,&c);
t=SSS(a,b,c);
area=AQRT(a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c,a,b,c);
printf("t=%.3f;area=%.3f\n",t,area);
}

```

9.4 给年份 **year**, 定义一个宏, 以判别该年份是否闰年 提示: 宏名可以定义为 **LEAP\_YEAR**, 形参为 **y**, 既定义宏的形式为

```
#define LEAP_YEAR(y) (读者设计的字符串)
```

在程序中用以下语句输出结果:

```

if (LEAP_YEAR(year)) printf("%d is a leap year",year);
else printf ("%d is not a leap year",year);
#define LEAPYEAR(y) y%4
main()
{int y;
scanf("%d",&y);
if(LEAPYEAR(y)) printf("%d is a not leap year\n",y);
else printf("%d is a lear year\n",y);
}

```

9.6 请设计输出实数的格式, 包括: (1)一行输出一个实数; (2)一行内输出两个实数; (3)一行内输出三个实数。实数用"**6.2f**"格式输出。

```

#define PR1(x) printf("%6.2f\n",x)
#define PR2(x) printf("%6.2f\t%6.2f\n",x,x)
#define PR3(x) printf("%6.2f\t%6.2f\t%6.2f\n",x,x,x)
main()
{float a;scanf("%f",&a);
PR1(a);PR2(a);PR3(a);
}

```

9.7 设计所需的各种各样的输出格式 (包括整数、实属、字符串等), 用一个文件名 "**fornat.h**", 把信息都放到这个文件内, 另编一个程序文件, 用命令 **#include "fornat.h"** 以确保能使用这些格式。

分别用函数和带参的宏, 从三个数中找出最大的数。

```

#define MAX(a,b,c) (a>((b>c)? b:c))? a:((b>c)? b:c)
main()
{float x,y,z,t;
scanf("%f,%f,%f",&x,&y,&z);
t=MAX(x,y,z);
printf("%.4f\n",t);
}

```

9.10 用条件编译方法实现以下功能:

输入一行电报文字, 可以任选两种输出, 一为原文输出; 一为将字母变成其下一字母 (如 'a' 变成 'b' ..... 'z' 变成 'a' 其它字符不变)。用命令来控制是否要译成密码。例如:

```
#define CHANGE 1
```

则输出密码。若:

```
#define CHANGE 0
```

则不译为密码，按原码输出。

```
#define CHANGE 1
```

```
main()
```

```
{char str[80],c;int i=0;
```

```
gets(str);
```

```
while(str[i]!='\0')
```

```
{
```

```
#if CHANGE
```

```
if(str[i]==90||str[i]==122) str[i]=str[i]-25;
```

```
else if(str[i]>=65&&str[i]<90||str[i]>=97&&str[i]<122) str[i]=str[i]+1;
```

```
#endif
```

```
i++;
```

```
}
```

```
puts(str);
```

```
}
```

或

```
#define CHANGE 1
```

```
main()
```

```
{char str[80],*c;int i=0;
```

```
gets(str);c=str;
```

```
while(*c!='\0')
```

```
{
```

```
#if CHANGE
```

```
if(*c==90||*c==122) *c=*c-25;
```

```
else if(*c>=65&&*c<90||*c>=97&&*c<122) *c=*c+1;
```

```
#endif
```

```
c++;
```

```
}
```

```
puts(str);
```

```
}
```

## 第十章 指针

10.1 输入三个整数，按由小到大的顺序输出。

```
main()
```

```
{ int a,b,c,*p1,*p2,*p3,t;
```

```
scanf("%d,%d,%d",&a,&b,&c);
```

```
p1=&a;p2=&b;p3=&c;
```

```
if(*p1>*p2)
```

```
{t=p1;p1=p2;p2=t;}
```

```
if(*p1>*p3)
```

```
{t=p1;p1=p3;p3=t;}
```

```
if(*p2>*p3)
```

```
{t=p2;p2=p3;p3=t;}
```

```
printf("%d,%d,%d\n",*p1,*p2,*p3);
```

```

}
或
main()
{int a,b,c,*p1,*p2,*p3,t;
scanf("%d,%d,%d",&a,&b,&c);
p1=&a;p2=&b;p3=&c;
if(a>b)
{t=*p1;*p1=*p2;*p2=t;}
if(a>c)
{t=*p1;*p1=*p3;*p3=t;}
if(b>c)
{t=*p2;*p2=*p3;*p3=t;}
printf("%d,%d,%d\n",a,b,c);
}

```

10.2 输入三个字符串，按由小到大的顺序输出

```

#define N 3
#define M 20
main()
{char str0[N][M],str1[M],*p,*q;
int i,l,m,n;
q=str0;
for(;p<q+N;p++)
gets(p);
l=strcmp(q,q+1);
if(l>0)
{strcpy(str1,q);strcpy(q,q+1);strcpy(q+1,str1);}
m=strcmp(q,q+2);
if(m>0)
{strcpy(str1,q);strcpy(q,q+2);strcpy(q+2,str1);}
n=strcmp(q+1,q+2);
if(n>0)
{strcpy(str1,q);strcpy(q+1,q+2);strcpy(q+2,q+1);}
for(p=q;p<q+N;p++)
puts(p);
}

```

10.3 输入 10 个整数，将其中最小的数与第一个数对换，把最大的数与最后一个数对换。  
写三个函数：①输入 10 个数；②进行处理；③输出 10 个数。

```

f(x,n)
int x[],n;
{int *p0,*p1,i,j,t,y;
i=j=x[0];p0=p1=x;
for(y=0;y<n;y++)
{if(x[y]>i)
{i=x[y];p0=&x[y];}

```

```

else if(x[y]<j)
{ j=x[y];p1=&x[y];} }
t=*p0;*p0=x[n-1];x[n-1]=t;
t=*p1;*p1=x[0];x[0]=t;
return;
}
main()
{int a[10],u,*r;
for(u=0;u<10;u++)
scanf("%d",&a[u]);
f(a,10);
for(u=0,r=a;u<10;u++,r++)
printf(" %d",a[u]);
printf("\n");
}

```

10.4 有  $n$  个整数, 使前面各数顺序向后移  $m$  个位置, 最后  $m$  个数变成前面  $m$  个数, 见图。  
写一函数: 实现以上功能, 在主函数中输入  $n$  个数和输出调整后的  $n$  个数。

```

#define N 10
void shift(p,x)
float *p;int x;
{ float a[N],*q,*o;int i;
o=a;q=p;
for(i=0;i<x;i++)
*(o+i)=*(q+N-x+i);
for(p=p+N-1;p>=q;p--)
*p=*(p-x);
for(i=0;i<x;i++)
*(q+i)=*(o+i);
return;}
main()
{ float shuzhu[N],*u,*v;
int h,i;u=v=shuzhu;
scanf("%f",&h);
for(;u<v+N;u++)
scanf("%f",u);
shift(v,h);
for(u=v;u<v+N;u++)
printf("%.2f ",*u);
printf("\n");
}

```

10.5 有  $n$  人围成一圈, 顺序排号。从第 1 个人开始报数 (从 1 到 3 报数), 凡报到 3 的人退出圈子, 问最后留下的是原来的第几号的那位。

```

#define N 5
main()

```

```

{int i,j,k,a[N+1],*p;
for(i=0,p=a;p<=a+N;i++,p++)
*p=i;
p=a+1;k=N;
for(i=0,j=1;k!=1;j++)
{if(p>(a+N))
p=a+1;
if(*p!=0)
i++;
if((i-3)==0)
{*p=0;i=0;k--;}
p++;
}
for(i=1;i<=N;i++)
if(a[i]!=0)
printf("The last number is %d\n",a[i]);}

```

10.7 有一字符串，包含  $n$  个字符。写一函数，将此字符串中从第  $m$  个字符开始的全部字符复制成为另一个字符串。

```

#include"stdio.h"
#define N 10
main()
{char a[N+1],b[N+1],*p,*q;
int m;
gets(a);
scanf("%d",&m);
p=a+m;q=b;
strcpy(q,p);
puts(q);
}

```

10.6 写一函数，求一个字符串的长度。在主函数中输入字符串，并输出其长度。

第十一章

11. 1 定义一个结构体变量（包括年、月、日）。计算该日在本年中是第几天，注意闰年问题。

解：Struct

```

{int year;
int month;
int day;
}date;
main()
{int days;
printf("Input year,month,day:");
scanf("%d,%d,%d",&date.year,&date.month,&date.day);
switch(date.month)
{case 1: days=date.day; break;

```

```

        case 2: days=date.day+31;        break;
        case 3: days=date.day+59;        break;
        case 4: days=date.day+90;        break;
        case 5: days=date.day+120;       break;
case 6: days=date.day+31;        break;
        case 7: days=date.day+181;       break;
        case 8: days=date.day+212;       break;
case 9: days=date.day+243;        break;
case 10: days=date.day+273;       break;
case 11: days=date.day+304;       break;
case 12: days=date.day+334;       break;
    }
    if((date.year%4==0&&date.year%100!=0||date.year%400==0)&&date.month
    h>=3)days+=1;
    printf("\n%d/%d is the %dth day in%d.",date.month,data.day,days,date,year);
}

```

11.2 写一个函数 **days**,实现上面的计算。由主函数将年、月、日传递给 **days** 函数,计算后将日数传回主函数输出。

解: struct y\_m\_d

```

{int year;
int month;
int day;
}date;
intdays(struct y_m_d date1)
{int sum;
switch(data.month)
{case 1:sum=date1.day;        break;
case 2:sum=date1.day+31;      break;
case 3:sum=date1.day+59;      break;
case 4:sum=date1.day+90;      break;
case 5:sum=date1.day+120;     break;
case 6:sum=date1.day+151;     break;
case 7:sum=date1.day+181;     break;
case 8:sum=date1.day+212;     break;
case 9:sum=date1.day+243;     break
case 10:sum=date1.day+243;    break
case 11:sum=date1.day+243;    break
case 12:sum=date1.day+243;    break
}
};

```

11.3 编写一个函数 **print**,打印一个学生的成绩数,该数组中有 5 个学生的数据记录,每个记录包括 num、name、score[3],用主函数输入这些记录,用 **print** 函数输出这些记录。

解:

```
#define N 5
```

```

struct student
{char num[6];
  char name[8];
  int score[4];
}stu[N];
main()
{int i,j ;
  for(i=0;i<N;i++)
  {printf("\Input score of student %d:\n",i+1);
    printf("no.:");
    scanf("%s",stu[i].num);
    printf("name:");
    scanf("%s",stu[i].name);
    for(j=0;j<3;j++)
    {printf("score%d:"j+1);
      scanf("%d",&stu[i].score[j]);
    }
    printf("\n");
  }
  print(stu);
}
print(struct student stu[6])
{int i,j;
  printf("%5s%10s",stu[i].num,stu[i].name);
  for(j=0;j<3;j++)
    printf("%9d",stu[i].score[j]);
  print("\n");
}

```

11.4 在上题的基础上，编写一个函数 **input**,用来输入 5 个学生的数据记录。  
解:

```

#define N 5
struct student
{char num[6];
  char name[8];
  int score[4]
}stu[N];
input(struct student stu[])
{int i,j;
  for(i=0;i<N;i++)
  {printf("input scores of student %d:\n",i+1);
    printf("NO.:");
    scanf("%s",stu[i].num);
    printf("name: ");
    scanf("%s", stu[i].name);
  }
}

```

```

for(j=0;j<3;j++)
{printf("score%d:",j++);
scanf("%d", &stu[i].score[j]);}
}
printf("\n");
}
}

```

11.5 有 10 个学生，每个学生的数据包括学号、姓名、3 门课的成绩，从键盘输入 10 个学生的数据，要求打印出 3 门课的总平均成绩，以及最高分的学生的数据（包括学号、姓名、3 门课成绩）

解：#define N 10

```

struct student
{char num[6]
char name[8]
int score[4]
float avr;
}stu[N];
main()
{int l,j,max,maxi,sum;
float average;
for(l=0;l<N;l++)
{printf("\nInput scores of student %d:\n",l+1);
printf("NO.:");
scanf("%s",stu[l].num);
printf("name");
scanf("%s",stu[l].name);
for(j=0;j<3;j++)
{printf("score %d:",j+1);
scanf("%d", &stu[l].score[j]);
}
}
average=0;
max=0;
maxi=0;
for(i=0;i<N;i++)
{sum=0;
for(j=0;j<3;j++)
sum+=stu[i].score[j];
stu[i].avr=sum/3.0;
average+=stu[i].avr;
if(sum>max)
{max=sum;
maxi=i;
}
}
}

```



```

}
average/=N;
printf("NO. name score1 score2 score3 average\n");
for(l=0;l<N;l++)
{printf("%5s%10s",stu[i].num, stu[i].name);
for(j=0;j<3;j++)
printf("%9d",stu[i].score[j]);
printf("%8.2f\n",stu[i].avr);
}
printf("average=%6.2f\n",average);
printf("The highest score is:%s,score total:%d."stu[maxi].name,max);
}

```

11.6 编写一个函数 **new**,对 **n** 个字符开辟连续的存储空间,此函数应返回一个指针,指向字符串开始的空间。**New(n)**表示分配 **n** 个字节的内存空间。

解: **new** 函数如下:

```

#define NULL 0
#define NEWSIZE 1000
char newbuf[NEWSIZE];
char *newp=newbuf;
char *new(int n)
{if (newp+n<=newbuf+ NEWSIZE)
    { newp= newp+n;
      return(newp-n);
    }
else
    return(NULL);
}

```

11.7 写一函数 **free**,将上题用 **new** 函数占用的空间释放。**Free(p)**表示将 **p** 指向的单元以后的内存段释放。

解:

```

#define Null 0
#define NEWSIZE 1000
char newbuf[NEWSIZE];
char *newp=newbuf;
free(char *p)
{if((p>=newbuf)&&(p<newbuf+NEWSIZE))
    newp=p;
}

```

11.8 已有 **a**、**b** 亮光链表,每个链表中的结点包括学好、成绩。要求把两个链表合并,按学号升序排列。

解:

```

#include<stdio.h>
#define NULL 0
#define LENsizeof(struct student)

```

```

struct student
{long num;
int scor;
struct student *next
};
struct student listA,listB;
int n,sum=0;
main()
{struct student *creat(void);
struct student *insert(struct student *,struct student *);
void print(struct student *);
stuct student *ahead , *bhead,*abh;
printf("\ninput list a:\n");
ahead=creat();
sum=sum+|n;
abh=insert(ahead,bhead);
print(abh);
}
struct student *creat(void)
{struct student *p1,*p2,*head;
n=0;
p1=p2=(struct student *)malloc(LEN);
printf("input number&scores of student:\n");
printf("if number ls 0,stop inputing.\n");
scanf("%ld,%d",&p1->num,&p1->score);
head=NULL;
while(p1->num!=0)
{n=n+1;
if(n==1)head=p1;
else p2->next =p1;
p2=p1;
p1=(struct student *)malloc(LEN);
scanf("%ld,,%d",&p1->num,&p1->score);
}
p2->next=NULL;
return(head);
}
struct student *insert(struct student *ah,struct student *bh)
{struct student *pa1 , *pa2,*pb1,*pb2;
pa2=pa1=ah;
pb2=pb1=bh;
do
{while((pb1->num>pa1->num)&&(pa1->next!=NULL))
{pa2=pa1;

```

```

pa1=pa1->next;
}
if(pb->num<=pa1->num)
{if(ah=pa1)
ah=pb1;
else pa2->next=pb1;
pb1=pb1->next;
pb2->next=pa1;
pa2=pb2;
pb2=pb1;
}
}
while((pa1->next!=NULL)||((pa1==NULL&&pb1!=NULL)));
if((pb1->num>pa1->num)&&(pa1->next==NULL))
    ap1->next=pb1;
return(ah);
}
void print(struct student *head)
{struct student *p;
printf("%ld%d\n",p->num,p->score);
p=p->next;
while(p!=NULL);
}

```

11.9 13 个人围成一圈，从第 1 个人开始顺序报号 1、2、3。凡报到“3”者退出圈子。找出最后留在圈子中的人原来的序号。

解：

```

#define N 13
struct person
{int number;
int nextp;
}link[N+1];
main()
{int l,count,h;
for(l=1;l<=N;l++)
{if(l==N)
    link[l].nextp=1;
else
    link[l].nextp=l+1;
link[l].number=l;
}
printf("\n");
count=0;
h=N;
printf("sequence that person2 leave the circle:\n");

```

```

while(count<N-1)
{
    l=0;
    while(l!=3)
    {
        h=link[h].nextp;
        if(link[h].number)
            l++;
    }
    printf("%4d",link[h].number);
    link[h].number=0;
    count++;
}
printf("\nThe last one is");
for(l=1;l<=N;l++)
    if(link[l].number)
        printf("%3d",link[l].number);
}

```

11.10 有两个链表 **a** 和 **b**, 设结点中包含学号、姓名。从 **a** 链表中删去与 **b** 链表中有相同学号的那些结点。

解:

```

#define LA 4
#define LB 5
#define NULL 0
struct student
{
    char num[6];
    char name[8];
    struct student *next;
}A[LA],b[LB];
main()
{
    struct student
    a[LA]={{"101","Wang"}, {"102","Li"}, {"105","zhang"}, {"106","Wei"}};
    struct
    studentb[LB]={ {"103","Zhang"}, {"104","Ma"}, {"105","Chen"}, {"107","Guo"},
    {"108","Lui"}};
    int i,j;
    struct student *p, *p1,*p2,*pt,*head1,*head2;
    head1=a;
    head2=b;
    printf("list a :\n");
    for(p1=head1,i=1;p1<a[LA];i++)
    {
        p=p1;
        p1->next=a[i];
        printf("%8s%8s\n",p1->num,p1->name);
        p1=p1->next;
    }
}

```

```

}
p->next=NULL;
printf("\n list b:\n");
for(p2=head2,l=1;p2<b+LB;l++)
{p=p2;
p2->next=b+l;
printf("%8s%8s\n",p2->num,p2->name);
p2=pa->next;
}
p->next=NULL;
printf("\n");
p1=head1;
while(p1!=NULL)
{p2=head2;
while(p2!=NULL&&strcmp(p1->num,p2->num)!=0)
p2=p2->next;
if(strcmp(p1->num,p2->num==0))
    if(p1==head1)
        head1=p1->next;
    else
        p->next=p1->next;
p=p1;
p1=p1->next;
}
p1=hedad1;
printf{"\n result:\n"};
while(p1!=NULL)
{printf("%7s %7s\n",p1->num,p1->name);
p1=p1->next;
}
}
}

```

11.11 建立一个链表，每个结点包括：学号、姓名、性别、年龄。输入一个年龄，如果链表中的结点所包含的年龄等于此年龄，则将此结点删去。

解：#define NULL 0

#define LEN sizeof(struct student)

struct student

{char num[6];

char name[8];

char sex[2];

int age;

stuct student \*next;

}stu[10];

main()

{struct student \*p,\*pt,\*head;

```

int l,length,iage,flag=1;
int find=0;
while(flag==1)
{printf( "input length of list (<10) : " );
scanf("%d",&length);
if(length<10)
flag=0;
}
for(l=0;l<lenth;l++)
{p=(struct student *)malloc(LEN);
if(l==0)
head=pt=p;
else
pt->next=p;
pt=p;
ptintf("NO:");
scanf("%s",p->num);
prntf("name:");
scanf("%s",p->name);
printf("sex:");
scanf("%s",p->sex);
printf("age:");
scanf("%s",p->age);
}
p->next=NULL;
p=head;
printf("\n NO. name sex age\n");
while(p!=NULL)
{printf("%4s%8s%6s%6d\n",p->num, p->name, p->sex, p->age);
p=p->next;
}
printf("Input age:");
scanf("%d",&iage);
pt=head;
p=pt;
if(pt->age==iage)
{p=pt->next;
head=pt=p;
find=1;
}
else
pt=pt->next;
while(pt!=NULL)
{if(pt->age==iage)

```

```

{p->next=pt->next;
find=1;
}
else p=pt;
pt=pt->next;
}
    if(!find)
printf("Not found%d.",iage);
    p=head;
    printf("\n NO.name sex age\n");
    while(p!=NULL)
    {
printf("%4s%8s",p->num,p->name);
printf("%6s%6d",p->sex,p->age);
p=p->next;
}
}

```

11.12 将一个链表按逆序排列，即将链头当链尾，链尾当链头。  
解：

```

# define NULL 0
struct stu
{int num;
    struct stu *next;
}
main()
{int len=1
struct stu *p1,*p2,*head,*new,*newhead;
p1=p2=head=(struct stu * )malloc(sizeof(strct stu));
printf("input number(0:list end):");
scanf("%d",&p1->num);
while(p1->num!=0)
{p1=(struct stu*)malloc(sizeof(struct stu));
printf("input number(n:listend):");
scanf("%d",&p1->num);
if(p1->num==0)
    p2->next=null;
else
    {p2->next=p1;
p2=p1;
len++;
}
}
p1=head;
prtnf("\n the original list:\n");

```

```

do
{printf("%4d",p1->num);
if(p1->next!=NULL)
p1=p1->next;
}
while(p1->next!=NULL)
{p2=p1;
p1=p1->next;
}
if(l==0)
newhead=new=p1;
else
new=nes->next=p1;
p2->next=NULL;
}
printf("\n\n The new listL\n");
p1=newhead;
for(l=0;l<len;l++)
{printf("4d,p1->num");
p1=p1->next,
}
printf("\n");
}

```

## 第 12 章

12. 1 编写一个函数 **getbits**，从一个 16 位的 单元中取出某几位（即该几位保留原值，其余位为 0）。函数调用形式为：**getbits(value,n1,n2)**其中 **value** 为该 16 位单元中的数据值，**n1** 为欲取出的起始位，**n2** 为欲取出的结束位。

解：main()

```

{unsigned int a;
int n1,n2;
printf("input an octal number:");
scanf("%o",&a);
printf("input n1,n2:");
scanf("%d,%d",&n1,&n2);
printf("result:%o\n",getbits(a,n1-1,n2));
}

getbits(unsigned value,int n1,n2)
{unsigned int z;
z=~0;
z=(z>>n1)&(z<<(16-n2));
return(z);;
}

```

12.2 写一个函数，对一个 16 位的二进制数驱除它的奇数位。

解：main()



```

{unsigned getbits(unsigned);
unsigned int a;
printf("\ninput an octal number:");
scanf("%o",&a);
printf("result:%o\n",getbits(a));
}
unsigned getbits(unsigned value)
{int l,j,m,n;
unsigned int z,a q;
z=0;
for(l=1;l<=15;l+=2)
{q=1;
for(j=1;j<=(16-l-1)/2;j++)
q=q*2;
a=value>>(16-l);
a=a<<15;
a=a>>15;
z=z+a*q;
}
return(z);
}

```

12.3 编一程序，检查一下年一所用的计算机系统的 C 编译在执行有移时是按照逻辑位移的原则，还有按照算术右移的原则。如果是逻辑右移，请编一函数实现算术右移。如果是算术右移，请编写一函数实现逻辑右移。

解：

```

main()
{int a,n,m;
a=~0;
if(a>>5)!=a
{printf("\nTurbo C,logical move!\n");
m=0;
}
else
{printf("\nTurbo C,arithmetic move!\n");
m=1;
}
printf("input an octal number:");
scanf("%o",&a);
printf("how mang digit move thwards the right ::");
scanf("%d",&n);
if(m==0)
printf("Arithmetic right move,result:%o\n",getbits1(a,n));
else
printf("Logical ritht move,result:%o", getbits2(a,n));
}

```

```

}
getbits1(unsigned value,int n)
{unsigned z;
z=~0;
z=z>>n;
z=~z;
z=z|(value>>n);
return(z);
}
egtbit2(unsigned valud,int n )
{unsigned z;
z=(~(1>>n)&(value>>n));
}

```

12.4 编一函数用来实现左右循环移位。函数名为 **move**，调用方法为：  
**move(value,n)** 其中 **value** 为要循环位移的数，**n** 为位移的位数。

解：

```

main()
{unsigned moveright(unsigned,int);
unsigned moveleft(unsigned,int);
unsigned a;
int n;
printf("\n input an octal number:");
scanf("%o",&a);
printf("input n:");
scanf("%d",&n);
if(n>0)
{moveright(a,n);
printf("result:%o\n",moveright(a,n));
}
else
{n=-n;
moveleft(a,n);
printf("result:%o\n",moveleft(a,n));
}
}
unsigned moveright(unsigned value,int n)
{unsigned z;
z=(value>>n)|(value<<(16-n));
return(z);
}
unsigned moveleft(unsigned value,int n )
{unsigned z;
z=(value>>(16-n)|(value<<n);
return(z);
}

```

```
}
```

12.5 设计一个函数，使给出一个数的原码，能得到该数的补码。

解：

```
main()
{unsigned int a;
unsigned int getbits(unsigned);
printf("\nInput an octal number:");
scanf("%o",&a);
printf("result:%o\n",getbits(a));
}
unsigned int getbits(unsigned value)
{unsigned int z;
z=value&0100000;
if(z==0100000)
    z=~value+1;
else
    z=value;
return(z);
}
```

### 第十三章

13.4 从键盘输入一个字符串，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。输入的字符串以“!”结束。

解：#include <stdio.h>

```
main()
{
File *fp;
Char str[100];
Int l=0;
If((fp=fopen("test","w")==NULL)
{printf("Can not 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]=str[i-32];
    fputc(str[i],fp);
    l++;
}
fclose(fp);
fp=fopen("test","r");
fgets(str,strlen(str)+1,fp);
printf("%s\n",str);
}
```

```
fclose(fp);
}
```

13.5 有两个磁盘文件”A”和”B”,各存放一行字母,要求把这两个文件中的信息合并(按字母顺序排列),输出到一个新文件”C”中。

解: #include<stdio.h>

```
main()
{
FILE *fp;
Int l,j,n,i1;
Char c[100],t ,ch;
If((fp=fopen("a1","r"))==NULL)
{printf("can not open the file\n");
exit(0);
}
printf("\n file A:\n");
for(l=0;(ch=fgetc(fp))!=EOF;l++)
{c[l]=ch;
putchar(c[l]);
}
fclose(fp);
i1=l;
if((fp=fopen("b1","r"))==NULL)
{printf("\n can not ipen the file");
exit(0);
}
printf("\n file B:\n");
for(l=i1;(ch=fgetc(fp))!=EOF;l++)
{c[l]=ch;
putchar(c[l]);
}
fclose(fp);
n=l;
for(i=0;i<n;i++)
for(j=l+1;j<n;j++)
if(c[i]>c[j])
{t=c[i];
c[i]=c[j];
c[j]=t;
printf("\n file C:\n");
fp=fopen("c1","w");
for(l=0;l<n;l++)
{putc(c[l],fp);
putchar(c[l]);
}
}
```

```

        fclose(fp);
    }

```

13.6 有 5 个学生，每个学生有 3 门课的成绩，从键盘输入以上数据（包括学生号、姓名、三门课成绩），计算出平均成绩，将原有数据和计算出的平均分数存放在磁盘文件 **stud** 中。

解：

```

#include<stdio.h>
struct student
{
    char num[10];
    char name[8];
    int score[3];
    float ave;
}stu[5];
main()
{
    int i,j,sum;
    FILE *fp;
    for(i=0;i<5;i++)
    {
        printf("\n input score of student%d:\n",i+1);
        printf("NO.:");
        scanf("%s",stu[i].num);
        printf("name:");
        scanf("%s",stu[i].name);
        sum=0;
        for(j=0;j<3;j++)
        {
            printf("score %d : ",j+1);
            scanf("%d",&stu[i].score[j]);
            sum+=stu[i].score[j];
        }
        stu[i].ave=sum/3.0;
    }
    fp=fopen("stud","w");
    for(i=0;i<5;i++)
    if(fwrite(&stu[i],sizeof(struct student),1,fp)!=1)
        printf("File write error\n");
    fclose(fp);
    fp=fopen("stud","r");
    for(i=0;i<5;i++)
    {
        fread(&stu[i],sizeof(struct student),1,fp);
        printf("%s,%s,%d,%d,%d,%6.2f\n",stu[i].num,stu[i].name,stu[i].score[0],
            stu[i].score[1], stu[i].score[2], stu[i].ave);
    }
}

```

13.7 将上题 **stud** 文件中的学生数据按平均分进行排序处理，并将已排序的学生数据存入一个新文件 **stu-sort** 中。

解：

```

#include <stdio.h>
#define N 10
struct student
{char num[10];
char name[8];
int score[3];
float ave;
}st[N],temp;
main()
{
FILE *fp;
int i,j,n;
if((fp=fopen("stud","r"))==NULL)
{printf("can not open the file");
exit(0);
}
printf("\n file 'stud':");
for(i=0;fread(&st[i],sizeof(struct student),1,fp)!=0;i++)
{printf("\n%8s%8s",st[i].num,st[i].name);
for(j=0;j<3;j++)
printf("%8d",st[i].score[j]);
printf("%10.1f",st[i].ave);
fclose(fp);
n=i;
for(i=0;i<n;i++)
for(j=i+1;j<n;j++)
if(st[i].ave<st[j].ave)
{temp=st[i];
st[i]=st[j];
st[j]=temp;
}
printf("\nnow:");
fp=fopen("stu-sort","w");
for(i=0;i<n;i++)
{fwrite(&st[i],sizeof(struct student),1,fp);
printf("\n%8s%8s",st[i].num,st[i].name);
for(j=0;j<3;j++)
printf("%8d",st[i].score[j]);
printf("%10.2f",st[i].ave);
fclose(fp);
}

```

13.8 将上题以排序的学生成绩文件进行插入处理。插入一个学生的 3 门课成绩，程序先计算新插入学生的平均成绩，然后将它按平均成绩高低顺序插入，插入后建立一个新文件。

解：#include <stdio.h>

```

struct student
{char num[10];
char name[8];
int score[3];
float ave;
}st[10],s;
main()
{FILE *fp, *fp1 ;
int l,j,t,n;
printf("\n NO.:");
scanf("%s",s.num);
printf("name:");
scanf("%s",s.name);
printf("score1,score2,score3:");
scanf("%d,%d,%d",&s.score[0], &s.score[1], &s.score[2]);
s.ave=(s.score[0]+s.score[1]+s.score[2])/3.0;
if((fp=fopen("stu_sort","r"))==NULL)
{printf("can not open file.");
exit(0);
}
printf("original data:\n");
for(l=0;l<n;l++)
{printf("\n%8s%8s",st[l].num,st[l].name);
for(j=0;j<3;j++)
printf("%8d",st[l].score[j]);
printf("%10.2f",st[l].ave);
}
n=l;
for(t=0;t<n;t++)
{printf("\nnow:\n");
fp1=fopen("sort1.dat","w");
for(l=t;l<n;l++)
{fwrite(&st[l],sizeof(struct student),1,fp1);
printf("\n%8s%8s",st[l].num,st[l].name);
for(j=0;j<3;j++)
printf("%8d",st[l].score[j]);
printf("%10.2f",st[l].ave);
}
fwrite(&s,sizeof(struct student),1,fp1);

printf("\n%8s%7s%7d%7d%7d%10.2f",s.num,s.name,s.score[0],s.score[1],s.score[2],s.ave);
for(l=t;l<n;l++)
{fwrite(&st[l],sizeof(struct student),1,fp1);

```

```

printf("\n %8s%8s",st[i].num,st[i].name);
for(j=0;j<3;j++)
printf("%8d",st[i].score[j]);
printf("10.2f",st[i].ave);
fclose(fp);
fclose(fp1);
}

```

13.9 上题结果仍存入原有的 stu\_sort 文件而不另建立新文件。

解: #include<stdio.h>

```

struct student
{char num[10];
char name[8];
int score[3];
float ave;
}st[10],s;
main()
{FILE *fp, *fp1;
int l ,j,t,n;
printf("\nNO.:");
scanf("%s",s.num);
printf("name:");
scanf("%s",s.name);
printf("score1,score2,score3:");
scanf("%d%d%d",&s.score[0]+&s.score[1]+&s.score[1], &s.score[2]);
s.ave=( s.score[0]+ s.score[1]+ s.score[2])/3.0;
if((fp=fopen("stu=sort","r"))==NULL)
{printf("can not open the file.");
exit(0);
}
printf("original data:\n");
for(l=0;fread(&st[l],sizeof(struct student),1,fp)!=0;l++)
{printf("\n%8s%8s",st[l].num,st[l].name);
for(j=0;j<3;j++)
printf("%8d",st[l].score[j]);
printf("%10.2f",st[l].ave);
}
fclose(fp);
n=l;
for(t=0;st[t].ave>s.ave+&&t<n;t++);
ptintf("\nnow:\n");
fp=fopen("stu_sort","w");
for(l=0;l<t;l++)
{fwrite(&st[l],sizeof(struct student),1,fp);
printf("\n%9s%8s%8d%8d%8d%10.2f",s.num,s.name,s.score[0],s.score[1]

```



```

s.score[2] s.ave);
for(l=t;l<n;l++)
{fwrit(&sr[i],sizeof(struct srudent),1,fp);
    printf("\n %8s%8s",st[i].num,st[i].name);
for(j=0;j<3;j++)
printf("%8d",st[i].score[j]);
printf("%10.2f",st[i].ave);
}

    fclose(fp);
}

```

**13.10** 有一磁盘文件 **employee**,内存放职工的数据。每个职工的数据包括: 职工姓名、职工号、性别、年龄、住址、工资、健康状况、文化程度。要求将职工名和工资的信息单独抽出来另建一个简明的职工工资文件。

解: #include<stdio.h>

```

struct employee
{char num[6];
char name[10];
char sex[2];
int age;
char addr[20];
int salary;
char health[8];
char class[10];
}en[10];
struct emp
{char name[10];
int salary;
}em_case[10];
main()
{FILE *fp1, *fp2;
int l,j;
if ((fp1=fopen("employee","r"))==NULL)
{printf("can not open the file.");
    exit(0);
}
printf("\n NO. name sex age addr salary health class\n");
for(l=0;fread(&em[i],sizeof(struct employee),1,fp1)!=p;l++)
{printf("\n%4s%8s%4s%6s%10s%6s%10s%8s",em[i].num,em[i].name,em[i].sex,
em[i].age, em[i].addr, em[i].salary, em[i].health, em[i].class);
strcpy(em_case[i].name, em[i].name);
em_case[i].salary=en[i].salary;
}
printf("\n\n*****");
if((fp2=fopen("emp_salary","wb"))==NULL)

```

```

{printf("can not open the file.");
exit(0);}
for(j=0;j<l;j++)
{if(fwrite(&em_case[j],sizeof(struct emp),1,fp2)!=1)
printf("error!");
printf("\n %12s%10d",em_case[j].name,em_case[j].salary);
}
printf("\n*****");
fclose(fp1);
fclose(fp2);
}

```

13.11 从上题的“职工工资文件”中删区一个职工的数据，再存回原文件。

解: #incude <stdio.h>  
#incude <string.h>  
struct employee  
{char name[10];  
int salary;  
}emp[20];  
main()  
{FILE \*fp;  
int l,j,n,flag;  
char name[10];  
int salary;  
if((fp=fopen("emp\_salary","rb"))==NULL)  
{printf("can not open file.");  
exit(0);  
}  
printf("\n original data:");  
for(l=0;fread(&emp[i],sizeof(struct employee),1,fp)!=0;l++)  
printf("\n %8s %7d",emp[i].name,emp[i].salary);  
fclose(fp);  
n=l;  
printf("\n input name deleted:");  
scanf("%s",name);  
for(flag=1,l=0;flag&&l<n;l++)  
{if(strcmp(name,emp[i].name)==0)  
{for(j=l;j<n-1;j++)  
{strcmp(name,emp[j].name)==0  
{for(j=l;j<n-1;j++)  
{strcpy(emp[j].name,emp[j+1].name);  
emp[j].salary=emp[j+1].salary;  
}  
flag=0;  
}  
}

```

        }
if(!flag)
        n=n-1;
else
printf("\n Now,the content of file:\n");
fp=fopen("emp-dalary","wb");
for(l=p;l<n;l++)
        fwrite(&emp[i],sizeof(struct employee),1,fp);
fclose(fp);
fp=fopen("emp_salary","r");
for(l=0;fread(&emp[i],sizeof(struct employee),1,fp)!=0;l++)
printf("\n%8s%7d",emp[i].name,emp[i].salary);
fclose(fp);
}

```

13.12 从键盘输入若干行字符（每行长度不等），输入后把它们存储到一磁盘文件中。再从该文件中读入这些数据，将其中小写字母转换成大写字母后在显示屏上输出。

解： #include<stdio.h>

```

main()
{int l,flag;
char str[80],c;
FILE *fp;
Fp=fopen("text","w");
Flag=1;
While(flag==1)
{printf("\n Input string:\n");
ges(str);
fprintf(fp,"%s",str);
printf("\nContinue?");
c=getchar();
if((c=='N')||(c=='n'))
flag=0;
getchar();
}
fclose(fp);
fp=fopen("text","r");
while(fscanf(fp,"%s",str)!=EOF)
{for(l=0;str[l]!='\0';l++)
        if((str[l]>='a')&& (str[l]<='z'))
                str[l]-=32;
        printf("\n%s\n",str);
}
fclose(fp);
}

```