第 5 章 循环结构程序设计

在许多问题中常常遇到需要重复处理的问题，就要用到循环结构，如要求计算全校每个学生的平均成绩，计算两个正整数的最大公约数和最小公倍数等都需要用到循环控制。循环结构是结构化程序设计的三种基本结构之一，在程序设计中对于那些需要重复执行的操作应该采用循环结构来完成，利用循环结构处理各类重复操作既简单又方便。C 语言中有三种循环语句，分别是 while 语句、do…while 语句和 for 语句。

5.1 while 循环语句

while 循环语句是用来实现“当型”循环结构的。它的特点是先判断表达式，后执行语句。

格式：

```
while(表达式)
   语句;
```

该语句的执行顺序是：先计算表达式的值，再判断其值是否为“真”（即非 0）。若结果为“真”，则执行语句；此过程重复执行，直到表达式的值为“假”（即为 0）时，结束循环。其流程图和 N-S 图分别如图 5-1 和 5-2 所示。

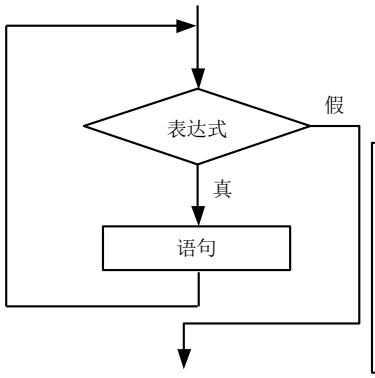

图 5-1while 循环结构流程图

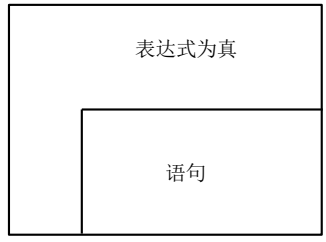

图 5-2 while 循环结构 N-S 图

说明：（1）while 后一对圆括号中的表达式可以是 C 语言中任意合法的表达式，但不能为空，由它来控制循环体是否执行。

（2）因为 while 循环语句是当型循环结构，所以循环体有可能一次也不执行。

（3）循环体可以是任何语句，但应有使循环趋于结束的语句。

【例 5.1】求 1 到 100 之间的所有奇数和。

程序如下：

```
#include "stdio.h"
main()
```

```
{
   int i=1,sum=0;
   while(i<=100)
      {
         sum+=i;
         i=i+2;
      }
   printf("%d",sum);
}
```

程序运行结果：2500

注意：循环体如果包含一个以上的语句，应该用花括弧括起来，以复合语句形式出现。

【例 5.2】通过键盘输入 10 个小于 50 的整数，求它们的平均值。

程序如下：

```
#include "stdio.h"
main()
{ int i=1,sum=0,x;
 float aver;
 while(i<=10)
{scanf("%d",&x);
if(x>=50)printf("please input again");
 else
{sum+=x;
  i++;}
}
aver=sum/10.0;
 printf("aver=%f",aver);
}
```

程序运行结果：10↙12↙-13↙3↙9↙-6↙40↙31↙0↙23↙
　　aver=10.0

注意：（1）“↙”表示回车符，起到结束每个数据输入的作用。

（2）如果循环体不是空语句，则 while 的括号后面不加分号。

5.2　do…while 循环语句

do…while 循环语句的特点是先执行，后判断。

格式：

```
do
   语句
while(表达式);
```

该语句的执行顺序是：首先执行一次循环语句，然后计算表达式的值。若结果为“真”（即非 0），则再次执行循环体。此过程重复执行，直到表达式的值为“假”（即为 0）时，循环结束。其流程图和 N-S 图分别如图 5-3 和 5-4 所示。

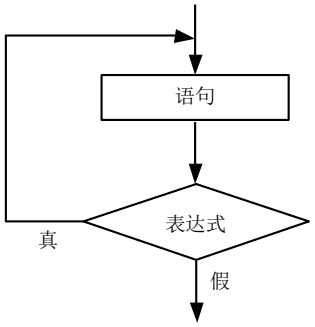

图 5-3　do…while 循环结构流程图

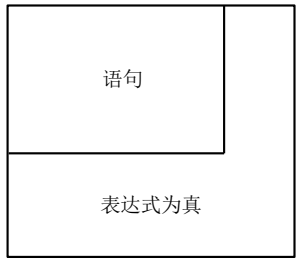

图 5-4　do…while 循环结构 N-S 图

说明：（1）因为 do…while 循环语句是先执行，后判断，所以循环体至少执行一次。

（2）循环体可以是任何语句，但应有使循环趋于结束的语句。

（3）在一般情况下，用 while 语句和 do…while 语句处理同一问题时，若二者的循环体部分是一样的，则它们的结果也一样。

【例 5.3】分别用 while 循环语句和 do…while 循环语句实现 10 以内自然数的和的求解，并做相应的比较。

程序如下：

```
#include "stdio.h"
main()
{
   int sum=0,i;
   scanf("%d",&i);
   while(i<=10)
      {
         sum+=i;
i++;     i++;
      }
   printf("%d",sum);
}
```

程序运行结果：1↙

```
sum=55
```

再运行一次：

```
11↙              11↙
sum=0
```

程序如下：

```
#include "stdio.h"
      main()
      {
         int sum=0,i;
         scanf("%d",&i);
         do
            {
               sum+=i;

            }while(i<=10);
         printf("%d",sum);
      }
```

程序运行结果：1↙

```
      sum=55
```

再运行一次：

```

      sum=11
```

比较：① do…while 语句最后的分号不可少，但 while 语句花括号后不可有分号。

② 当循环条件第一次判断条件就为真时，while 和 do…while 语句在执行过程中没有什么区别；而当循环条件第一次判断条件就为假时，while 语句一次也不执行，而 do…while 语句至少执行一次。

【例 5.4】计算 x 的值使 1+4+7+10+13+…+x 的和小于 200，求出最大的 x。

程序如下：

```
#include "stdio.h"
main()
{  int x=1,sum=1;
do
     {  x=x+3;
        sum+=x;
     }while(sum<=200);
   printf("x=%d",x-3);
}
```

运行结果：x=31

注意：循环体中要有使循环趋于结束的语句，否则会导致死循环。

5.3　for 循环语句

C 语言中 for 循环语句的使用最为灵活，不仅可以用于循环次数已经确定的情况，而且可以用于循环次数不确定的情况，它完全可以代替 while 循环。

格式：for(表达式 1;表达式 2;表达式 3)语句;

该语句的执行顺序是：先计算表达式 1，接着计算表达式 2，若表达式 2 的值为非 0，则执行循环体；然后计算表达式 3，再计算表达式 2。若表达式 2 的值仍为非 0，则继续执行循环体，否则退出循环。其流程图和 N-S 图分别如图 5-5 和 5-6 所示。

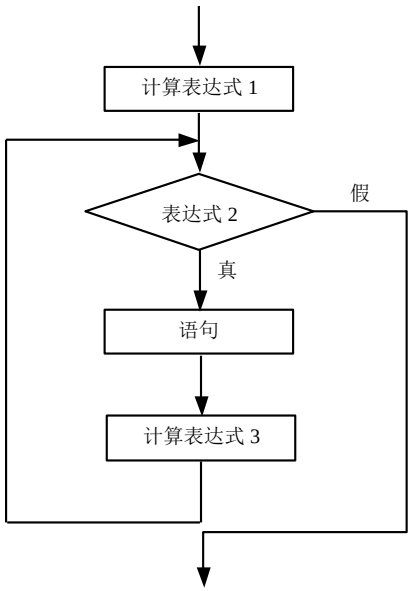

图 5-5　for 循环结构流程图

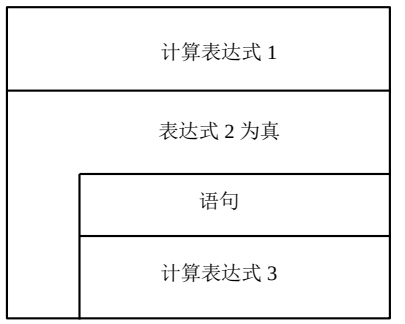

图 5-6　for 循环结构 N-S 图

说明：（1）当表达式 1 或表达式 2 省略时，其后的分号不能省略。

（2）循环体中要有使循环趋于结束的语句。

【例 5.5】计算 n!的值。

程序如下：

```
#include "stdio.h"
main()
{
int mup=1,i,n;
   scanf("%d",&n);
   for(i=1;i<=n;i++)mup*=i;
   printf("mup=%d",mup);
}
```

运行程序并输入：5↙

程序运行结果：mup=120

注意：若循环体不是空语句，不能在 for 语句的圆括号后加分号。

【例 5.6】输出 Fibonacci 数列的前 20 个数。该数列具有两个特点：第一个特点是前两个数都是 1；第二个特点是从第 3 个数开始，每个数都是其前两个数之和。

程序如下：

```
#include "stdio.h"
main()
{  long int n1,n2;
int i;
 n1=1; n2=1;
for ( i=1; i<=10; i++)
{
printf("%10ld  %10ld",n1,n2);
if (i%2==0) printf("\n");
  n1=n1+n2;
n2=n2+n1;   }
}
```

运行结果如图 5-7 所示。

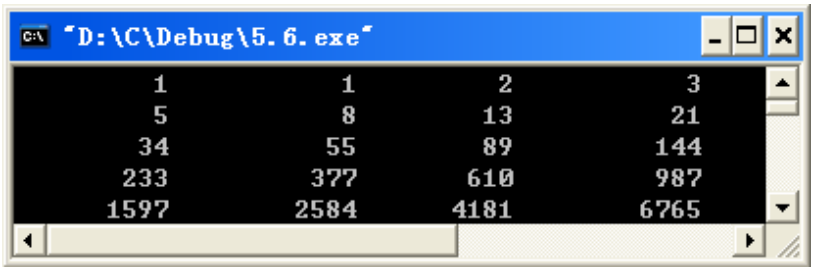

图 5-7　例 5.6 运行结果

注意：循环 10 次，每次输出一对数。每行输出 4 个数。

5.4　循环的嵌套

一个循环体内又包含另一个完整的循环结构称为循环的嵌套。while 循环、do…while 循

环和 for 循环都可以互相嵌套。

格式：　① while（）　② do　③ for(; ;)

```
{…                {…              {…
 while（）         do               for( ; ; )
 {…}               {…}              {…}
 …                 while( );        …
          }                 } while( );          }
```

④ while()　⑤ for(;;)　⑥ do

```
{…                {…              {
 do                while()  …
{…}                {…}               for( ; ; )
 while( );         …                 {…}
 …                 }                 …
          }                                      }while( );
```

说明：（1）嵌套的循环控制变量一般不应同名，以免造成混乱。

（2）循环嵌套要注意正确地使用缩进式书写格式来明确循环嵌套的层次关系，以增加程序的可读性。

【例 5.7】全班有 20 个学生，每个学生考 5 门课程，要求分别统计出每个学生各门课的平均成绩。

分析：首先考虑求一个学生各门课的平均成绩，设置循环控制变量 i 控制课程数，其变化从 1 到 5，每次增量为 1。每个学生的处理过程都一样，因此只需对上述流程重复执行 20 遍即可。设置循环变量 j 控制学生人数，其变化从 1 到 20，每次增量也为 1。

程序如下：

```
#include "stdio.h"
main()
{
  int i,j,score,sum;
  float aver;
  j=1;
  while(j<=20)
{
sum=0;
for(i=1;i<=5;i++)
 {
  printf("Enter No.%d the score %d:",j,i);
                              /*屏幕提示:输入 j 号学生的 i 门课程成绩*/
scanf("%d",&score);           /*输入成绩*/
sum=sum+score;                /*求成绩的和*/
}
aver=sum/5.0;                     /*求成绩的平均值*/
printf("No.%d aver=%f\n",j,aver);    /*输出 j 号学生的平均成绩*/
```

```
    j++;
      }
}
```

程序运行部分结果如图 5-8 所示。

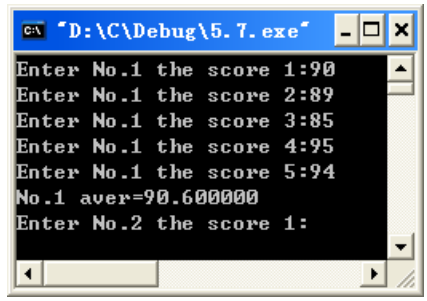

图 5-8　例 5.7 运行结果

注意：外层循环应“完全包含”内层循环，不能发生交叉。

【例 5.8】在 1～5 中取出 3 个互不相同的整型数，输出其和能被 4 整除的个数。

程序如下：

```
#include "stdio.h"
main()
{int i,j,k,n=0;
 for(i=1;i<=5;i++)
    for(j=i+1;j<=5;j++)
      for(k=j+1;k<=5;k++)
        if((i+j+k)%4==0)
{n++;
            printf("i=%d j=%d k=%d \n",i,j,k);}
printf("n=%d\n ",n);
}
```

运行结果如图 5-9 所示。

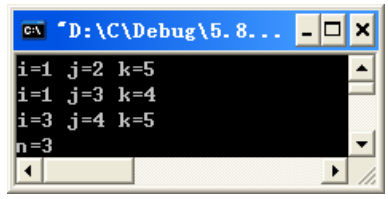

图 5-9　例 5.8 运行结果

注意：通过本题掌握三重 for 循环的用法。

5.5　循环的退出

循环的退出有 break 语句、continue 语句和 goto 语句，这是三种常见的循环退出语句。

5.5.1　break 语句

break 语句可以用来实现从循环体内跳出的功能，即提前结束循环。

格式：`break;`

该语句的功能是：

执行 break 语句，结束整个包含它的最内层循环。

说明：（1）break 语句只能用于 switch 语句或循环语句中。

（2）在嵌套的循环结构中使用时，break 语句只能跳出包含它的最内层循环，不能同时跳出多层循环。

【例 5.9】在 200 以内的正整数中，找出既是 15 的倍数又是 7 的倍数的最大值。

程序如下：

```
#include "stdio.h"
main()
{  int i;
```

```
   for(i=200;i>=0;i--)
   if(i%15==0&&i%7==0)break;
   printf("the result is i=%d",i);
}
```

运行结果：the result is i=105

注意：等于号“==”和赋值号“=”的区别。

【例 5.10】找出 100～200 之间第一个能被 16 整除的数。

程序如下：

```
#include "stdio.h"
main()
{
int i=100;
   while(i<=200)
   {  if(i%16==0)                          /*判断该数是否能被 16 整除*/
        {  printf("%d",i);
           break; }
      else
i++;
   }
}
```

程序运行结果：112

【例 5.11】输入 5 个学生的百分制成绩，分别输出其对应的等级（90～100 为 A，89～70 为 B，69～60 为 C，低于 60 为 D）。

程序如下：

```
#include "stdio.h"
main()
{int a,i=1;
char y;
 while(i<=5)
  {scanf("%d",&a);
switch(a/10)
{case 10:
case 9:y='A';break;
case 8:
 case 7:y='B';break;
  case 6:y='C';break;
  default:y='D';break;
}
printf("y=%c\n",y);
    i++;}
}
```

运行结果：72↙

```
    y=B
```

……

注意：break 语句在 switch 语句中的用法。

5.5.2 continue 语句

continue 语句的作用是结束本次循环，即跳过循环体中下面尚未执行的语句，接着继续下一次是否执行循环体的判定。

格式：

```
continue;
```

该语句的执行顺序是：执行 continue 语句，结束本次循环，判断是否继续循环。

说明：（1）continue 语句只能用于循环语句中。

（2）continue 语句和 break 语句都实现了程序执行方向上的无条件转移，但是 continue 语句只结束本次循环，而不是终止整个循环的执行；break 语句则是结束整个循环过程，不再判断执行循环的条件是否成立。

【例 5.12】输入 10 个整数，输出其中负数的个数及所有负数的和。

分析：本题要求统计出负数的个数，并计算其和。定义一个整型变量 sum 用来存放所有负数的累加和，再定义一个整型变量 num 用来存放负数的个数。

程序如下：

```
#include "stdio.h"
main()
{
   int i,sum=0,num=0,x;
   printf("input 10numbers:");
for(i=0;i<10;++i)
 { scanf("%d",&x);
    if(x>=0) continue;
 sum+=x;
  num++;
  }
if(num)
 printf("num=%d,sum=%d",num,sum);
else
 printf("num is 0");
}
```

程序运行结果：

```
input 10numbers:-3↙1↙3↙-1↙-4↙6↙-7↙9↙0↙-14↙
num=5,sum=-29
```

5.6 用 goto 语句构成循环

goto 语句是无条件转移语句，它的作用是无条件转移。

格式：`goto 语句标号;`

说明：（1）语句标号是用标识符表示，它的命名规则与变量名相同。

例如，“goto loop;”是合法的。

（2）goto 语句和 if 语句配合使用，可实现循环功能。

（3）结构化程序设计方法主张限制使用 goto 语句，因为滥用 goto 语句将会影响程序结构的清晰，降低可读性。

【例 5.13】从键盘输入 10 个整数，求其累加和。可用 goto 语句和 if 语句配合实现。

程序如下：

```
  main()
{
int i=1,sum=0,x;
     loop:if(i>10)goto end;
 printf("enter a data:");
 scanf("%d",&x);
sum+=x;
 i++;
goto  loop;
     end: printf("sum=%d",sum);
  }
```

程序运行结果：

```
enter a data:3↙
enter a data:5↙
enter a data:2↙
enter a data:4↙
enter a data:10↙
enter a data:30↙
enter a data:15↙
enter a data:18↙
enter a data:20↙
enter a data:22↙
sum=129
```

说明：loop 和 end 都是语句标号。

5.7　循环结构程序设计举例

【例 5.14】求 1～100 之间的全部素数。

分析：（1）素数求解可采用如图 5-10 所示的算法：先求出数 m 的平方根 k，然后让该数 m 被 2 到 k 除。

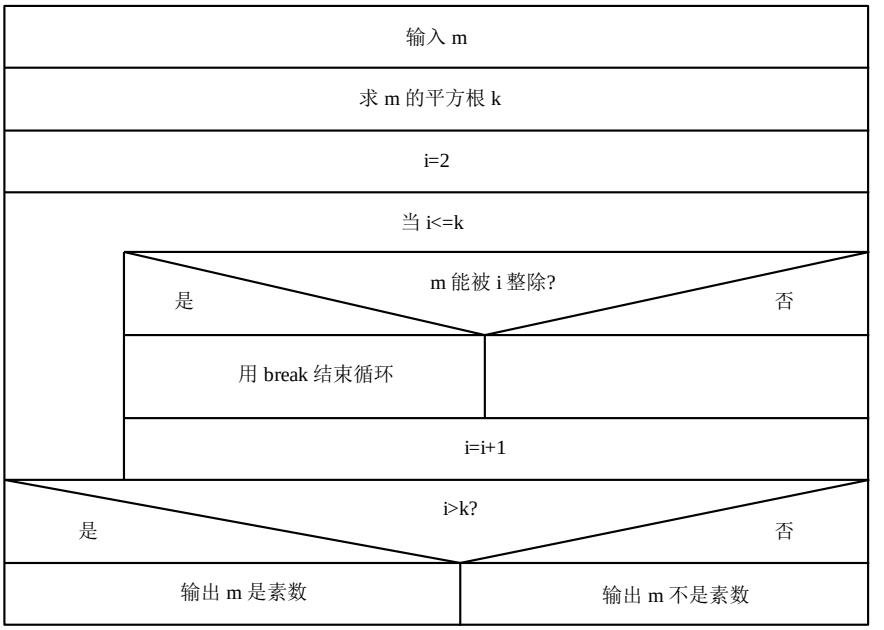

图 5-10　求素数算法的 N-S 图

- 如果 m 能被 2～k 之间的任何一个整数整除，则提前结束循环，说明该数不是素数，此时的 i 必然小于或等于 k。
- 如果 m 不能被 2～k 之间的任何一个整数整除，则不提前结束循环，在结束循环后，i=k+1，说明该数是素数。

（2）本题用一个嵌套的 for 循环即可实现其功能，内层循环用来判断是否是素数，外层循环产生 1～100 的自然数。

程序如下：

```
#include "stdio.h"
#include "math.h"        /*包含数学函数的头文件*/
 main()
 {
   int k,m,i,n=0;
   for(m=1;m<=100;m=m+2)
      {
    k=sqrt(m);   /*sqrt()是求 m 的平方根的函数*/
for(i=2;i<=k;i++)
  if(m%i==0)break;
if(i>k){printf("%-4d",m);n++;}
     /*判断是否是正常退出循环，为真，则 m 是素数，输出并计数*/
if(n%10==0)printf("\n");}/*判断一行是否输出 10 个素数，若是，则换行*/
printf("\nn=%d\n",n);
}
```

程序运行结果：

```
1   3   5   7   11  13  17  19  23  29
31  37  41  43  47  53  59  61  67  71
73  79  83  89  97
n=25
```

【例 5.15】将原文转换为密文。通常在互联网上传送信息时，为保证安全，发送者先按一定的规律将原文转换为密文，然后发送给接收者，接收者收到密文后，再将密文按约定的规律译回原文。例如，转换规律为：字母转换为其后第 6 个字母，非字母不变。

分析：（1）先判断该字符是否是大写字母或者是小写字母，若是，则将其值加 6（变成其后的第 6 个字母）。

（2）若加 6 之后字符值大于'Z'或'z'，则表示原来的字母在'T'或't'之后，应按图 5-11 所示的规律将它转换为'A'～'F'（或'a'～'f'）之一。方法是使其减 26。

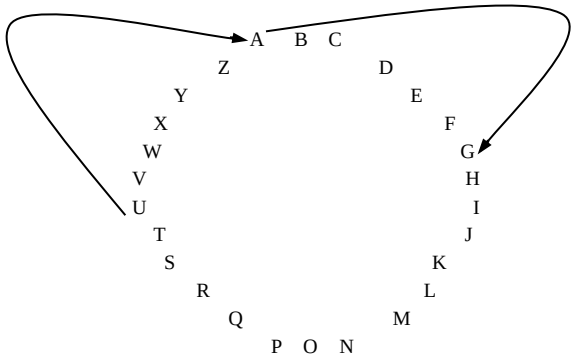

图 5-11　转换规则示意图

程序如下：

```
#include<stdio.h>
main()
{ char ch;
   while ((ch=getchar())!='\n')
{ if((ch>='a'&&ch<='z')||(ch>='A'&&ch<='Z'))
  {ch=ch+6;
          if(ch>'Z'&&ch<='Z'+6||ch>'z')ch=ch-26;}
printf("%c",ch);
}}
```

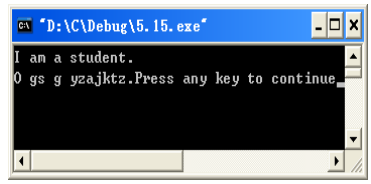

图 5-12　例 5.15 运行结果

输入“I am a student.”后，程序运行结果如图 5-12 所示。

【例 5.16】百马百瓦问题。有 100 匹马，驮 100 块瓦，大马驮 3 块，小马驮 2 块，两个马驹驮 1 块。问大马、小马、马驹各多少？编程列出所有可能的驮瓦方案。

分析：设大马、小马、马驹各为 x、y、z 只，根据题目要求，可以用下列方程表示：

x+y+z＝100

3x+2y+z/2＝100

这是一个不定方程，未知数多于方程数，此题有若干个解。要解此不定方程应先固定一个变量的值，然后求其他两个值。即对其余两个变量的各可能值一一测定，看它是否满足方程。显然这是一个穷举问题。利用穷举法，只要将各种可能的组合一一测试，将符合条件的组合输出即可。

程序如下：

```
#include "stdio.h"
main()
{int  x,y,z;
```

```
for(z=98; z>=2;z=z-2)
for(y=50; y>=1; y--)
for(x=0; x<=33; x++)
if(z+x+y==100&&3*x+2*y+z/2==100)
printf("x=%2d,y=%2d,z=%2d\n",x,y,z);
}
```

程序运行结果如图 5-13 所示。

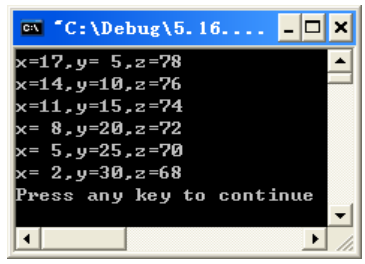

图 5-13　例 5.16 运行结果

【例 5.17】小猴吃桃问题。有一天小猴摘下了若干个桃子，当即吃掉一半，还不过瘾，又多吃了一个；第二天接着吃了剩下的桃子的一半后又多吃一个；以后每天都吃尚存桃子的一半零一个，到第 10 天早上要吃时只剩下一个了。问小猴第一天共摘下了多少个挑子？

分析：这是一个“递推”问题，先从最后一天推出倒数第二天的桃子，再从倒数第二天的桃子推出倒数第三天的桃子…… 。

设第 n 天的桃子为 x，它是 $n-1$ 天的桃子数的一半减 1 个，即：

$$x_n=\frac{1}{2}x_{n-1}-1$$

那么 $n-1$ 天的桃子数为：

$x_{n-1}=(x_n+1)\times 2$　　（递推公式）

已知第 10 天的桃子数 x_{10} 为 1，根据递推公式可得第 9 天的桃子数 x_9 为 4，……。

程序如下：

```
#include "stdio.h"
main()
{
int i,x;
x=1;                        /*第 10 天的桃子数*/
for(i=9; i>=1;i--)
{x=(x+1)*2 ;
printf( "i=%2d,x=%d\n",i,x);  } /*第 i 天的桃子数为 x 只*/
}
```

程序运行后显示结果如图 5-14 所示。

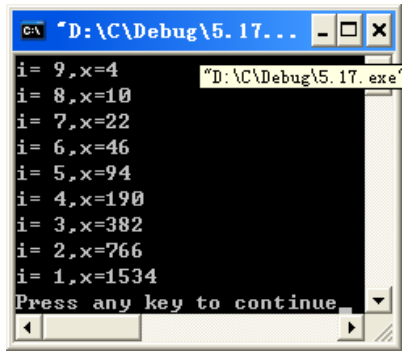

图 5-14　例 5.17 运行结果

【例 5.18】用迭代法求某数 a 的平方根。已知求平方根的迭代公式为：$x_{n+1}=(x_n+a/x_n)/2$

分析：利用以上迭代公式求 a 的平方根的算法步骤如下。

（1）设定一个值给 x_0 作为初值，这里取 $a/2$ 作为 x_0 的初值，利用迭代公式 $x_1=(x_0+a/x_0)/2$ 求出一个 x_1。

（2）把新求得的 x_1 的值赋给 x_0，用此新的 x_0 再求出一个新的 x_1。

（3）如此重复步骤（2），直到前后两次求出的 x_0 和 x_1 满足以下关系:$|x_1-x_0|<10^{-6}$。

（4）输出 a 的平方根的值，即 x_1 的值。

程序如下：

```
#include "stdio.h"
#include "math.h"
main()
{
float a,x0,x1;
printf("Input a: "); scanf("%f",&a);
if (a<0)
    printf(" Input error!\n");        /*不能求负数的平方根*/
else
{   x0=a/2;
    x1=(x0+a/x0)/2;
    do
    {   x0=x1;
        x1=(x0+a/x0)/2;
    }while (fabs(x0-x1)>1e-6);
    printf("sqrt(%f)=%f, 标准 sqrt(%f)=%f\n",a,x1,a,sqrt(a));
}
}
```

执行以上程序，输入为 5 时，输出结果如图 5-15 所示。

图 5-14　例 5.18 运行结果

可以看到，此程序的结果和标准平方根函数 sqrt()的结果基本没有差异。

习　题

一、选择题

1．设 int n=1，则执行循环“while(n++<5);”后 n 的值是：

A．6　　B．5　　C．1　　D．−264

2．执行语句“for(i=0;i++<3;)prinf("%d"，i);”后，程序输出为：

A．2345　　B．012　　C．123　　D．234

3. 在下述程序中，判断 i<j 共执行的次数是：

```
#include "stdio.h"
main()
{int i=0,j=10,k=2,s=0;
for(;;)
   {  i+=k;
      if(i>j)
         {  printf("%d",s);
            break;}
      s+=i;}   }
```

A．4　　B．7　　C．5　　D．6

4. 下列程序的输出结果是：

```
#include "stdio.h"
main()
{int x=0,y=5,z=3;
  while(z- -&&++x<50)
   y--;
printf("%d,%d,%d\n", x,y,z);  }
```

A．3,2,0　　B．3,2,−1　　C．4,3,−1　　D．5,−2,−5

5. 以下程序的功能是：按顺序读入 10 名学生 4 门课程的成绩，计算出每位学生的平均分并输出，程序如下：

```
main()
{ int n,k;
float score,sum,ave;
sum=0.0;
 for(n=1;n<=10;n++)
 { for(k=1;k<=4;k++)
 { scanf(" %f ",&score);  sum+=score;  }
ave=sum/4.0;
  printf(" NO%d:%f\n ",n,ave);
}
}
```

上述程序运行后结果不正确，调试中发现有一条语句出现在程序中的位置不正确。这条语句是：

A．sum=0.0;　　B．sum+=score;

C．ave=sum/4.0;　　D．printf(" NO%d:%f\n " ,n,ave);

6. 以下程序中，while 循环的循环次数是：

```
main()
{ int i=0;
while(i<10)
 { if(i<1) continue;
  if(i==5) break;
   i++;
   }
   ......
}
```

A．1　　B．10

C．6　　D．死循环，不能确定次数

二、填空题

1. 下面程序的功能是：计算 1～10 之间的奇数之和与偶数之和，将该程序补充完整。

```
#include "stdio.h"
main()
{int a,b,c,i;
 a=c=0;
 for(i=0;i<=10;i+=2)
 { a+=i;
;
   c+=b;
}
printf("偶数之和=%d\n",a);
printf("奇数之和=%d\n",c-11); }
```

2. 执行下面程序段后，x 和 i 的值分别是和。

```
main()
{int x,i;
  for(i=1,x=1;i<=20;i++)
  {  if(x>=10)break;
  if(x%2==1)
  {  x+=5;continue;}
     x-=3;
}}
```

3. 程序运行后的输出结果是。

```
main()
{ int  x=l5;
 while (x>10&&x<50)
{ x++;
 if(x/3) { x++;break; }
 else continue;
   }
   printf("%d\n",x);
}
```

4. 有以下程序

```
#include  <stdio.h>
main()
{ char c;
while((c=getchar())!='?')
 putchar(--c);}
```

程序运行时，如果从键盘输入：Y?N?<回车>，则输出结果为。

5. 程序的输出结果是。

```
#include <stdio.h>
main()
{ int i=0,a=0;
while(i<20)
{ for(;;)
```

```
{ if((i%10)==0) break;
 else  i--;
 }
  i+=11; a+=i;
  }
  printf("%d\n",a);  }
```

三、读程题

给出以下程序的输出结果

1.
```
#include <stdio.h>
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);} }
```

2.
```
#include <stdio.h>
main()
{int i,b,k=0;
   for(i=1;i<=5;i++)
   {  b=i%2;
      while(b--==0)k++;
      }
printf("%d,%d",k,b);
}
```

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

4.
```
#include <stdio.h>
main()
{ int i=10, j=0;
do
{ j=j+i; i--;}
while(i>2);
printf("%d\n",j);
}
```

5.
```
#include <stdio.h>
main()
{ int n1,n2;
scanf("%d",&n2);
while(n2!=0)
{ n1=n2%10;
n2=n2/10;
printf("%d",n1);
}
}
```

程序运行后，如果从键盘上输入 1234，则输出结果为。

四、编程题

1．从键盘输入一批整数，最后一个数为 0，编程找出其中的最大数和最小数，并输出。

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。

3．求下列分数序列的前 20 项之和：2/1+3/2+5/3+8/5+13/8+...。

4．打印图形。

```
    1
   123
  12345
 1234567
123456789
```

5．两个乒乓球队进行比赛，各出 3 人。甲队为 A，B，C3 人，乙队为 X，Y，Z3 人。以抽签决定比赛名单。有人向队员打听比赛名单，A 说他不和 X 比，C 说他不和 X，Z 比，请编程找出 3 对选手的名单。

6．输入两个正整数 m 和 n，求其最大公约数和最小公倍数。