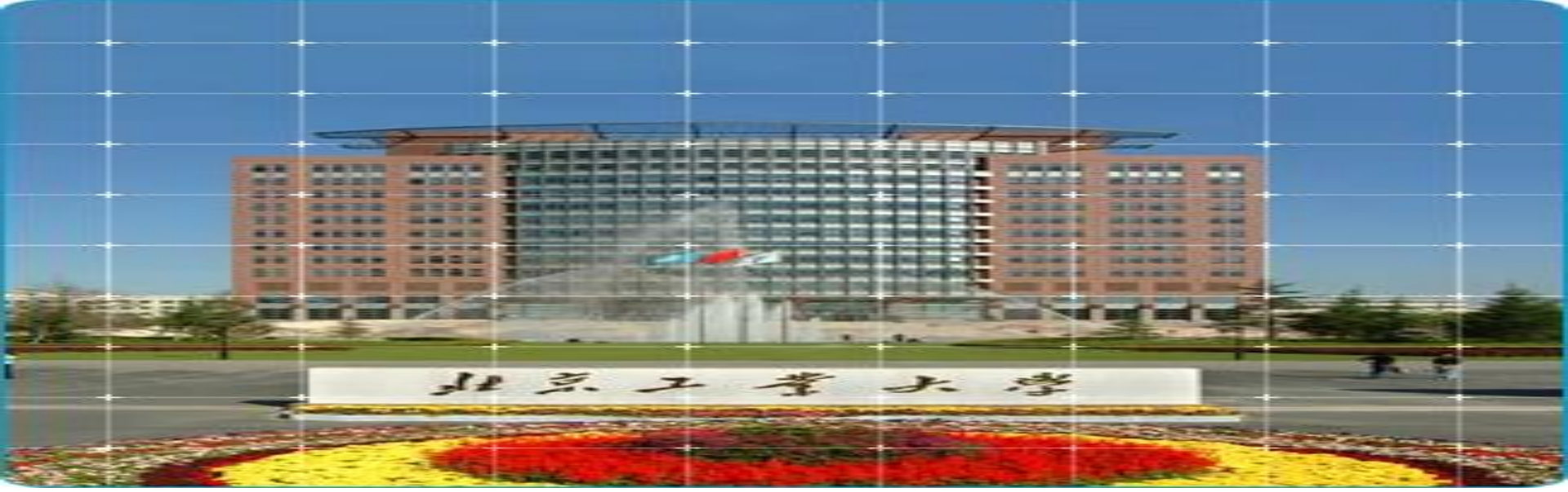


C语言程序设计

第 2 章　循环结构

2.1 求解100以内所有偶数的和（for语句）

2.2 求解n!（for语句、while语句）

2.3 求解两个数的最大公约数(while语句)

2.4 用格里高利公式求π的近似值（do-while语句）

2.5 统计一个整数的位数（do-while语句）

2.6 判断素数（break 和 continue 语句）

2.7 求1! + 2! + …. + 100!（嵌套循环）

2.8 说谎族和诚实族（综合案例）

本章问题

- 使用for、while 和do-while语句实现循环结构?
- while 和do-while语句有什么不同?
- 如何使用break语句处理多循环条件?
- 如何实现多重循环?

2.1 求偶数和

求解100以内所有偶数的和。

sum=2+4+6+…

2.1.1　问题分析

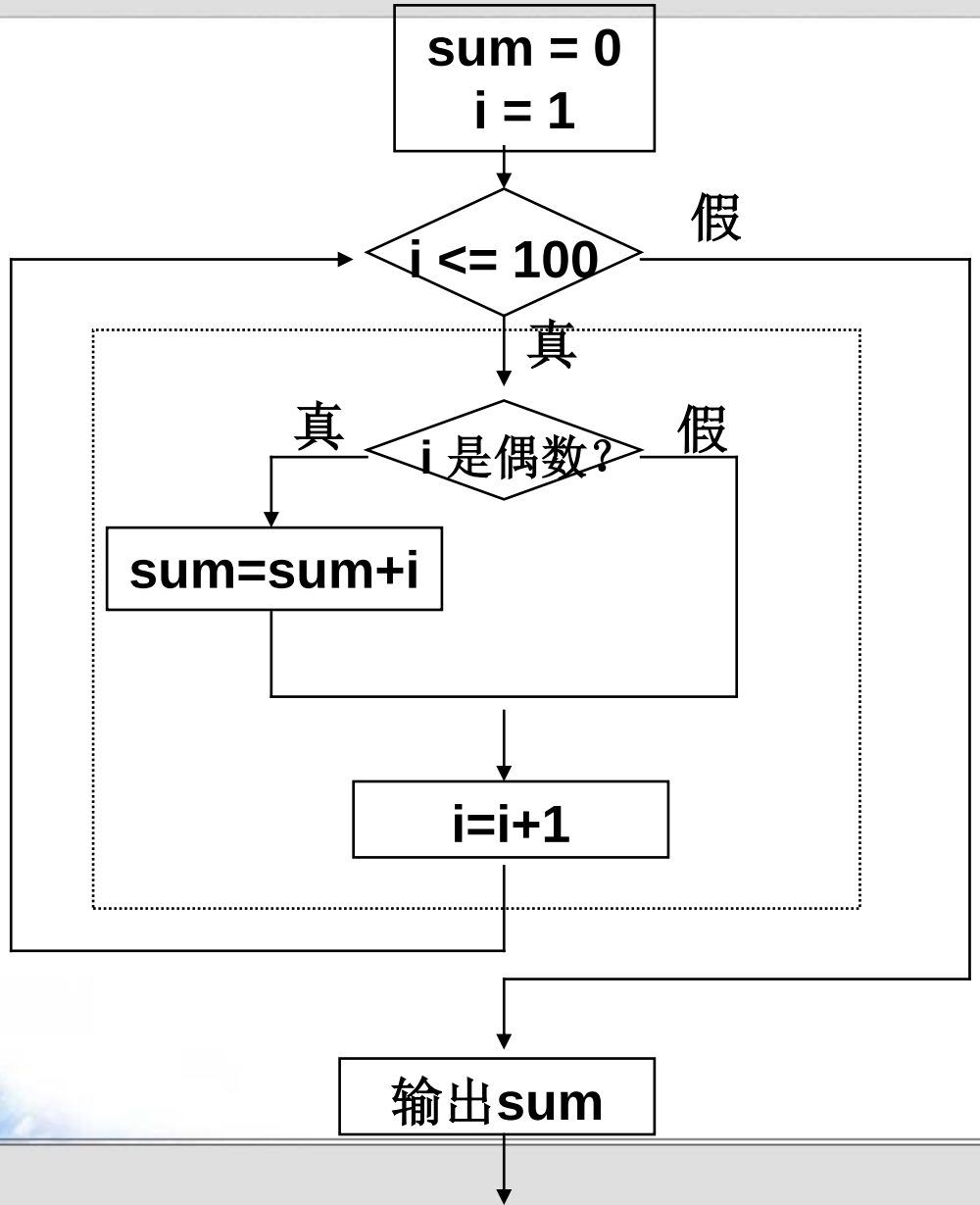
2.1.2　for语句

2.1.1问题分析

求在一定范围内（1～100）、满足一定条件(偶数)的若干整数的和，求累加和。

思路：设置一个变量(sum)，其初值为0，然后在1～100的数中(i)寻找偶数，将它们一个一个累加到sum中。

- 一步累加：sum = sum + i;
- 重复累加，用循环语句实现，在循环过程中：
  - (1) 判别 i 是不是偶数：用分支控制语句来实现。
  - (2) 对循环次数进行控制：通过 i 值的变化


sum = 0
i = 1
i <= 100
假
真
真
i 是偶数?
假
sum=sum+i
i=i+1
输出sum

程序实现

```
#include <stdio.h>
int main(void)
{
  int i, sum = 0;

  for(i = 1; i <= 100; i++)
        if (i%2 == 0)
              sum = sum + i;
     printf("%d\n", sum);
     return 0;
}
```

2.1.2 for语句

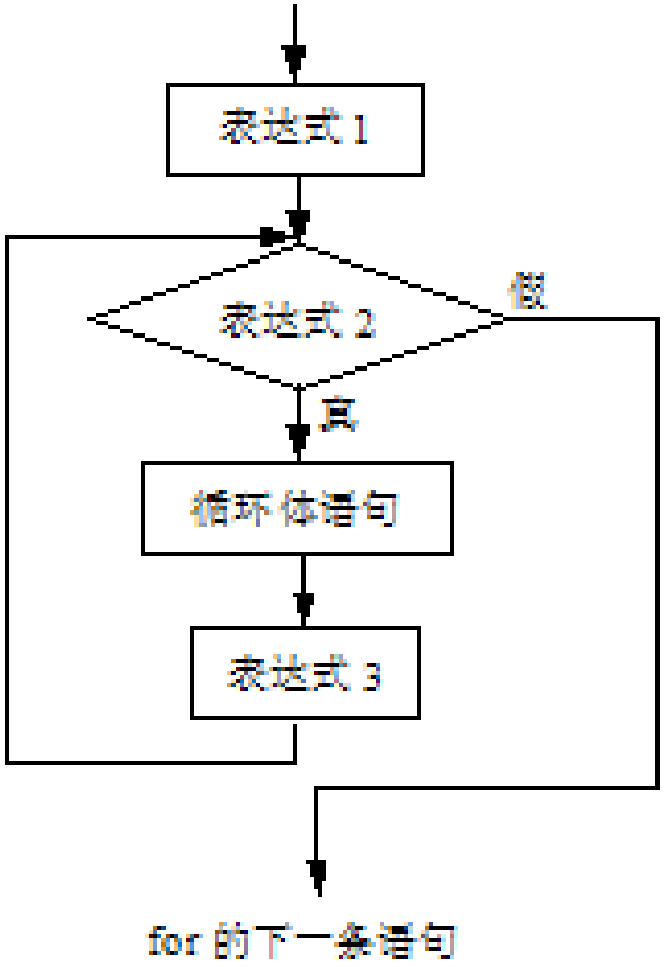
for (表达式1; 表达式2;表达式3)
　　循环体语句

实现C语句的重复执行

3个表达式、循环体语句

！ 书写顺序和执行顺序不同

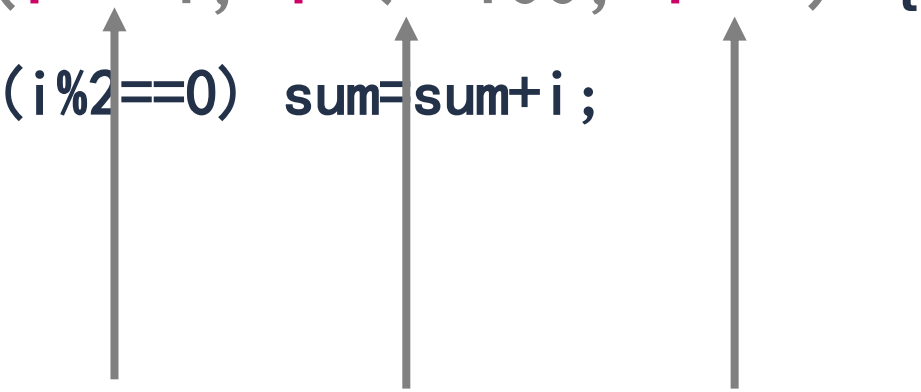
！ 表达式1只执行一次



for语句中的循环变量

循环(控制)变量：for语句中，通过改变或判断某个变量的值来控制循环的执行

```
for (i = 1; i <= 100; i ++) {
    if(i%2==0) sum=sum+i;
}
```



for语句的说明

```
for (i = 1; i <= 100; i ++) {
    if(i%2==0) sum=sum+i;
}
```

表达式1：给循环变量赋初值，指定循环的起点。

i = 1

表达式2：给出循环的条件，判断循环是否达到终点？

i <=100

表达式3：设置循环的步长，改变循环变量的值，从而可改变表达式2的真假性。

i ++

循环体语句：被反复执行的语句，一条语句。

2.2 求解n!

n! = 1×2×3×4×…×n

2.2.1 问题分析

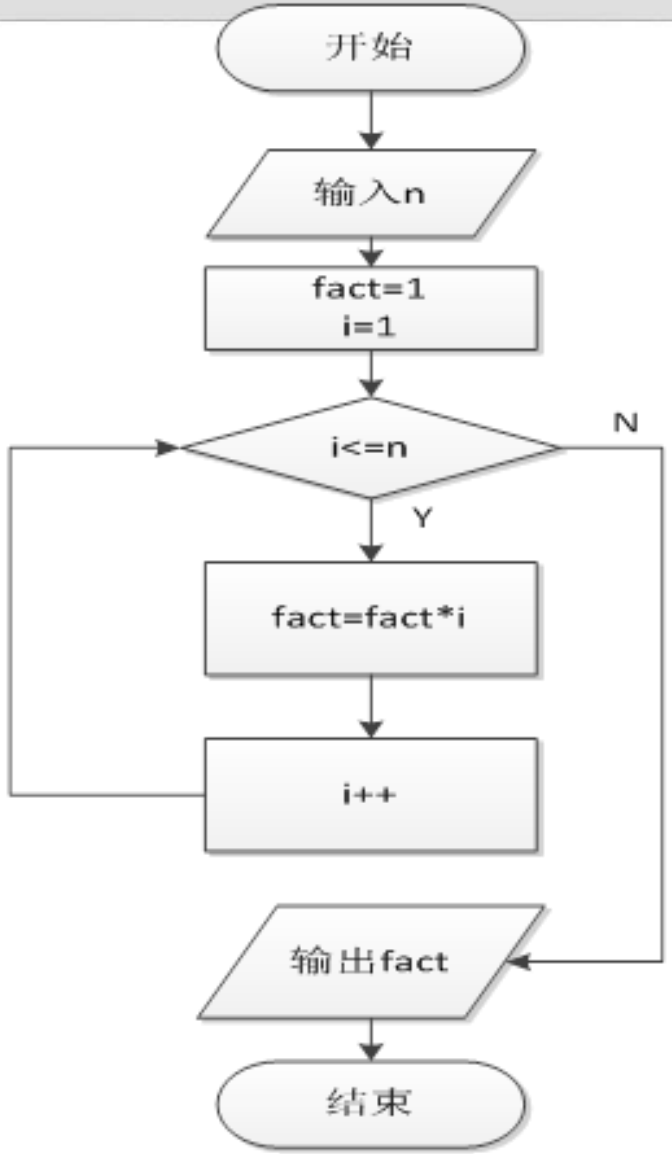
2.2.2 for语句、while语句

2.2.1问题分析

求在一定范围内（1～n）、若干整数的乘积，求累乘。

思路：设置一个变量(fact)，其初值为1，然后在1～n的数中依次将它们累乘到fact中。

- 乘积操作：fact=fact* i;
- 重复累乘，用循环语句实现，在循环过程中通过i值的变化对循环次数进行控制；
- 分别使用for语句和while语句实现循环结构；


开始
输入n
fact=1
i=1
i<=n
N
Y
fact=fact*i
i++
输出fact
结束

for 语句实现 while 语句

```
#include <stdio.h>
main( )
{
 int i, n,fact;
   printf ("input n:");
   scanf ("%d", &n);
   fact=1;
for(i=1;i<=n;i++)
fact = fact*i;
printf("%d! = %d\n", n, fact);
return 0;
 }
```

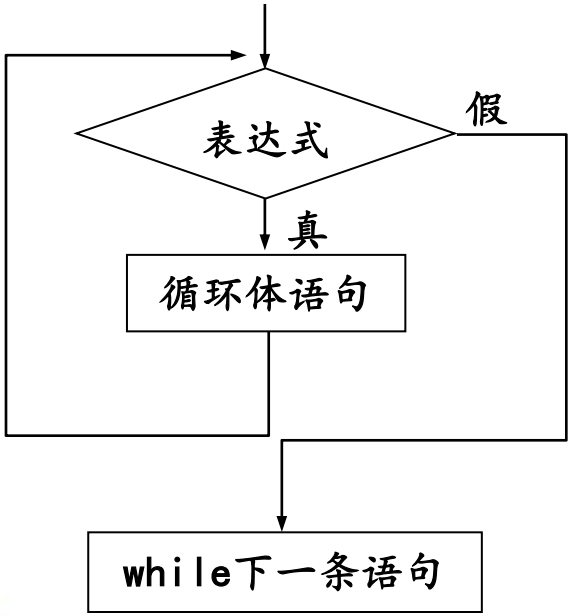
```
#include <stdio.h>
main( )
{
 int i, n,fact;
   printf ("input n:");
   scanf ("%d", &n);
   fact=1; i=1;
while(i<=n)
    {fact = fact*i;
     i++;}
printf("%d! = %d\n", n, fact);
return 0;
 }
```

2.2.2 while 语句

一条语句

while（条件）

循环体语句；



循环条件

循环体

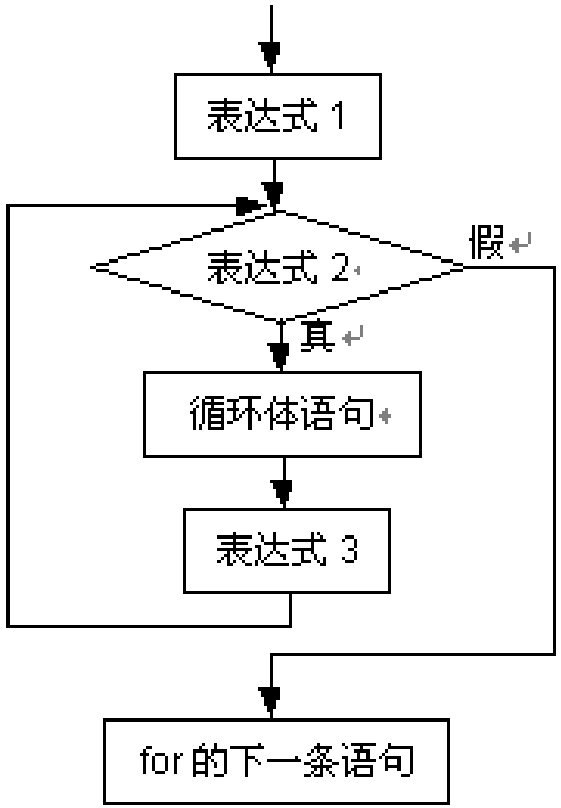


图 4-2　for 语句的执行流程

while 语句说明

while 语句和for语句

都是在循环前先判断条件

把for语句改写成while语句

for(表达式1；表达式2；表达式3)

循环体语句

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

while 和 for 的比较

```
for (i = 1; i <= n; i++)
    sum = sum + i;
```

```
i = 1;                循环变量赋初值
while (i <= n) {      循环条件
    sum = sum + i;
    i++;              循环变量的改变
}
```

循环体

2.3 求解最大公约数

输入两个整数，求解这两个数的最大公约数。

最大公约数：是能够同时整除这两个整数的最大的正整数。

2.3.1 问题分析

2.3.1问题分析

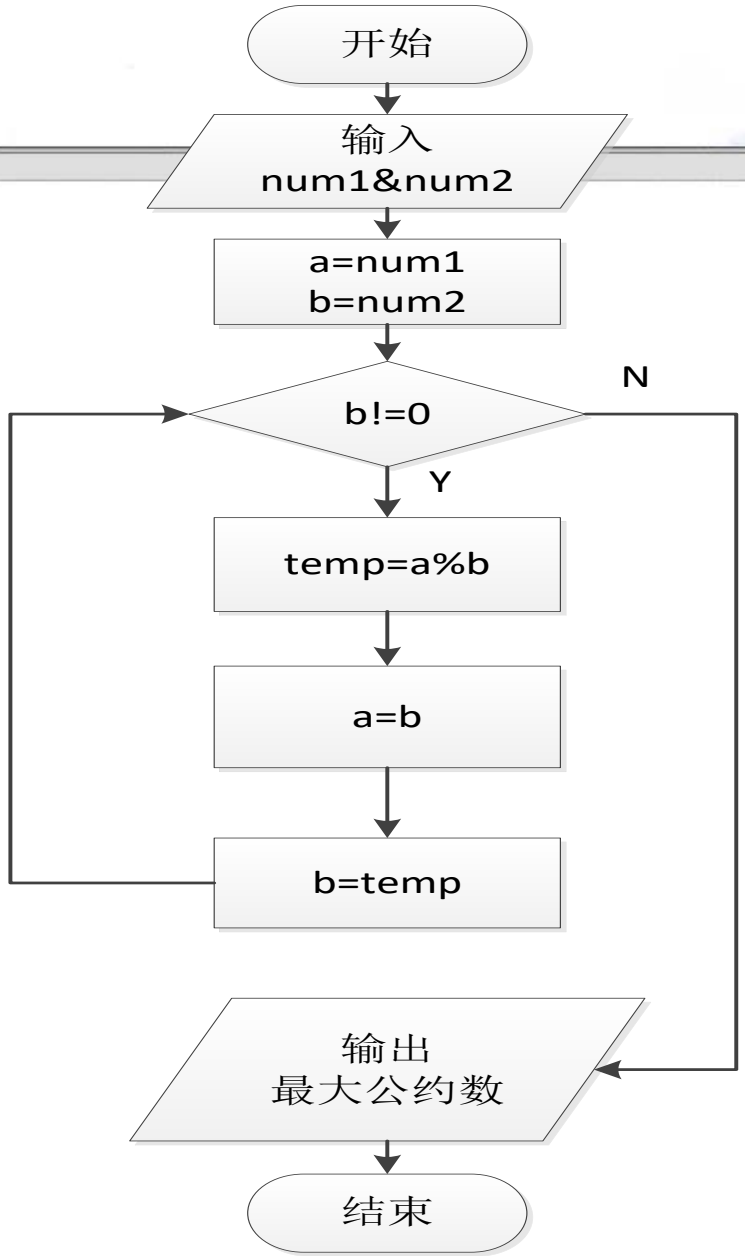
采取辗转相除法求两个数的最大公约数，辗转相除法基于如下原理：两个整数的最大公约数等于其中较小的数和两数相除余数的最大公约数。

思路：

- 将num1和num2赋值给变量a和b。
- 判断条件b!=0是否满足，

  如果满足，重复执行操作：temp=a%b, a=b, b=temp。

  如果不满足，即b=0，则输出a的值，即为num1和num2的最大公约数；


开始
输入
num1&num2
a=num1
b=num2
b!=0
N
Y
temp=a%b
a=b
b=temp
输出
最大公约数
结束

```
#include <stdio.h>
main( )
{int num1,num2;
   int a,b,temp;
printf("Input num1 & num2:");
   scanf("%d%d", &num1, &num2) ;
   a=num1;   b=num2;
while (b!=0 )
  { temp = a%b ;
    a = b;
    b = temp ;    }
if (num1!=0&&num2!=0)
printf("%d\n",a);
 }
```

2.4 用格里高利公式求π的近似值

使用格里高利公式求π的近似值，要求精确到最后一项的绝对值小于10^{-4}。

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

2.4.1 问题分析

2.4.2 do-while语句

2.4.1问题分析

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

这是一个数列求和的问题，即求累加和。

关键点1：数列的通项如何表示？

```
item = flag*1.0/i ;
flag = -flag ;
i = i+2 ;
```

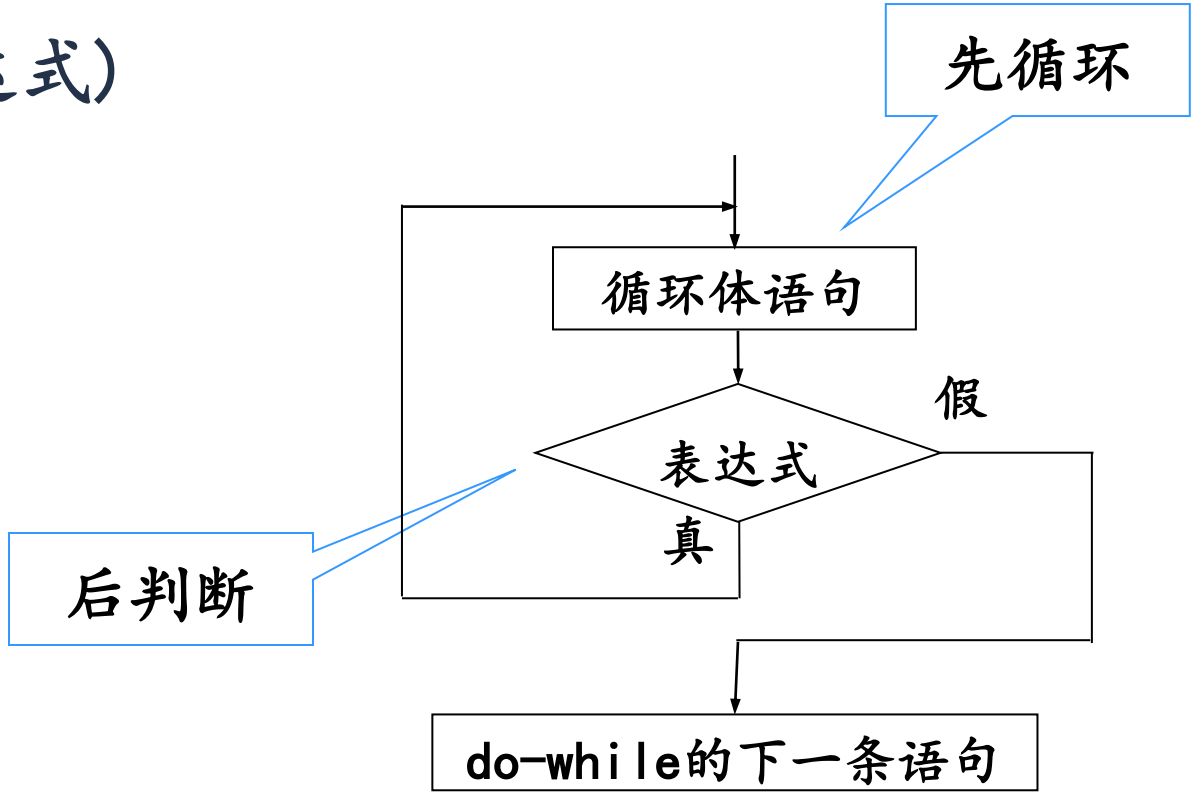
关键点2：循环继续的条件？

精确到最后一项的绝对值小于10^{-4} ，循环次数不确定

```
#include <stdio.h>
#include<math.h>
 main( )
{
    double item, sum, pi;
    int flag, i;
    i = 1;
    flag=1;
    sum=0;
    do {
        item = flag*1.0/i ;
        sum = sum + item;
        flag = -flag ;
        i = i+2 ;
    } while ( fabs (item) >= 0.0001 );
   pi = sum * 4 ;
   printf("pi = %lf\n", pi);
}
```

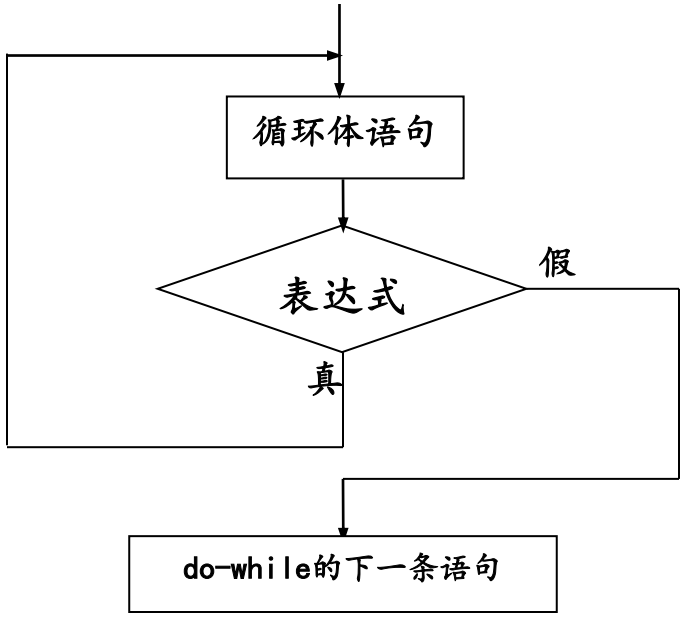
2.4.2 do - while 语句

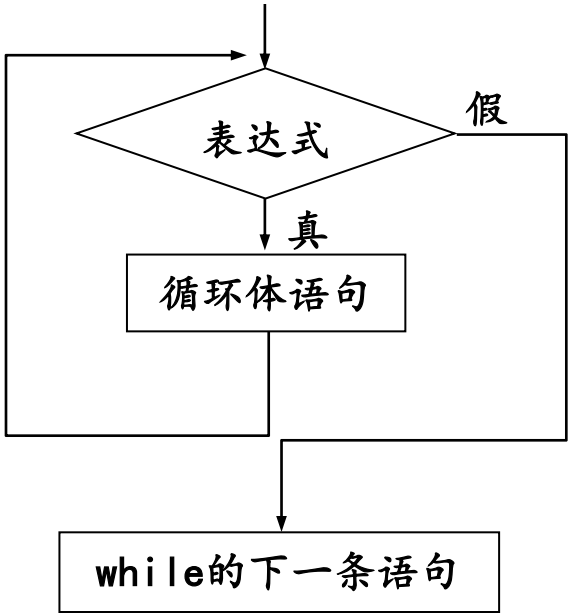
```
do {
    循环体语句
} while (表达式)
```



while 和 do-while 的比较

- while 是先判别条件，再决定是否循环；
- do-while 是先至少循环一次，然后再根据循环的结果决定是否继续循环。





循环结构程序设计

- 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
  - 通过其他条件控制循环，考虑while或do-while

2.5 统计一个整数的位数

从键盘读入一个整数，统计该数的位数。

2.5.1 问题分析

2.5.2 do - while语句

程序实现

```
int main(void)
{   int count, number;
    count = 0;
    printf("Enter a number: ");
    scanf ("%d", &number) ;
    if (number < 0)    number = -number;
    do {
            number = number / 10;
                    count ++;
    } while (number != 0);
    printf("It contains %d digits.\n", count);
      return 0;
}
```

```
Enter a number: 12534
It contains 5 digits.
```

```
Enter a number: -99
It contains 2 digits.
```

```
Enter a number: 0
It contains 1 digits.
```

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

2.6 判断素数

输入一个正整数m，判断它是否为素数。

2.6.1　问题分析

2.6.2　break语句 和continue语句

2.6.1 问题分析

算法：除了1和m，不能被其它数整除。

```
for(i = 2; i <= m/2; i++)
        if(m % i == 0) break;    /*如果m能被某个i整数，则m不是素数，提前结束循环*/
if(i > m/2) printf("yes\n")    /*如果循环正常退出，说明m不能被任何一个i整除，则m是素数*/
else printf("no\n");
```

m		%2	%3	%4	%5	%(m-1)
不是素数	||	=0	=0			
是素数	&&	!=0	!=0			

m不可能被大于 m/2 的数整除

i 取值 [2, m-1] 、 [2, m/2] 、 [2, $\sqrt{m}$]

```
int main(void)
{  int i, m;
 printf(“Enter a number: ");
 scanf ("%d", &m);
 for (i = 2; i <= m/2; i++)
        if (m % i == 0)  break;
 if (i > m/2&&m!=1 )
      printf("%d is a prime number! \
n", m);
 else
}
```

```
Enter a number: 9
No
```

```
Enter a number: 11
11 is a prime number!
```

循环条件?
循环的结束条件?

```
for (i = 2; i <= m/2; i++)
    if (m % i == 0) printf("No!  n");
    else printf("%d is a prime number!\n", m);
```

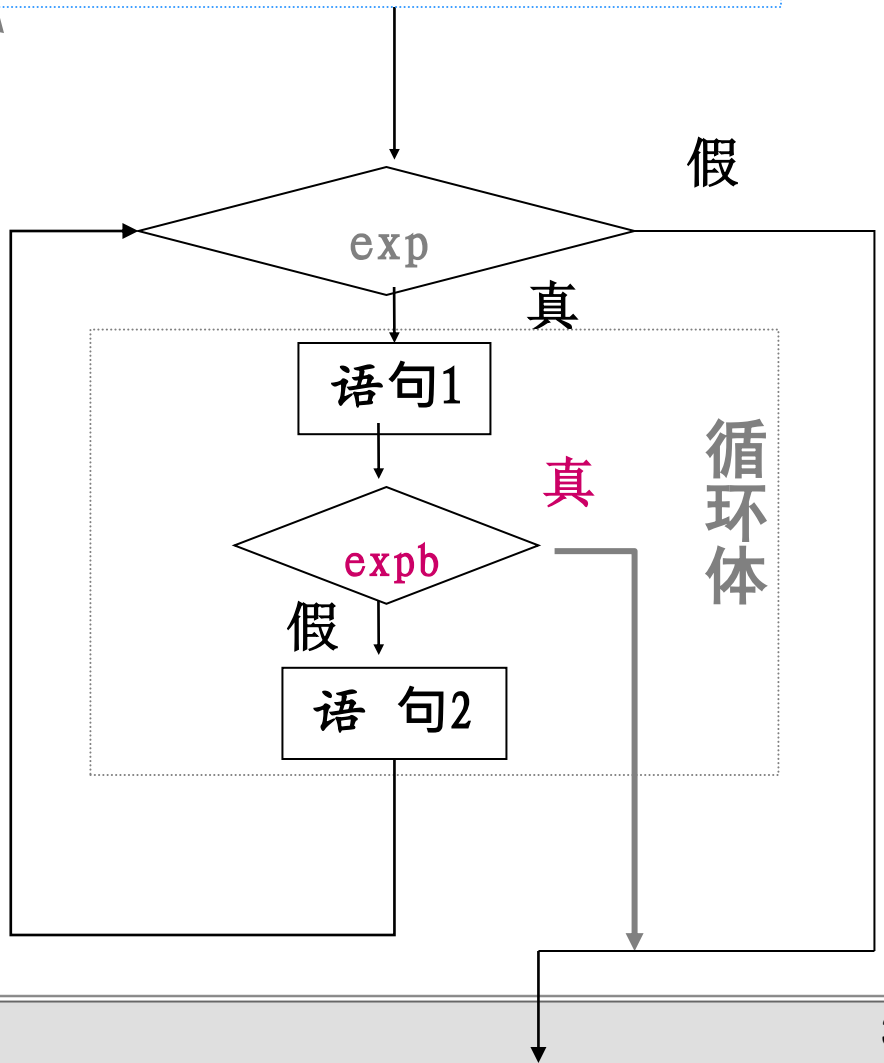
2.6.2 break 语句

```
for (i = 2; i <= m/2; i++)
if (m % i == 0)   break;
if (i > m/2 )   printf("Yes");
else   printf("No!  n");
```

```
while(exp){
    语句1
    if (expb)   break;
    语句2
}
```

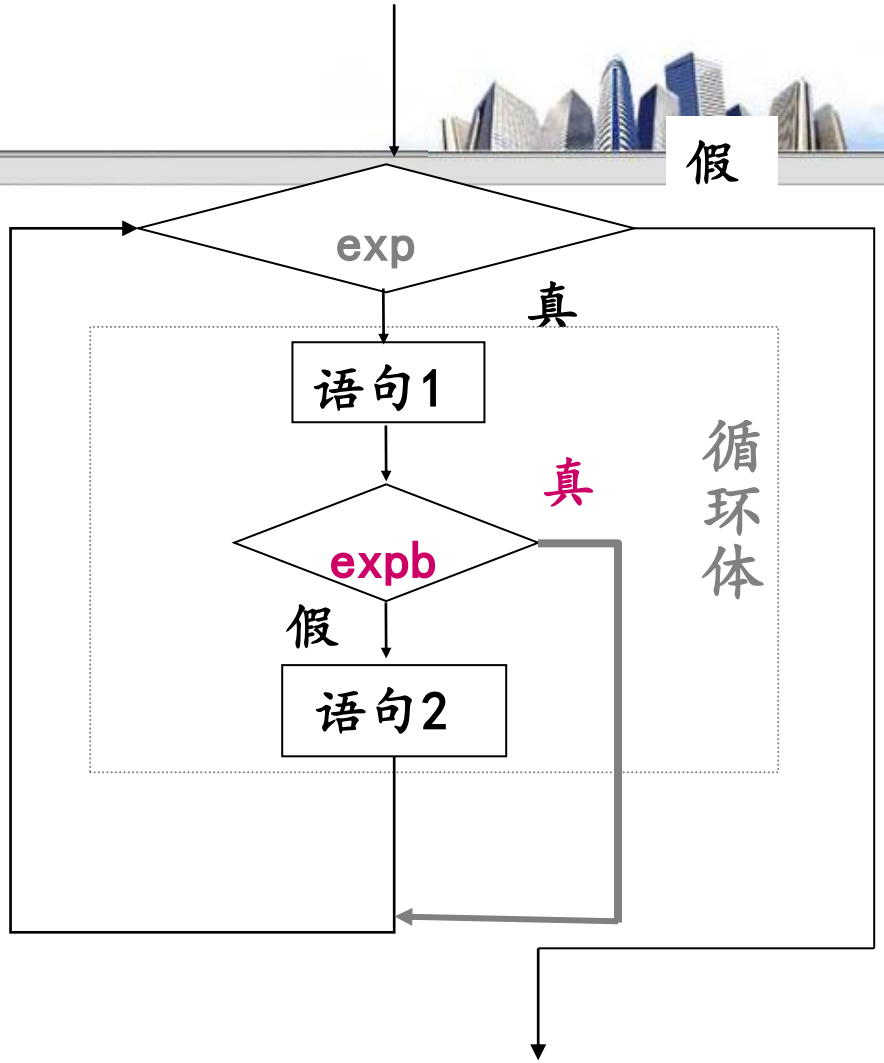
当循环有多个出口时：

- 表示循环条件
- 区分结束条件



continue 语句

```
while(exp){
    语句1
    if (expb)  continue;
    语句2
}
```



跳过continue后面的语句，继续下一次循环

break 和 continue

```
#include "stdio.h"
int main(void)
{  char c;
   int i;
   for (i = 0; i < 10; i++)
   {
        c = getchar();
        if (c == '\n')continue;
        putchar(c);
   }
}
```

```
abc↙
efgh ↙
123 ↙
```

```
abc
```

```
abcefgh1
```

break 和 continue

以下程序段输出100~200之间所有能被3整除的数。

```
for(i = 100;i <= 200;i++) {
      if( i % 3 != 0 )
            continue;
      printf(“%d  ”,i);
}
```

```
for(i = 100;i <= 200;i++)
    if(i % 3 == 0)
        printf(“%d  ”,i);
```

2.7 求1! + 2! + …. + 100!

```
for (i = 1; i <= 100; i++){
    item = i !
    sum = sum + item;
}
```

2.7.1嵌套循环

用循环计算 i 的阶乘

2.7.1 嵌套循环

```
for (i = 1; i <= 100; i++)
   {
      item = i !
      sum = sum + item;
}
```

```
for(i = 1; i <= 100; i++) {
    item = 1;
    for (j = 1; j <= i; j++)
        item = item * j;
    sum = sum + item;
}
```

```
#include <stdio.h>
int main(void)
{ int i, j;
  double item, sum;      sum = 0;
    for(i = 1; i <= 100; i++)
     {item = 1;
      for (j = 1; j <= i; j++)
          item = item * j;
      sum = sum + item;
  }
 printf("1! + 2! + 3! + … + 100! = %e\n", sum);
}
```

分析嵌套循环的执行过程

```
for(i = 1; i <= 100; i++) {
    item = 1;
    for (j = 1; j <= i; j++)
      item = item * j;
    sum = sum + item;
}
```

外层循环变量 i 的每个值

内层循环变量 j 变化一个轮次；

内外层循环变量不能相同

分别用 i 和 j

```
for (i = 1; i <= 100; i++)
    for (j = 1; j <= i; j++)
        printf ("%d %d\n", i, j );
```

i = 1	j = 1	输出 1 1 （第 1 次输出）
i = 2	j = 1	输出 2 1 （第 2 次输出）
	j = 2	输出 2 2 （第 3 次输出）
.....		
i = 100	j = 1	输出 100 1 （第 4951 次输出）
	j = 2	输出 100 2 （第 4952 次输出）
	.....	
	j = 100	输出 100 100 （第 5050 次输出）

2.8诚实族和说谎族

诚实族和说谎族是来自两个荒岛的不同民族，诚实族的人永远说真话，而说谎族的人永远说假话。

有一天小明遇到来自这两个民族的三个人，为了调查这三个人都是什么族的，小明问了他们一个问题，以下是他们的对话：

问：“你们是什么族？”，第一个人答：“我们之中有两个来自诚实族。”第二个人说：“不要胡说，我们三个人中只有一个是诚实族的。”第三个人听了第二个人的话后说：“对，就是只有一个诚实族的。”

请根据他们的回答判断他们分别是哪个族的

1. 问题分析

三个人分别是诚实族还是说谎族？

2. 解题思路

枚举法：将三个人可能出现的组合枚举一遍，找出满足条件的解。

解题思路

- 定义变量a, b, c分别表示3个人；
- 确定每个人的取值范围：0或1（1代表诚实族，0代表说谎族）；
- 对三个人的所有组合方式进行遍历，找出满足条件的解；

三重嵌套循环，遍历a, b, c的所有组合

```
for (a = 0; a <= 1; a++)
    for (b = 0; b <= 1; b++)
        for (c = 0; c <= 1; c++)
```

循环体：检查是否满足条件

```
(a && a+b+c==2 || !a && a+b+c!=2) &&
                  (b && a+b+c==1 || !b && a+b+c!=1) &&
                  (c && a+b+c==1 || !c && a+b+c!=1)
```

```
#include<stdio.h>
main( )
{    int a,b,c;
 for(a=0;a<=1;a++)          /*穷举各种可能性*/
   for(b=0;b<=1;b++)
     for(c=0;c<=1;c++)
      if( (a && a+b+c==2 || !a && a+b+c!=2) &&
                  (b && a+b+c==1 || !b && a+b+c!=1) &&
                  (c && a+b+c==1 || !c && a+b+c!=1))
                {   printf("A is a %s.\n",a?"honest":"liar");
                    printf("B is a %s.\n",b?"honest":"liar");
                    printf("C is a %s.\n",c?"honest":"liar");
                }
}
```

练习

1. 输入一批学生成绩，求最高分和平均分（当输入的成绩为负数时，输入结束）。
2. 输入一个正整数n,再输入n个整数，判断它们是否为素数。素数是只能被1和自身整除的正整数，1不是素数，2是素数。
3. 某地需要搬运砖块，已知男人一人搬3块，女人一人搬2块，小孩两人搬一块。问用45人正好搬45块砖，有多少种搬法？

本章总结

- 循环结构以及循环执行
  - while语句
  - do-while语句
  - for语句
- break语句与continue语
- 嵌套循环的使用
- 循环结构程序的综合设计
  - 几个常用的算法

- 正确理解 while语句和 do-while语句的执行机制；
- 掌握 break 和 continue 语句的作用方式；
- 掌握嵌套循环的执行机制与设计方法；
- 能合理运用循环语句熟练编写循环结构类的程序； 熟练掌握几个常用的算法；