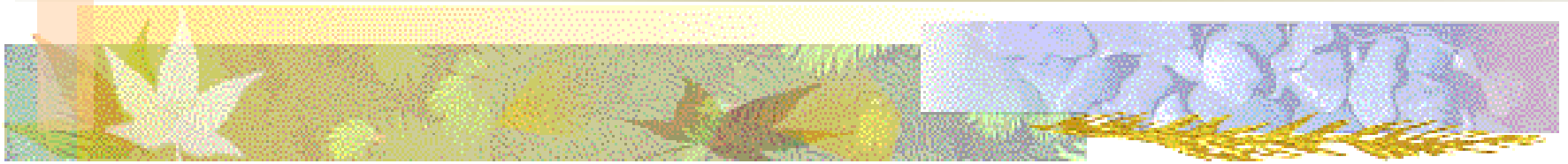
基本控制结构（续）



哈尔滨工业大学
苏小红
sxh@hit.edu.cn

```
while(1)
{
  printf("1. a + b\n");
  printf("2. a - b\n");
  printf("3. a * b\n");
  printf("4. a / b\n");
  printf("5. a %% b\n");
  printf("0. exit\n");
  printf("Input choice:");
  scanf("%d", &choice);
  switch(choice)
  {
     ……
  }
}
```

- 数位拆分与计算V6.0
  - 输出一个菜单，用户输入1~5，选择执行不同的运算，输入0则退出程序

```
switch (choice)
{
    case 1: printf("a+b=%d\n", a+b);  break;
    case 2: printf("a-b=%d\n", a-b);  break;
    case 3: printf("a*b=%d\n", a*b);  break;
    case 4:
            if (b != 0)
               printf("a/b=%d\n", a/b);
            else
            {
               printf("Division by zero!\n");
               exit(0);
            }
            break;
    case 5:
            if (b != 0)
               printf("a%%b=%d\n", a%b);
            else
            {
               printf("Division by zero!\n");
               exit(0);
            }
            break;
    case 0: exit(0);
    default:
            printf("Input error!\n");
}
```

```
while(1)
{
  printf("1. a + b\n");
  printf("2. a - b\n");
  printf("3. a * b\n");
  printf("4. a / b\n");

  printf("0. exit\n");
  printf("Input choice:");
  scanf("%d", &choice);
  switch(choice)
  {
     ……
  }
}
```

- **浮点数四则运算**
  - 输出一个菜单，用户输入1~5，选择执行不同的运算，输入0则退出程序

```
switch (choice)
{
    case 1: printf("a+b=%d\n", a+b);  break;
    case 2: printf("a-b=%d\n", a-b);  break;
    case 3: printf("a*b=%d\n", a*b);  break;
    case 4:
            if (fabs(b) <= 1e-6)
            {
               printf("Division by zero!\n");
               exit(0);
            }
            else
            {
               printf("a/b=%d\n", a/b);
            }
            break;
    case 0: exit(0);
    default:
            printf("Input error!\n");
}
```

单选题

设置

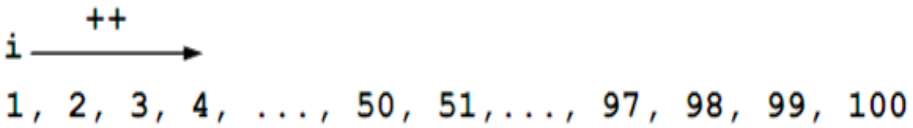
C语言里的循环语句中的循环条件是什么条件？

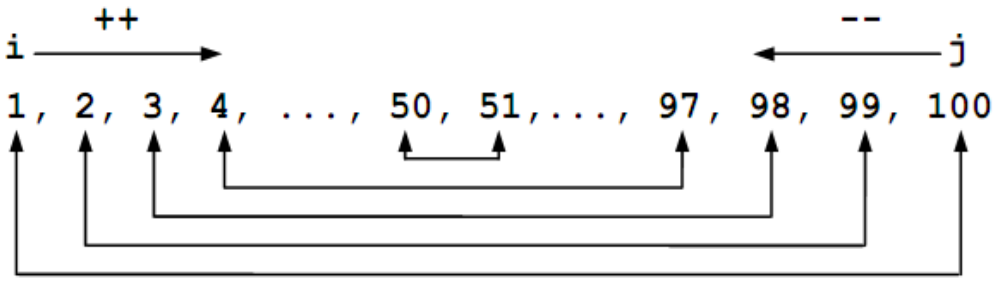
A 循环继续条件

B 循环结束条件

提交

如何快速计算1+2+3……+100的值？






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

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

逗号运算符

讨论题

- 修改程序使其快速计算1+2+3……+n的值，n从键盘输入。

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

能否推广到n?

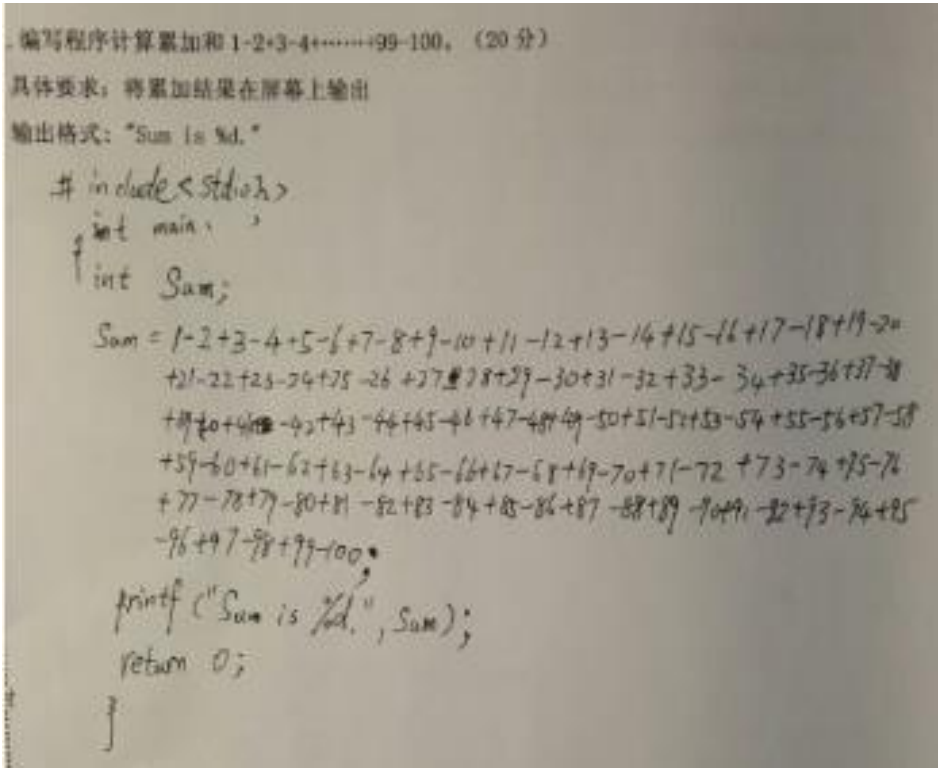
```
#include  <stdio.h>
int main()
{
   int i, j, sum = 0, n;
   printf("Input n:");
   scanf("%d", &n);
   sum = (n%2==0) ? 0 : (n+1)/2;
   for (i=1,j=n; i<j; i++,j--)
   {
      sum = sum + i + j;
   }
   printf("sum=%d\n", sum);
   return 0;
}
```

条件运算符

头脑风暴

- 快速计算1−2+3−4+5−……n的值，n从键盘输入。

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



头脑风暴

- 快速计算1−2+3−4+5−……n的值，n从键盘输入。

```
#include  <stdio.h>
int main()
{
   int i, sum = 0, n, m;
   printf("Input n:");
   scanf("%d", &n);
   m = (n % 2 == 0) ? n : n-1;
   sum = (n % 2 == 0) ? 0 : n;
   for (i=1; i<=m; i+=2)
   {
        sum = sum + i - (i+1);
   }
   printf("sum=%d\n",sum);
   return 0;
}
```

```
#include <stdio.h>
int main()
{
    int sum = 0, n;
    printf("Input n:");
    scanf("%d", &n);

    if (n % 2 == 0)
        sum = (n/2)*(-1);
    else
        sum = (n/2)*(-1) + n;

    printf("sum=%d\n", sum);
    return 0;
}
```

$$\sum_{i=1}^{n} i = 1+2+3+\ldots+n$$

$$n! = 1\times 2\times 3\times \ldots \times n$$

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

sum = sum + i

p = p * i

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

循环实现累加累乘

1!，2!，3!，…，n!

```
#include  <stdio.h>
int main()
{
   int   i, n;
   long  p = 1;
   printf("Input n:");
   scanf("%d", &n);
   for (i=1; i<=n; i++)
   {
        p = p * i;
        printf("%ld\n", p);
   }

   return 0;
}
```

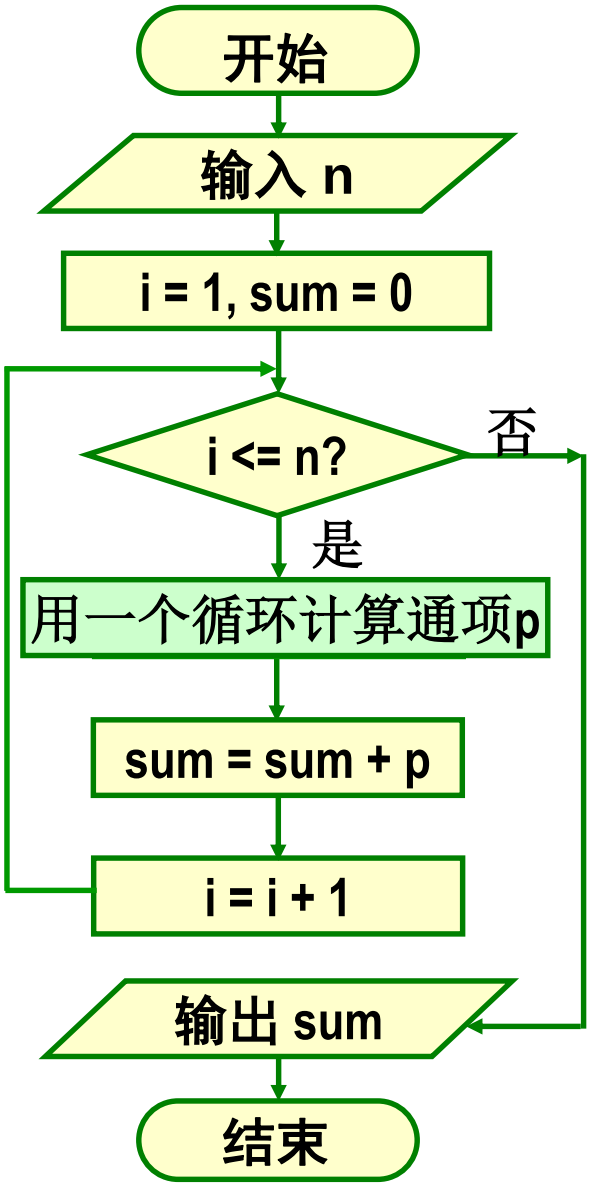
$$i! = (i\text{-}1)!*i$$

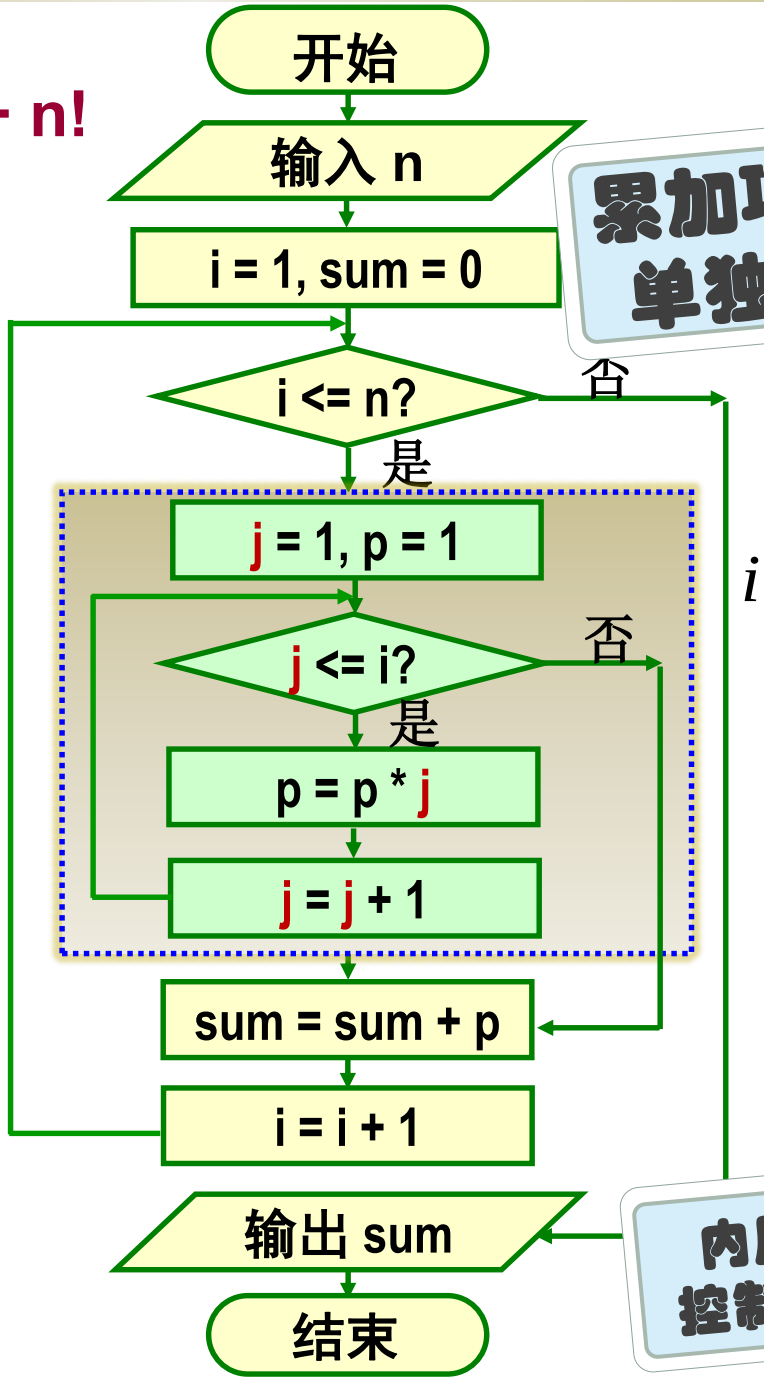
如何计算并输出1! + 2! + 3! … + n! ?

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

累加项的构成规律：
利用前项计算后项

计算并输出1! + 2! + 3! … + n!



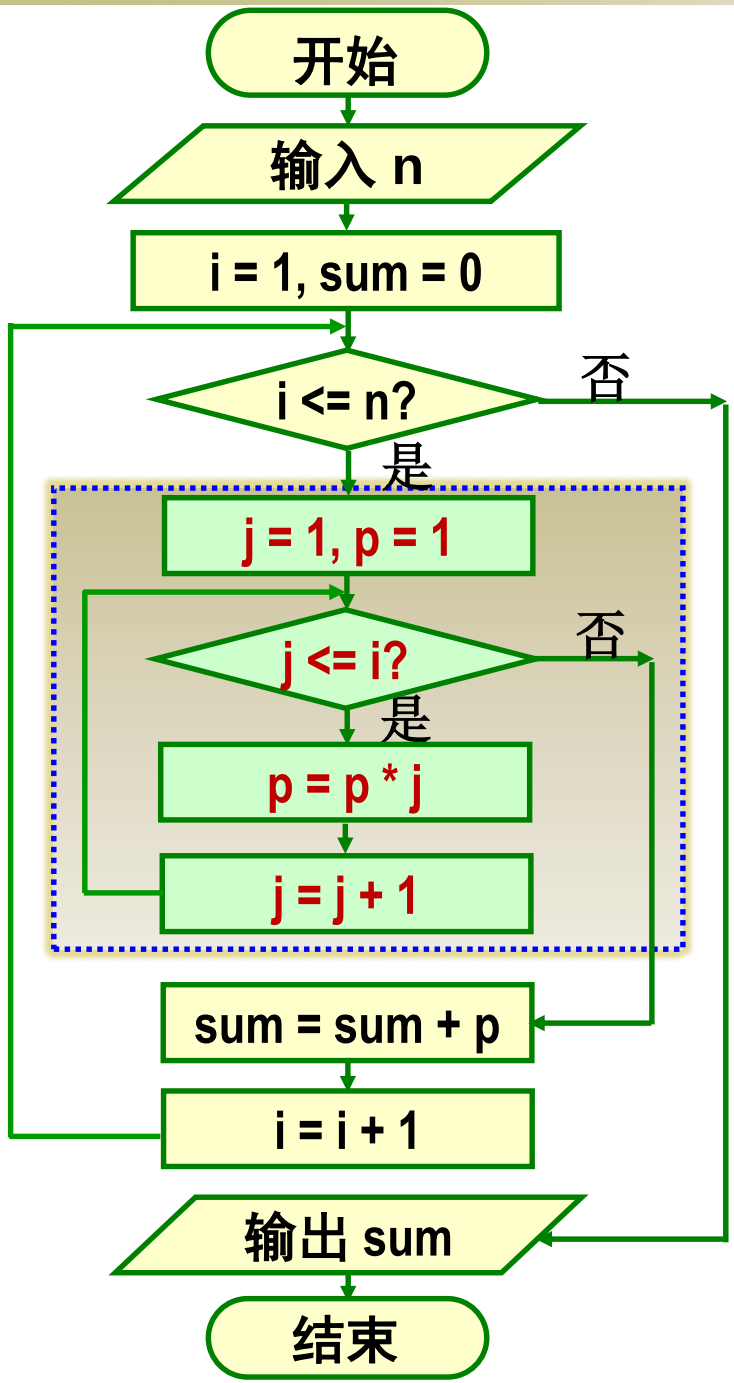


累加项构成规律：
单独计算累加项

$$i!=1\times2\times3\times\ldots\times i$$

$$i! = (i\text{-}1)!*i$$

内层和外层循环
控制变量不能同名



嵌套循环（Nested Loop）

```
#include  <stdio.h>
int main()
{
    int  i, j, n;
    long p, sum = 0;
    printf("Input n:");
    scanf("%d", &n);
    for (i=1; i<=n; i++)
    {
        p = 1;
        for (j=1; j<=i; j++)
        {
            p = p * j;
        }
        sum = sum + p;
    }
    printf("sum = %ld\n", sum);
    return 0;
}
```

嵌套循环是如何执行的?

思考——这个程序是做什么的？

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

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

1*1	1*2	1*3	1*4	1*5	1*6	1*7	1*8	1*9
2*1	2*2	2*3	2*4	2*5	2*6	2*7	2*8	2*9
3*1	3*2	3*3	3*4	3*5	3*6	3*7	3*8	3*9
4*1	4*2	4*3	4*4	4*5	4*6	4*7	4*8	4*9
5*1	5*2	5*3	5*4	5*5	5*6	5*7	5*8	5*9
6*1	6*2	6*3	6*4	6*5	6*6	6*7	6*8	6*9
7*1	7*2	7*3	7*4	7*5	7*6	7*7	7*8	7*9
8*1	8*2	8*3	8*4	8*5	8*6	8*7	8*8	8*9
9*1	9*2	9*3	9*4	9*5	9*6	9*7	9*8	9*9

总循环次数 = 内循环次数 × 外循环次数

思考——这个呢？

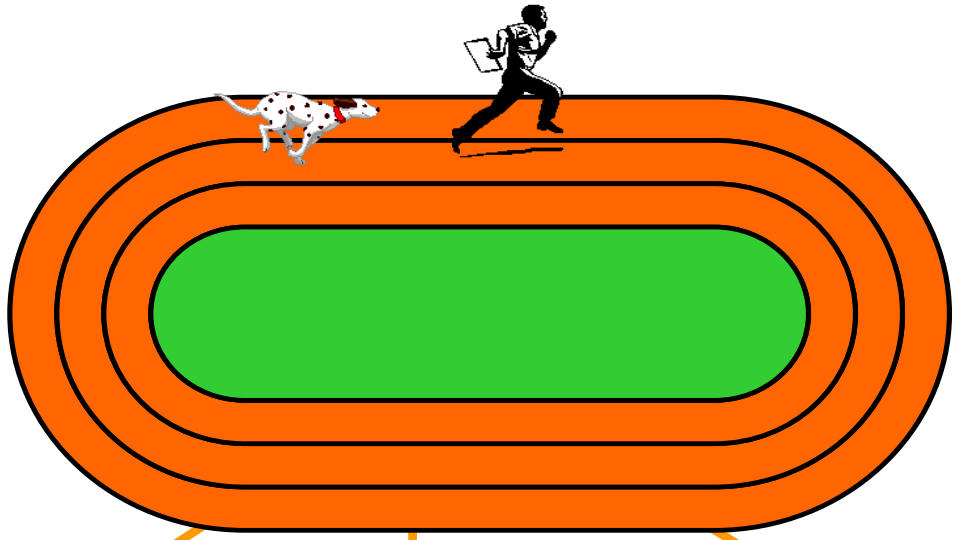
```
#include  <stdio.h>
int main()
{
   int  m, n;
   for (m=1; m<=9; m++)
   {
        for (n=1; n<=m; n++)
        {
           printf("%4d", m*n);
        }
        printf("\n");
   }
   return 0;
}
```

总循环次数 = ？

```
1
2   4
3   6   9
4   8   12  16
5   10  15  20  25
6   12  18  24  30  36
7   14  21  28  35  42  49
8   16  24  32  40  48  56  64
9   18  27  36  45  54  63  72  81
```

1*1								
2*1	2*2							
3*1	3*2	3*3						
4*1	4*2	4*3	4*4					
5*1	5*2	5*3	5*4	5*5				
6*1	6*2	6*3	6*4	6*5	6*6			
7*1	7*2	7*3	7*4	7*5	7*6	7*7		
8*1	8*2	8*3	8*4	8*5	8*6	8*7	8*8	
9*1	9*2	9*3	9*4	9*5	9*6	9*7	9*8	9*9

还有哪些循环控制方式？



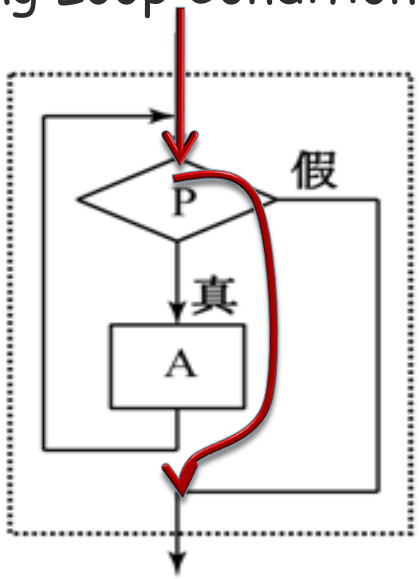
条件控制
Condition Controlled
循环次数未知

计数控制
Counter Controlled
循环次数已知
确定性循环

标记控制
Sentinel Controlled
循环次数未知

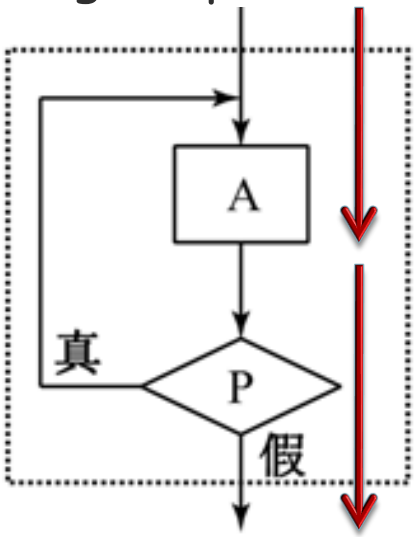
当型和直到型循环有何区别？

Testing Loop Condition **First**



- 当**第一次**测试循环条件就为**假**时？

Testing Loop Condition **last**



直到型循环的循环条件是**循环继续**条件还是**循环终止**条件？

条件控制的循环

- 判断数字位数V1.0：从键盘输入一个int型数据，编写程序判断该整数共有几位数字。

```
#include <stdio.h>
int main()
{
   int a, b;
   int counter = 1;
   printf("Input a number:");
   scanf("%d", &a);
   b = a / 10;
   while (b != 0)  //直到为0为止
   {
       counter++;
       b = b / 10; //不断缩小10倍
   }
   printf("%d bits\n", counter);
   return 0;
}
```

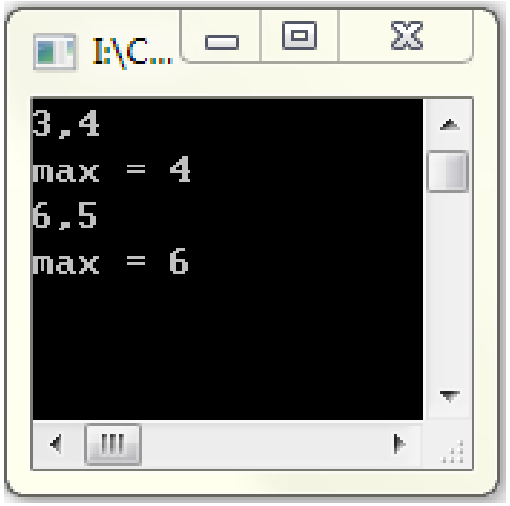
```
#include <stdio.h>
int main()
{
   int a, b;
   int counter = 0;
   printf("Input a number:");
   scanf("%d", &a);
   b = a;
   do
   {
       counter++;
       b = b / 10;  //不断缩小10倍
   }while (b != 0);  //直到为0为止
   printf("%d bits\n", counter);
   return 0;
}
```

while还有什么用？

- 一次运行测试多组数据

```
#include <stdio.h>
int main()
{
    int a, b, max;
    while (scanf("%d,%d", &a, &b) == 2)
    {
        if (a > b)  max = a;
        else        max = b;
        printf("max = %d\n", max);
    }
    return 0;
}
```

scanf的返回值是什么？




```
max = a > b ? a : b;
```

程序测试与程序调试

教材5.10.1，6.6.2节

哈尔滨工业大学

苏小红

sxh@hit.edu.cn

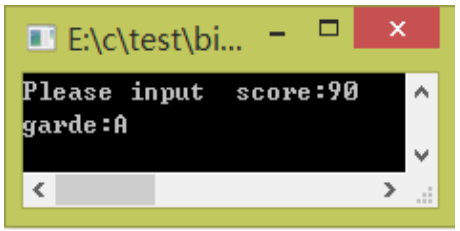
```c
#include<stdio.h>
int main()
{
  int score, mark;
  printf("Please input  score:");
  scanf("%d", &score);
  mark = score / 10;
  switch (mark)
  {
    case 0:
    case 1:
    case 2:
    case 3:
    case 4:
    case 5:    printf("garde:E\n");
               break;
    case 6:    printf("garde:D\n");
               break;
    case 7:    printf("garde:C\n");
               break;
    case 8:    printf("garde:B\n");
               break;
    case 9:
    case 10:   printf("garde:A\n");
               break;
    default:   printf("error!\n");
  }
  return 0;
}
```

白盒测试

- 程序测试
  - 给定一组测试用例，通过运行被测程序，检查程序输出是否与预期结果一致

输入数据	预期结果	实际输出
0，15，55	E	E
65	D	D
75	C	C
85	B	B
95，100	A	A
-10，110	error!	error!

边界测试



```c
#include<stdio.h>
int main()
{
  int score, mark;
  printf("Please input  score:");
  scanf("%d", &score);
  mark = score / 10;
  switch (mark)
  {
    case 0:
    case 1:
    case 2:
    case 3:
    case 4:
    case 5:    printf("garde:E\n");
               break;
    case 6:    printf("garde:D\n");
               break;
    case 7:    printf("garde:C\n");
               break;
    case 8:    printf("garde:B\n");
               break;
    case 9:
    case 10:   printf("garde:A\n");
               break;
    default:   printf("error!\n");
  }
  return 0;
}
```

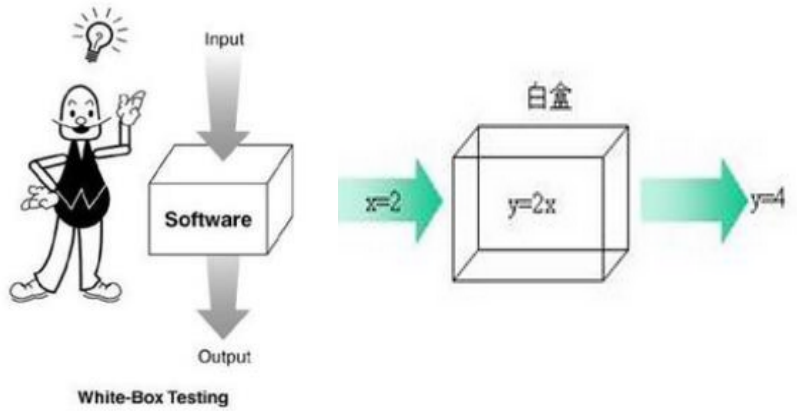
- 测试用例的选取方法
  - 尽量覆盖所有分支（路径）
  - 应考虑到合法的输入和不合法的输入以及各种边界条件

```c
mark = (score>=0 && score<=100) ? score/10 : -1;
```

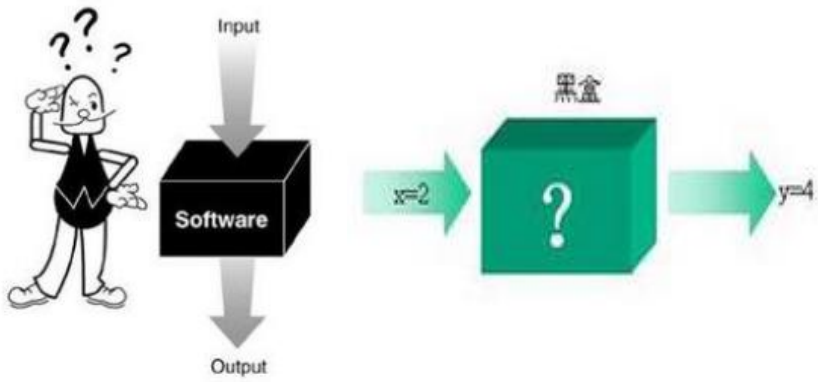
输入数据	预期结果	实际输出
0，15，55	E	E
65	D	D
75	C	C
85	B	B
95，100	A	A
-10，110	error!	error!
-5	error!	E
105	error!	A

程序测试

- 测试的分类
  - 白盒测试（结构测试） —— 测试早期或重要路径
  - 黑盒测试（功能测试） —— 测试后期或重要功能
- 测试用例的选取方法
  - 尽量覆盖所有分支和路径
  - 边界测试
    - 非法输入
    - 极端的临界点



White-Box Testing



Black-Box Testing

多选题

已知闰年的条件是
((year%4==0) && (year%100!=0)) || (year%400==0)
请问测试这个判断闰年的程序，至少需要测试下面哪些数据才能覆盖上面复合条件中的所有分支？

- A 2000
- B 2100
- C 2016
- D 2015
- E -1
- F a

提交

设置

```
int year, month;
printf("Input year,month:");
scanf(“%d, %d”, &year, &month); //怎样输入数据?
switch (month)
{
  case 1: case 3: case 5: case 7: case 8: case 10: case 12:
          printf("31 days\n");
          break;
  case 2:
          if ((year%4==0 && year%100!=0) || (year%400==0))
              printf("29 days\n”);
           else
              printf("28 days\n");
           break;
  case 4: case 6: case 9: case 11:
          printf("30 days\n");
          break;
  default:
          printf("Input error!\n");
          exit(0);
}
```

```
printf("Input year,month:");
n = scanf("%d,%d", &year, &month);
if (n != 2)
{
    printf("Input error!\n");
    return 0;  //exit(0);
}
```


```
printf("Input year,month:");
n = scanf("%d,%d", &year, &month);
while (n != 2)
{
     while (getchar() != '\n');
     printf("Input year,month:");
     n = scanf("%d,%d", &year, &month);
}
```

遇到不正确使用或非法数据输入时仍能保护自己避免出错的能力——健壮性

不能盲目依赖测试去发现bug，而是以测试驱动编程，预防为主

```
printf("Input year,month:");
n = scanf("%d,%d", &year, &month);
while (n != 2)
{
     while (getchar() != '\n');
     printf("Input year,month:");
     n = scanf("%d,%d", &year, &month);
}
```

```
do{
     while (getchar() != '\n'); //????
     printf("Input year,month:");
     n = scanf("%d,%d", &year, &month);
} while (n != 2);
```

```
do{
     printf("Input year,month:");
     n = scanf("%d,%d", &year, &month);
     if (n != 2) while (getchar() != '\n');
} while (n != 2);
```

多选题

设置

何时需要用到scanf函数的返回值？

A 需要判断scanf()是否成功读入指定个数的数据的时候

B 需要判断是否读入了非法字符的时候

C 需要增强程序的健壮性和容错能力的时候

D 需要判断scanf()读入的数据类型是否正确的时候

提交

程序测试的实质

- 能否对输入数据的所有可能取值都进行测试？
- 程序测试的实质——抽样检查
  - 测试是减少缺陷、提高软件质量的重要手段，但提高软件质量不能只依赖于测试
- 测试只能证明程序有错，不能证明程序无错

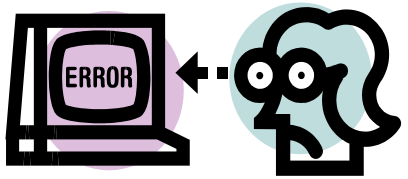
——E.W.Dijkstra

程序测试的目的

- 测试的目的
  - 用少量的测试用例，找出尽可能多的Bug
  - 成功的测试在于发现迄今为止尚未发现的Bug
- 测试人员的主要任务
  - 站在使用者角度，通过不断使用（包括非常规使用），尽可能多地找Bug
  - 测试的过程就像黑客的攻击过程

程序中常见的出错原因

编译错误	语法错误（Syntax Error）
链接错误	函数名写错，缺少包含文件、或者包含文件的路径错误等
运行时错误	运行结果与预期不一致 程序无法正常运行



程序改错1：计算圆周率

- 利用 $\frac{\pi}{4}=1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\cdots$，编程计算π的近似值，直到最后一项的绝对值小于$10^{-4}$时为止。

```
#include  <math.h>
#include  <stdio.h>
int main()
{
    double pi=0,term=1,sign=1.0;
    int    n = 1;
    do{
        term = sign / n; //单独计算
        pi = pi + term;
        sign = -sign;
        n = n + 2;
    }while (fabs(term) >= 1e-4);
    printf("pi = %f\n", pi * 4);
    return 0;
}
```

pi = 3.141793

```
#include  <math.h>
#include  <stdio.h>
int main()
{
   double pi=0,term=1,sign=1.0;
   int    n = 1;
   do{
      pi = pi + term;
      sign = -sign;
      n = n + 2;
      term = sign / n; //单独计算
   }while (fabs(term) >= 1e-4);
   printf("pi = %f\n", pi * 4);
   return 0;
}
```

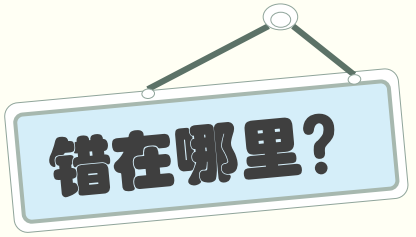
为啥结果不一样?

✖ pi = 3.141393

程序改错2：计算sinx

- 利用sinx $\approx x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\frac{x^9}{9!}$ +…，计算sinx（x为弧度值）的值，直到最后一项的绝对值小于10^{-5}时为止，输出e的值并统计累加

```
#include  <stdio.h>
#include  <math.h>
int main()
{
    int n = 1, count = 1;
    double x, sum, term;
    printf("Input x:");
    scanf("%lf", &x);
    sum = x;
    term = x;
    do{
        term = term * pow(-1,count) *x*x/((n+1)*(n+2)); //利用前项求后项
        sum = sum + term;
        n = n + 2;
        count++;
    } while (fabs(term) >= 1e-5);
    printf("sin(x) = %f, count = %d\n", sum, count);
    return 0;
}
```



程序改错2：计算sin*x*

- 利用$\sin x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}$，计算sin*x*（*x*为弧度值）的值，直到最后一项的绝对值小于10^{-5}时为止，输出e的值并统计累加的项

```
#include  <stdio.h>
#include  <math.h>
int main()
{
    int n = 1, count = 1;
    double x, sum, term;
    printf("Input x:");
    scanf("%lf", &x);
    sum = x;
    term = x;
    do{
        term = -term * x * x / ((n + 1) * (n + 2 ));
        sum = sum + term;
        n = n + 2;
        count++;
    } while (fabs(term) >= 1e-5);
    printf("sin(x) = %f, count = %d\n", sum, count);
    return 0;
}
```

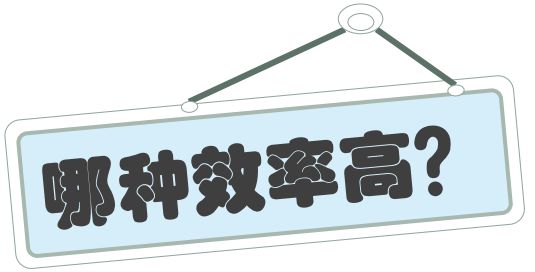
小结

- 累加项的前后项之间无关
  - 1*2*3 + 3*4*5 +… + 99*100*101

```
sum = sum + i*(i+1)*(i+2)  (i=1; i<=99; i+=2)
```

  - 1 + 3 + 5 + … + 2n-1

```
sum = sum + 2*i-1; (i=1; i<=n; i++)
sum = sum + i;     (i=1; i<=2*n-1; i+=2)
```



- 累加项的前后项之间有关
  - $x^0 + x^1 + x^2 + \ldots + x^n$

```
sum = sum + pow(x, i);  (i=0; i<=n; i++)
sum = sum + term;  sum初值为x^0
term = term * x;   term初值为1
```

如果sum是long型，千万不要用pow()强转为long后的结果累加
非要用，就将sum定义为double，累加结束后再强转

调试方法

调试方法

- **逆向推理**：粗分细找，逆向推理，观察中间结果，定位大致的范围

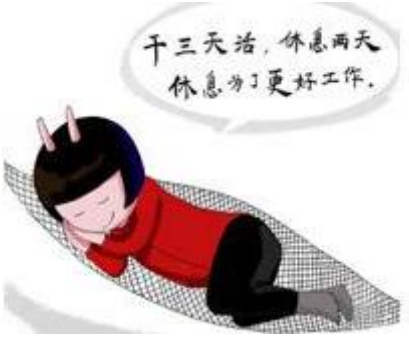
- **分治排除**：用注释切掉一些代码，调试无误后再打开注释

- **缩减输入**：缩减输入数据，设法找到导致失败的最小输入

世界上最好的调试工具
就是那些有经验的人

讨论题

- 中国有句俗语叫“三天打鱼两天晒网”，某人从1990年1月1日起开始“三天打鱼两天晒网”，即工作三天，然后再休息两天。从键盘任意输入某年某月某天，编程判断他是在工作还是在休息。
  - 如果是在工作，则输出“He is working”
  - 如果是在休息，则输出“He is having a rest”
  - 输入提示信息： "Input year,month,day:"




```
输入y, m, d
for (i=1990; i<y; i++)
{
    累计从1990开始到y-1年的总天数到sum中
}
for (i=1; i<m; i++)
{
    累计第y年从1月到m-1月的总天数到sum中
}
累计第y年第m月的天数d到sum中
以5天为一个周期，根据余数决定是在工作还是在休息
```

课后作业

- 已知1900年1月1日是星期一，编程从键盘输入一个年份y，计算在1901年1月1日至y年12月31日间共有多少个星期天落在每月的第一天上？
  - 输入提示信息： "Input year:"