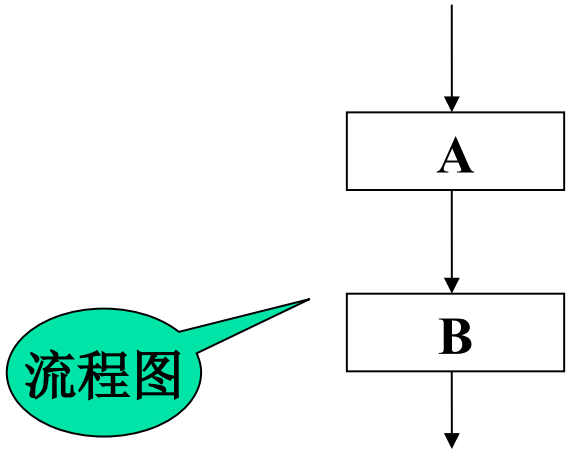


# 控制语句

- 程序的三种基本结构
  - 顺序结构
  - 选择结构
  - 循环控制结构
- 选择结构程序设计
- 循环结构程序设计
- 辅助控制语句

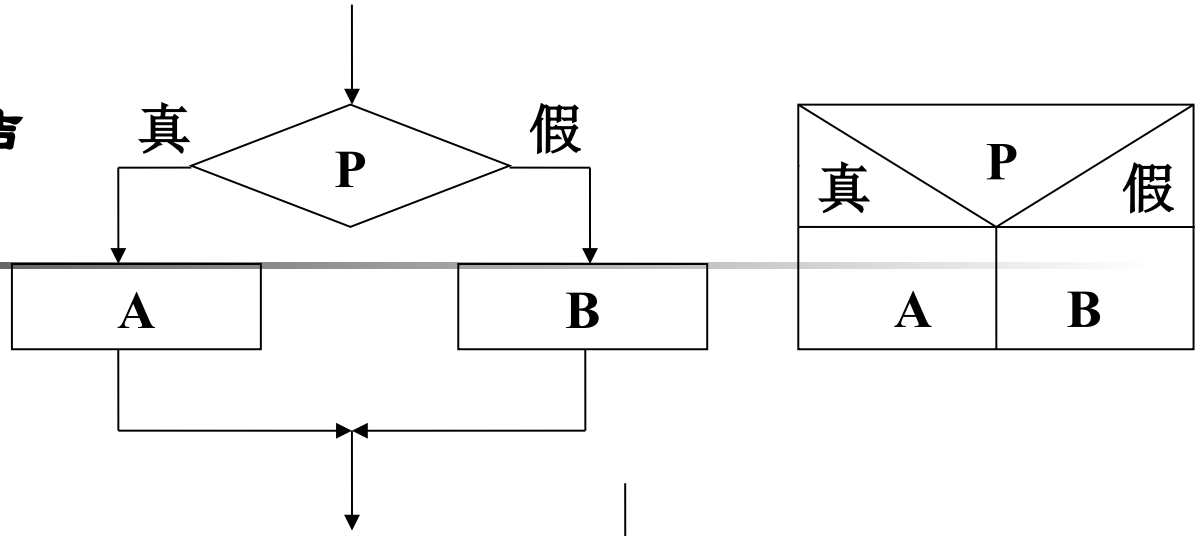
# 程序的三种基本结构

- 结构化程序设计
  - 基本思想：任何程序都可以用三种基本结构表示，限制使用无条件转移语句（ **goto** ）
  - 优点：结构清晰，易读，提高程序设计质量和效率
- 三种基本结构
  - 顺序结构

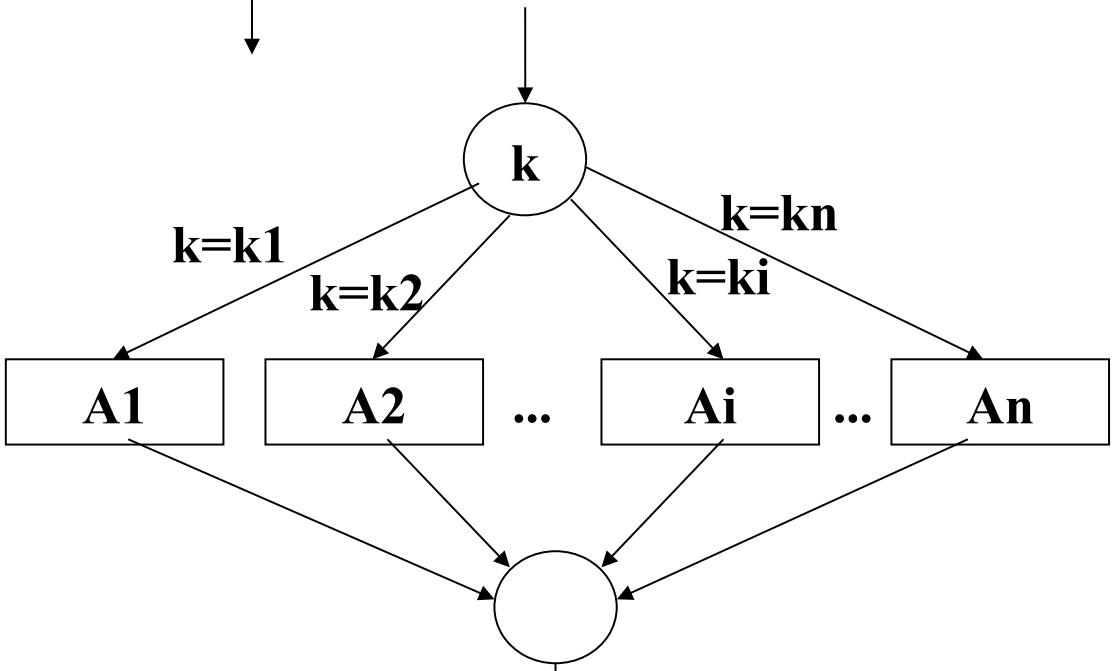


❖选择结构

- 二分支选择结构



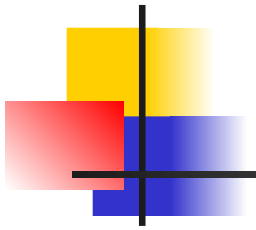

- 多分支选择结构




❖循环结构

A,B,A1….An 可以是一个简单语句，也可以是一个基本结构

# C 程序流程控制语句



- 分支
  - if( )~else~
  - switch
- 循环
  - for( )~
  - while( )~
  - do~while( )
- 辅助控制
  - continue
  - break
  - goto
  - return

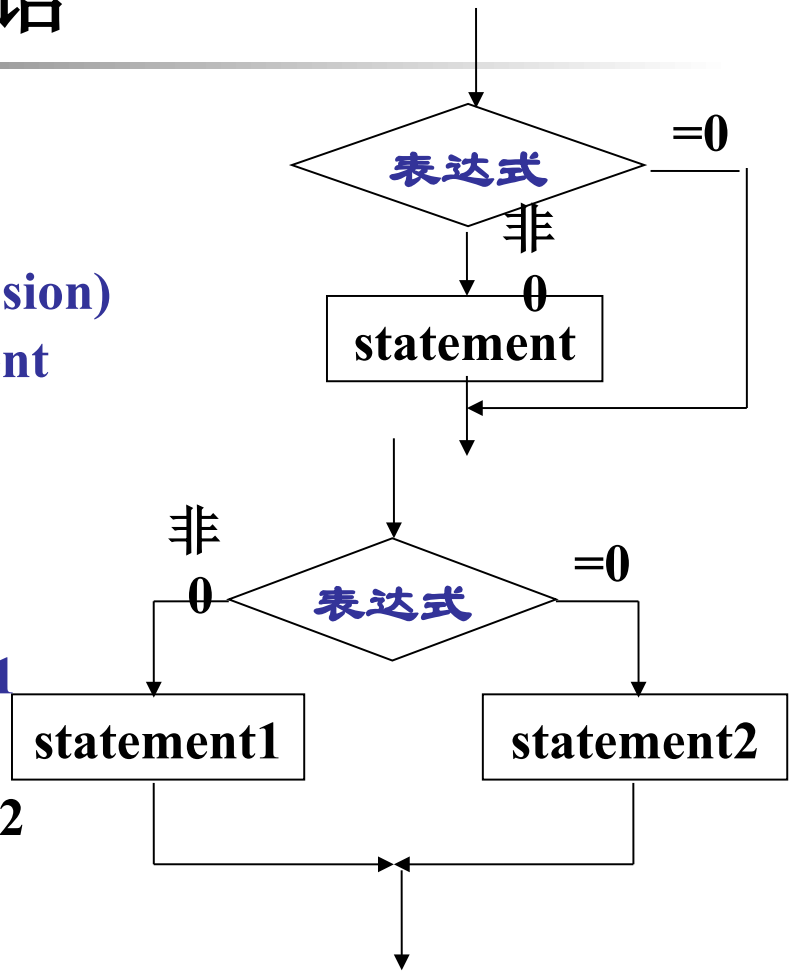
# 选择结构程序设计

## (一) if 语句（条件选择语句）

### if 语句的三种形式

- 形式一：
  - 格式： if (expression)
    statement
  - 执行过程：

```
if (x>y){
    printf("%d",x);
}
```




- 形式二：
  - 格式： if (expression)
    statement1
    else
    statement2
  - 执行过程：




例：

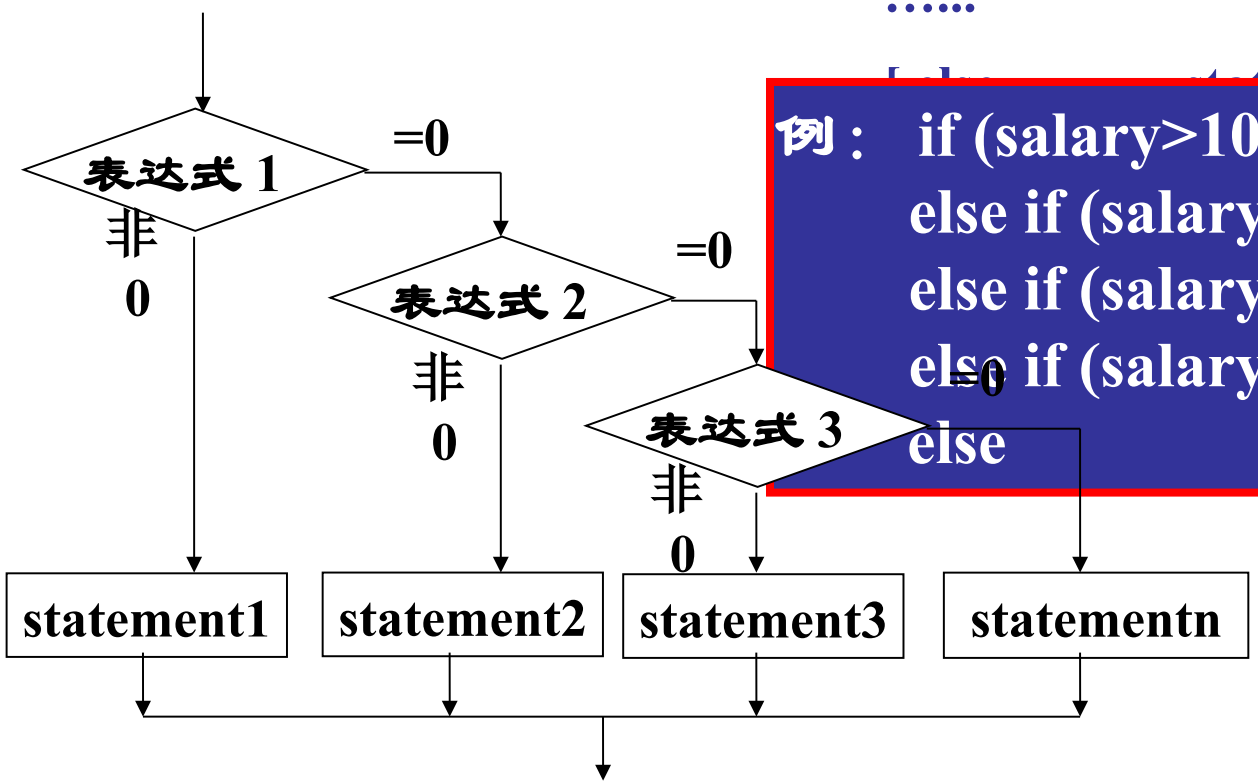
```
if (x>y){
    max=x;
}
else{
    max=y;
}
```

# 形式三：

➢格式：

```
if ( expr1 )          statement1
else if (expr2 )
statement2
else if (expr3 )
statement3
……
```

➢执行过程：




例：
```
if (salary>1000)        index=0.4;
else if (salary>800)    index=0.3;
else if (salary>600)    index=0.2;
else if (salary>400)    index=0.1;
else                    index=0;
```

- 说明：
  - if 后面的表达式类型任意
  - 语句可以是复合语句
  - if(x) ⇔ if(x!=0)
    if(!x) ⇔ if(x==0)

例：
```
#include <stdio.h>
int  main( )
  {
     int  x, y, z;
     scanf("%d,%d", &x, &y);
     if (x>y)
     { z=x; x=y; y=z;}
     else
     { x++;   y++;   }
     printf("%d,%d\n", x, y);
     return 0;
  }
```



- if 语句嵌套：

```
if (expr1)
    if (expr2)
        statement1
    else
        statement2
```

内嵌 if

```
if (expr1)
    if (expr2)
        statement1
else
    statement3
```

内嵌 if

```
if (expr1)
    statement1
else
    if(expr3)
        statement3
    else
        statement4
```

内嵌 if

```
if (expr1)
    if (expr2)  statement1
    else        statement2
else
    if(expr3)   statement3
    else        statement4
```

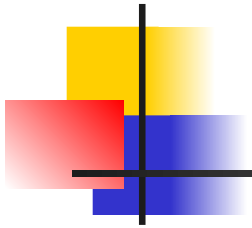
内嵌 if

内嵌 if

## 例：输入两数并判断其大小关系

```
#include <stdio.h>
void main( )
{
   int  x, y;
   printf("Enter integer x,y:");
   scanf("%d,%d", &x, &y);
   if (x!=y)
     if (x>y)  printf("X>Y\n");
     else     printf("X<Y\n");
   else
     printf("X==Y\n");
}
```

```
运行：  Enter  integer
x,y:12,23↵
          X<Y
         Enter  integer  x,y:12,6↵
          X>Y
         Enter  integer  x,y:12,12↵
```



- if ～else 配对原则：

  缺省{}时， else 总是和它上面离它最近的未配对的 if 配对。

- 实现 if～else 正确配对方法：加{}

```
if(……)
    if(……)
        if(……)
        else……
    else……
else……
```

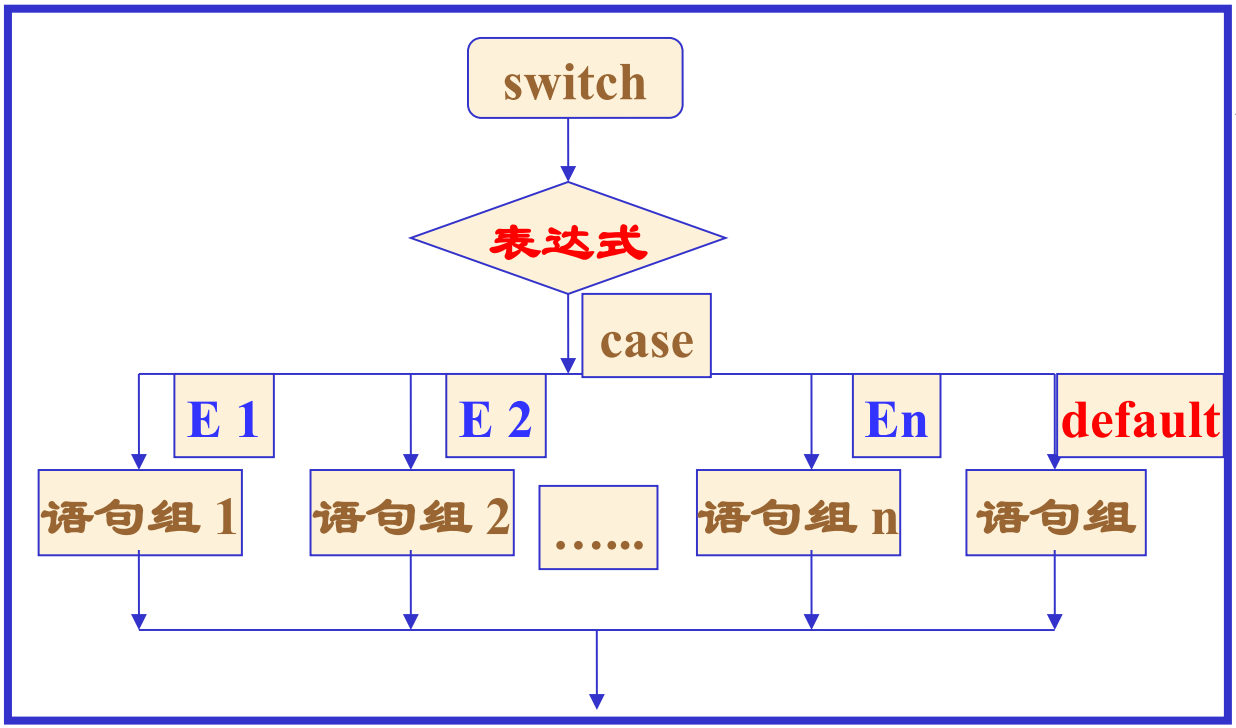
例：
```
if (a==b)
    if(b==c)
        printf("a==b==c");
else
    printf("a!=b");
```

修改：
```
if (a==b)
{
    if(b==c)
        printf("a==b==c");
}
else
    printf("a!=b");
```

# （二） switch 语句（开关分支语句）

❖一般形式：

❖执行过程：




switch( 表达式 )

{ case E1:

语句组 1;

break;

case E2:

语句组 2;

break;

…….

case En:

语句组 n;

break;

[default:

语句组 ;

break;]

❖说明：

- E1,E2,…En 是常量表达式，且值必须互不相同，起语句标号作用，用 break 才能跳出
- case 后可包含多个可执行语句，且不必加 { }
- switch 可嵌套
- 多个 case 可共用一组执行语句

例：

```
……
    case  ‘A’:
    case  ‘B’:
    case  ‘C’:
                printf(“score>60\n”);
                break;
    ……..
```

例： switch(score)

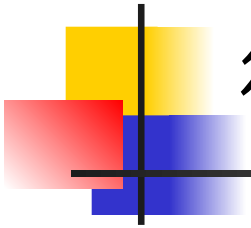
```
{
        case  5:  printf("Very good!");
        case  4:  printf("Good!");
        case  3:  printf("Pass!");
        case  2:  printf("Fail!");
        default :  printf("data error!");
}
```

运行结果： score 为 5 时，输出：

Very good! Good! Pass! Fail! data error!

```
例 : int main( )
    {  int  x=1;     int  y=0;
       int  a=0;     int  b=0;
       switch(x)
       {
           case  1:
                    switch(y)
                     {
                         case 0:   a++;  break;
                         case 1:   b++;  break;
                     }
            case  2:  a++; b++; break;
            case  3:  a++; b++;
        }
       printf("\n a=%d,b=%d",a,b);
       return 0 ;
     }
```

运行结果：
a=2,b=1



# 循环结构程序设计

C 语言可实现循环的语句：

- 用 **goto** 和 **if** 构成循环
- **while** 语句
- **do ~ while** 语句
- **for** 语句

# （一） goto 语句及用 goto 构成循环

❖goto 语句一般格式：

```
        goto    语句标
号;
        ……
标号：语句;
```

❖功能：无条件转移语句

❖说明：

- 不能用整数作标号
- 标号只能出现在 goto 所在函数内，且唯一
- 标号只能加在可执行语句前面
- 限制使用 goto 语句

例：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);
    return 0;
}
```

循环初值

循环条件

循环变量增值

循环终值

循环体

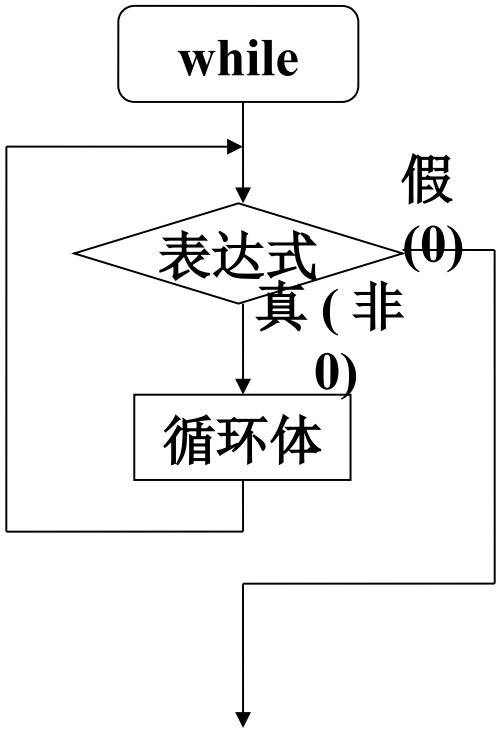
sum=0+1
sum=1+2=3
sum=3+3=6
sum=6+4
……
sum=4950+100=5050

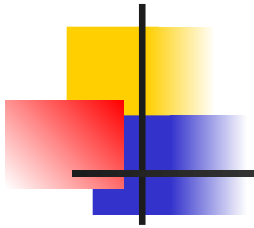
# （二） while 语句

❖一般形式：

```
while( 表达式 )
        循环体语句;
```

❖执行流程：





❖特点：先判断表达式，后执行循环体

❖说明：

– 循环体有可能一次也不执行

- 循环体可为任意类型语句

- 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, return, goto

❖无限循环： while(1)

循环体；

例：用 while 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);
    return 0;
}
```

循环初值

循环条件

循环终值

循环变量增值

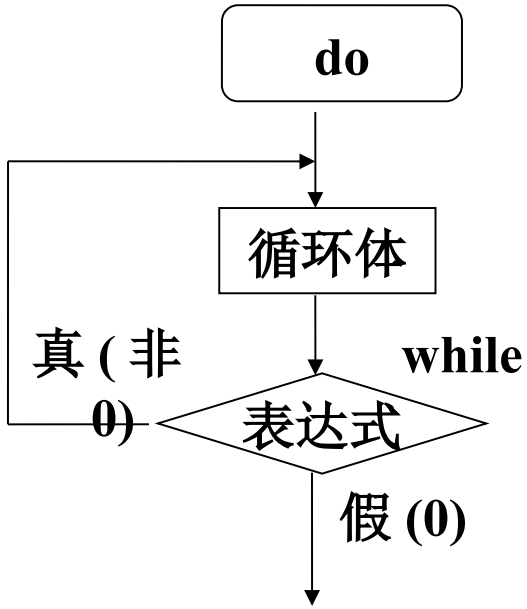
循环体

# （三） do~while 语句

❖一般形式：

```
do
        循环体语
句；
 while( 表达式 );
```

❖执行流程：

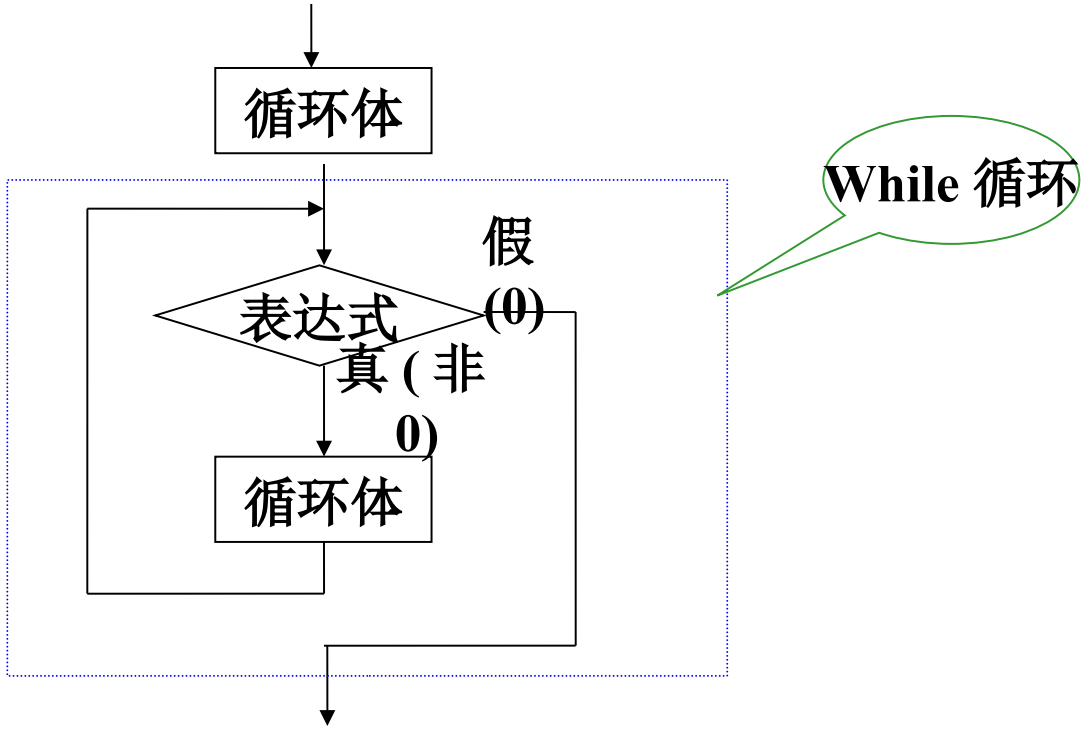


❖特点：先执行循环体，后判断表达式

❖说明：

- 至少执行一次循环体

-do~while 可转化成 while 结构



例：用 do~while 循环求 $\sum_{n=1}^{100} n$

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

循环初值

循环变量增值

循环条件

循环体

循环终值

例： while 和 do~while 比较

```
int main( )
{
   int  i, sum=0;
   scanf("%d", &i);
   do
   {
        sum+= i;
        i++;
   } while (i<=10);
   printf("%d", sum);
   ruturn 0;
}
```

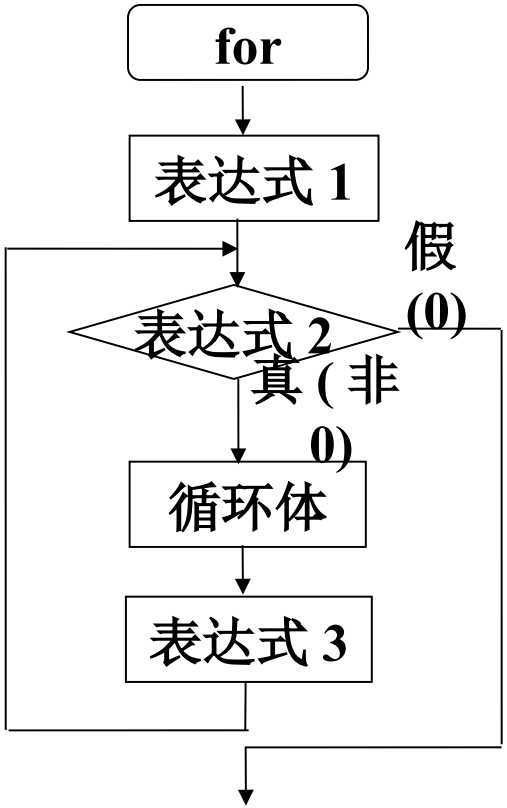
```
int main( )
{
   int  i, sum=0;
   scanf("%d", &i);
   while (i<=10)
   {
        sum+= i;
        i++;
   }
   printf("%d", sum);
   ruturn 0;
}
```

# （四）for 语句

❖一般形式：

```
for([expr1] ;[ expr2] ;[ expr3])
    循环体语句;
```

❖执行流程：



❖for 语句一般应用形式：

```
for( 循环变量赋初值；循环条件；循环变量改变 )
{
        循环体语句；
}
```

例：用 for 循环求 $\sum_{n=1}^{100} n$

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

❖说明：

-for 语句中 expr1, expr2 ,expr3 类型任意，都可省略，但分号；不可省

- 无限循环： for(;;)

-for 语句可以转换成 while 结构

```
expr1;
while(expr2)
{
        循环体语句;
        expr3;
}
```

# （五）循环的嵌套

- 三种循环可互相嵌套，层数不限
- 外层循环可包含两个以上内循环，但不能相互交叉
- 嵌套循环的执行流程
- 嵌套循环的跳转

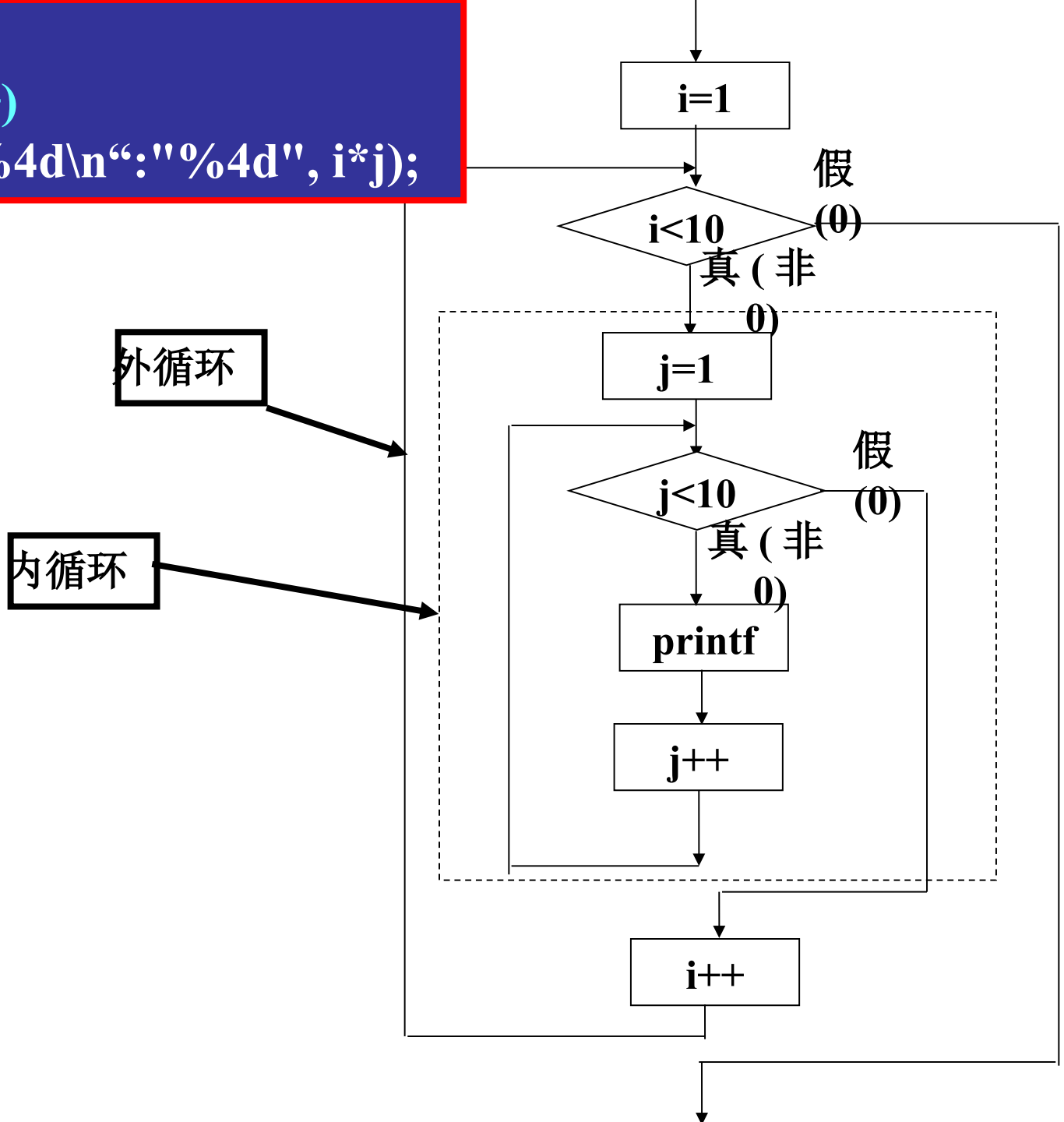
  禁止：

  - 从外层跳入内层
  - 跳入同层的另一循环

例：循环嵌套，输出九九表

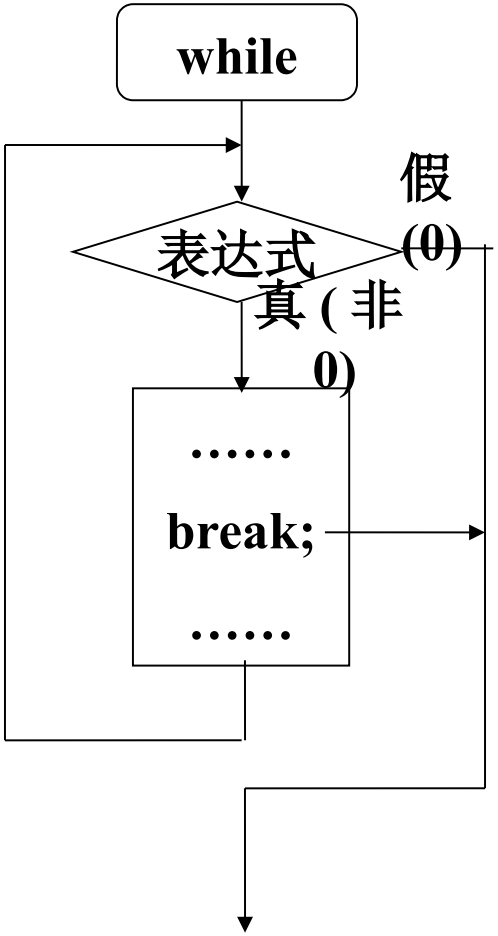
```
#include <stdio.h>
int main( )
{
    int  i, j;
   for (i=1; i<10; i++)
      printf("%4d", i);
   printf("\n--------------------------------\n");
   for (i=1; i<10; i++)
      for (j=1; j<10; j++)
            printf((j==9)?"%4d\n" : "%4d", i*j);
   return 0;
}
```

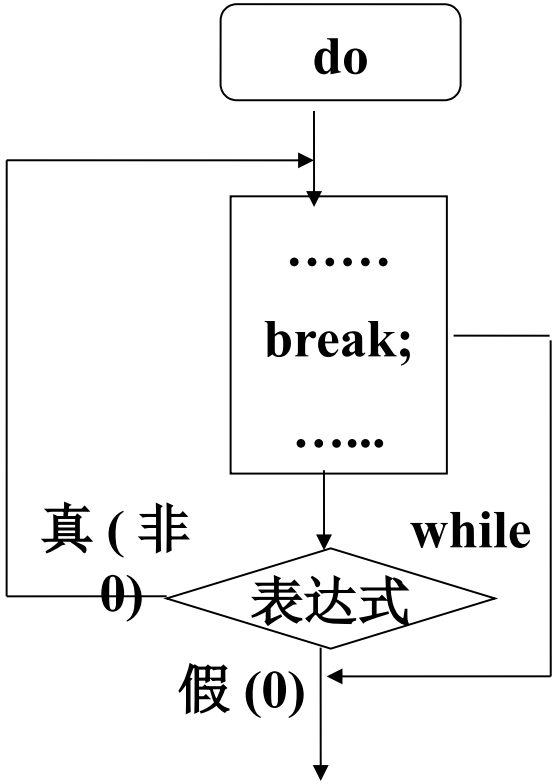
```
for (i=1; i<10; i++)
  for (j=1; j<10; j++)
    printf((j==9)?"%4d\n“:"%4d", i*j);
```

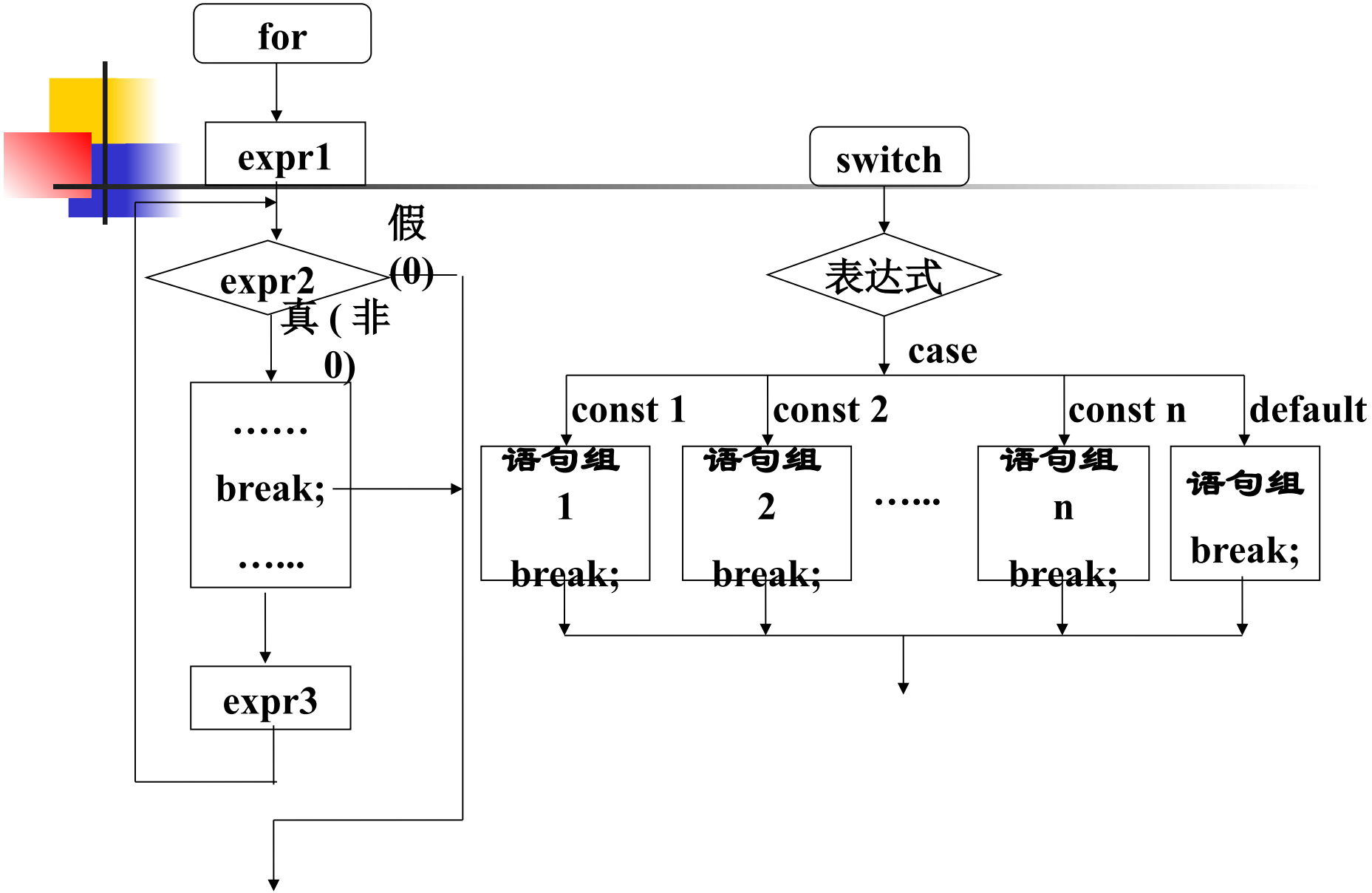


# 辅助控制语句

- **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


while
假
(0)
表达式
真 ( 非
0)
……
break;
……


do
……
break;
……
真 ( 非
0)
while
表达式
假 (0)

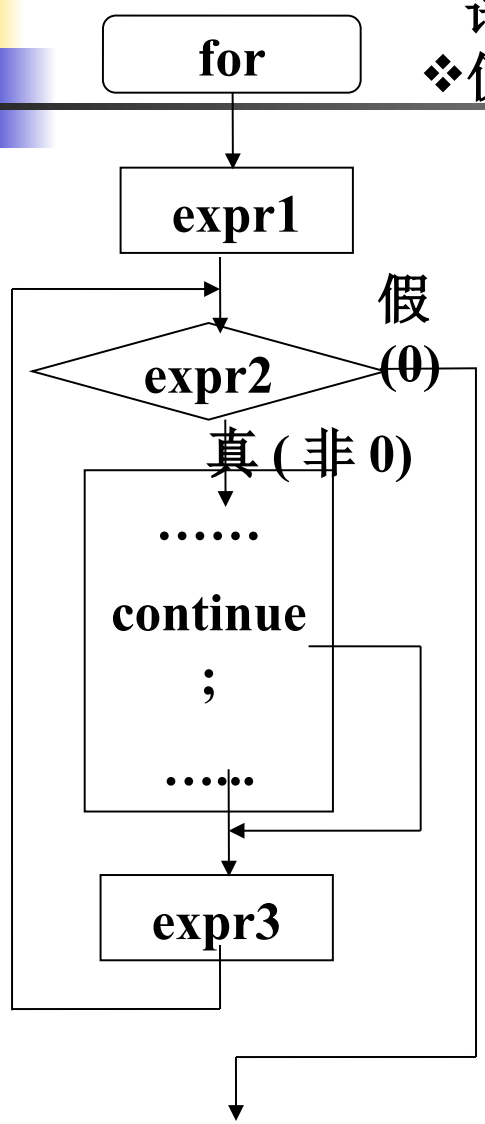

for
expr1
expr2
假
(0)
真 ( 非
0)
……
break;
……
expr3
switch
表达式
case
const 1
const 2
const n
default
语句组
1
break;
语句组
2
break;
……
语句组
n
break;
语句组
break;

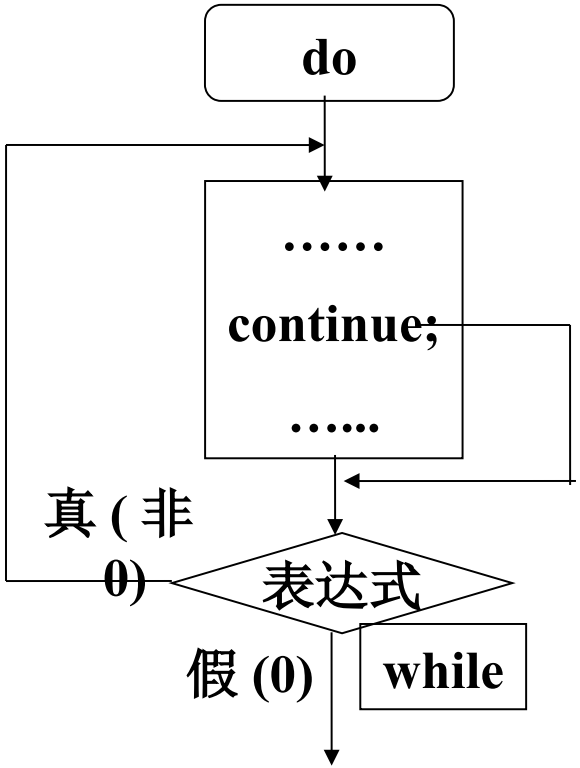
举例：从半径 =1 开始，输出圆的面积直到面积大于 100 。

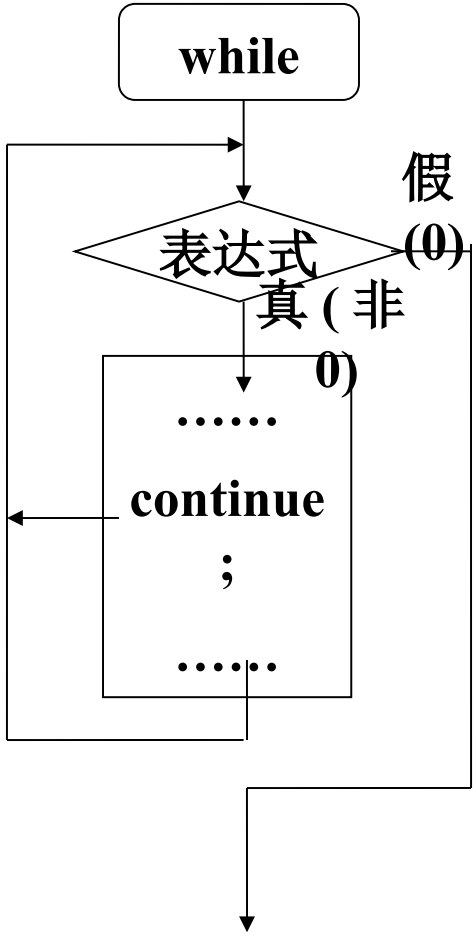
```
#define  PI  3.14159
int main( )
{   int  r;
    float  area;
    for (r=1; r<=10; r++)
    {
        area = PI*r*r;
       if (area>100)
           break;
       printf("r=%d, area=%.2f\n", r, area);
    }
     return 0;
}
```

# ➢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
- ❖仅用于循环语句中



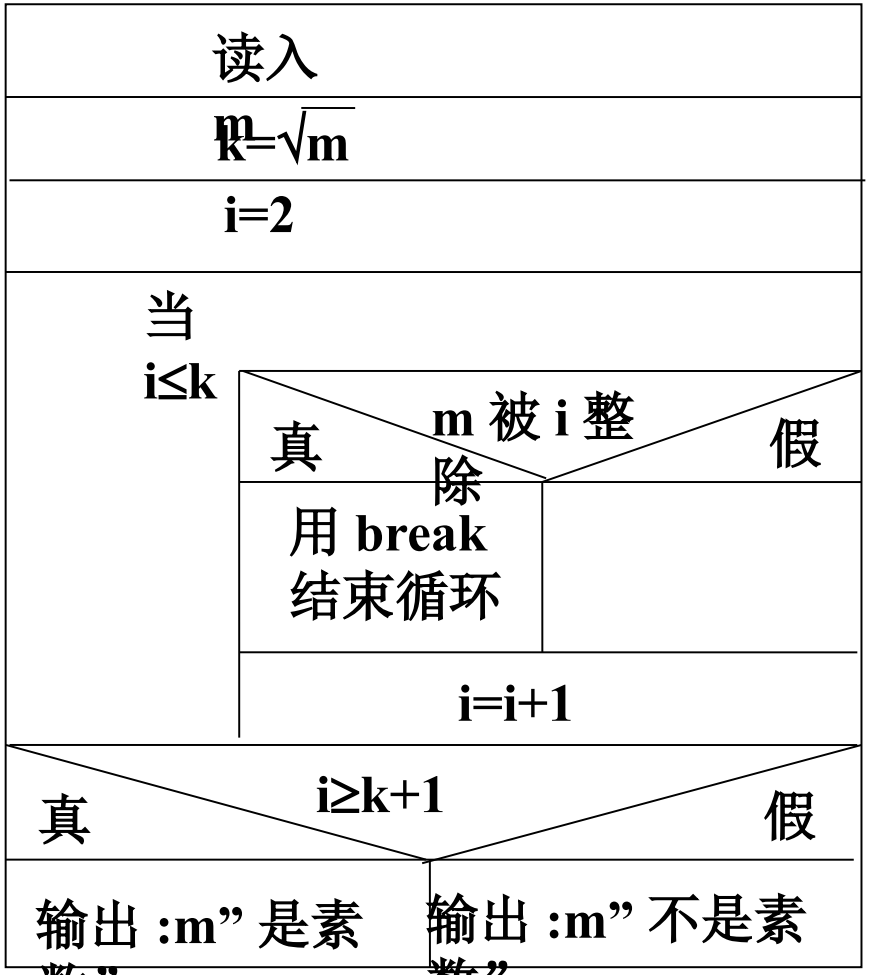



continue 举例：输入的十个整数，求其中正数的个数及其平均值

```
#include <stdio.h>
int main( )
{   int  i, a;
    int  num=0;
    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);
    return 0;
}
```

# 例：判断 m 是否素数



# 例：判断 m 是否素数

```
#include <stdio.h>
#include <math.h>
int main( )
{   int m, k, i=2;
    scanf_s("%d", &m);
    k=(int)sqrt(m);
    while(i<=k)
    {
            if(m%i==0) break;
            i++;
    }
    if(i>=(k+1)) printf(“%d is  a prime number”, m);
    else printf(“%d is not a prime number”, m);
     return 0;
}
```

# 例：输出 100-200 的素数（每行 5 个数）

```
int main( ){
        int m, k, i=2, j, count = 0;
        for (m = 101; m<200; m +=2 ){
                i=2;  k=(int)sqrt(m);
                while(i<=k){ if(m%i==0) break; i++;}
                if(i>=(k+1)){
                        count++;
                        printf("%5d", m);
                        if (count%5 == 0) printf("\n");
                }
        }
        return 0 ;
}
```

- 用迭代法求 $x$ 的值，当最后一项小于 $10^{-6}$ 则停止计算。

$$x = \sum_{n=1}^{+\infty} \frac{2^n}{n!}$$

```
#include <stdio.h>
#include <math.h>
int main( )
{
        double sum=0, term=0;
        int n=1;
        term = 2.0/n;
        while (fabs(term) >= 1e-6)
        {       sum += term;
                n++;
                term =term * 2.0/n;
        }
        printf("%lf",sum);
        return 0 ;
}
```

- 编写程序，求满足公式的变量 κ 的最大值：
  2⊥κ <= μ 其中， μ 是程序输入的一个正整数。
- 4294967295

```
#include <stdio.h>
int main (void)
{
        unsigned int m,k,temp;
        scanf("%u", &m);
        k=0;
        while(m>1)
        {
                temp=m%2;
                m=m-temp;
                m=m/2;
                k++;
        }
        printf("%u", k);
        return 0;
}
```

# 小结

- 从程序执行的流程来看， 程序可分为三种最基本的结构： 顺序结构，分支结构以及循环结构。
- C 语言的条件分支语句包括：

  (1) if 语句主要用于单向选择。

  (2) if-else 语句主要用于双向选择。

  (3) if-else-if 语句和 switch 语句用于多路选择。
- C 语言的循环语句包括：
  - while 语句
  - do ~ while 语句
  - for 语句
- C 语言的辅助控制语句包括：

  continue 、 break 、 goto 、 return 等