第四章
循环结构

关于第一次测验2班：第1-4章

- 3.23(周四) 7:50之前到计算中心12机房，不刷卡
- 带校园卡或身份证，放于桌上备查；
- 考试机位将在3.22(周三)20:00之前在钉钉群公布
- 进入机房后，打开机器，选择win7，启动后；点击“OMS客户端”程序，等待教师启动考试
- 考试分两场：理论(8:10-8:55)、实验(8:55-9:25)，中间自动切换，不要退出OMS
- 理论考试期间，不能打开除OMS系统之外的任何窗口；实验考试，只能打开Dev-C++等C编译器。违反规定，按照作弊处理

关于上机

- 程序书写风格
  - 缩入：例如，统一缩入四个空格、一个Tab
  - 复合语句中 {} 的对齐
  - 适当的空行
  - 变量名命名
  - 变量名使用
  - ……

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

int main(void)
{
    int denominator, flag ;
    double item, pi ;

    flag = 1 ;
    denominator = 1 ;

    item = 1.0 ;
    pi = 0.0 ;

    while (fabs(item) >= 0.000025)
    {
        item = flag * 1.0 / denominator ;
        pi += item ;
        flag = -flag ;
        denominator += 2 ;
    }

    pi *= 4 ;
    printf("pi=%.5lf\n", pi);

    return 0;

}
```

上机问题

- 关系运算符“==”与赋值运算符“=”

  n == 0 与 n = 0 的区别？

- 逻辑运算符：&& 与 || 的区别?

  &&：两个条件均为真，结果为真

  ||：有一个条件为真，结果为真

- for 语句、if-else语句的逻辑错误

上机问题

- scanf的格式控制符

  输入： 2□+□3

  scanf(“%d%c%d”, &a, &op, &b)　　op=‘□’ ×

  应该使用

  scanf(“%d□%c□%d”, &a, &op, &b)

  scanf(“%d%c%d\n”, &a, &op, &b)： 不要加“\n”

- http://www.cplusplus.com/reference/cstdio/scanf/

上机问题

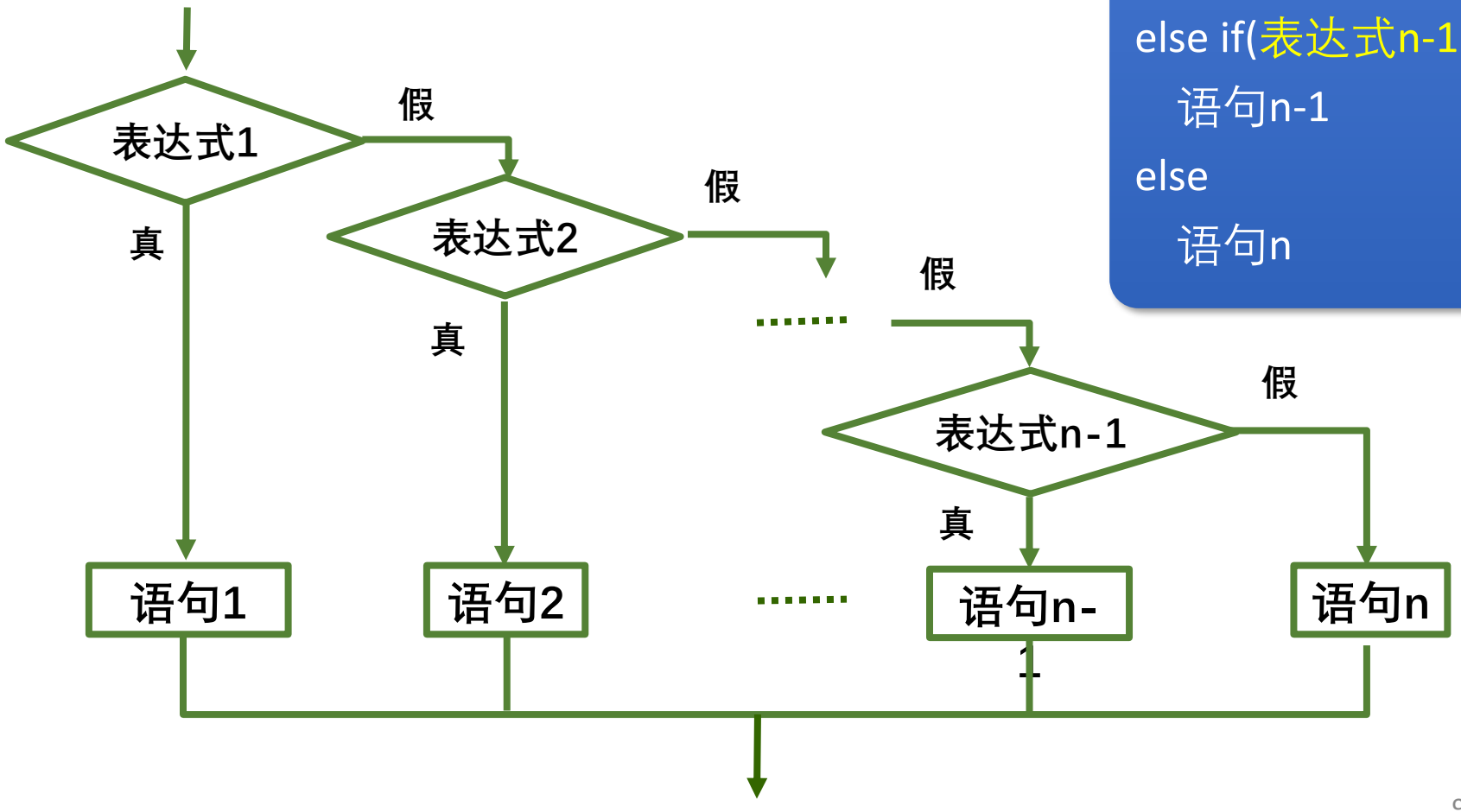
- 判断 a < x < b：　x > a && x < b
- 字符型常量：
  - 0（整数）；‘0’（字符常量）
  - ‘ ’（字符常量空格）；“ ”（字符串常量空格）
- 嵌套的分支结构：复合语句大括号匹配 { }
- 浮点运算：尽可能用 double 类型变量
- 编程：多练习，熟能生巧

回顾分支结构

- if
- if/else
- if/else if/…/else if/else
- switch

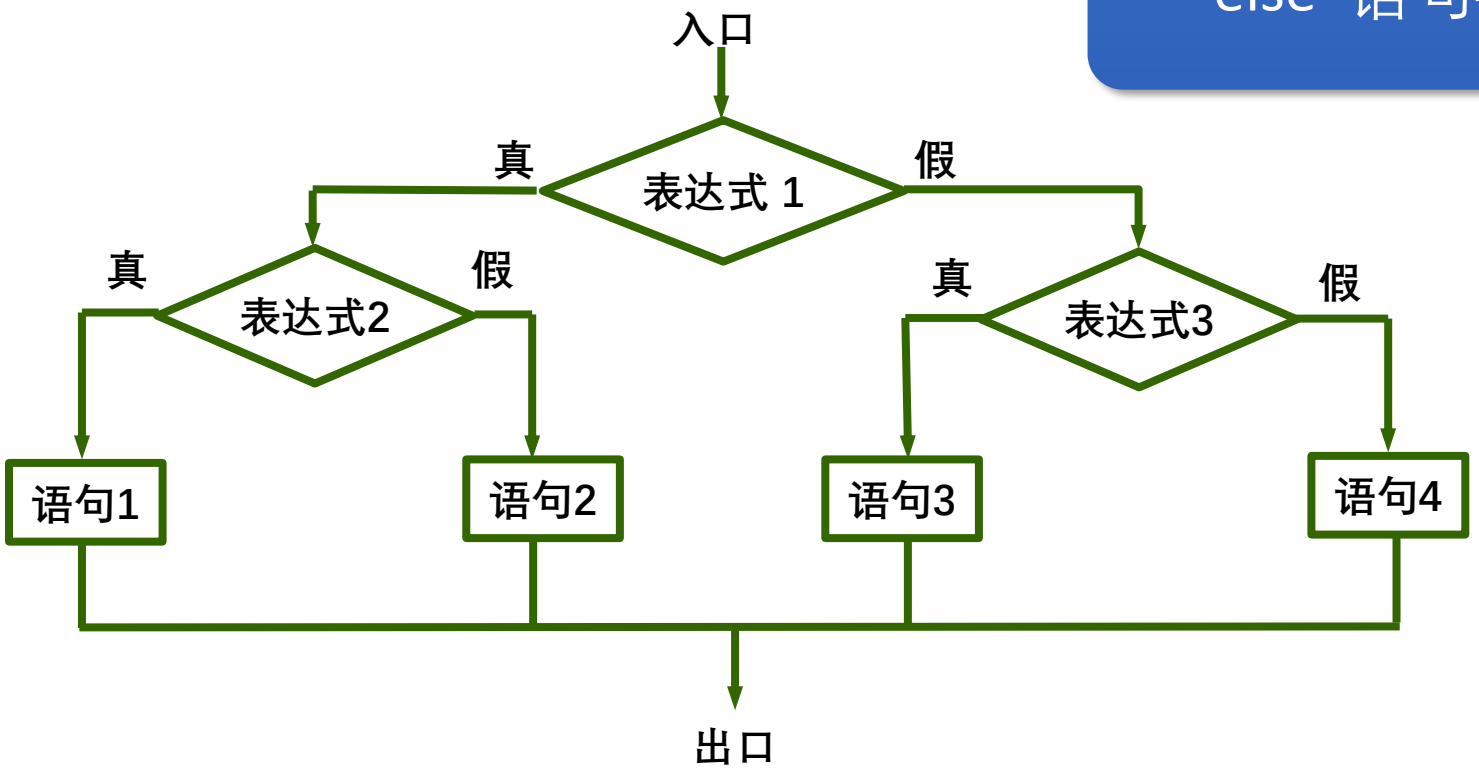
if/else if语句的流程图

```
if (表达式1)
    语句1
else if(表达式2)
    语句2
……
else if(表达式n-1)
    语句n-1
else
    语句n
```



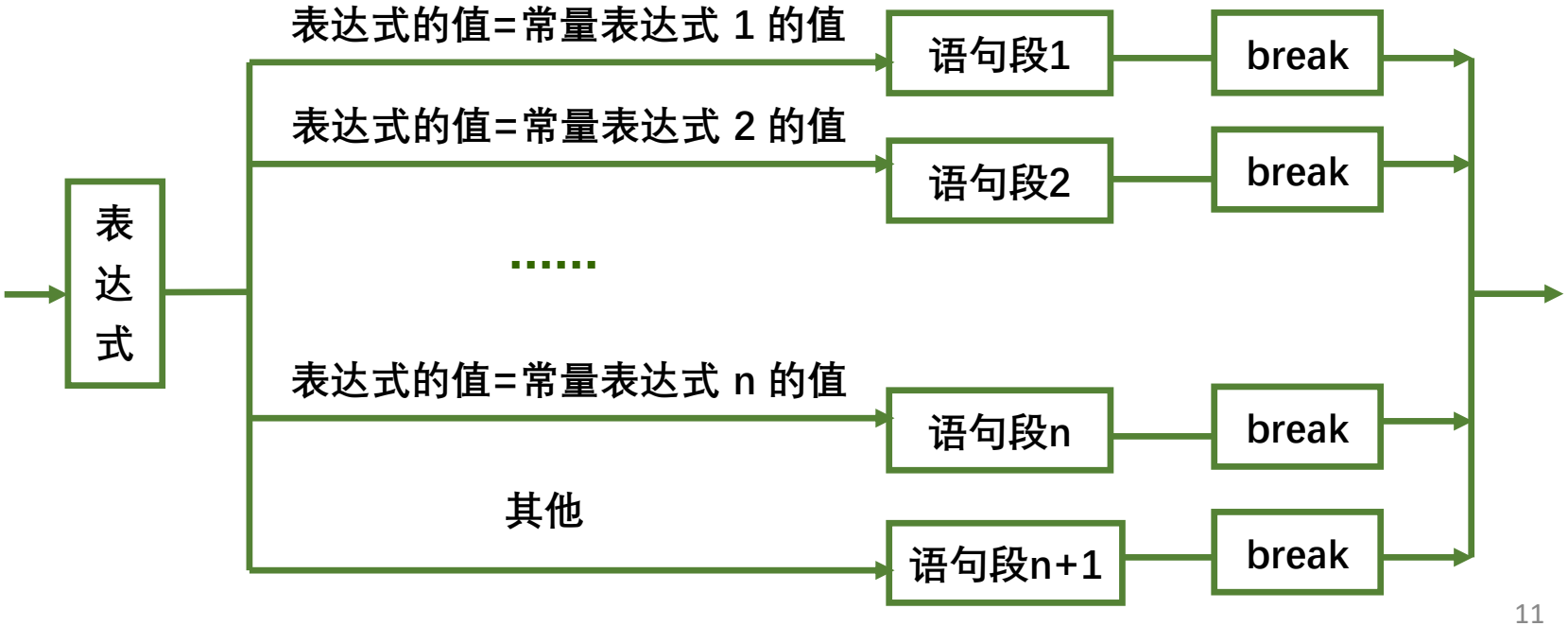
嵌套的 if – else 语句

```
if(表达式1)
    if(表达式2) 语句1
    else  语句2
else
    if(表达式3) 语句3
    else  语句4
```



switch语句流程图

- 从哪个语句段开始执行?
  - 有n+1个入口可以选择
  - 分别对应n个case的入口和1个缺省(default）的入口
- 根据表达式的值决定执行入口
  - 如果表达式等于常量表达式k，那么从语句段k开始执行



循环结构内容

- while循环结构
- do-while循环结构
- break语句和continue语句
- 多重/嵌套循环

循环结构内容

- while循环结构
- do-while循环结构
- break语句和continue语句
- 多重/嵌套循环

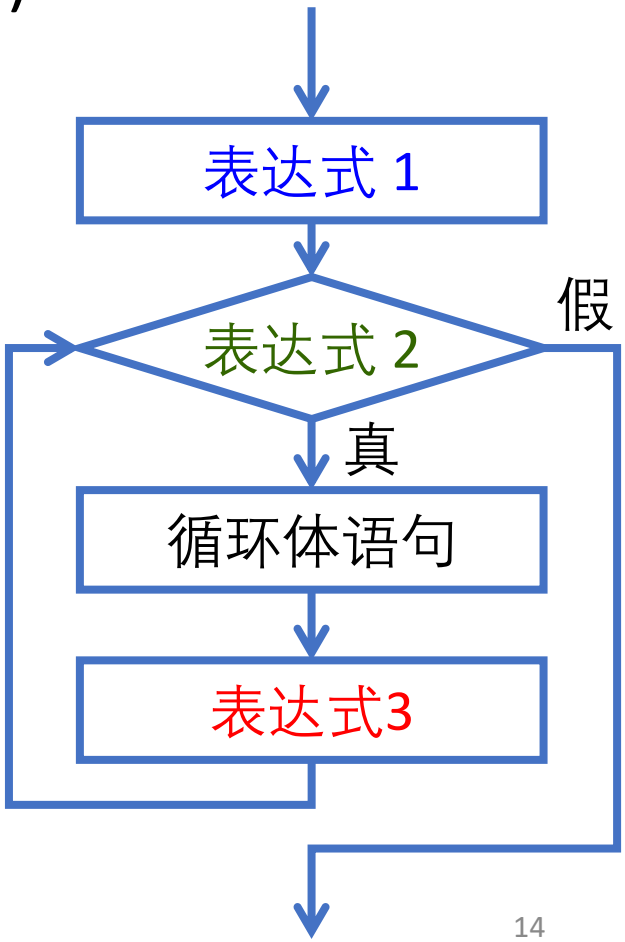
for语句回顾

适用于：给定次数的循环

for(表达式1； 表达式2； 表达式3)

循环体语句

```
for( i = 1 ; i <= n ; i ++ )
{
    item = 1.0 / i;
    sum = sum + item;
}
```



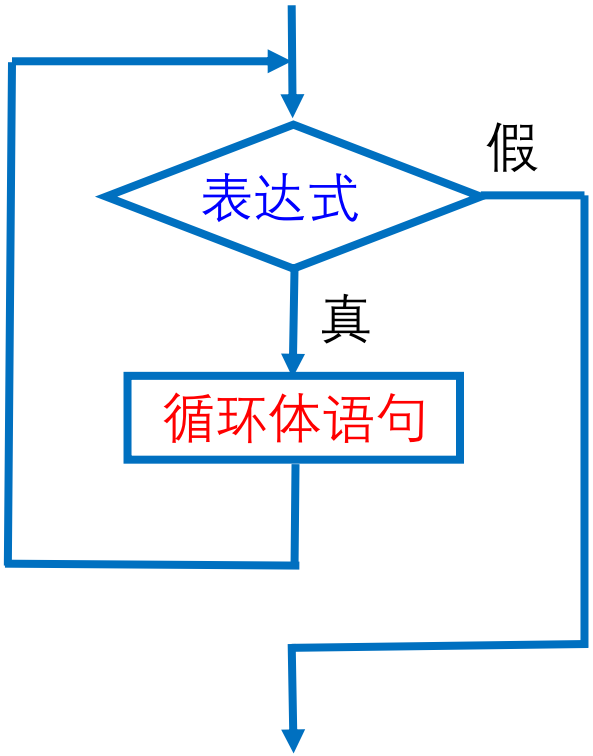
while循环语句

while(表达式)

循环体语句

当表达式成立的时候

执行循环操作

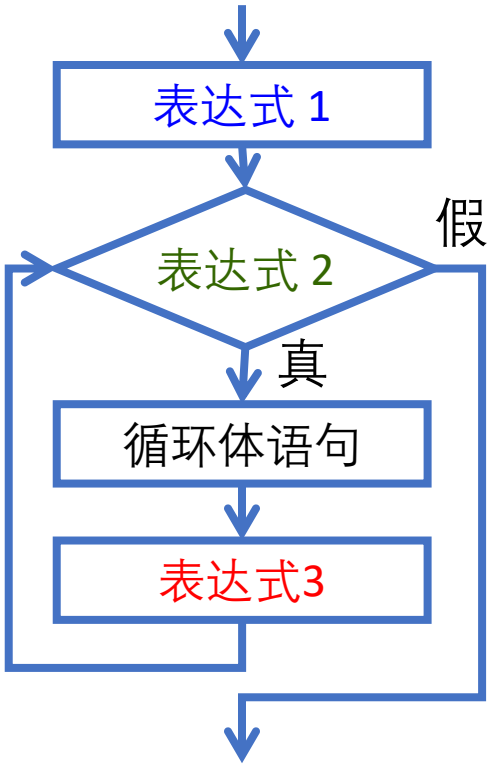


适用于：**不能显式地确定次数的循环**

for与while对比

for(表达式1； 表达式2； 表达式3)

循环体语句

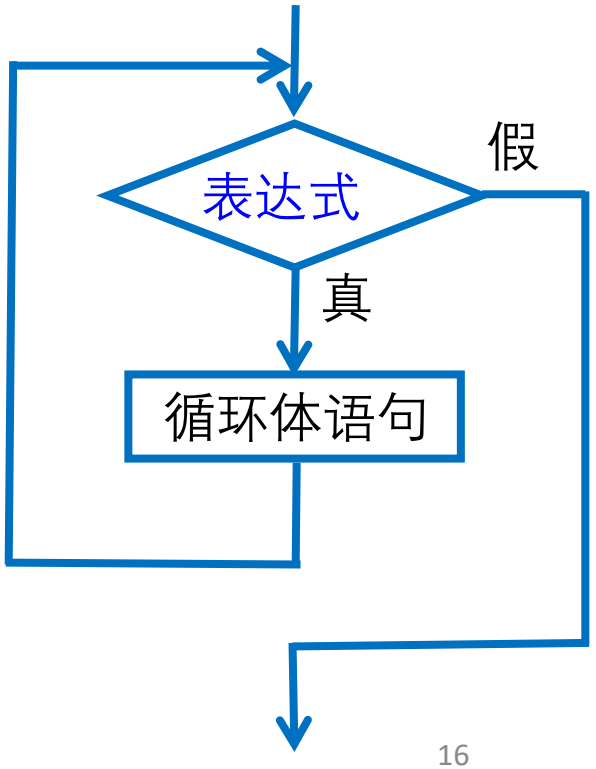


将for改为while

```
表达式1;
while( 表达式2 )
{
    循环体语句
    表达式3;
}
```

while(表达式)

循环体语句



while语句应用(1)

[例4-1，P70] 使用格雷戈里公式求π，要求最后一项绝对值小于10^{-4}。

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots$$

double sum, item;

int denominator, flag;

用格雷戈里公式求π

```
sum = 0.0; denominator = 1; flag = 1;
item = 1.0;
```

变量的初始化非常重要!

```
while( fabs(item) >= 0.0001 )
{
    item = flag * 1.0/denominator;
    sum = sum + item;
    flag = -flag;          /* 准备下一项 */
    denominator = denominator + 2;
}

printf( "pi = %.4f\n", sum * 4 );
```

while语句应用(2)

[例4-2，P72] 输入一批学生成绩，以负数作为结束标志，计算平均成绩，统计不及格人数。

学生个数未知
适合while循环

```
int num, failed;
double grade, sum;

num = failed = 0;
sum = 0.0;

printf("Enter grades:");
```

变量的初始化非常重要!

统计学生成绩

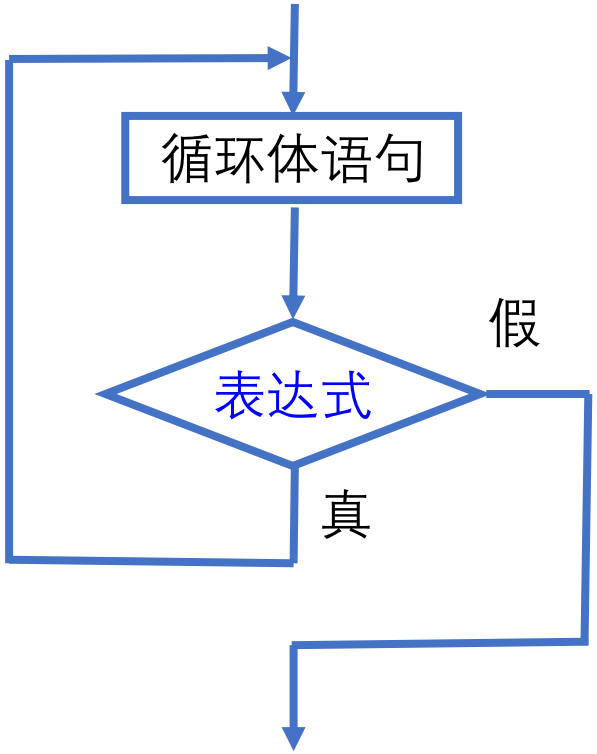
```
scanf("%lf", &grade);

while( grade >= 0 )
{
   sum = sum + grade;
   num ++;
   if( grade < 60 )        failed ++;

   printf("Enter next grade (end with a negative number):")
   scanf("%lf", &grade);
}

if( num != 0 )    /* 检查分母是否为零，良好的编程习惯 */
   printf("Grade average is %f\n", sum/num);

printf("Number of failures is %d\n", failed);
```

循环结构内容

- while循环结构
- do-while循环结构
- break语句和continue语句
- 多重/嵌套循环

do-while语句

```
do {
    循环体语句
} while( 表达式 )
```

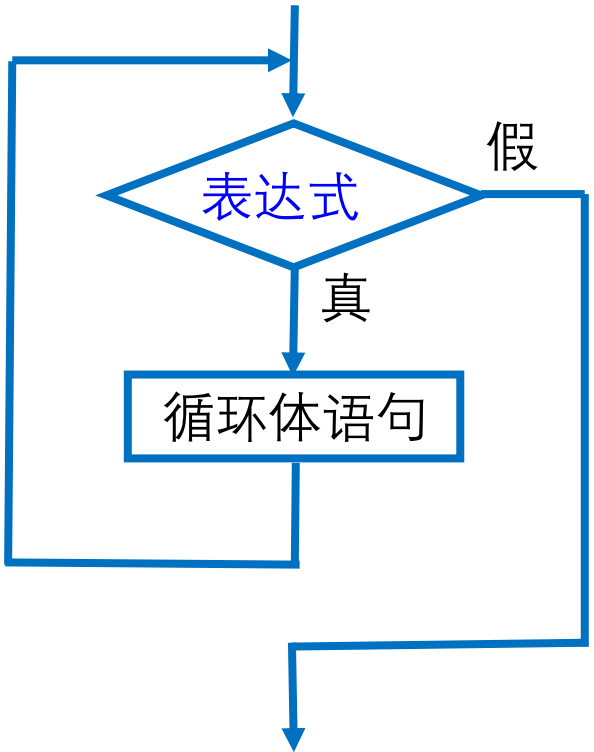


先执行循环操作

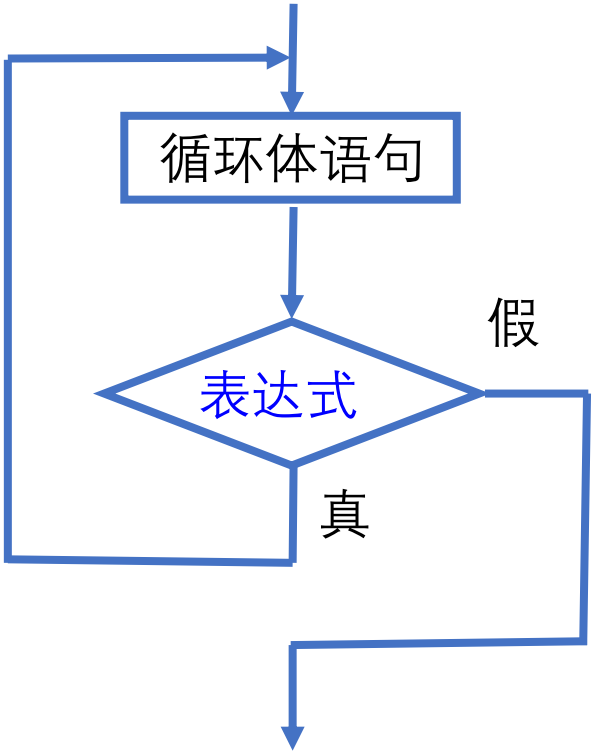
直到表达式为假

while 与 do - while

```
while( 表达式 ) {
    循环体语句
}
```




```
do {
    循环体语句
} while( 表达式 )
```



do-while语句应用

[例4-3，P74] 计算一个整数的位数。例如：0的位数为1，-99的位数为2。

```
int number, count;

printf(“Enter a number: ”);
scanf(“%d”, &number);
```

算法：如何计算一个整数的**位数**？

思想：不断用**10**去除它，直到**零**为止

计算整数的位数

```
count = 0;

do {
    number = number / 10;
    count ++;
} while ( number!=0 );

printf("it contains %d digitals.\n", count);
```

改为下面的while循环如何?

```
count = 1;
while ( abs(number) > 9 )
{
  count ++;
  number = number / 10;
}
```

循环结构内容

- while循环结构
- do-while循环结构
- break语句和continue语句
- 多重/嵌套循环

循环体中break和continue语句

[例4-4,P76] 素数的判定问题：除了1和自身，不能被别的数整除

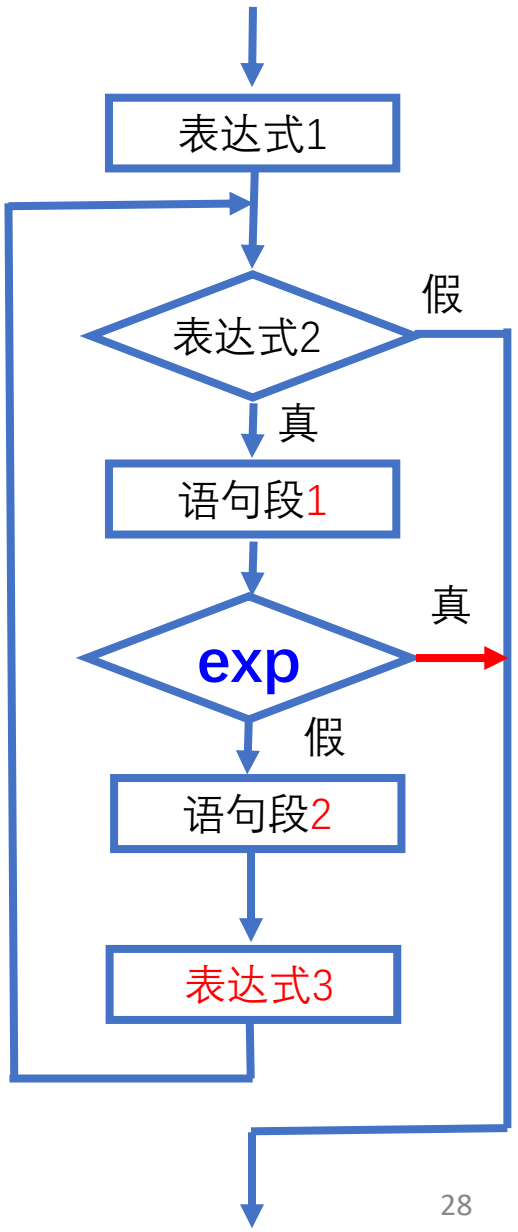
```
int i, m;

scanf(“%d”, &m);

for( i = 2; i <= m/2; i++ )
    if( m % i == 0 )
        break;

if( i > m/2 && m != 1 )
    printf("Yes\n");
else
    printf("No!\n");
```

注：用中间变量替换m/2，减少不必要的计算

```
int   mid_m
mid_m = m/2
```

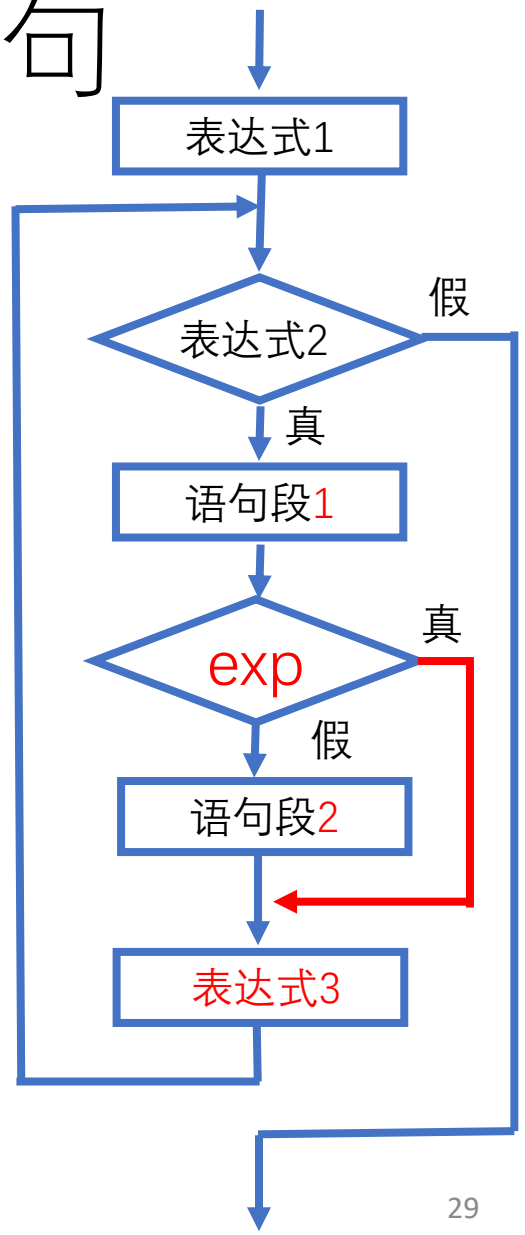
for循环中的break语句

```
for( 表达式1；表达式2；表达式3 )
{
    语句段1
    if( exp )
        break;
    语句段2
}
/* break：终止整个循环 */
```



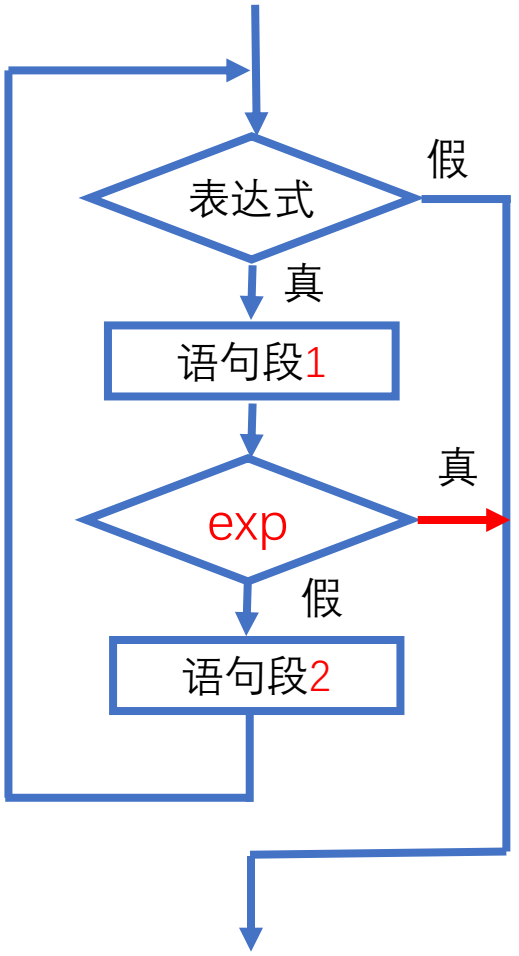
for循环中的continue语句

```
for( 表达式1；表达式2；表达式3 )
{
   语句段1
   if( exp )
     continue;
   语句段2
}
/* continue：终止循环的一步 */
```



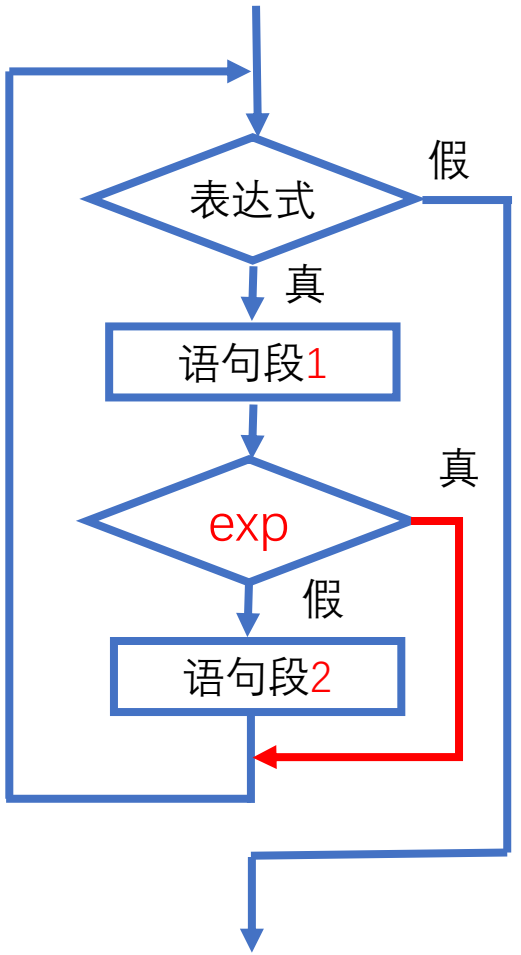
while循环中的break语句

```
while( 表达式 )
{
    语句段1

    if( exp )
        break;

    语句段2
}
/* break：终止整个循环 */
```



while循环中的continue语句

```
while( 表达式 )
{
    语句段1

    if( exp )
      continue;

    语句段2
}
/* continue：终止循环的一步 */
```



循环体中break和continue语句

- 适用于所有的循环结构
  - for, while, do-while

- 用于灵活控制循环结构的执行流程

循环结构内容

- while循环结构
- do-while循环结构
- break语句和continue语句
- 多重/嵌套循环

多重嵌套循环

- for, while, do-while 能够相互嵌套
  - 构成多重循环
- 处理多维数据
- 处理复杂过程

多重嵌套循环

假设有一幅宽度为width、高度为height的图像，
使用for-for嵌套语句，对它的每一个像素进行处理

```
for( i = 0; i < width; i ++ )
{
    for( j = 0; j < height; j ++ )
    {
        /* 处理坐标为(i,j)的图像像素 */
        …… ……
    }
}
```

省略{ }

循环体被执行了多少次?

width * height

多重嵌套循环

假设有一幅宽度为width、高度为height的图像，
使用for-for嵌套语句，对它的每一个像素进行处理

```
for( i = 0; i < width; i ++ )
for( j = 0; j < height; j ++ )
{
    /* 处理坐标为(x,y)的图像像素 */
    … … …
}
```

循环体被执行了多少次?

width * height

循环结构的设计

- 循环程序的实现要点：
  - 归纳出哪些操作需要反复执行？ 循环体
  - 这些操作在什么情况下重复执行？ 循环条件
- 选用合适的循环语句

  for while do-while
- 循环具体实现时考虑（循环条件）：
  - 事先给定循环次数，首选for
  - 通过其它条件控制循环，考虑while或do-while
    - 至少执行1次，用do-while
    - 否则，用while

[例4-8，P85] n个成绩的最高分

```
int n, i, mark, max;

printf(“Enter n: ”);
scanf (“%d”, &n);
printf(“Enter %d marks: ”, n);

scanf (“%d”, &mark);
max = mark;

/* 剩余成绩循环处理 */
/* 次数已知，适合for语句 */
```

[例4-8，P85] n个成绩的最高分

```
for( i = 1; i < n; i++ )
{
    scanf (“%d”, &mark);

    if( max < mark )
        max = mark;
}

printf("Max = %d\n", max);
```

[P86] 输入一批成绩，以负数结尾求最高分

```
int mark, max;

printf("Enter marks: ");

scanf ("%d", &mark);
max = mark;

/* 剩余成绩循环处理 */
/* 次数未知，适合while语句 */
```

[P83] 输入一批成绩，以负数结尾求最高分

```
while( mark >= 0 )
{
    if( max < mark )
        max = mark;

    scanf ("%d", &mark);
}

printf("Max = %d\n", max);
```

请思考：如果第一次就是输入的负数，结果是否合适？如何修改？

[例4-4 P76] 将整数按数字逆序输出

例如：x = 12345 的逆序为 54321

如何得到呢?

从低位开始逐个计算

5 = x % 10, x ← x/10 = 1234

4 = x % 10, x ← x/10 = 123

3 = x % 10, x ← x/10 = 12

2 = x % 10, x ← x/10 = 1

1 = x % 10, x ← x/10 = 0 [结束]

[例4-4 P76] 将整数按数字逆序输出

```
int x;

printf(“Enter x: ”);
scanf (“%d”, &x);

/* 对每一位数字进行循环处理 */
/* 次数未知，适合while语句 */
```

[例4-4 P76] 将整数按数字逆序输出

```
while( x != 0 )
{
    printf( "%d", x%10 );
    x = x/10;
}
```

如果输入x = 0，结果如何？

[例4-4 P76] 将整数按数字逆序输出

```
/* 用 do-while 实现 */
do {
    printf( "%d", x%10 );
    x = x/10;
} while( x != 0 )
```

用 do-while 实现更好，对x = 0也正确!

求100以内的素数每行输出10个

- 需要考察的整数范围

  2,3,4,…,100

- 素数：没有真因子

  对于整数 m，真因子的范围是：

  - 1 < 真因子 < m
  - 1 < 真因子 <= m/2：仍旧保守的边界
  - 如非素数，必有真因子满足：1 < 真因子 <= $m^{1/2}$

- 每行输出10个
  - 当素数个数是10的倍数时，输出换行符：printf("\n");

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

int count, m, n, i;

count = 0;
for( m = 2; m<100; m++ )
{
    n = sqrt(m);
    for( i=2; i<=n; i++ )
        if( m%i==0 )
            break;
    if( i<=n )
        continue;
    printf("%6d", m);
    count ++;
    if( count%10==0 )
        printf("\n");
}
```

[例4-9, P87]计算并输出：斐波那契数列前10项

- 1 1 2 3 5 8 13 21 …
- 从第3项起，等于前2项之和

```
int x1, x2, x, i;
x1 = x2 = 1;
printf("%6d%6d", x1, x2);
```

[例4-11, P86]计算并输出：斐波那契数列前10项

```
for( i=3; i<=10; i++ )
{
    x = x1 + x2;
    printf("%6d",x);
    x1 = x2;
    x2 = x;
}
printf("\n");
```

[例4-11, P90]穷举算法(搬砖)

- 男人：3块/人
- 女人：2块/人
- 小孩：1块/2人
- 问：45人搬45块砖，有多少种搬法？

int men, women, child;

[例4-11, P90]穷举算法(搬砖)

```
for( men=0; men<=45; men++ )
for( women=0; women<=45; women++ )
for( child=0; child<=45; child++ )
  if( men + women + child == 45 &&
      men * 3 + women * 2 + child * 0.5 == 45 )
  {
    printf("men=%d, women=%d, child=%d\n",
           men, women, child);
  }
```

合计执行多少次循环操作?

46*46*46

改进[1]

```
for( men=0; men<=45; men++ )
for( women=0; women<=45; women++ )
{
   child = 45-men-women;
   if( child>=0 &&
      men*3+women*2+child*0.5==45 )
   {
     printf("men=%d, women=%d, child=%d\n",
           men, women, child);
   }
}
```

合计执行多少次循环操作?

46*46

改进[2]

编写循环程序时
注意代码的执行效率

```
for( men=0; men<=15; men++ )
for( women=0; women<=22; women++ )
{
    child = 45-men-women;
    if( men*3+women*2+child*0.5==45 )
    {
      printf("men=%d, women=%d, child=%d\n",
            men, women, child);
    }
}
```

合计执行多少次循环操作?

16*23

要点

- for
- while
- do-while
- break
- continue
- 多重循环
  - 循环次数的计算
  - 执行效率