



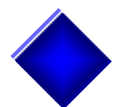
C程序设计案例教程

▶ 单元6： 循环结构程序设计

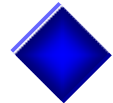




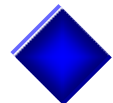
第6章 循环控制



概述



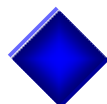
goto语句以及用goto语句构成循环



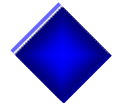
while语句



do-while语句



循环的嵌套



for语句



几种循环的比较



break语句和continue语句



程序举例



§ 6.1 概述

程序中凡涉及求阶乘、累加、排序等问题都要用循环解决，因为程序中的某一程序段要重复执行若干次。

C语言实现循环的语句：

- ❖ goto 和if组合
- ❖ while语句
- ❖ do while语句
- ❖ for语句





§ 6.2 goto语句以及用goto语句构成循环

❖作用：无条件地转移到语句标号指向的语句去执行。

❖一般格式：

```
goto 语句标号;  
.....  
语句标号: 语句;
```

❖说明：

- 语句标号的命名规则同标识符的一样。

`goto loop;` (✓) `goto 100;` (✗)

- 与循环语句配合使用时，可从循环体内跳到循环体外，而不能从循环体外跳到循环体内。
- goto语句转移的方向可前可后。
- 与if语句配合使用时可构成循环。
- goto 语句违背结构化程序设计的原则，因此应限制使用。



例6.1 用if 和goto语句构成循环，求

$$\sum_{n=1}^{100} n$$

```
#include <stdio.h>
int main( )
{  int i,sum=0;
   i=1;
loop: if(i<=100)
    {
      sum+=i;
      i++;
      goto loop;
    }
printf("%d",sum);
}
```

循环初值

循环终值

循环条件

循环变量增值

循环体





例 从键盘输入一组数据，以0结束输入，求数据和。

```
#include <stdio.h>
int main()
{
    int number,sum=0;
    read_loop: scanf("%d",&number);
                if(!number) goto print_sum;
                sum+=number;
                goto read_loop;
print_sum: printf("The total sum is %d\n",sum);
}
```



§ 6.3 while语句

while语句实现“当型”循环结构。

没有“;”

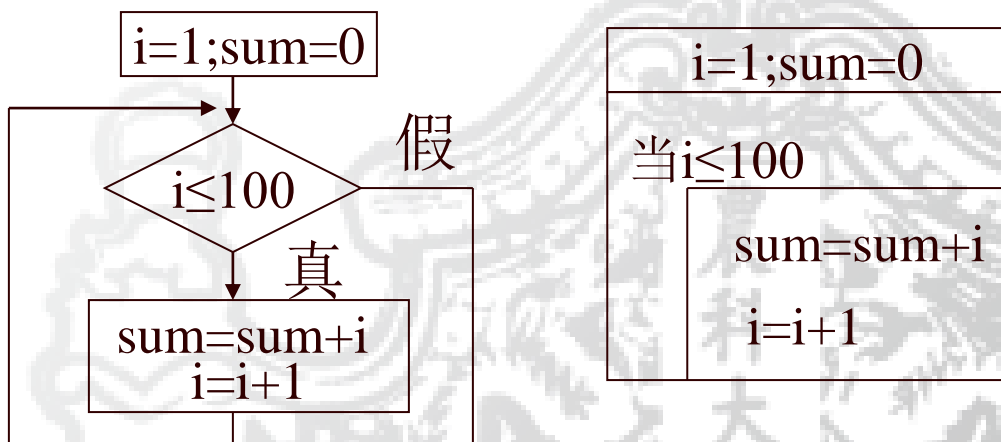
❖ 一般形式:

```
while(表达式)  
    循环体语句;
```

❖ 功能: 先判断表达式, 若为真则执行循环体, 再判断表达式, 重复上述过程, 直到表达式为假时退出循环。

❖ 求

$$\sum_{n=1}^{100} n$$





例6.2 用while语句构成循环，求

$$\sum_{n=1}^{100} n$$

```
#include <stdio.h>
int main( )
{
    int i,sum=0;
    i=1;
    while(i<=100)
    {
        sum=sum+i;
        i++;
    }
    printf("%d", sum);
}
```

循环初值

循环终值

循环条件

循环体

循环变量增值

農科大學



❖ 关于while循环语句：

- while循环先判断表达式，后执行循环体。循环体有可能一次也不执行。
- 循环体若包含一个以上语句，应该用{}括起来。
- 循环体应包含有使循环趋向结束的语句；
- 下列情况，退出while循环
 - ◆ 条件表达式不成立（为零）
 - ◆ 循环体内遇 break , goto
- 无限循环

```
while(1)
```

```
    循环体；
```

```
    程序设计I
```





while语句例子

```
▶ 例：  
▶ #include <stdio.h>  
▶ int main()  
▶ {  
▶     int n=0;  
▶     while (n<=2)  
▶     {  
▶         n++;  
▶         printf("%d", n);  
▶     }  
▶     return 0;  
▶ }
```

▶ 输出结果： 123

【讨论】

- 1) 若去while语句中的 { } , 结果： 3;
- 2) 若将n++移到printf语句中，结果： 012



例 用while语句显示1~10的平方

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

```
1*1=1
2*2=4
3*3=9
4*4=16
5*5=25
6*6=36
7*7=49
8*8=64
9*9=81
10*10=100
```



§ 6.4 do-while语句

❖ do-while语句实现“当型”循环结构。

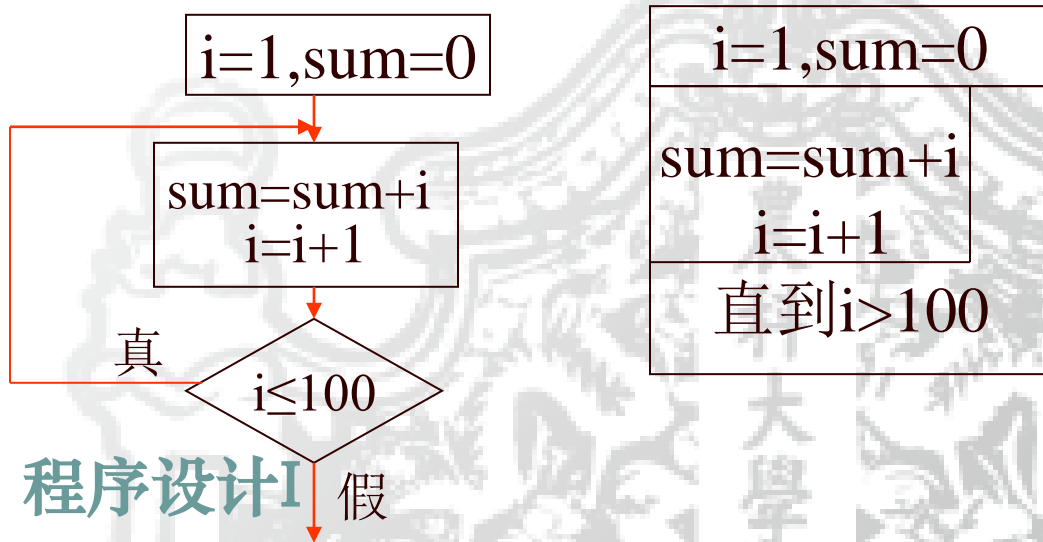
❖ 一般形式：

```
do  
循环体语句;  
while(表达式);
```

有“;”

❖ 功能：先执行循环体，然后判断表达式。若为真，则再次执行循环体，否则退出循环。

❖ 求 $\sum_{n=1}^{100} n$





例6.3 用do-while语句构成循环，求

$$\sum_{n=i}^{100} n$$

```
#include <stdio.h>
int main( )
{ int i,sum=0;
  scanf("%d",&i);
  do
  { sum+=i;
    i++;
  }
  while(i<=100);
  printf("%d",sum);
}
```

输入: 1
输出: 5050 5050

输入: 101
输出: 101 0

循环体

循环条件

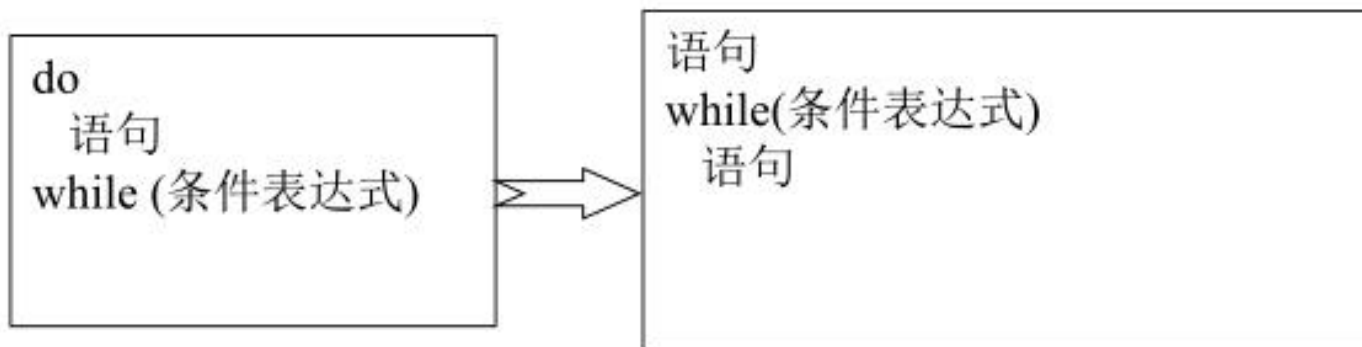
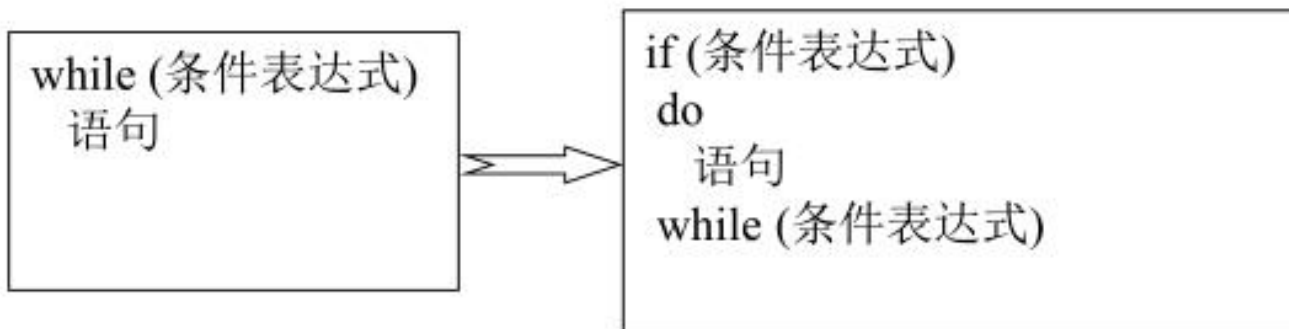
```
#include <stdio.h>
int main( )
{int i,sum=0;
  scanf("%d",&i);
  while(i<=100)
  { sum+=i;
    i++;
  }
  printf("%d",sum);
}
```

比较

- do-while 循环，循环体至少执行一次；
- while和do-while可以解决同一问题，两者可以互换。
- while后的表达式一开始就为假时，两种循环结果不同。



do-while与while比较





- ▶ 例6.4 从键盘输入一个整数，分析以下程序运行结果。
- ▶ 思路：若X为整数，则 $X\%10$ 可得X的个位数， $X/10$ 可得X的个位数移去后形成的新数。
- ▶ 如：
 $125\%10=5 \rightarrow 125/10=12 \rightarrow 12\%10=2 \rightarrow 12/10=1 \rightarrow 1\%10=1 \rightarrow 1/10=0$

- ▶ `#include <stdio.h>`
- ▶ `int main()`
- ▶ `{ int num,c;`
- ▶ `printf("请输入一个整数：");`
- ▶ `scanf("%d",&num);`
- ▶ `do`
- ▶ `{ c=num%10; /*取得num的个位数*/`
- ▶ `printf("%d",c); /*输出num的个位数*/`
- ▶ `}`
- ▶ `while((num/=10)>0); /*直到num/10为0*/`
- ▶ `printf("\n");`
- ▶ `return 0;}`

【讨论】 如果`while(num/=10>0)`，
结果如何？

陷入无限循环（输出无数个6）

```
do
{ c=num%10;
printf("%d",c);
num/=10;
}
while(num>0);
```

- ▶ 输出结果：输入12456，输出65421（将各位数字反序显示出来）



§ 6.5 for语句

for 语句是 C 语言中**最为灵活**，**使用最广泛**的循环语句，可完全替代while，do-while语句。

❖ 一般形式

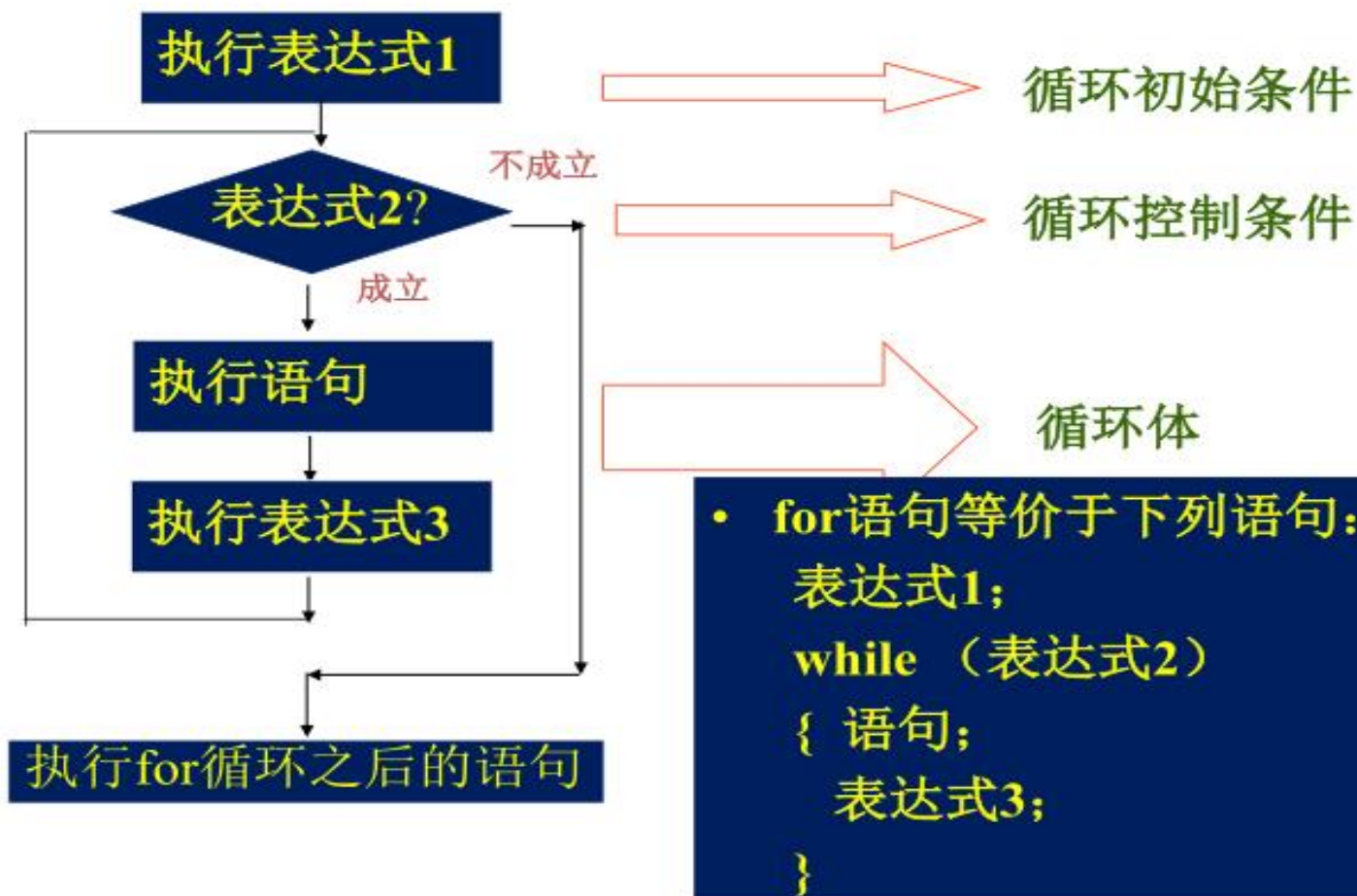
```
for(表达式1; 表达式2; 表达式3)  
    循环体语句;
```

❖ 常用形式

```
for(循环变量赋初值; 循环条件; 循环变量增值)  
    循环体语句;
```

★ 几点说明:

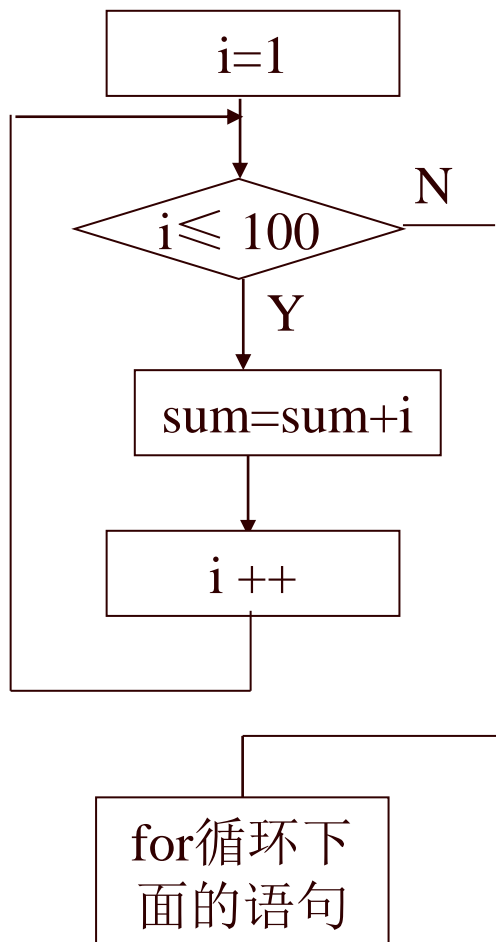
- ❖ for语句中表达式1、2、3类型任意，都可省略，但分号“;”不能省。
- ❖ 无限循环：for(;;)不断执行循环体，循环不终止。





例 用for语句构成循环，求

$$\sum_{n=1}^{100} n$$



```
#include <stdio.h>
int main( )
{
    int i,sum=0;
    for(i=1;i<=100;i++)
        sum+=i;
    printf("%d",sum);
}
```

循环步长



❖ 几种形式:

- 省略表达式1: 应在for之前为变量赋初值。
- 省略表达式2: 循环条件始终为“真”，循环不终止。一般是关系表达式(如 $i \leq 100$)或逻辑表达式(如 $a < b \ \&\& \ x < y$), 但也可以是数值表达式或字符表达式
- 省略表达式3: 应另外设法使程序能够结束。
- 省略表达式1、3: 完全等同于while语句。
- 三个表达式都省略: 无初值, 不判断条件, 循环变量不增值, 死循环。

```
/*省略表达式1*/  
main( )  
{ int i,sum=0;  
  i=1;  
  for(;i<=100;i++);  
    sum=sum+i;  
  printf("%d",sum);  
}
```

```
/*正常形式*/  
int main( )  
{ int i ,sum=0;  
  for(i=1;i<=100;i++)  
    sum=sum+i;  
  printf("%d",sum);  
}
```



❖ 几种形式:

- 省略表达式1: 应在for之前为变量赋初值。
- 省略表达式2: 循环条件始终为“真”，循环不终止。一般是关系表达式(如 $i \leq 100$)或逻辑表达式(如 $a < b \ \&\& \ x < y$), 但也可以是数值表达式或字符表达式
- 省略表达式3: 应另外设法使程序能够结束。
- 省略表达式1、3: 完全等同于while语句。
- 三个表达式都省略: 无初值, 不判断条件, 循环变量不增值, 死循环。

```
/*省略表达式3*/  
main( )  
{ int i,sum=0;  
  for(i=1;i<=100;)  
    {sum=sum+i;  
      i++;}  
  printf("%d",sum);  
}
```

```
/*正常形式*/  
int main( )  
{ int i ,sum=0;  
  for(i=1;i<=100;i++)  
    sum=sum+i;  
  printf("%d",sum);  
}
```



❖ 几种形式:

- 省略表达式1: 应在for之前为变量赋初值。
- 省略表达式2: 循环条件始终为“真”，循环不终止。一般是关系表达式(如 $i \leq 100$)或逻辑表达式(如 $a < b \ \&\& \ x < y$), 但也可以是数值表达式或字符表达式
- 省略表达式3: 应另外设法使程序能够结束。
- 省略表达式1、3: 完全等同于while语句。
- 三个表达式都省略: 无初值, 不判断条件, 循环变量不增值, 死循环。

```
/*省略表达式1、3*/  
int main( )  
{ int i,sum=0;  
  i=1;  
  for(;;i<=100;)  
    {sum=sum+i;  
     i++;}  
  printf("%d",sum);  
}
```

```
/*正常形式*/  
int main( )  
{ int i ,sum=0;  
  for(i=1;i<=100;i++)  
    sum=sum+i;  
  printf("%d",sum);  
}
```



- 表达式1、3可以是与循环无关的表达式，也可以是逗号表达式。

```
for ( s=0 , i=1 ; i<=100 ; i++ ) s=s+i;
```

- 表达式2可以是关系、逻辑、算术、字符表达式，非0时，执行循环体，为0时退出循环。

```
/*表达式是逗号表达式*/  
int main()  
{ int i,j,k;  
  for(i=0,j=100;i<=j;i++,j--)  
  { k=i+j;  
    printf("%d+%d=%d\n",i,j,k);  
  }  
}
```



```
#include<stdio.h>
int main()
{ char c;
  for(;(c=getchar())!='\n');)
    printf("%c ",c);
}
```

运行情况：

Computer (输入)

Computer (输出)

而不是：

Ccoommpppuutteerr

注意：

getchar()函数的作用是从计算机终端（一般为键盘）获取一个无符号字符。从终端键盘向计算机输入时，是在按 Enter 键以后才将一批数据一起送到内存缓冲区中去的。Enter 键也是一个字符 ‘\n’，它也会连同输入的内容一起放到缓存区，也就是说输入 ‘Computer’ 并回车后，缓存区中就有了 ‘Computer\n’。

在本例中，省略表达式1、3完全等同于while语句，即在每次循环中，程序读取一个缓冲区中的字符并输出，直到读取到 ‘\n’，循环终止。



```
#include<stdio.h>
int main()
{ char c;
  for(;(c=getchar())!='\n');
    printf("%c ",c);
}
```

输入 ‘Computer’ 并回车后，缓存区中就有了

‘Computer\n’

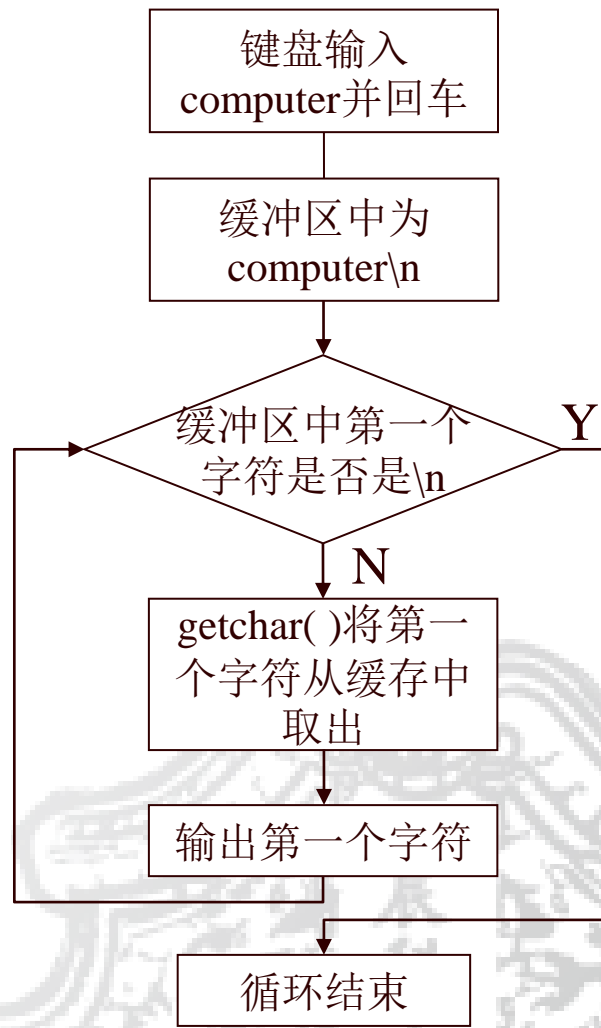
第一次循环getchar()取出 ‘c’ 并输出；

第二次循环getchar()取出 ‘o’ 并输出；

.....

第八次循环getchar()取出 ‘r’ 并输出；

第九次循环getchar()取出的是 ‘\n’ ，循环结束





▶ **【例一】** 以下程序中执行几次循环？

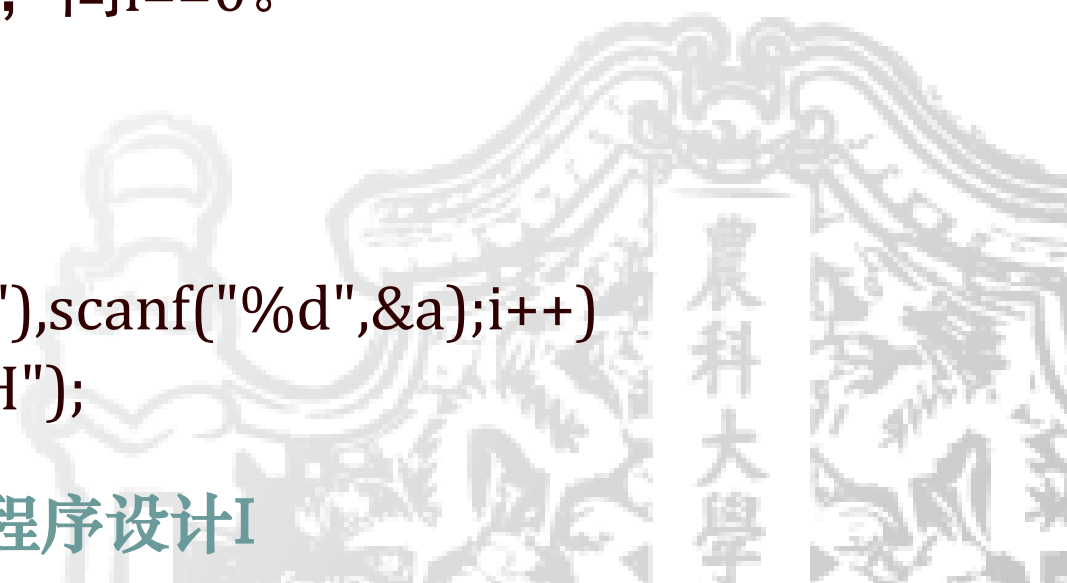
```
▶ int main()  
▶ { int i;  
▶   for (i=2;i==0; )  
▶     printf("%d", i++);  
▶ }
```

▶ **【讨论】** 若 $i==2$ ，结果？，循环执行？次。

▶ i 为其他值时，同 $i==0$ 。

▶ **【例二】**

```
▶ int main()  
▶ { int a,i;  
▶   for(i=0;printf("a="),scanf("%d",&a);i++)  
▶     printf("H");  
▶ }
```





§ 6.6 循环的嵌套

- ❖ 定义：一个循环体内又包含了另一个完整的循环结构，称为循环的嵌套。
- ❖ 三种循环可以互相嵌套，层数不限。
- ❖ 嵌套循环的执行流程。

```
(1) while()
    { .....
      while()
      { .....
        }
      .....
    }
```

```
(2) do
    { .....
      do
      { .....
        }
      while();
    .....
  }
```

```
(3) for(;;)
    { .....
      for(;;)
      { .....
        }
    }
```

```
(4) while()
    { .....
      do
      { .....
        }
      while();
    .....
  }
```

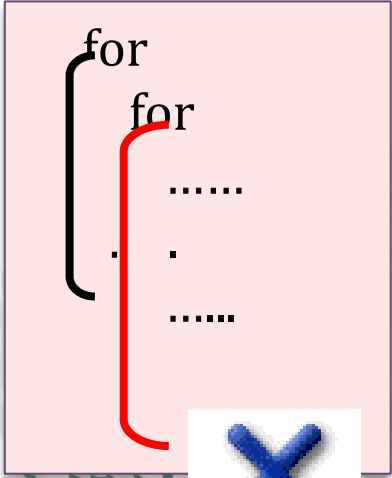
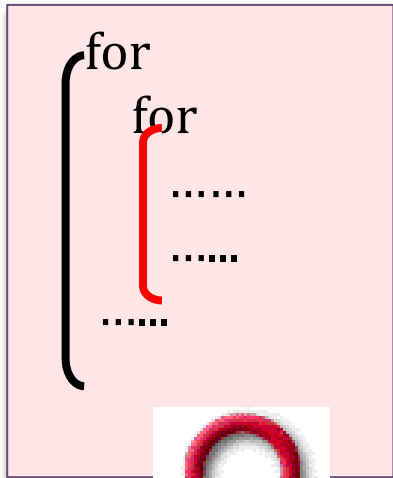


```
(5)
for(;;)
{ .....
  while()
  { .....
  }
  .....
}
```

```
(6)
do
{ .....
  for(;;)
  { .....
  }
  .....
}
while();
```

```
(7)
for(;;)
{ .....
  do
  { .....
  }
  while();
  .....
  while()
  { .....
  }
  .....
}
```

❖ 循环可以互相嵌套，但不能相互交叉。





例 计算3组1~4累加的和

```
#include <stdio.h>
int main()
{ int i,j,s=0;
  for(i=1;i<=3;i++)
    for(j=1;j<=4;j++)
      s=s+j;
  printf("s=%d\n",s);
  return 0;
}
```

结果: s=30

s=s+j; 执行多少次?

i	j	s=0
1	1	1
1	2	3
1	3	6
1	4	10
2	1	11
2	2	13
2	3	16
2	4	20
3	1	21
3	2	23
3	3	26
3	4	30



例 循环嵌套，输出九九乘法表

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
.....									
9	9	18	27	36	45	54	63	72	81



例 循环嵌套，输出九九乘法表

```
int 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++)
      printf((j==9)?"%4d\n":"%4d",i*j);
}
```

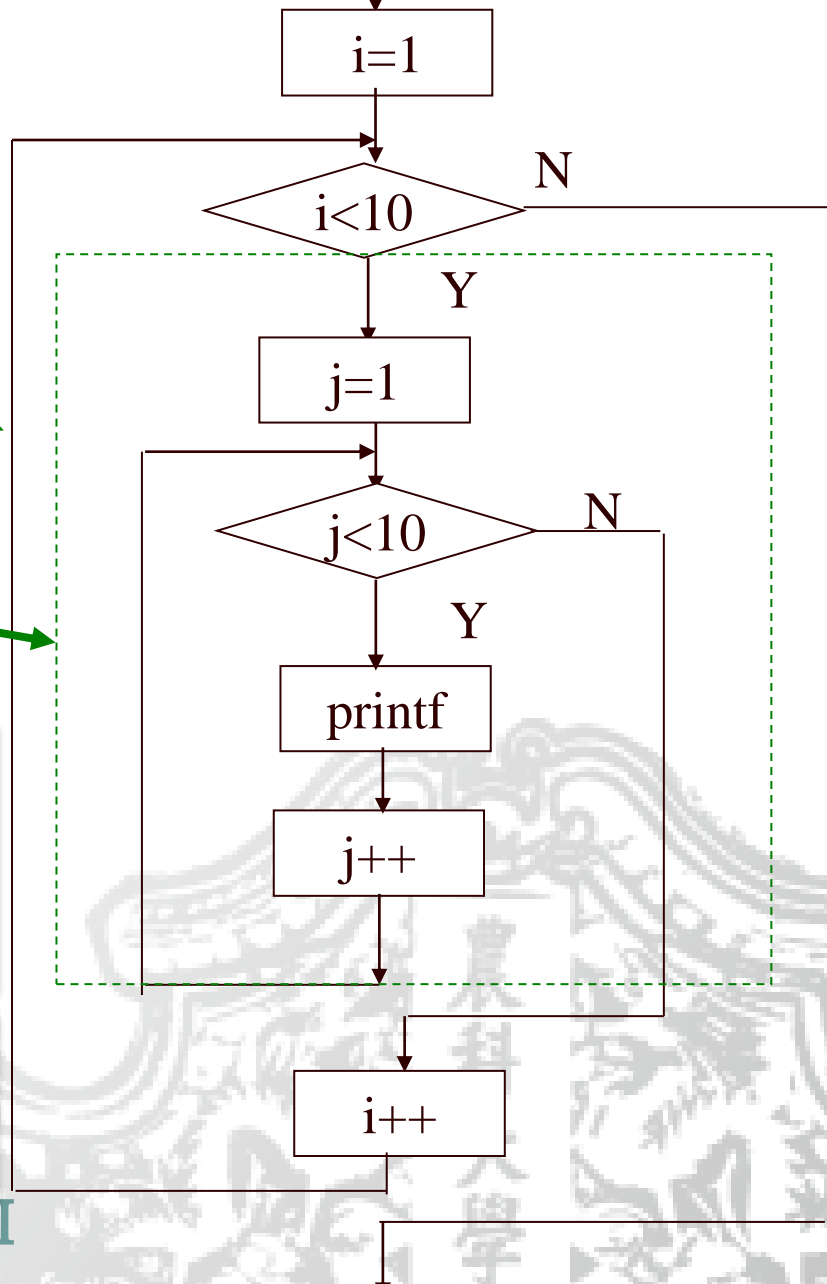
条件运算符：
j=9，输出i*j，再回车换行。
j≠9，只输出i*j。



❖ 九九乘法表逻辑框图

外循环

内循环



```
for(i=1;i<10;i++)  
  for(j=1;j<10;j++)  
    printf((j==9)?"%4d\n":"%4d",i*j);
```

```
printf((j==9)?"%4d\n":"%4d",i*j);  
执行多少次？
```



§ 6.7 几种循环的比较

语句	适用范围	说明
if~goto	无条件循环	该循环不能用 break 和 continue 语句控制
while	只知道结束条件而无法确定执行次数的情况下。	① 循环变量初始化在循环体之前 ② 循环体中应包含使循环结束的语句
do-while	至少执行一次时	③ 可以用 break 和 continue 语句控制
for	已知执行次数或者已知初值，终值，步长时。	使用比较灵活



§ 6.8 break语句和continue语句

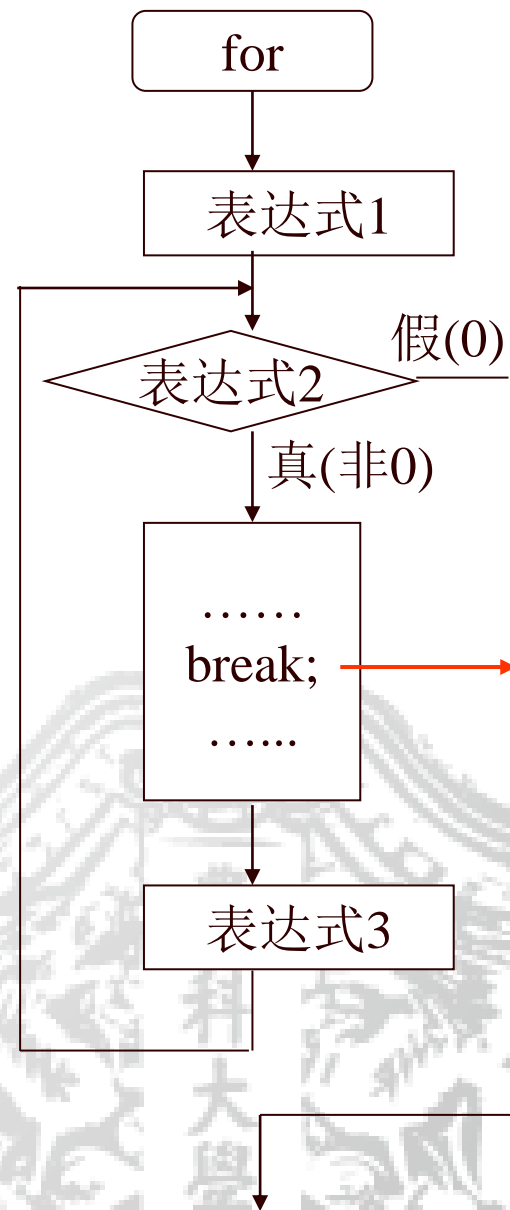
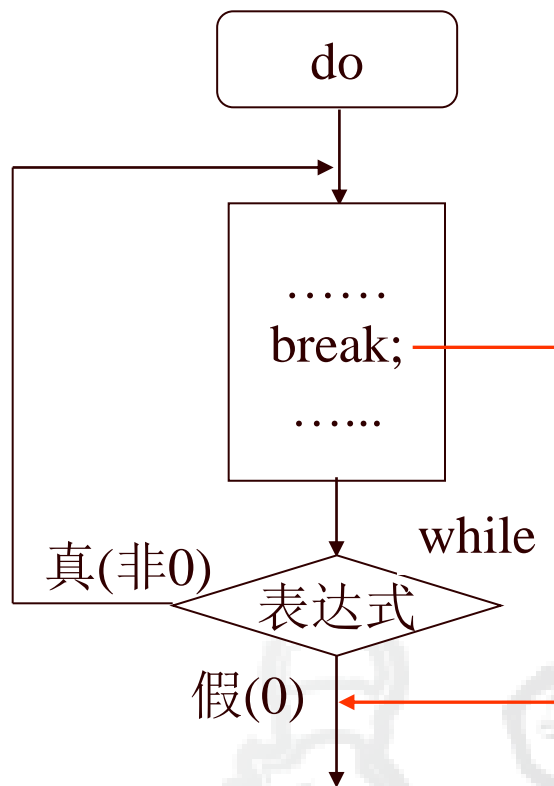
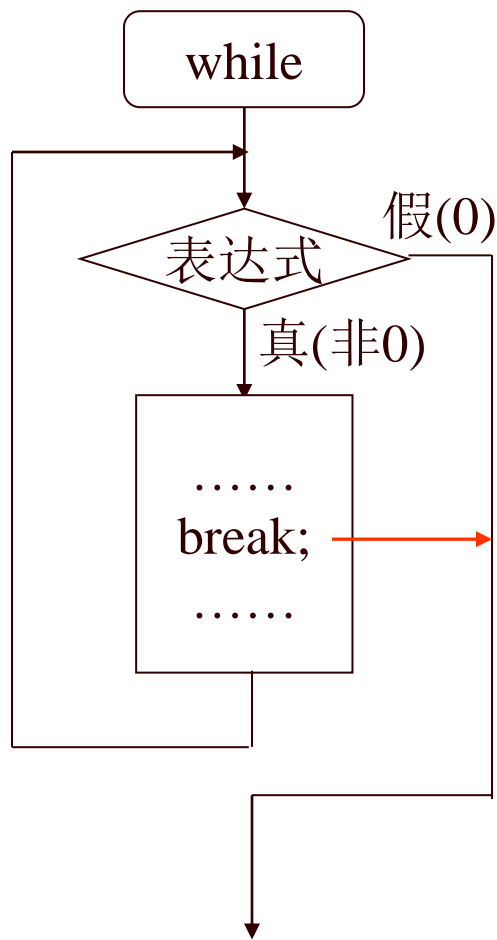
★break语句

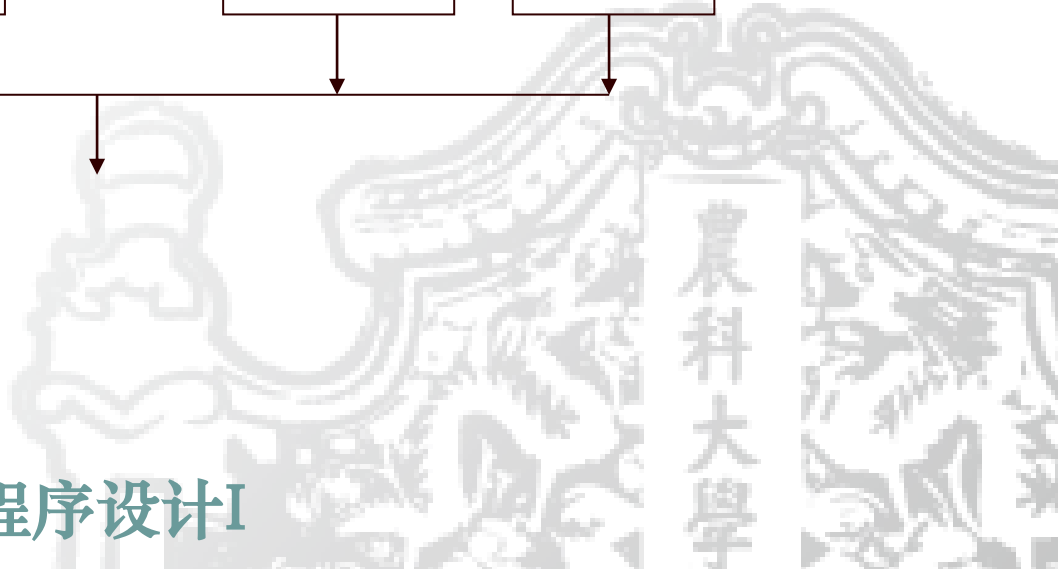
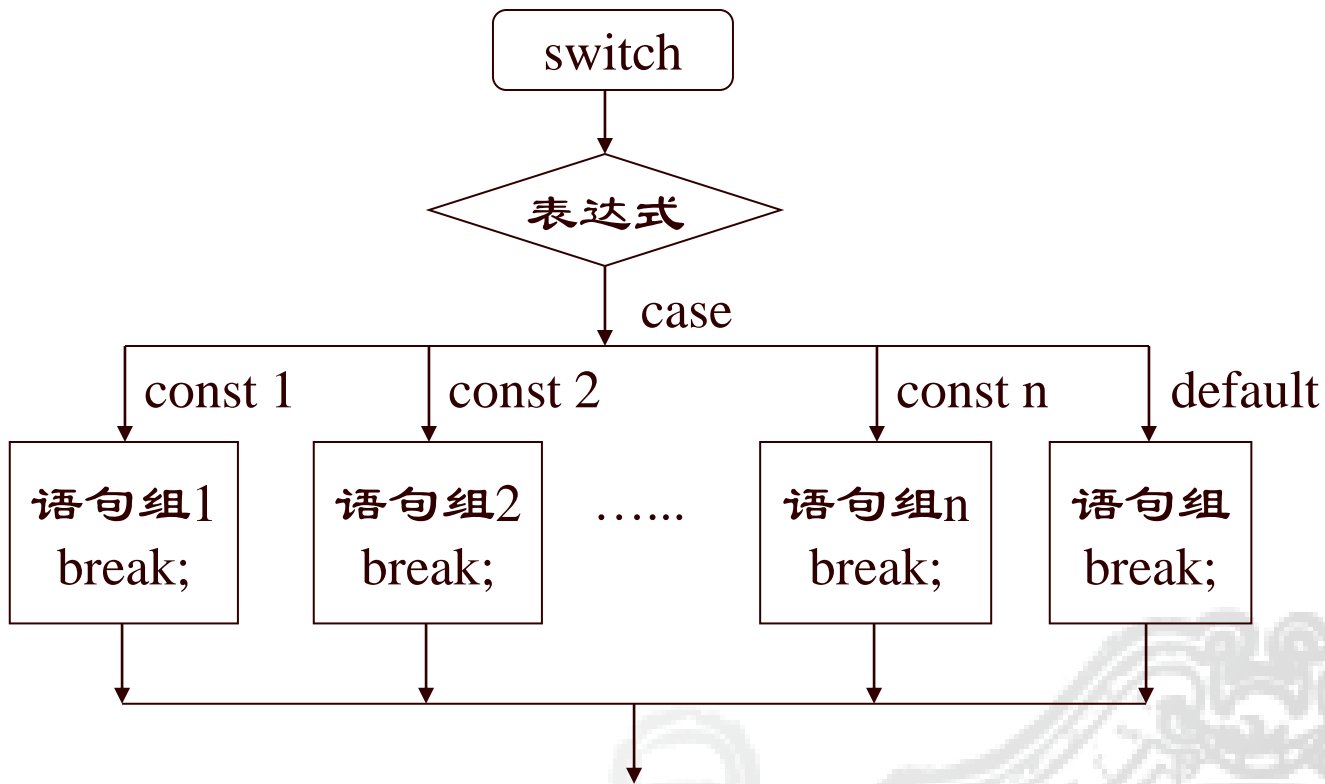
- ❖ 功能：在循环语句和switch语句中,终止并跳出循环体或开关体。
- ❖ 说明：
 - break只能终止并跳出最近一层的结构。
 - break不能用于循环语句和switch语句之外的任何其它语句之中。
- ❖ 一般形式：**break;**





❖ 流程形式:





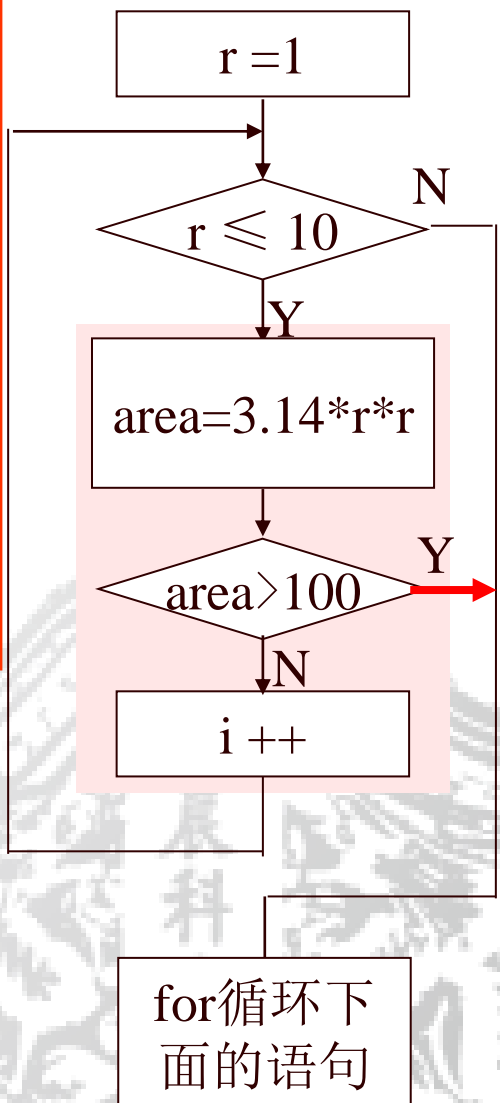


例 break举例：输出圆面积，面积大于100时停止

```
#include <stdio.h>
#define PI 3.14159
int main( )
{ int r ;
  float area;
  for(r=1;r<=10;r++)
  {
    area=PI*r*r ;
    if(area>100) break;
    printf("r=%d,area=%f\n",r,area);
  }
  return 0;
}
```

输出：
r=1, area=3.14
r=2, area=12.57
r=3, area=28.27
r=4, area=50.27
r=5, area=78.85
/* 当 r=6时，
area=113.10
故此值没有被输出 */

使用break跳出整个循环





例 break举例：小写字母转换成大写字母,直至输入非字母字符

```
#include <stdio.h>
int main()
{char c;
  while(1)
  { c=getchar();
    if(c>='a' && c<='z') putchar(c-32);
    else
      if(c>='A' && c<='Z') putchar(c);
      else break;
  }
}
```

break 使无限循环结束



★ continue 语句

- ❖ 功能：结束本次循环，跳过循环体中尚未执行的语句，进行下一次是否执行循环体的判断。
- ❖ continue 语句仅用于循环语句中。

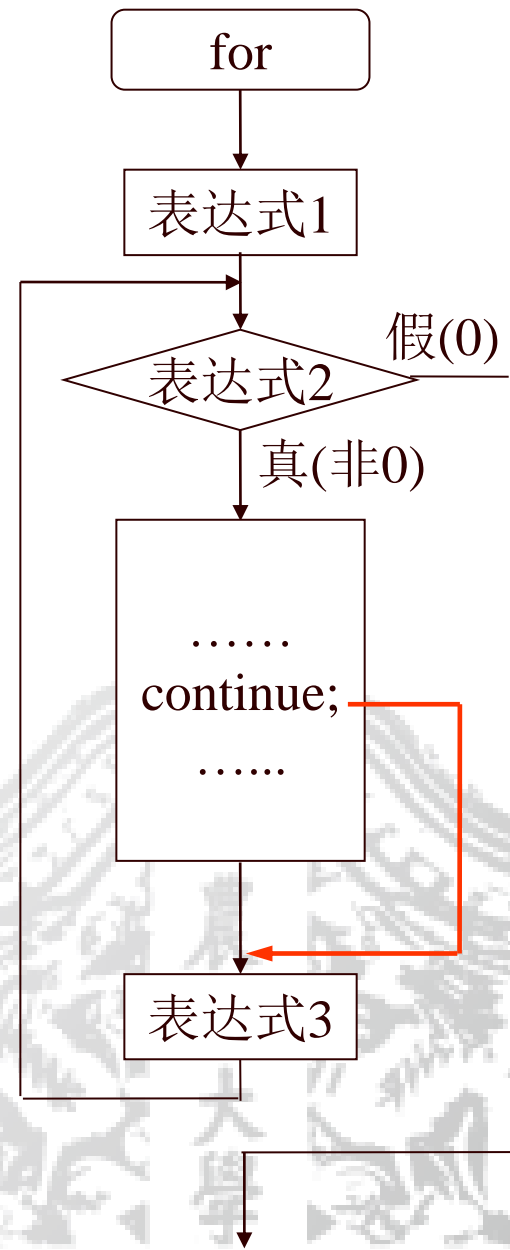
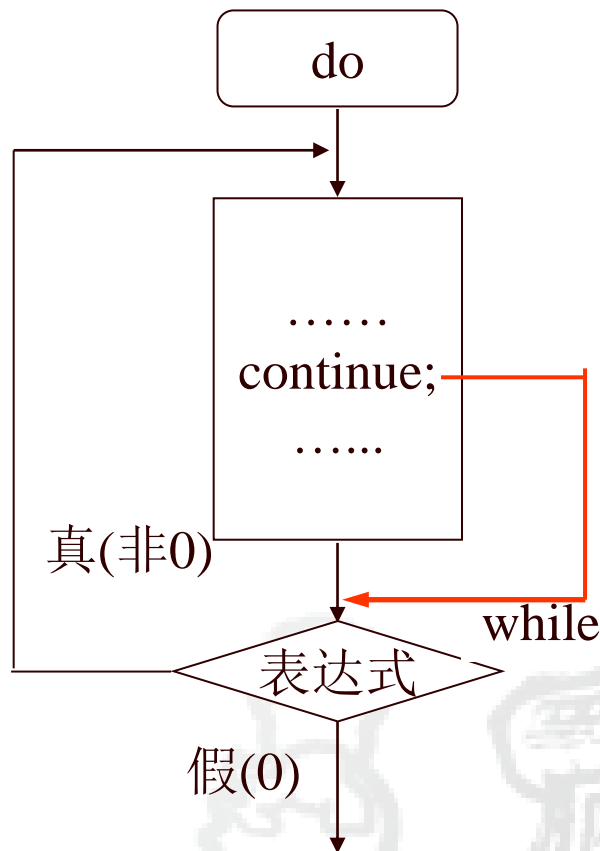
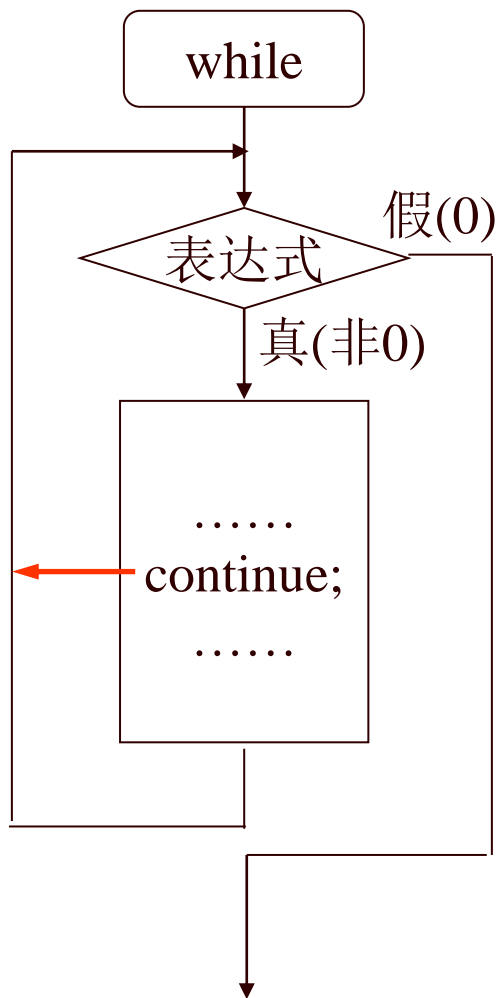


break 和 continue 语句的区别

- ❖ continue 语句只结束本次循环，break 语句则是结束整个循环。
- ❖ continue 语句只用于 while, do-while, for 循环语句中，break 语句还可以用于 switch 语句中。



❖ 流程形式:

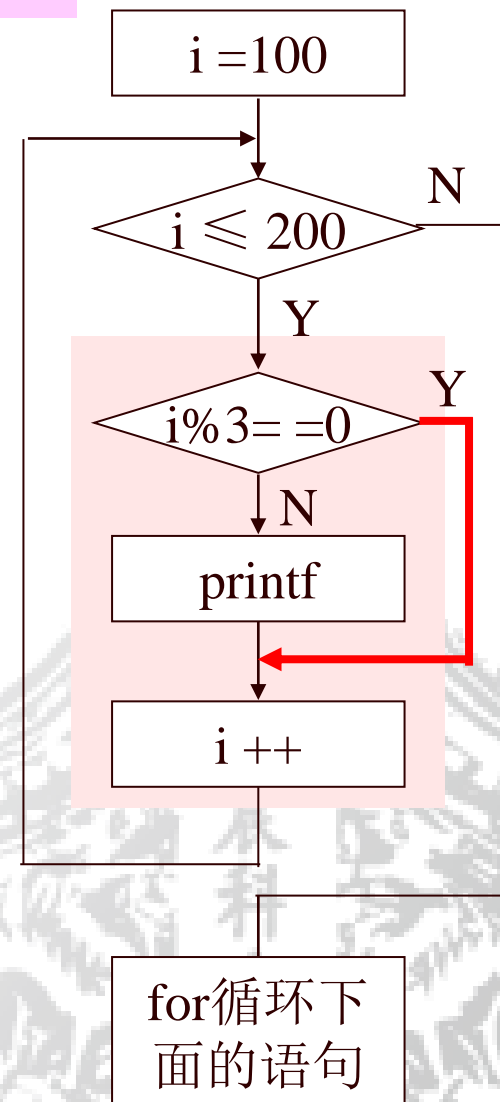




例6.5 把100~200之间不能被3整除的数输出

```
#include <stdio.h>
int main()
{ int i;
  for(i=100;i<=200;i++)
  { if(i%3==0)
    continue;
    printf("%d\n",i);
  }
  return 0;
}
```

运行结果：
100, 101,
103, 104,
106, 107,
109, 110,
112, 113,
115, 116,
118, 119,



使用continue
结束本次循环



例 求输入的十个整数中正数的个数及其平均值

```
#include <stdio.h>
int main()
{   int i,num=0,a;
    float sum=0;
    for(i=0;i<10;i++)
    {   scanf("%d",&a);
        if(a<=0) continue;
        num++;
        sum+=a;
    }
    printf("%d plus integer's sum :%6.0f\n",num,sum);
    printf("Mean value:%6.2f\n",sum/num);
}
```



例6.7 求Fibonacci数列：1、1、2、3、5、8.....前40个数。

Fibonacci数列可以用一个有趣的古典数学问题来描述：有一对兔子，出生三个月后每个月都生一对兔子。小兔子长到三个月后每个月又生一对兔子。假设所有的兔子都不死，问每个月的兔子总数是多少对？

$$F_1 = 1 \quad (n = 1)$$

$$F_2 = 1 \quad (n = 2)$$

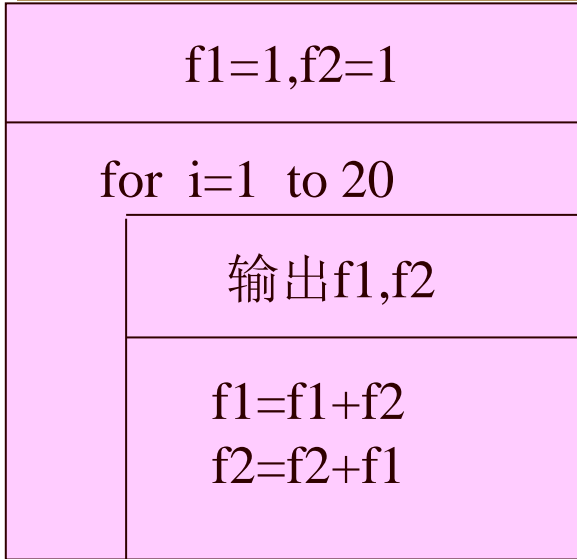
$$F_n = F_{n-1} + F_{n-2} \quad (n \geq 3)$$

月数	小兔	中兔	老兔	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
...

$$F_1 = 1 \quad (n = 1)$$

$$F_2 = 1 \quad (n = 2)$$

$$F_n = F_{n-1} + F_{n-2} \quad (n \geq 3)$$



```

#include <stdio.h>
int main()
{ int f1,f2, f3;
  int i;
  f1=1; f2=1;
  printf("%d %d ",f1,f2);
  for(i=3;i<=20;i++)
  {
    f3 = f1+f2;
    f1=f2;
    f2=f3;
    printf("%d ",f3);
    if (i%4==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 雨水淋湿了算术书的一道题，8个数字只能看清3个，第一个数字虽然看不清，但可看出不是1。编程求其余数字是什么？
- ▶ $[\square * (\square 3 + \square)]^2 = 8\square\square 9$
- ▶ 分析：设分别用A、B、C、D、E五个变量表示自左到右五个未知的数字。其中A的取值范围为2~9，其余取值范围为0~9。条件表达式即为给定算式。
- ▶ `#include <stdio.h>`
- ▶ `int main() {`
- ▶ `int A,B,C,D,E;`
- ▶ `for (A=2;A<=9;A++)`
- ▶ `for (B=0;B<=9;B++)`
- ▶ `for (C=0;C<=9;C++)`
- ▶ `for (D=0;D<=9;D++)`
- ▶ `for (E=0;E<=9;E++)`
- ▶ `if (A*(B*10+3+C)*A*(B*10+3+C)==8009+D*100+E*10)`
- ▶ `printf("%d%d%d%d%d\n",A,B,C,D,E);`
- ▶ `return 0;}`
- ▶ 结果：32864



- ▶ **【例7】** 张三、李四、王五三个棋迷，定期去文化宫下棋。张三每五天来一次，李四每六天来一次，王五每九天来一次。问每过多少天他们才能一起在文化宫下棋？
- ▶ **分析：**此问题实际上是求最小公倍数的数学问题。设结果为 x ，其取值范围为 $0 \rightarrow \infty$ 。
- ▶ 因上限为无限大，计数值不能预先确定，故用while循环结构更合适。

- ▶ `#include <stdio.h>`
- ▶ `int main()`
- ▶ `{ int x=1;`
- ▶ `while (1)`
- ▶ `{ if (x%5==0 && x%6==0 && x%9==0)`
- ▶ `{ printf("x=%d\n",x);`
- ▶ `break;}`
- ▶ `x++;}`
- ▶ `return 0;}`

结果： $x=90$



循环结构嵌套 - 跳出内循环

信息与电气工程学院

```
#include <stdio.h>
int main()
{ int a[10],i,j,t;
  for(j=0;j<9;j++)
    for(i=0;i<9;i++)
      {
        if(i==j)
        {
          printf("* ");
          continue;
        }
        printf("%d ",a[i]);
      }
}
```

```
#include <stdio.h>
int main()
{ int a[10],i,j,t;
  for(j=0;j<9;j++)
    for(i=0;i<9;i++)
      {
        if(i==j)
        {
          printf("* ");
          break;
        }
        printf("%d ",a[i]+10);
      }
}
```

序 }



循环结构嵌套 - 跳出内循环

信息与电气工程学院

```
#include <stdio.h>
int main()
{ int a[10],i,j,t;
  for(j=0;j<9;j++)
  {
    for(i=0;i<9;i++)
    {
      if(i==j)
      {
        printf("*");
        printf("%d ",a[i]);
      }
      else printf("%d ",a[i]);
    }
    printf("\n");
  }
}
```





循环结构嵌套 - 跳出外循环

```
#include <stdio.h>
int main()
{ int a[10],i,j,t=0;
  for(j=0;j<9;j++)
  {
    for(i=0;i<9;i++)
    {
      if(i==j)
      {
        t=1;
        break;
      }
    }
    if (t==1)
    {
      printf("%d ",a[i]+10);
      break;
    }
  }
}
```

```
#include <stdio.h>
int main()
{ int a[10],i,j,t=0;
  for(j=0;j<9;j++)
  {
    for(i=0;i<9;i++)
    {
      if(i==j)
      {
        t=1;
        continue;
      }
    }
    if (t==1)
    {
      printf("%d ",a[i]+10);
      continue;
    }
  }
}
```




- ▶ **【练习2】** 设有程序段
- ▶ `int k=10;`
- ▶ `while(k=0) k=k-1;`
- ▶ 下面描述正确的是 ()
- ▶ A) While循环执行10次
- ▶ B) 循环是无限循环
- ▶ C) 循环体语句一次也不执行
- ▶ D) 循环体语句执行一次





▶ **【练习3】** 下面程序的执行结果 ()

▶ main ()

▶ {

▶ int x=10;

▶ do{x--;} while(--x);

▶ printf("%d\n",x--);

▶ }

▶ A) 0 B) -1 C) 1 D) 8

