

# 第5章 循环结构程序设计

5.1 为什么需要循环控制

5.2 用**while**语句实现循环

5.3 用**do-while**语句实现循环

5.4 用**for** 语句实现循环

5.5 循环的嵌套

5.6 几种循环的比较

5.7 改变循环执行的状态

5.8 循环程序举例

## 5.7 改变循环执行的状态

5.7.1 用**break**语句提前终止循环

5.7.2 用**continue**语句提前结束本次循环

5.7.3 **break**语句和**continue**语句的区别

## 5.7.1 用break语句提前终止循环

- **break**语句可以用来从循环体内跳出循环体，即提前结束循环，接着执行循环下面的语句

## 5.7.1 用break语句提前终止循环

例**5.4** 在全系**1000**学生中，征集慈善募捐，当总数达到**10**万元时就结束，统计此时捐款的人数，以及平均每人捐款的数目。

## 5.7.1 用break语句提前终止循环

➤ 编程思路：

◆ 循环次数不确定，但最多循环**1000**次

- 在循环体中累计捐款总数

- 用**if**语句检查是否达到**10**万元

- 如果达到就不再继续执行循环，终止累加

◆ 计算人均捐款数

## 5.7.1 用break语句提前终止循环

➤ 编程思路：

- ◆ 变量**amount**，用来存放捐款数
- ◆ 变量**total**，用来存放累加后的总捐款数
- ◆ 变量**aver**，用来存放人均捐款数
- ◆ 定义符号常量**SUM**代表**100000**

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

```
#define SUM 100000
```

```
int main()           指定符号常量SUM代表100000
```

```
{ float amount,aver,total;  int i;
```

```
  for (i=1,total=0;i<=1000;i++)
```

```
  { printf("please enter amount:");
```

```
    scanf("%f",&amount);
```

```
    total= total+amount;
```

```
    if (total>=SUM) break;
```

```
  }
```

```
  aver=total / i;
```

```
  printf("\num=%d\naver=%10.2f\n"  
        ,i,aver);
```

```
  return 0;
```

```
}
```

```
#include <stdio.h>
#define SUM 100000
int main()
{ float amount,aver,total;  int i;
  for (i=1,total=0;i<=1000;i++)
  { printf("please enter amount:");
    scanf("%f",&amount); 应该执行1000次
    total= total+amount;
    if (total>=SUM) break;
  }
  aver=total / i;
  printf("\num=%d\naver=%10.2f\n"
        ,i,aver);

  return 0;
}
```



```
#include <stdio.h>
#define SUM 100000
int main()
{ float amount,aver,total;  int i;
  for (i=1,total=0;i<=1000;i++)
  { printf("please enter amount:");
    scanf("%f",&amount);
    total= total+amount;
    if (total>=SUM) break;
  }
  aver=total / i;
  printf("\num=%d\naver=%10.2f\n",i,aver);

  return 0;
}
```

达到10万，提前结束循环

```

#include <stdio.h>
#define SUM 100000
int main()
{
    float amount, aver, total;
    for (i=1, total=0; i<=SUM; i++)
    {
        printf("please enter amount:");
        scanf("%f", &amount);
        total = total + amount;
        if (total >= SUM) break;
    }
    aver = total / i;
    printf("num=%d\n", i);
    printf("aver=%10.2f\n", aver);

    return 0;
}

```

```

please enter amount:12000
please enter amount:24600
please enter amount:3200
please enter amount:5643
please enter amount:21900
please enter amount:12345
please enter amount:23000
num=7
aver= 14669.71

```

实际捐款人数

```

#include <stdio.h>
#define SUM 100000
int main()
{
    float amount, aver, total;    int i;
    for (i=1, total=0; i<=1000; i++)
    {
        printf("please enter amount:");
        scanf("%f", &amount);
        total = total + amount;
        if (total >= SUM) break;
    }
    aver = total / i;
    printf("num=%d \naver = %f\n", i, aver);

    return 0;
}

```

只能用于循环语句和switch  
语句之中，而不能单独使用

## 5.7.2 用**continue**语句提前结束本次循环

- 有时并不希望终止整个循环的操作，而只希望提前结束本次循环，而接着执行下次循环。这时可以用**continue**语句

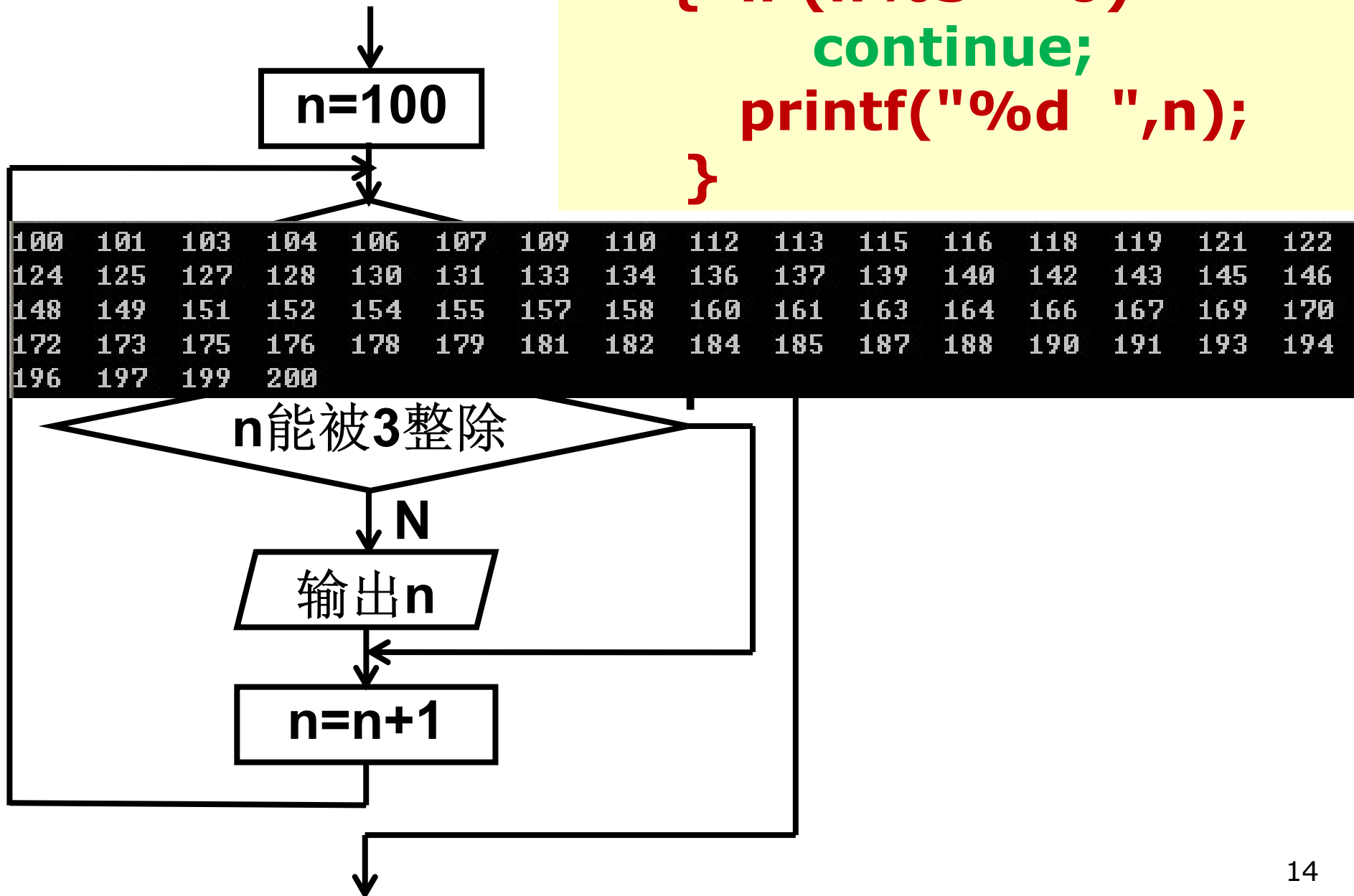
## 5.7.2 用continue语句提前结束本次循环

例**5.5** 要求输出**100~200**之间的不能被**3**整除的数。

➤编程思路：

- ◆对**100**到**200**之间的每一个整数进行检查
- ◆如果不能被**3**整除，输出，否则不输出
- ◆无论是否输出此数，都要接着检查下一个数(直到**200**为止)。

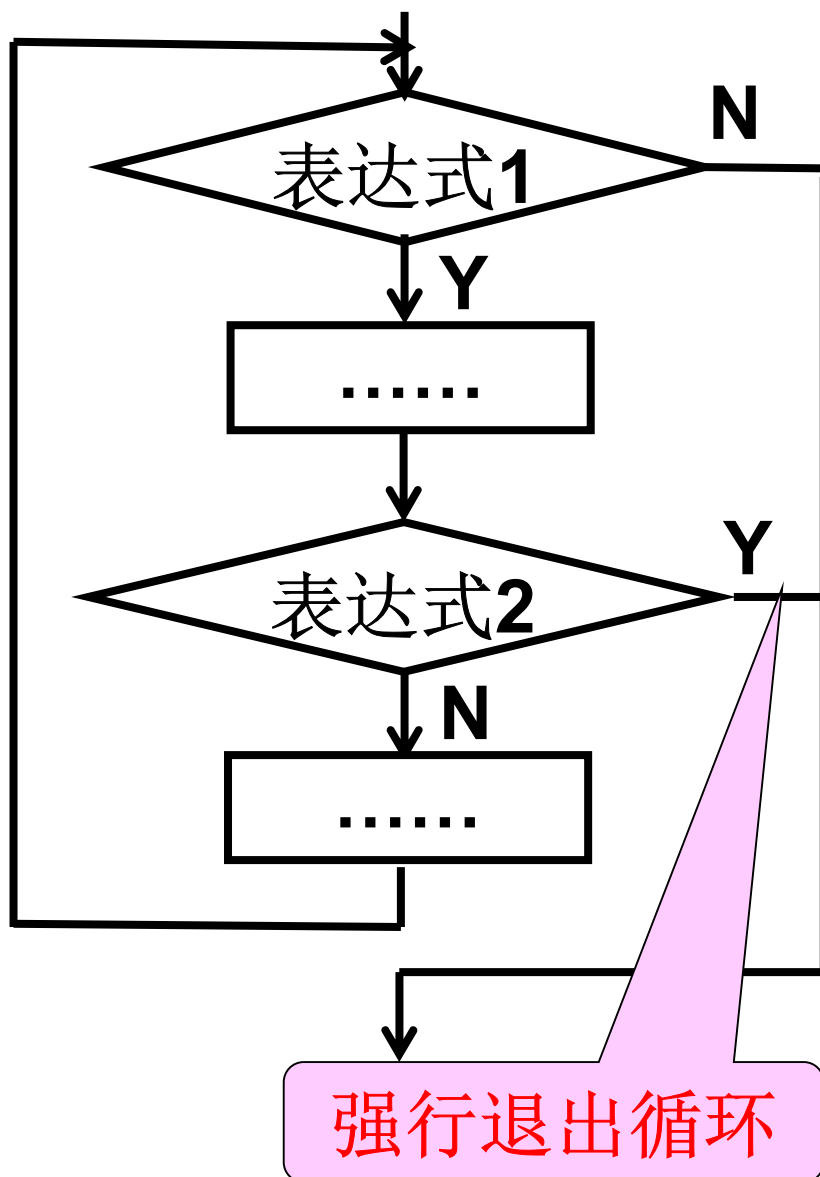
```
for(n=100;n<=200;n++)  
{ if (n%3==0)  
    continue;  
    printf("%d ",n);  
}
```



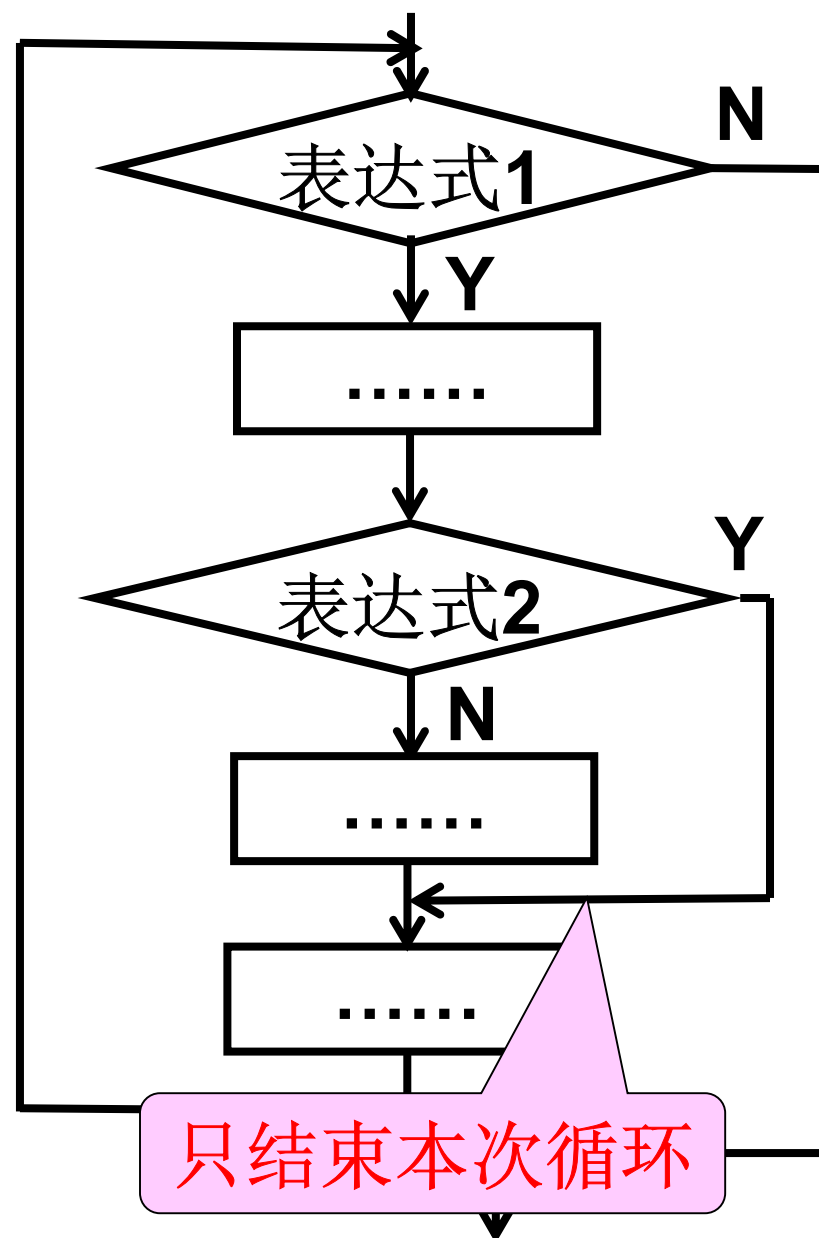
## 5.7.3 break语句和continue语句的区别

- **continue**语句只结束本次循环，而不是终止整个循环的执行
- **break**语句结束整个循环过程，不再判断执行循环的条件是否成立

## break语句



## continue语句





例**5.6** 输出以下**4\*5**的矩阵。

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>2</b>	<b>4</b>	<b>6</b>	<b>8</b>	<b>10</b>
<b>3</b>	<b>6</b>	<b>9</b>	<b>12</b>	<b>15</b>
<b>4</b>	<b>8</b>	<b>12</b>	<b>16</b>	<b>20</b>

➤ 解题思路:

- ◆ 可以用循环的嵌套来处理此问题
- ◆ 用外循环来输出一行数据
- ◆ 用内循环来输出一列数据
- ◆ 按矩阵的格式(每行**5**个数据)输出

```
#include <stdio.h>
int main()
{ int i,j,n=0;
  for (i=1;i<=4;i++) 累计输出数据的个数
    for (j=1;j<=5;j++,n++)
    { if (n%5==0) printf ("\n");
      printf ("%d\t",i*j);
    }
  printf ("\n");
  return 0;
}
```

控制一行内输出5个数据

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

```
int main()
```

```
{ int i,j,n=0;
```

双重循环

```
for (i=1;i<=4;i++)
```

```
for (j=1;j<=5;j++,n++)
```

```
{ if (n%5==0) printf ("\n");
```

```
printf ("%d\t",i*j);
```

```
}
```

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

```
return 0;
```

```
}
```

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

```
int main()
```

```
{ int i,j,n=0;
```

```
for (i=1;i<=4;i++) 控制输出4行
```

```
for (j=1;j<=5;j++,n++)
```

```
{ if (n%5==0) printf ("\n");
```

```
printf ("%d\t",i*j);
```

```
}
```

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

```
return 0;
```

```
}
```

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

```
int main()
```

```
{ int i,j,n=0;
```

控制每行中输出5个数据

```
for (i=1;i<=4;i++)
```

```
for (j=1;j<=5;j++,n++)
```

```
{ if (n%5==0) printf ("\n");
```

```
printf ("%d\t",i*j);
```

```
}
```

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

```
return 0;
```

```
}
```

```

#include <stdio.h>
int main()
{  int i,j,n=0;
    for (i=1;i<=4;i++)    i=1时
        for (j=1;j<=5;j++,n++)
            {  if (n%5==0) printf ("\n");
                printf ("%d\t",i*j);
            }
    printf("\n");    j由1变到5
                    i*j的值是1,2,3,4,5
    return 0;
}

```

```
#include
```

```
int main
```

```
{ int i,j,n=0;
```

```
for (i=1;i<=4;i++)
```

```
for (j=1;j<=5;j++,n++)
```

```
{ if (n%5==0) printf ("\n");
```

```
printf ("%d\t",i*j);
```

```
}
```

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

```
return 0;
```

```
}
```

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20

如何修改程序，不输出第一行的空行？

j也由1变到5

i\*j的值是2,4,6,8,10



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

```
int main()
```

```
{ int i,j,n=0;
```

```
for (i=1;i<=4;i++)
```

```
for (j=1;j<=5;j++,n++)
```

```
{ if (n%5==0) printf ("\n");
```

```
if (i==3 && j==1) break;
```

```
printf ("%d\t",i*j);
```

```
}
```

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

```
return 0;
```

```
}
```

1	2	3	4	5
2	4	6	8	10
4	8	12	16	20

遇到第3行第1列，  
终止内循环

```
#include
int main
```

1	2	3	4	5
2	4	6	8	10
6	9	12	15	
4	8	12	16	20

原来第3行第1个  
数据3没有输出

```
    0;
    i<=4;i++)
    for (j=1;j<=5;j++,n++)
    { if (n%5==0) printf ("\n");
      if (i==3 && j==1) continue;
      printf ("%d\t",i*j);
    }
    printf("\n");
    return 0;
}
```

## 5.8 循环程序举例

例**5.7**用  $\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots$  公式求  $\pi$  的近似值，直到发现某一项的绝对值小于  **$10^{-6}$**  为止(该项不累计加)。

## 5.8 循环程序举例

➤ 解题思路:

◆ 求 $\pi$  近似值的方法很多, 本题是一种

◆ 其他方法:

$$\pi \approx \frac{22}{7}$$

$$\frac{\pi^2}{6} \approx \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{n^2}$$

$$\frac{\pi}{2} = \frac{2 \times 2}{1 \times 3} \times \frac{4 \times 4}{3 \times 5} \times \frac{6 \times 6}{5 \times 7} \times \cdots \times \frac{(n-1)^2}{n \times (n+2)}$$

## 5.8 循环程序举例

$$\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

- 每项的分子都是**1**
- 后一项的分母是前一项的分母加**2**
- 第**1**项的符号为正，从第**2**项起，每一项的符号与前一项的符号相反

$$\frac{1}{n} \quad \Rightarrow \quad -\frac{1}{n+2}$$

## 5.8 循环程序举例

**sign=1, pi=0, n=1, term=1**

**当 term  $\geq 10^{-6}$**

**pi=pi+term**

**n=n+1**

**sign=-sign**

**term=sign/n**

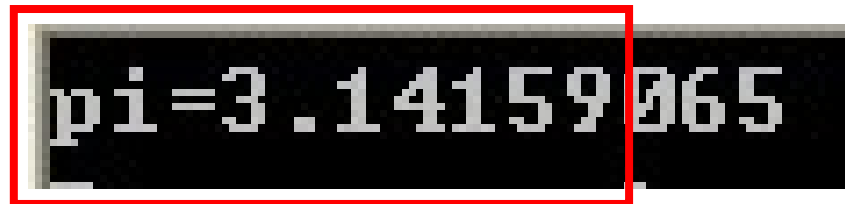
**pi=pi\*4**

**输出 pi**

```

#include <stdio.h>
#include <math.h>
int main()
{ int sign=1; double pi=0,n=1,term=1;
  while(fabs(term)>=1e-6)
  { pi=pi+term;      求绝对值的函数
    n=n+2;
    sign=-sign;
    term=sign/n;
  }
  pi=pi*4;          只保证前5位小数是准确的
  printf("pi=%10.8f\n",pi);
  return 0;
}

```



```
#include <stdio.h>
#include <math.h>
int main()
{ int sign=1; double pi=0,n=1,term=1;
  while(fabs(term)>=1e-6) 改为1e-8
  { pi=pi+term;
    n=n+2;
    sign=-sign;
    term=sign/n;
  }
  pi=pi*4;
  printf("pi=%10.8f\n",pi);
  return 0;
}
```

A terminal window with a black background and white text showing the output of the program with the original 1e-6 tolerance: pi=3.14159065.A terminal window with a black background and white text showing the output of the program with the modified 1e-8 tolerance: pi=3.14159263.



**例5.8** 求费波那西(**Fibonacci**)数列的前**40**个数。这个数列有如下特点：第**1**、**2**两个数为**1**、**1**。从第**3**个数开始，该数是其前面两个数之和。即：

$$\begin{cases} F_1 = 1 & (n = 1) \\ F_2 = 1 & (n = 2) \\ F_n = F_{n-1} + F_{n-2} & (n \geq 3) \end{cases}$$

➤这是一个有趣的古典数学问题：

◆有一对兔子，从出生后第**3**个月起每个月都生一对兔子。

◆小兔子长到第**3**个月后每个月又生一对兔子。

◆假设所有兔子都不死，问每个月的兔子总数为多少？

第几个月	小兔子对数	中兔子对数	老兔子对数	兔子总数
1	1	0	0	1
2	0	1	0	1
3	1	0	1	2
4	1	1	1	3
5	2	1	2	5
6	3	2	3	8
7	5	3	5	13
⋮	⋮	⋮	⋮	⋮

**f1=1,f2=1**

**输出f1,f2**

**For i=1 to 38**

**f3=f1+f2**

**输出f3**

**f1=f2**

**f2=f3**

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

```
int main()
```

```
{ int f1=1,f2=1,f3; int i;
```

```
  printf("%12d\n%12d\n",f1,f2);
```

```
  for(i=1; i<=38; i++)
```

```
  { f3=f1+f2;
```

```
    printf("%12d\n",f3);
```

```
    f1=f2;
```

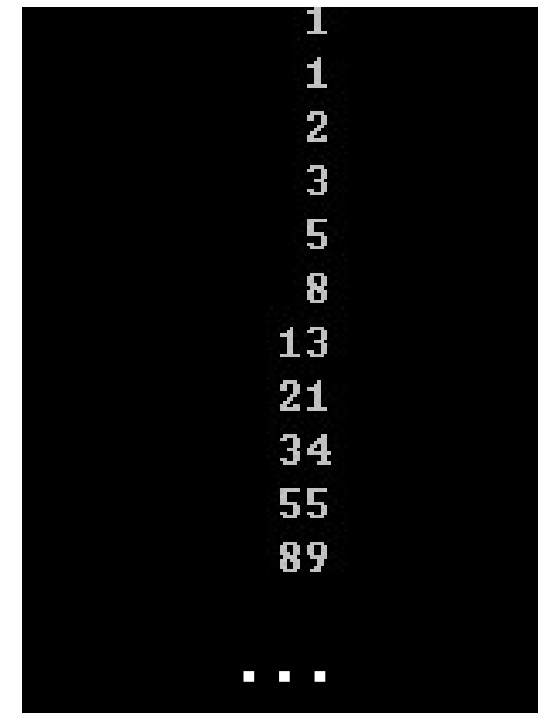
```
    f2=f3;
```

```
  }
```

```
  return 0;
```

```
}
```

代码可改进



```
1
1
2
3
5
8
13
21
34
55
89
...
```

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

```
int main()
```

```
{ int f1=1,f2=1; int i;
```

```
  for(i=1; i<=20; i++)
```

```
  { printf("%12d %12d ",f1,f2);
```

```
    if(i%2==0) printf("\n");
```

1	1	2	3
5	8	13	21
34	55	89	144
233	377	610	987
1597	2584	4181	6765
10946	17711	28657	46368
75025	121393	196418	317811
514229	832040	1346269	2178309
3524578	5702887	9227465	14930352
24157817	39088169	63245986	102334155

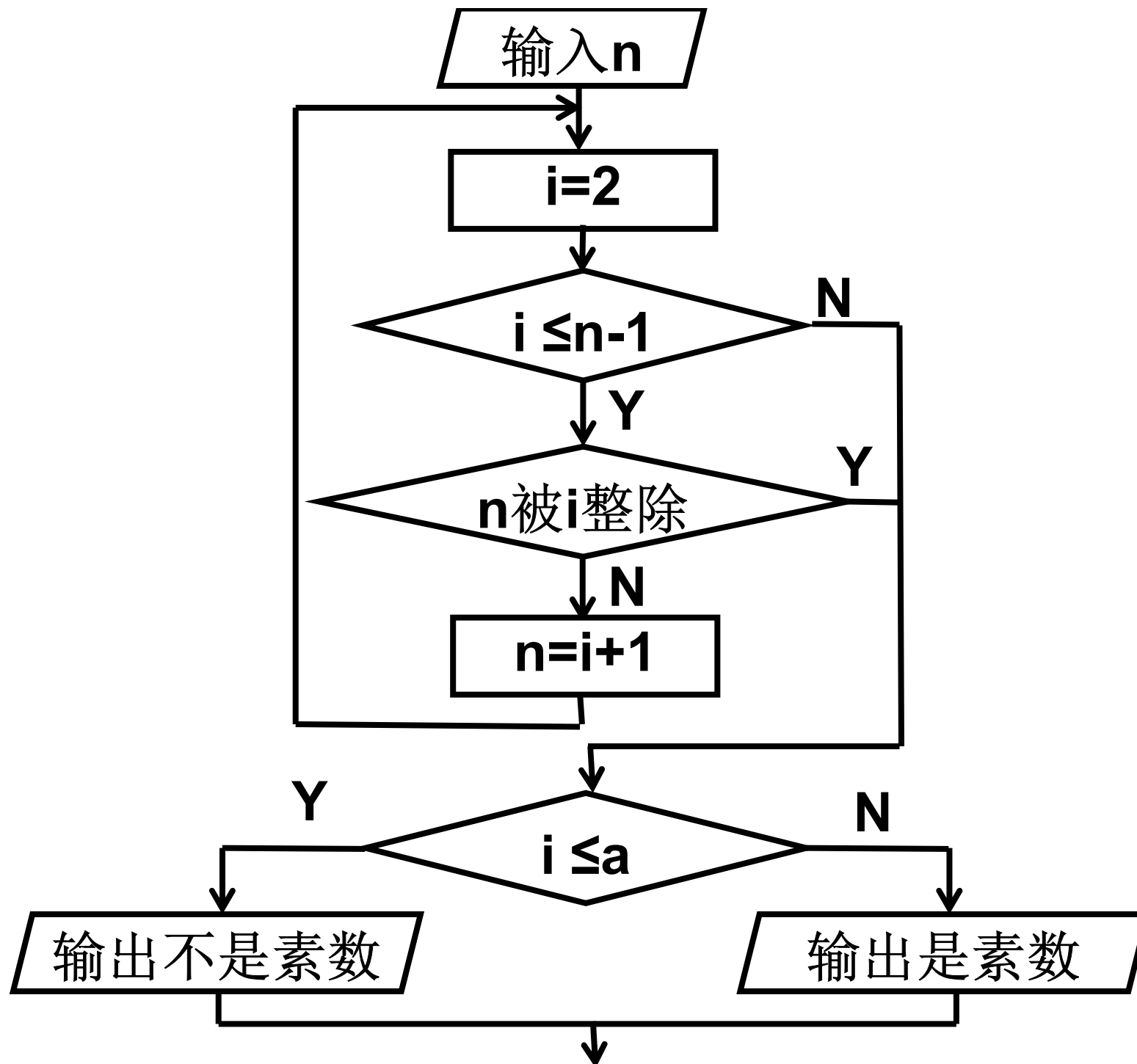
例**5.9**输入一个大于**3**的整数**n**，判定它是否素数(**prime**，又称质数)。

➤解题思路：

◆让**n**被**i**整除(**i**的值从**2**变到**n-1**)

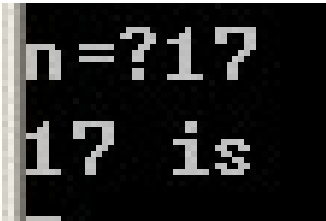
◆如果**n**能被**2~(n-1)**之中任何一个整数整除，则表示**n**肯定不是素数，不必再继续被后面的整数除，因此，可以提前结束循环

◆注意：此时**i**的值必然小于**n**





```
#include <stdio.h>  
int main()  
{ int n,i;  
    printf("n=?"); scanf("%d",&n);  
    for (i=2;i<=n-1;i++)  
        if(n%i==0) break;  
    if(i<n) printf("%d is not\n",n);  
    else printf("%d is\n",n);  
    return 0;  
}
```



A terminal window with a black background and white text. The first line shows the prompt 'n=?' followed by the input '17'. The second line shows the output '17 is'.



A terminal window with a black background and white text. The first line shows the prompt 'n=?' followed by the input '327'. The second line shows the output '327 is not'.

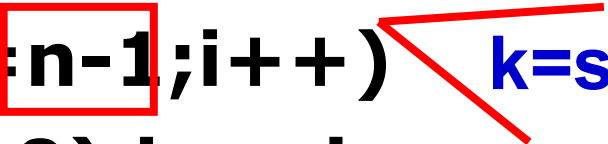
```

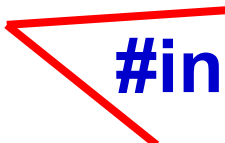
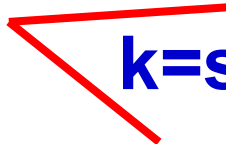
#include <stdio.h>
int main()
{
    int n,i;
    printf("n=?"); scanf("%d",&n);
    for (i=2;i<=n-1;i++)
        if(n%i==0) break;
    if(i<n) printf("%d is not\n",n);
    else printf("%d is\n",n);
    return 0;
}

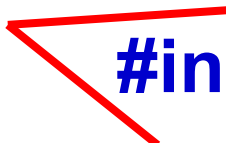
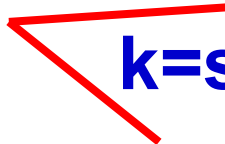
```

$\sqrt{n}$

$k=\text{sqrt}(n);$



```
#include <stdio.h>
int main()  #include <math.h>
{ int n,i,k;
  printf("n=?"); scanf("%d",&n);
  for (i=2; i<=k; i++)  k=sqrt(n);
    if(n%i==0) break;
  if(i<n) printf("%d is not\n",n);
  else printf("%d is\n",n);
  return 0;
}
```

```
#include <stdio.h>
int main()  #include <math.h>
{ int n,i,k;
  printf("n=?"); scanf("%d",&n);
  for (i=2; i<=k; i++)  k=sqrt(n);
    if(n%i==0) break;
  if(i<=k) printf("%d is not\n",n);
  else printf("%d is\n",n);
  return 0;
}
```

**例5.10 求100~200间的全部素数。**

➤ **解题思路：**

◆ **使用例5.9的算法**

◆ **在例5.9程序中只要增加一个外循环，先后对100~200间的全部整数一一进行判定即可**

.....

```
for(n=101;n<=200;n=n+2)
```

```
{ k=sqrt(n);
```

```
  for (i=2;i<=k;i++)
```

只对奇数进行检查

```
    if (n%i==0) break;
```

```
  if (i>=k+1)
```

```
  { printf("%d ",n);
```

```
    m=m+1;
```

控制每行输出10个数据

```
  }
```

```
  if(m%10==0) printf("\n");
```

```
}
```

.....

例**5.11** 译密码。为使电文保密，往往按一定规律将其转换成密码，收报人再按约定的规律将其译回原文。



- 非字母字符保持原状不变
- 输入一行字符，要求输出其相应的密码

➤ 解题思路：问题的关键有两个：

(1) 决定哪些字符不需要改变，哪些字符需要改变，如果需要改变，`c=getchar();`

◆ 处理的方法是：输入一个字符给字符变量c，先判定它是否字母(包括大小写)，若不是字母，不改变c的值；若是字母，则还要检查它是否'**W**'到'**Z**'的范围内(包括大小写字母)。如不在此范围内，则使变量c的值改变为其后第**4**个字母。如果在'**W**'到'**Z**'的范围内，则应将它转换为**A~D**(或**a~d**)之一的字母。



➤ 解题思路：问题的关键有两个：

(1) 决定哪些字符不需要改变，哪些字符需要改变，如果需要改变，应改为哪个字符

```
if((c>='a' && c<='z') || (c>='A' && c<='Z'))
```

先判定它是否字母(包括大小写)，若不是字母，不改变**c**的值；若是字母，则还要检查它是否'**W**'到'**Z**'的范围内(包括大小写字母)。如不在此范围内，则使变量**c**的值改变为其后第**4**个字母。如果在'**W**'到'**Z**'的范围内，则应将它转换为**A~D**(或**a~d**)之一的字母。

➤ 解题思路：问题的关键有两个：

(1) 决定哪些字符不需要改变，哪些字符需要改变，如果需要改变，应改为哪个字符

```
if(c>='W' && c<='Z' || c>='w' && c<='z')  
    c=c+4-26;  
else c=c+4;
```

否 'W'到'Z'的范围内 (包括大小写字母)。如不在此范围内，则使变量**c**的值改变为其后第**4**个字母。如果在**'W'到'Z'**的范围内，则应将它转换为**A~D(或a~d)**之一的字母。

➤ 解题思路： 问题的关键有两个：

(2) 怎样使**c**改变为所指定的字母？

◆ 办法是改变它的**ASCII**值

◆ 例如字符变量**c**的原值是大写字母'**A**'，想使**c**的值改变为'**E**'，只需执行“**c=c+4**”即可，因为'**A**'的**ASCII**值为**65**，而'**E**'的**ASCII**值为**69**，二者相差**4**

```
char c;  
c=getchar();  
while(c!='\n')  
{ if((c>='a' && c<='z') || (c>='A' &&  
                                     c<='Z'))  
    { if(c>='W' && c<='Z' || c>='w' &&  
                                             c<='z')  
        c=c-22;  
      else c=c+4;  
    }  
    printf("%c",c);  
    c=getchar();  
}
```

可以改进程序



```
China!  
Glmre!
```

```
char c;
while((c=getchar())!='\n')
{ if((c>='A' && c<='Z') || (c>='a' &&
                                c<='z'))
    { c=c+4;
      if(c>='Z' && c<='Z'+4 || c>'z')
          c=c-26;      不能少
    }
    printf("%c",c);
}
```