选择结构

教材5.3~5.9节

MOOC第4周

哈尔滨工业大学

苏小红

sxh@hit.edu.cn

如何描述一个判断条件？

关系运算符	数学运算符	优先级	结合性
<	<	高	从左向右
>	>		
<=	≤		
>=	≥		
==	=	低	
!=	≠		

- 计算关系表达式的值
  - true ←→1
  - false ←→0
- 判断表达式（不限于关系表达式）的真假
  - 非0 ←→ true
  - 0 ←→ false

如何描述一个复杂的条件？

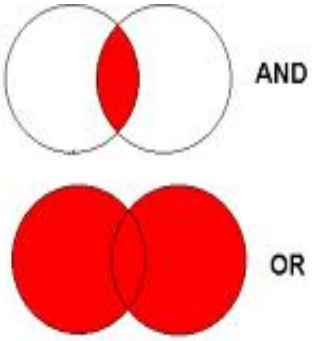
- a > b > c 的值是真是假？
- 如何测试“b在a和c之间”？
- 如何测试“b不在a和c之间”

- 美女，帅哥，亦师亦友
- 无论黑猫白猫，能抓耗子的就是好猫

&& 与（AND）：“A 并且 B”

|| 或（OR）：“或者A 或者 B”

! 非（NOT）



- 逻辑非 → 算术 → 关系 → 逻辑与 → 逻辑或 → 赋值
- 圆括号优先级最高

逻辑运算符的短路特性

- 若表达式值可由左操作数计算，则不再计算右操作数
- “短路”特性：使得表达式中的某些操作可能不被计算

a && b

把最有可能为假的简单条件写在表达式最左边

a || b

把最有可能为真的简单条件写在表达式最左边

a && b++ ≠ **b++ && a**

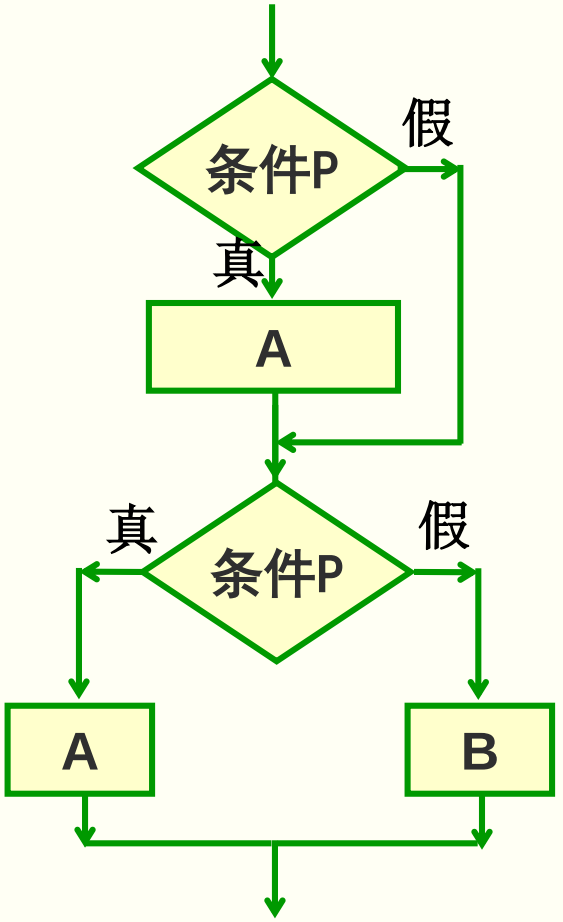
不建议使用多用途、复杂而晦涩难懂的复合表达式

思考题

- 一位百岁老人一生只过了25个生日，是何原因？
- 从键盘上任意输入一个年份，编程判断这一年他是否能过上生日？
  - 判断某年year是闰年的条件是满足下列二者之一
    - （1）能被4整除，但不能被100整除
    - （2）能被400整除
    - 如何表示“能被某数整除”？用什么运算？

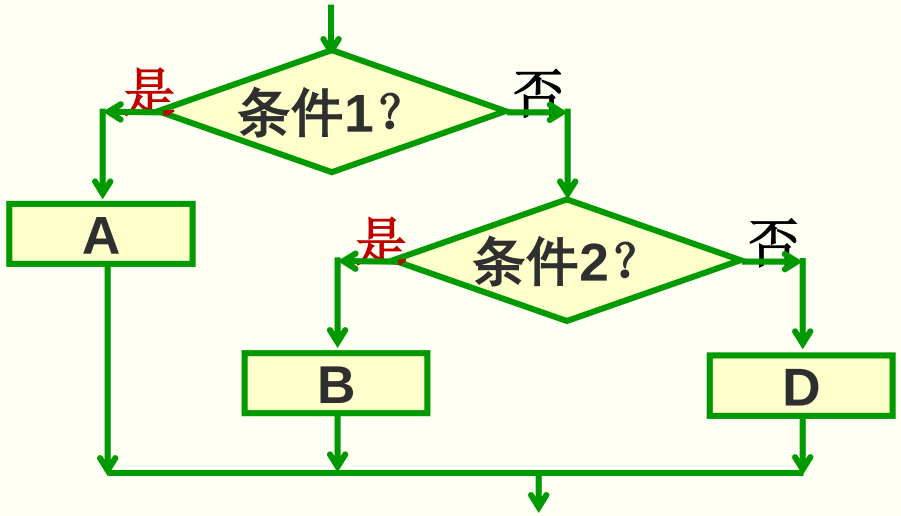

```
((year%4==0) && (year%100!=0)) || (year%400==0)
```

```c
#include <stdio.h>
int main()
{
   int n;
   printf("Input a year:");
   scanf("%d", n);
   if (n%4 = 0 && n%100 != 0)
      printf("%d is a leap year!\n", n);
   if (n%400 = 0)
      printf("%d is a leap year!\n", n);
   else
      printf("%d is not a leap year!\n", n);
   return 0;
}
```



错在哪里?

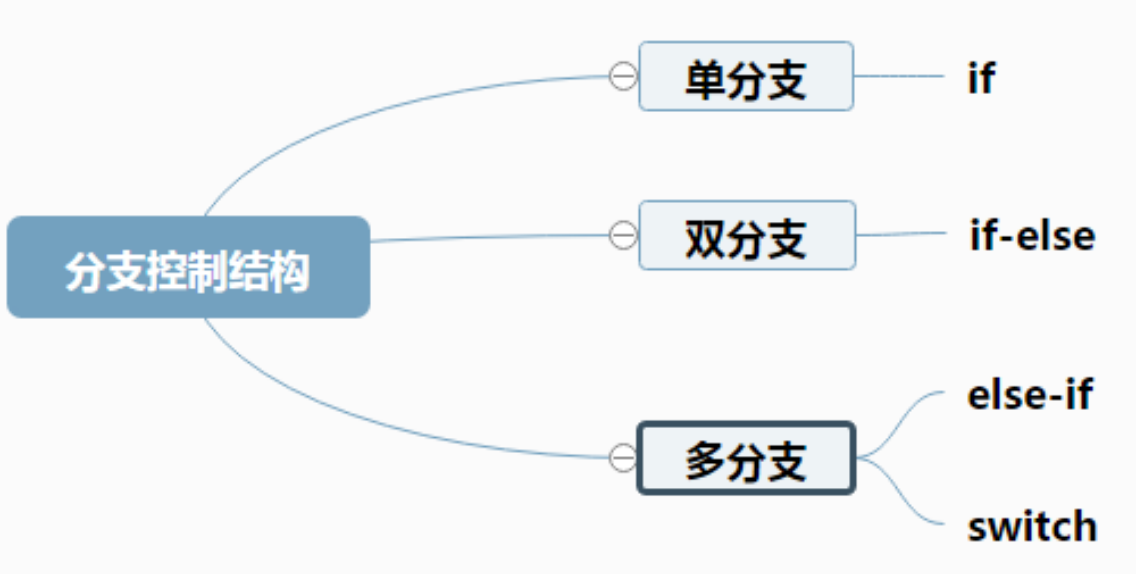
```c
#include <stdio.h>
int main()
{
   int n;
   printf("Input a year:");
   scanf("%d", &n);
   if (n%4 == 0 && n%100 != 0)
      printf("%d is a leap year!\n", n);
   else if (n%400 == 0)
      printf("%d is a leap year!\n", n);
   else
      printf("%d is not a leap year!\n", n);
   return 0;
}
```



把**最可能发生**的情况放在**前面**

闰年相关的问题v2.0

- 闰年相关的问题v2.0
  - 从键盘输入你的出生年月，编程输出你出生的月份有多少天?
    - 输入提示信息： "Input year,month:"



```
if (month==1 || month==3 || month==5 || month==7 ||
    month==8 ||month==10 || month==12)
{
    printf("31 days\n");
}
else if (month==4 || month==6 || month==9 || month==11)
{
    printf("30 days\n");
}
else if (month == 2)
{
    if ((year%4==0 && year%100!=0) || (year%400==0))
        printf("29 days\n"); //闰年的2月有29天
    else
        printf("28 days\n"); //平年的2月有28天
}
else
{
    printf("Input error!\n");
    exit(0);
}
```

复合语句：被当作一条语句看待

```
switch (month)  //int或char型表达式
{
  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);
}
```

语句标号作用

常量必须互不相同

顺序无关紧要

课上练习

- 数位拆分与计算V5.0
  - 考虑除数为0
- 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

循环结构

教材6.1~6.5节

MOOC第5周

哈尔滨工业大学

苏小红

sxh@hit.edu.cn

循环有什么用？

- 3秒钟倒计时开始


```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int main()
{
   printf("3");
   Sleep(1000);

   printf("2");
   Sleep(1000);

   printf("1");
   Sleep(1000);

   printf("0\n");
   Sleep(1000);

   system("pause");
   return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int main()
{
   printf("3");
   Sleep(1000);

   printf("\r2");
   Sleep(1000);

   printf("\r1");
   Sleep(1000);

   printf("\r0\n");
   Sleep(1000);

   system("pause");
   return 0;
}
```

循环有什么用？

- 3秒钟倒计时开始
- 如果60秒倒计时，怎么破？

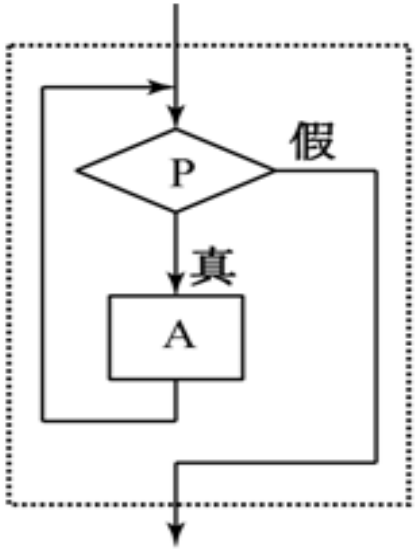

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int main()
{
   printf("3");
   Sleep(1000);

   printf("2");
   Sleep(1000);

   printf("1");
   Sleep(1000);

   printf("0\n");
   Sleep(1000);

   system("pause");
   return 0;
}
```

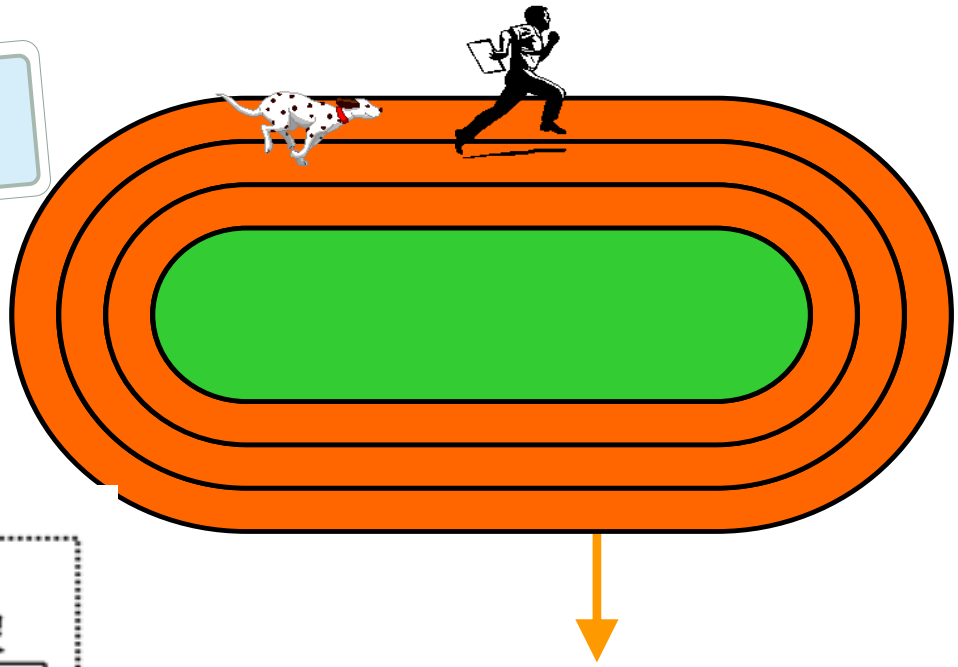
```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int main()
{
   system("cls");
   printf("3");
   Sleep(1000);
   system("cls");
   printf("2");
   Sleep(1000);
   system("cls");
   printf("1");
   Sleep(1000);
   system("cls");
   printf("0\n");
   Sleep(1000);
   system("pause");
   return 0;
}
```

循环有什么用？

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int main()
{
   system("cls");
   printf("3");
   Sleep(1000);
   system("cls");
   printf("2");
   Sleep(1000);
   system("cls");
   printf("1");
   Sleep(1000);
   system("cls");
   printf("0\n");
   Sleep(1000);
   system("pause");
   return 0;
}
```

Testing Loop Condition **First**





计数控制

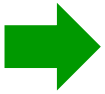
Counter Controlled

循环次数已知

- **循环继续**条件vs**循环终止**条件？

for语句实现的倒计时程序

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int main()
{
   system("cls");
   printf("3");
   Sleep(1000);
   system("cls");
   printf("2");
   Sleep(1000);
   system("cls");
   printf("1");
   Sleep(1000);
   system("cls");
   printf("0\n");
   Sleep(1000);
   system("pause");
   return 0;
}
```



```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
int main()
{                          循环继续条件
   int count;
   for (count=3; count>=0; count--)
   {
       system("cls");
       printf("%d", count);
       Sleep(1000);
   }
   printf("\n");
   system("pause");
   return 0;
}
```

如何计算1+2+3……+100的值？

- 计数控制——Loop is controlled by a counter（计数器）

i=1,2,…99,100 → i<=100

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

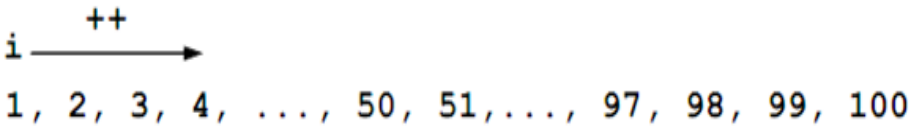
循环是如何**执行**的？

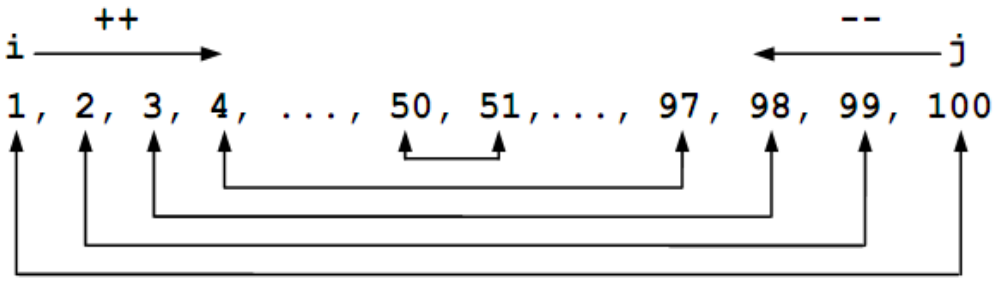
i=100,99…,2,1 → i>=1

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

如何控制循环不会成为**死循环**？

如何快速计算1+2+3……+100的值？





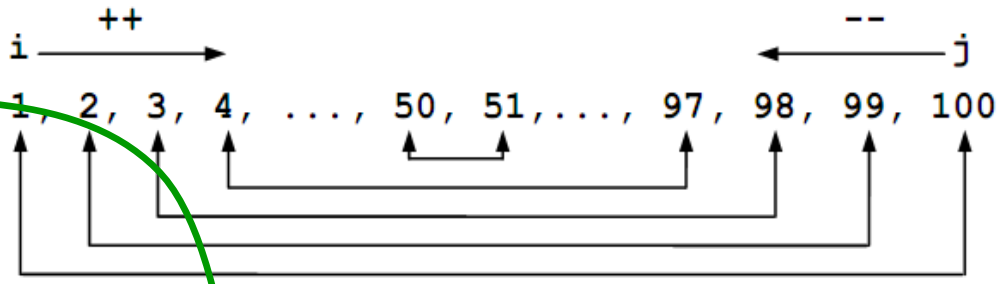

```
#include  <stdio.h>
int main()
{
   int i, sum = 0;
   for (i=1; i<=100; i++)
   {
        sum = sum + i;
   }
   printf("sum=%d\n",sum);
   return 0;
}
```

```
#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;
}
```

如何快速计算1+2+3……+100的值？

逗号运算符（Comma Operator）

表达式1， 表达式2， …， 表达式n



- 顺序求值运算符
- 主要用在循环语句中→同时对多个变量赋初值

```
#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;
}
```

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

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

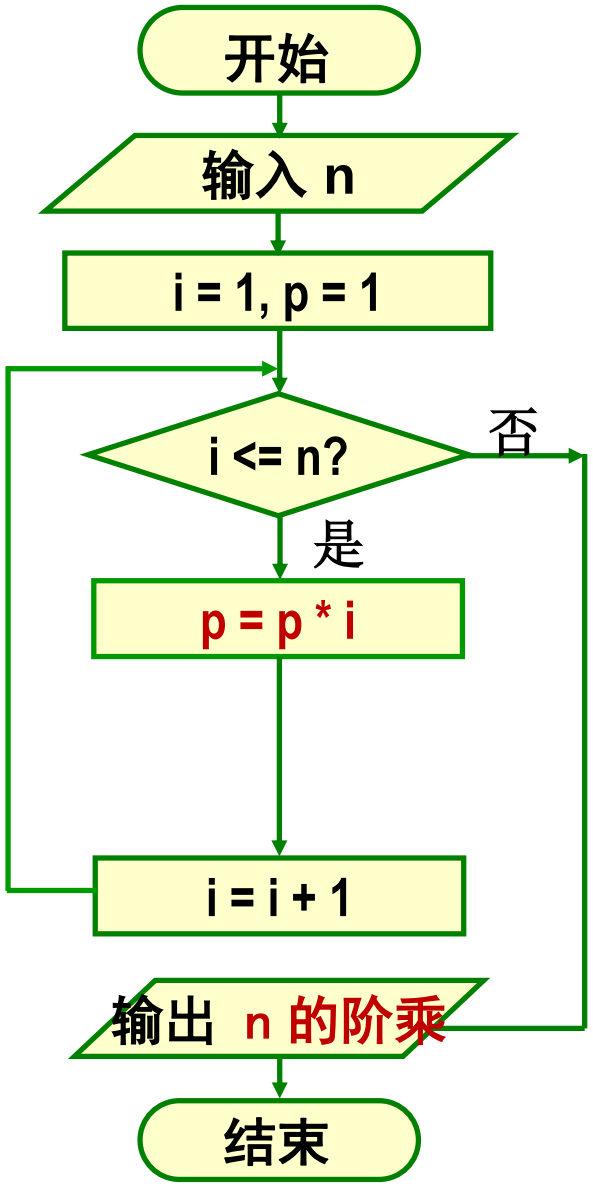
```
#include  <stdio.h>
int main()
{
   int i, sum = 0, n;
  printf("Input n:");
  scanf("%d", &n);
   for (i=1; i<=n; i++)
   {
       sum = sum + i;
   }
   printf("sum=%d\n",sum);
   return 0;
}
```

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

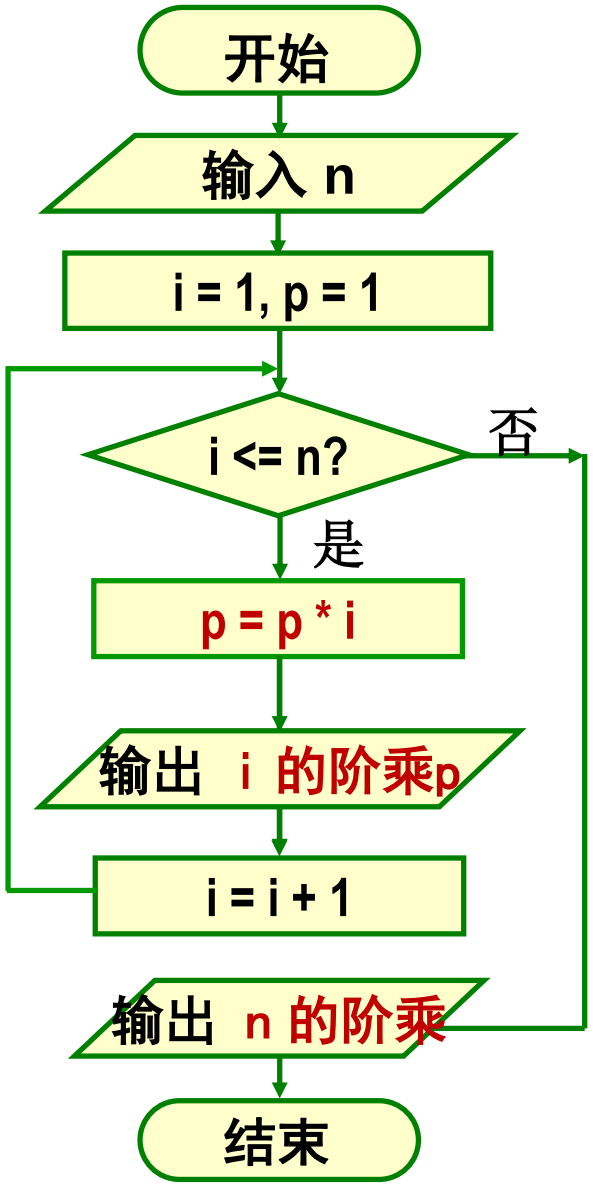


计算并输出n!

```
#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;
}
```

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

循环实现累加累乘

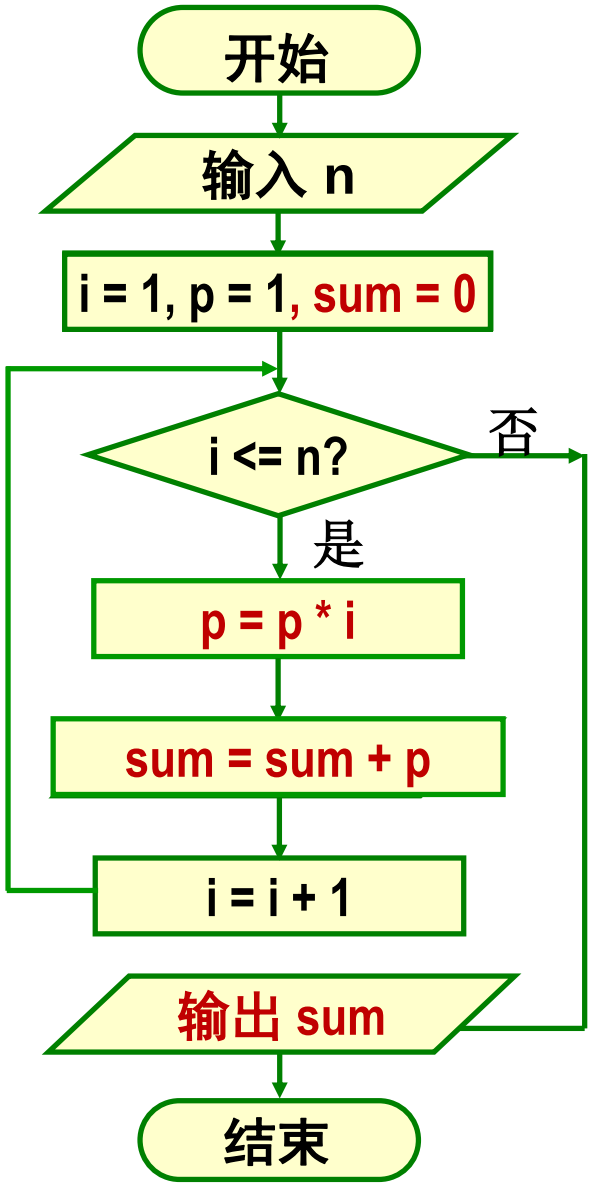


计算并输出1!, 2!, 3!, … , n!

```
#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;
}
```

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

循环实现累加累乘



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;
}
```

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

循环实现累加累乘

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

前后项有关

- 利用前项计算后项
- p = p * i
- sum = sum + p

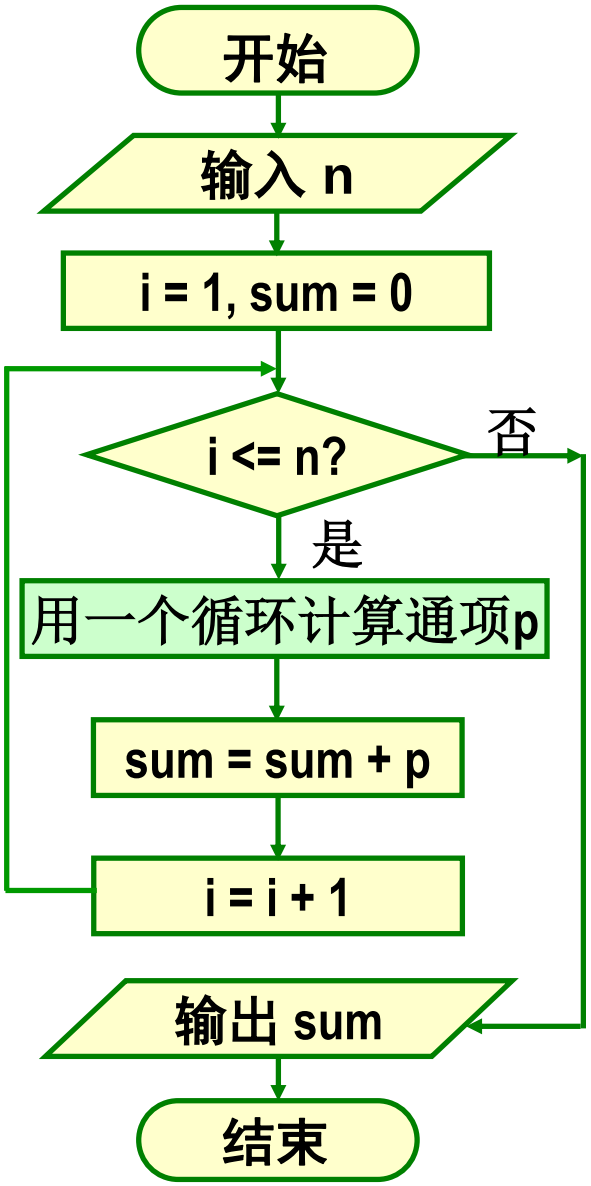
$$i! = (i-1)! * i$$

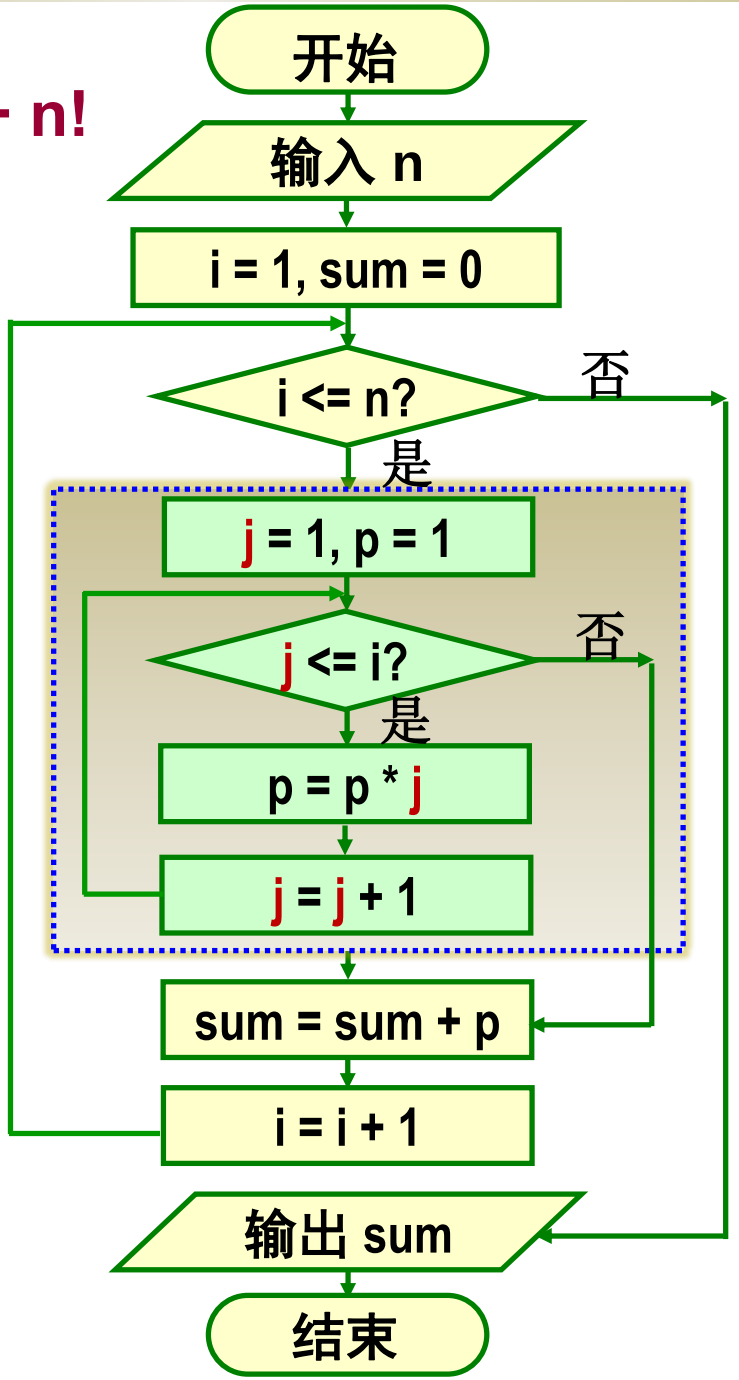
前后项无关

- 单独计算累加项p
- p = ?
- sum = sum + p

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

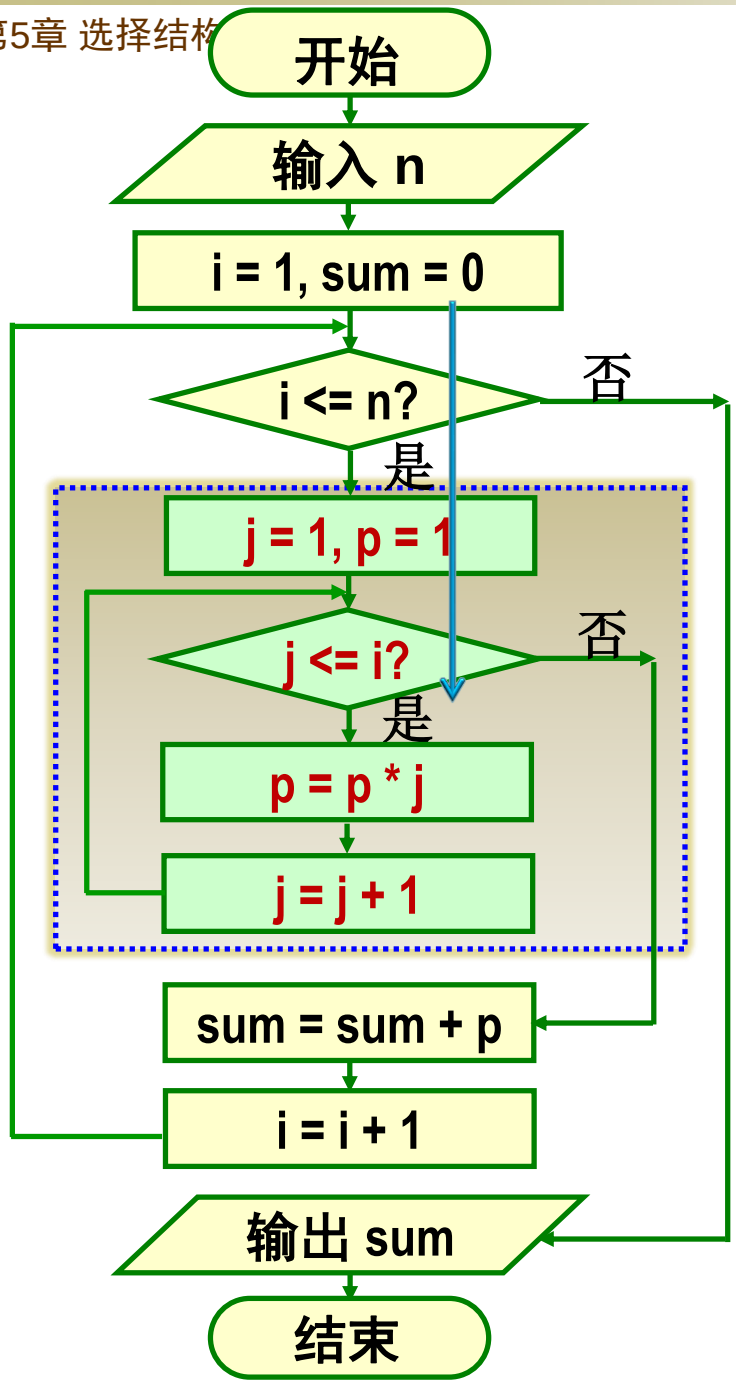
计算并输出1! + 2! + 3! … + n!





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

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



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;
}
```

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

嵌套循环是如何执行的?

思考题——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

内层和外层循环控制变量不能同名

总循环次数 = 内循环次数 × 外循环次数

思考题——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;
}
```

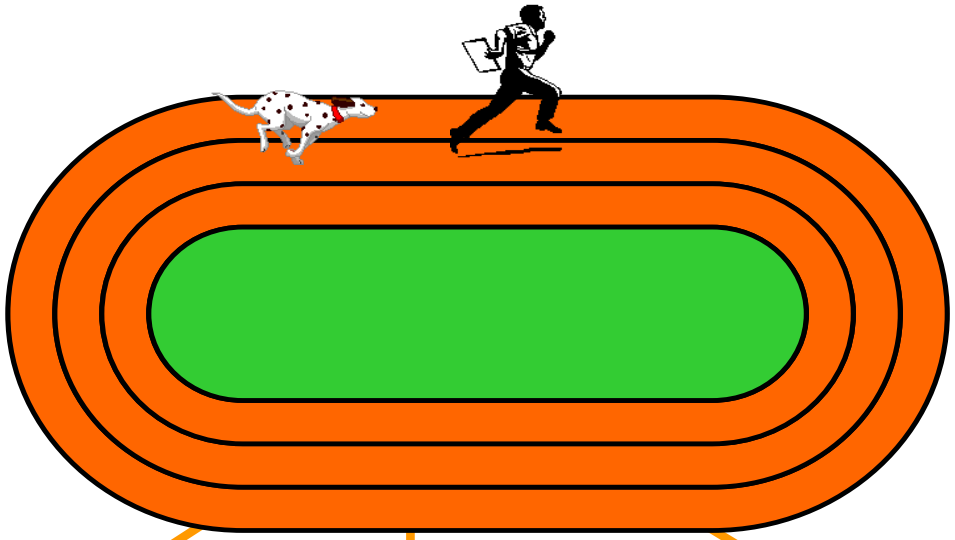
总循环次数 = ？

```
for (m=1; m<=1000; m++)
{
        ;
}
```

```
for (m=1; m<=1000; m++);
```

```
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
```

还有哪些循环控制方式？



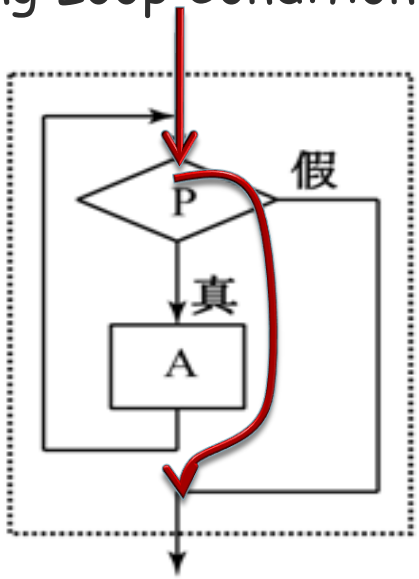
条件控制
Condition Controlled
循环次数未知

计数控制
Counter Controlled
循环次数已知

标记控制
Sentinel Controlled
循环次数未知

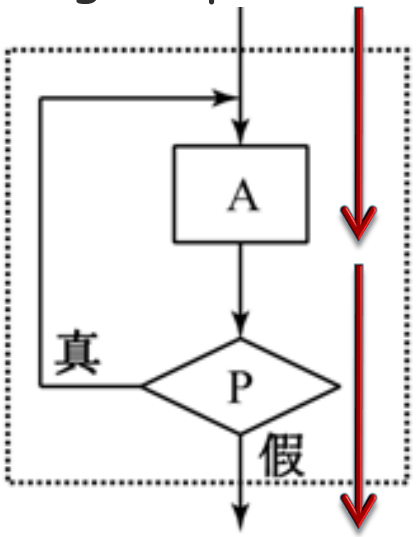
当型和直到型循环有何区别？

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？

C语言中的循环语句


```
#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;
}
```

while语句实现？

do-while语句实现？

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;
}
```

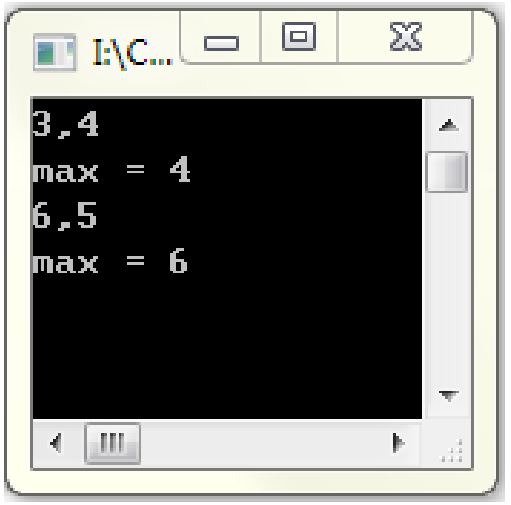
```
#include <stdio.h>
int main()
{
   int a, b;
   int counter = 0;
   printf("Input a number:");
   scanf("%d", &a);
   b = a;
   do
   {
       counter++;
       b = b / 10;  //不断缩小10倍
   }while (b != 0);  //直到为0为止
   printf("%d bits\n", counter);
   return 0;
}
```

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;
```

购物街“看商品猜价格”游戏

- 每次运行程序可以猜多个数，每个数最多可猜10次，若10次仍未猜对，则停止本次猜数，判断用户是否继续猜下一个数，若是，则计算机重新随机生成一个数让用户猜；否则算法结束。

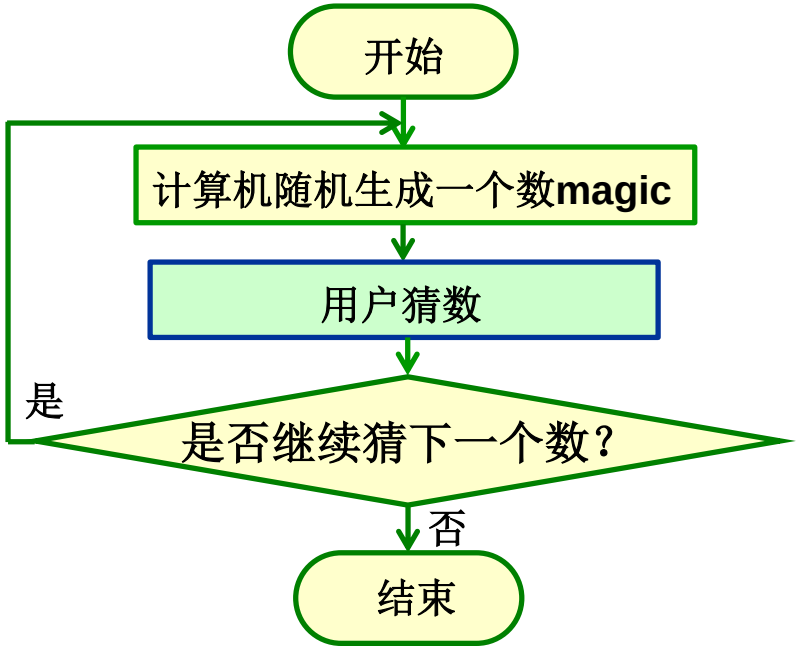
Guess the price of a piece of goods（an integer, [1,100]）

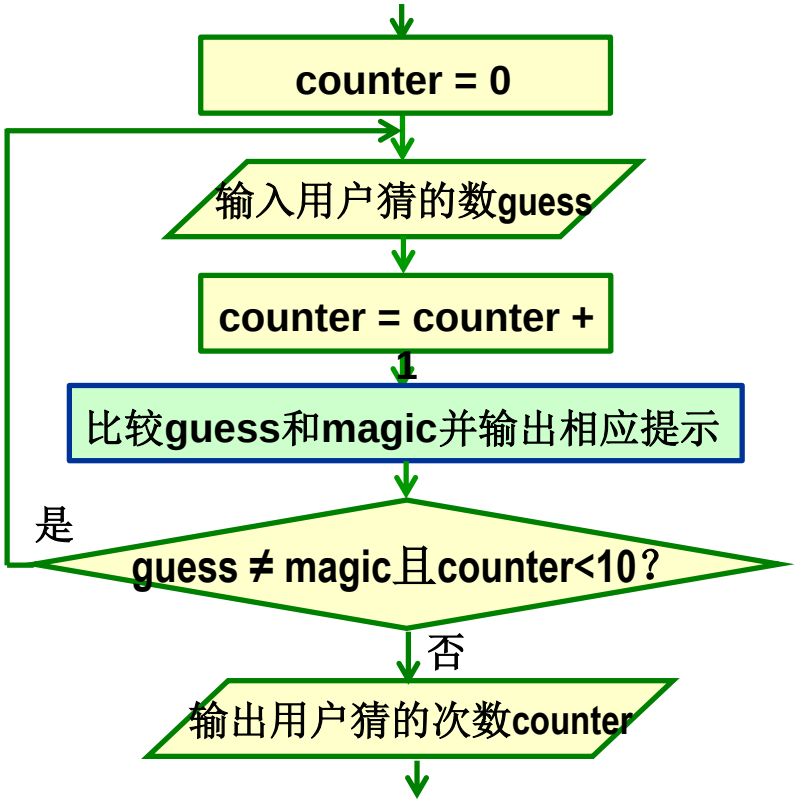
Shopping street Game

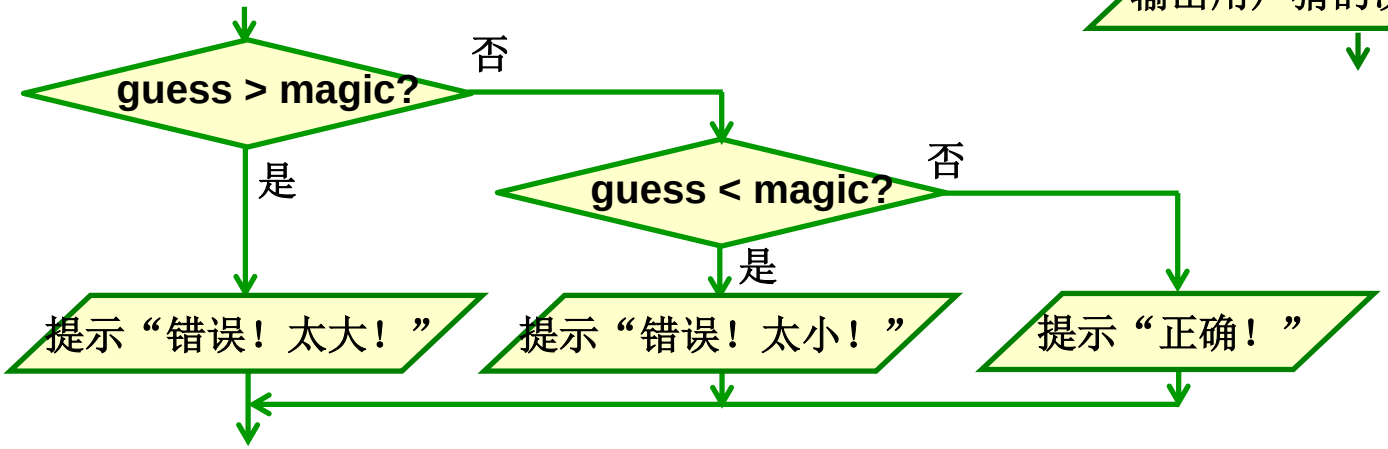
Right: Congratulations!
商品归你了！

Wrong: greater or less? 太大了！
or 太小了！

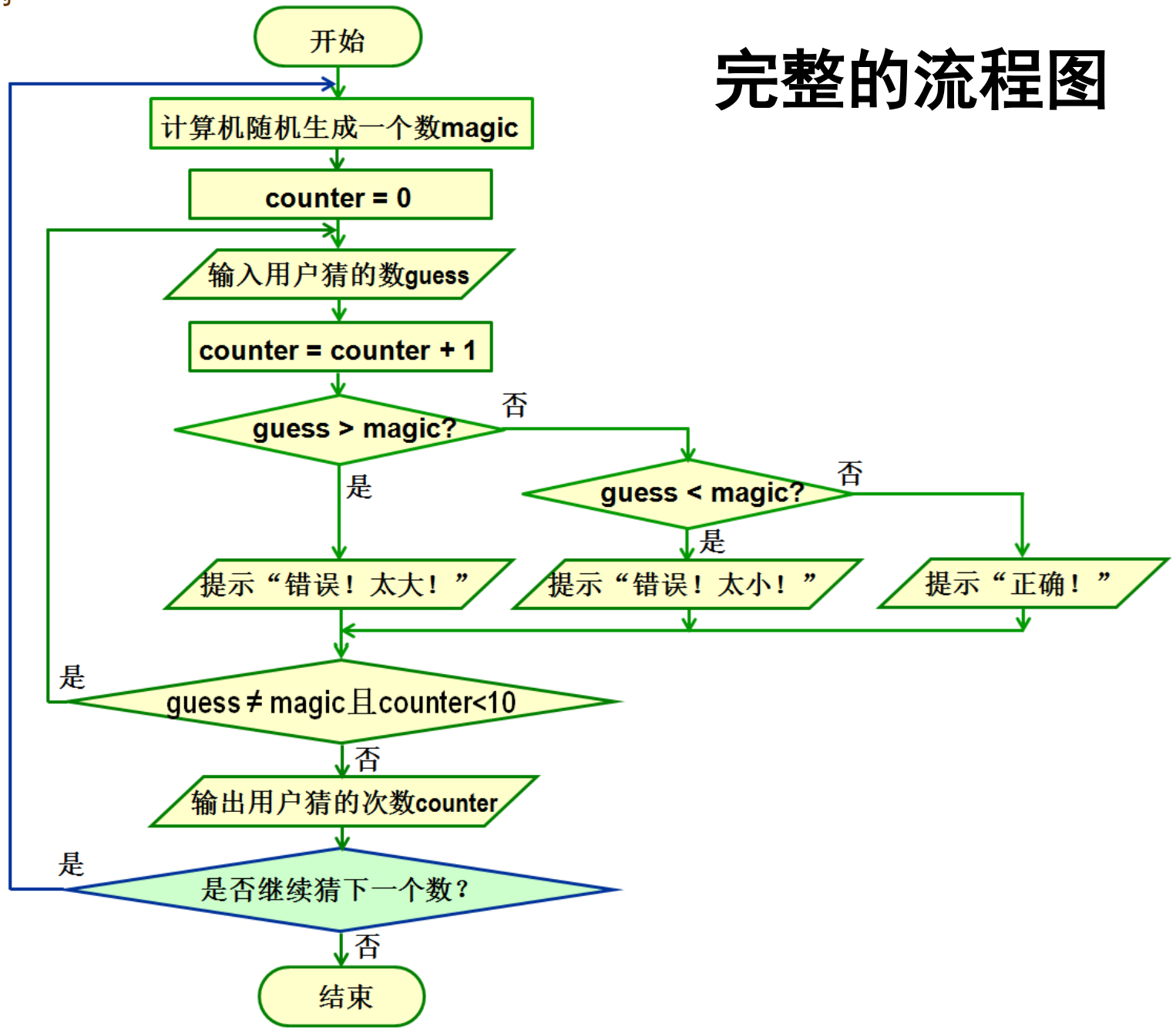
自顶向下、逐步求精


开始
计算机随机生成一个数magic
用户猜数
是否继续猜下一个数？
是
否
结束

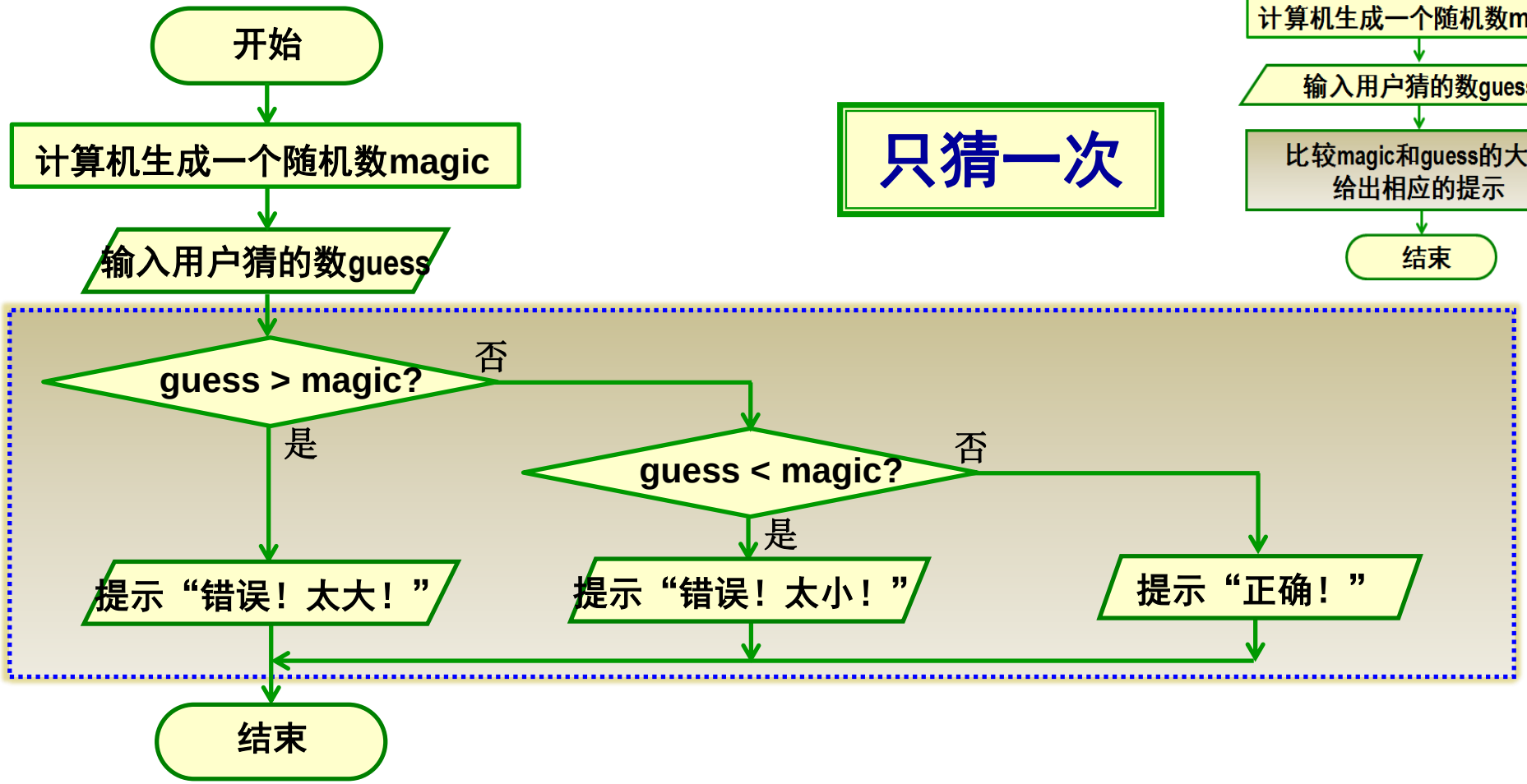

counter = 0
输入用户猜的数guess
counter = counter + 1
比较guess和magic并输出相应提示
guess ≠ magic且counter<10？
是
否
输出用户猜的次数counter


guess > magic?
否
是
guess < magic?
否
是
提示“错误！太大！”
提示“错误！太小！”
提示“正确！”

完整的流程图


开始
计算机随机生成一个数magic
counter = 0
输入用户猜的数guess
counter = counter + 1
guess > magic?
否
是
guess < magic?
否
是
提示“错误！太大！”
提示“错误！太小！”
提示“正确！”
是
guess ≠ magic且counter<10
否
输出用户猜的次数counter
是
是否继续猜下一个数？
否
结束

自底向上设计

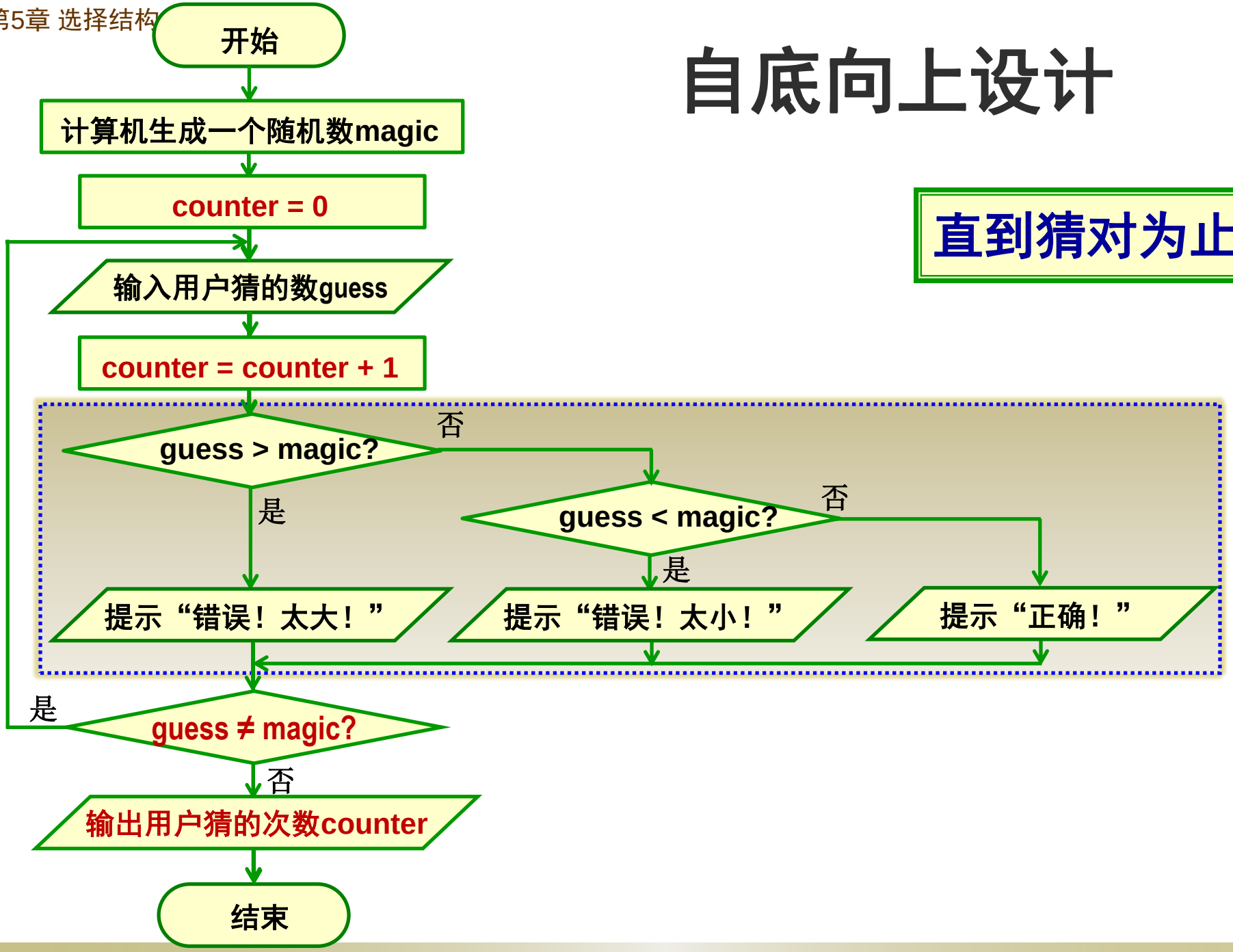

开始
计算机生成一个随机数magic
输入用户猜的数guess
guess > magic?
否
是
guess < magic?
否
是
提示“错误！太大！”
提示“错误！太小！”
提示“正确！”
结束

只猜一次

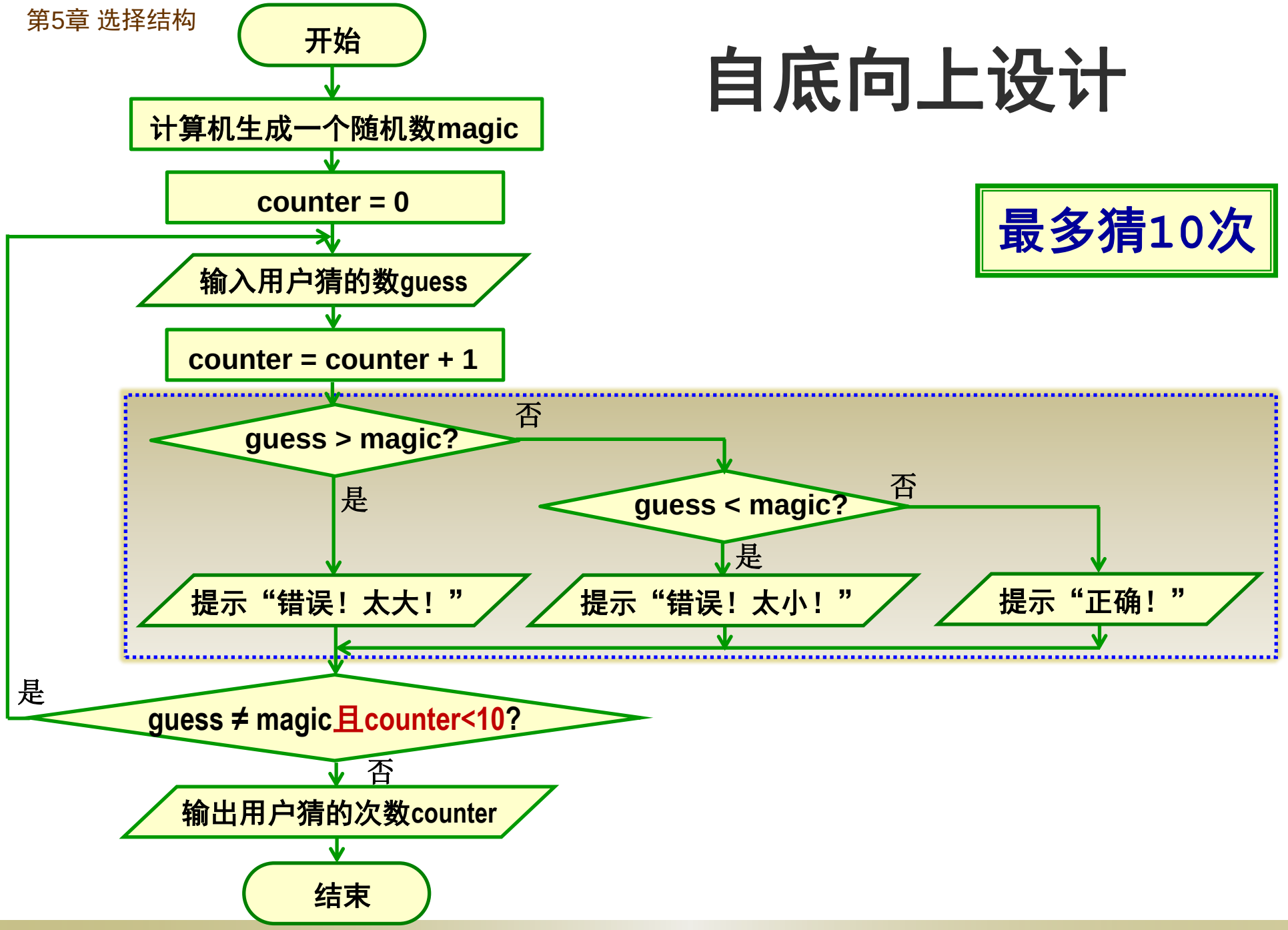
开始
计算机生成一个随机数magic
输入用户猜的数guess
比较magic和guess的大小
给出相应的提示
结束

自底向上设计

直到猜对为止



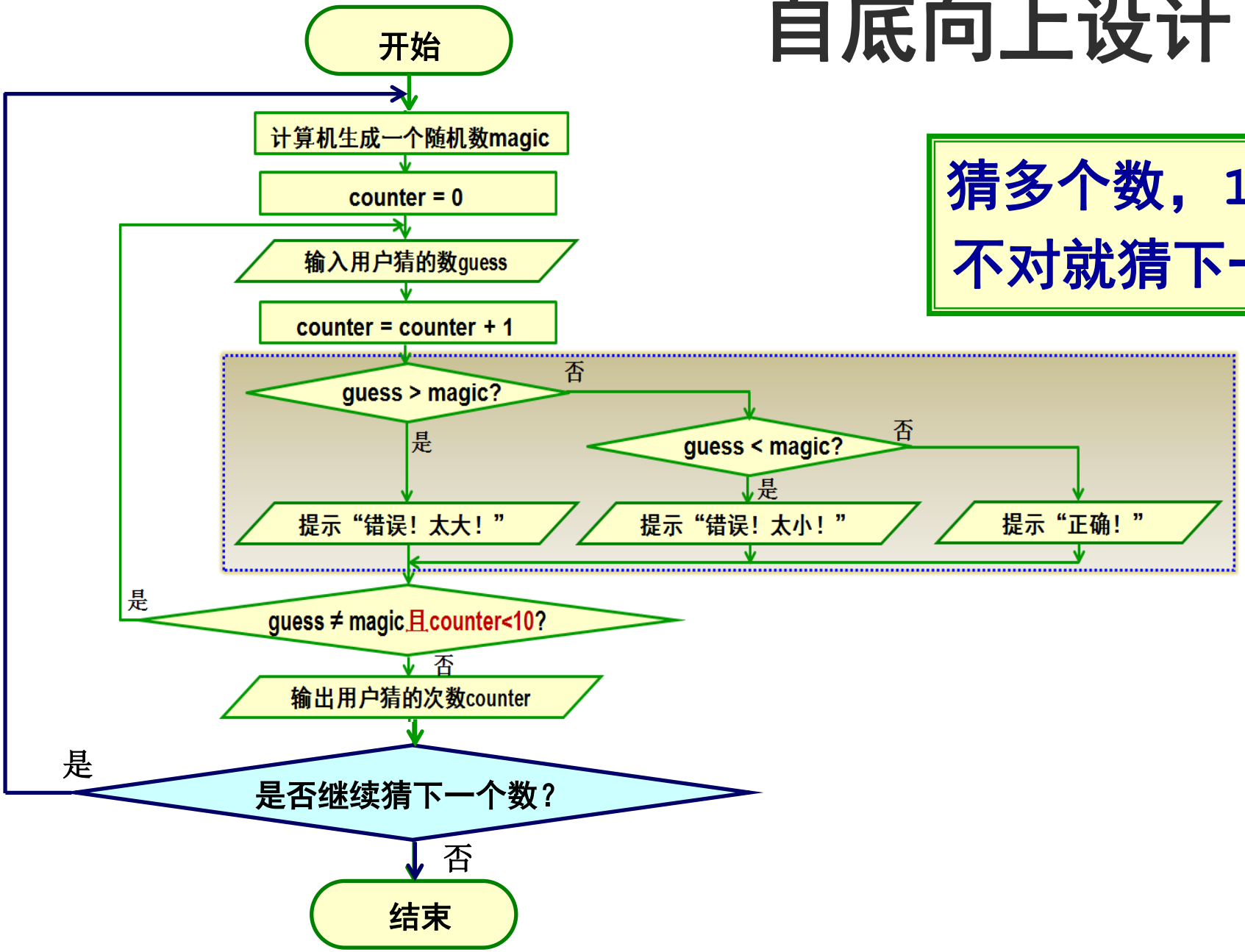
自底向上设计



最多猜10次

自底向上设计

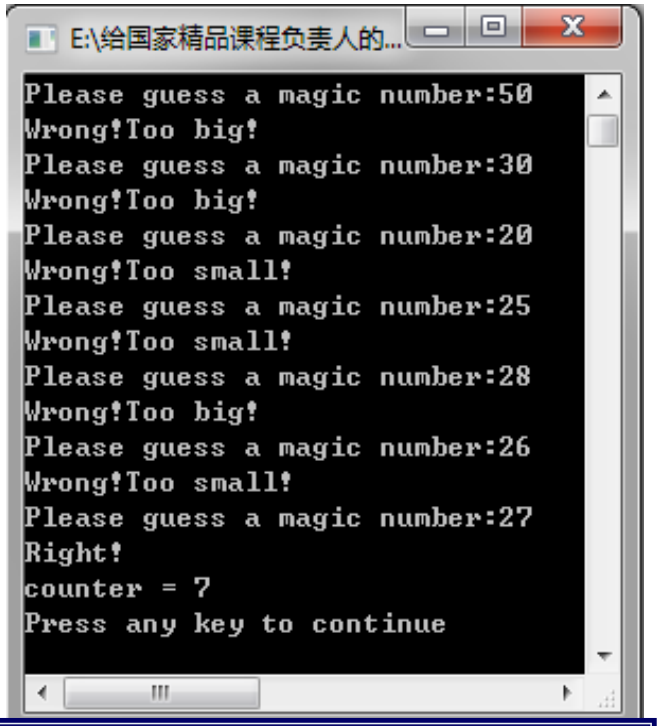
猜多个数，10次猜不对就猜下一个数



```c
#include  <stdlib.h>
#include  <stdio.h>
#include  <time.h>
int main()
{
   int  magic;
   int  guess;
   int  counter;
   srand(time(NULL));//什么作用?
   magic = rand() % 100 + 1;
   counter = 0;
   do{
     printf("Please guess a magic number:");
     scanf("%d", &guess);
     counter++;
     if (guess > magic)
        printf("Wrong! Too big!\n");
     else if (guess < magic)
        printf("Wrong! Too small!\n");
     else
        printf("Right!\n");
   }while (guess != magic);
   printf("counter = %d \n", counter);
  return 0;
}
```

若输入了非数字字符?

直到猜对为止

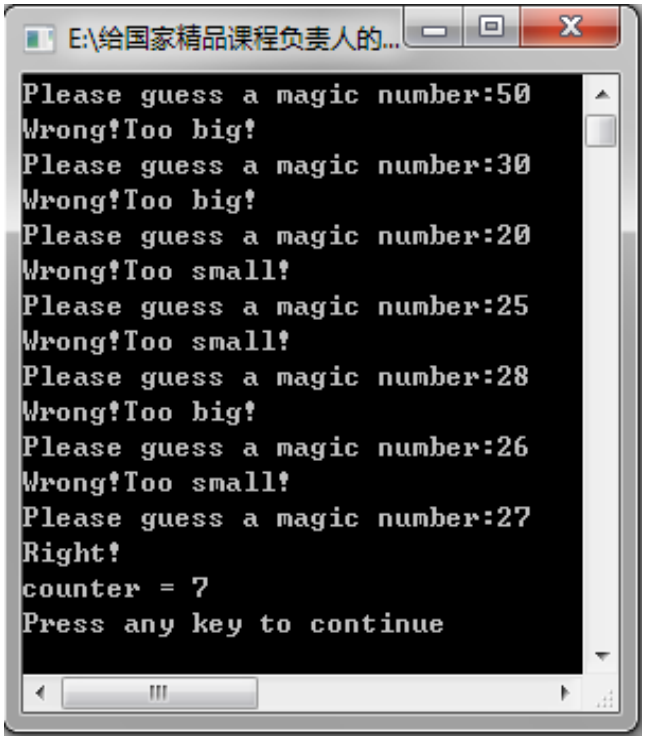


scanf()按指定格式读缓冲区中数据，若读取失败，则缓冲区中的非数字字符不会被读走，一直处于判断、读取、判断、读取、…（死机）

```c
#include  <stdlib.h>
#include  <stdio.h>
#include  <time.h>
int main()
{
   int  magic;
   int  guess;
   int  counter;
   srand(time(NULL));//什么作用?
   magic = rand() % 100 + 1;
   counter = 0;
   do{
     printf("Please guess a magic number:");
     scanf("%d", &guess);
     counter++;
     if (guess > magic)
        printf("Wrong! Too big!\n");
     else if (guess < magic)
        printf("Wrong! Too small
     else
        printf("Right!\n");
   }while (guess != magic);
   printf("counter = %d \n", cou
  return 0;
}
```

若输入了非数字字符?

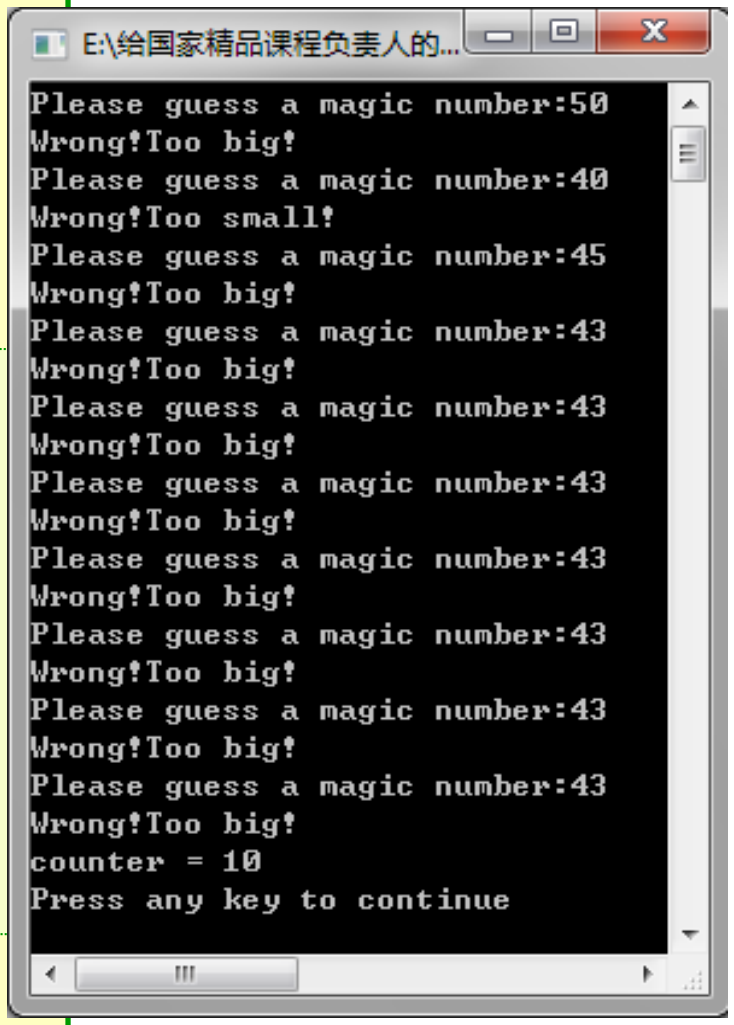
直到猜对为止




```c
while (scanf("%d", &guess) != 1)
{
   while (getchar() != '\n');//什么作用?
   printf("Please guess a magic number:");
}
```

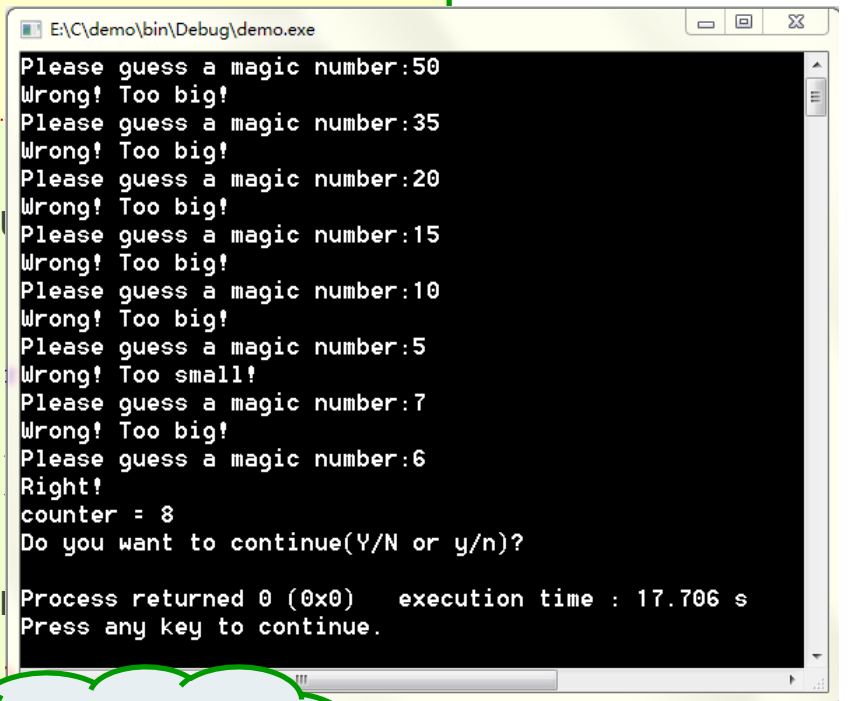
```c
#include  <stdlib.h>
#include  <stdio.h>
#include  <time.h>
int main()
{
   int  magic;
   int  guess;
   int  counter;
   srand(time(NULL));
   magic = rand() % 100 + 1;
   counter = 0;
   do{
     printf("Please guess a magic number:");
     scanf("%d", &guess);
     counter++;
     if (guess > magic)
        printf("Wrong! Too big!\n");
     else if (guess < magic)
        printf("Wrong! Too small!\n");
      else
         printf("Right!\n");
   }while (guess != magic)&& counter < 10);
   printf("counter = %d \n", counter);
  return 0;
}
```

最多猜10次

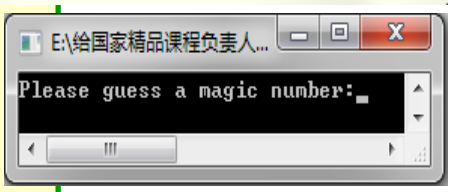


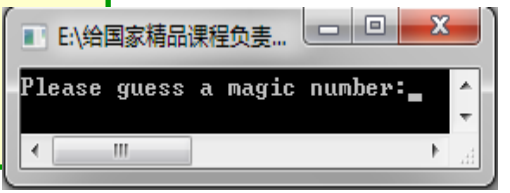
猜多个数

```
……
srand(time(NULL));
do{
        magic = rand() % 100 + 1;
        counter = 0;
        do{
                printf("Please guess a magic nu
                scanf("%d", &guess);
                counter ++;
                if (guess > magic)
                    printf("Wrong! Too big!\n"
                else if (guess < magic)
                    printf("Wrong! Too small!\n
                else
                    printf("Right!\n");
        }while (guess != magic && coun
        printf("counter = %d\n", counter);
        printf("Do you want to continue(Y/N or y/n)?\n");
        scanf(" %c", &reply);//忽略空白字符，或前面加getchar
}while ((reply == 'Y') || (reply == 'y'));
```



若不加空格会怎样？

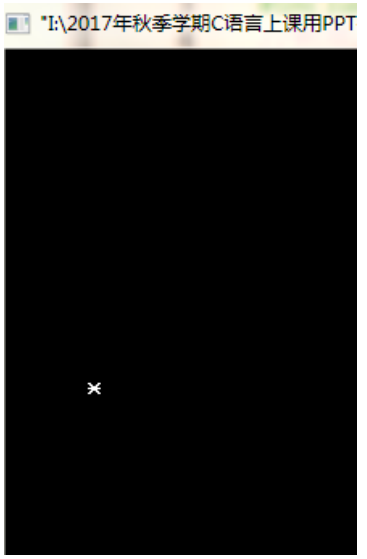




飞机游戏V1.0版

- 飞机用*显示
- 控制飞机移动方式
  - 用scanf()输入a,s,d,w改变,y坐标

```
while (1)
{
    system("cls"); //清屏
    用循环输出空行和空格的方式在第x行第y列输出'*'
    scanf("%c", &input);  //等待键盘输入
    if (input == 'a') y--; //左
    if (input == 'd') y++; //右
    if (input == 'w') y--; //上
    if (input == 's') y++; //下
}
```



x

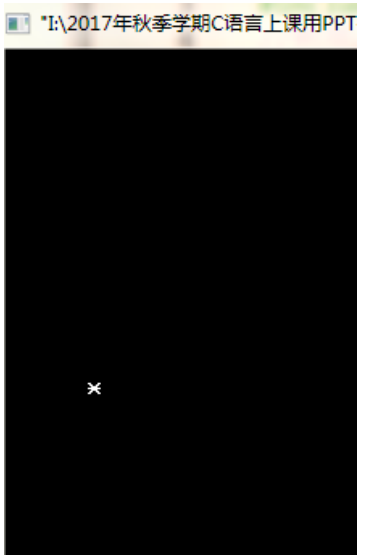
y

飞机游戏V1.0版

- 控制飞机移动方式
  - 用getch()输入a,s,d,w改变,y坐标
  - #include <conio.h>

```
while (1)
{
    system("cls"); //清屏
    用循环输出空行和空格的方式在第x行第y列输出'*'
    input = getch(); //试试看你发现了什么?
    if (input == 'a') y--;
    if (input == 'd') y++;
    if (input == 'w') x--;
    if (input == 's') x++;
}
```

x



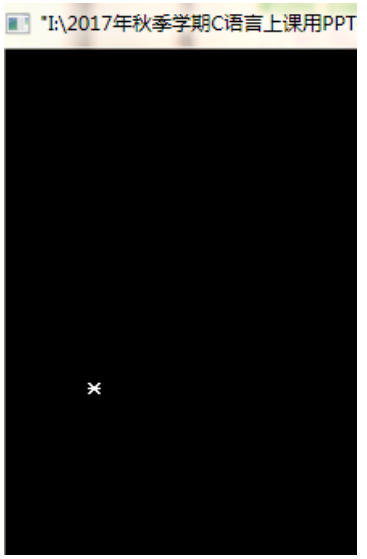
y

飞机游戏V1.0版

- 控制飞机移动方式
  - 用kbhit()检测是否有键盘输入
  - #include <conio.h>

```
while (1)
{
    system("cls"); //清屏
    用循环输出空行和空格的方式在第x行第y列输出'*'
    if (kbhit()) //没有键盘输入就循环显示'*'
    {
        input = getch();
        if (input == 'a') y--;
        if (input == 'd') y++;
        if (input == 'w') x--;
        if (input == 's') x++;
    }
}
```

x

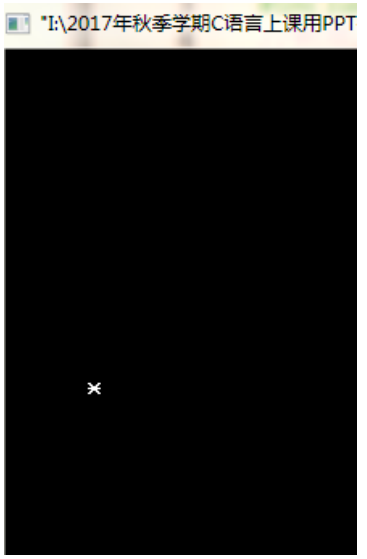


y

飞机游戏V1.0版

- **控制飞机移动方式**
  - 用上下左右键控制飞机移动
  - 键盘扫描码：键盘上的每一个键都有两个唯一的数值进行标志

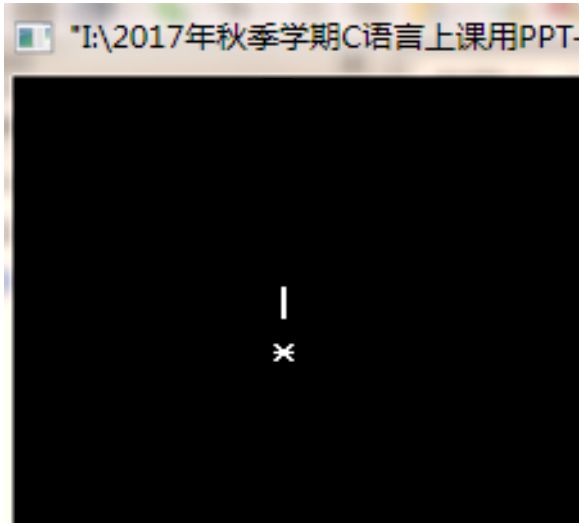
```
while (1)
{
    system("cls"); //清屏
    用循环输出空行和空格的方式在第x行第y列输出'*'
    if (kbhit()) //没有键盘输入就循环显示'*'
    {
        input = getch();
        if (input == 75) y--; //左
        if (input == 77) y++; //右
        if (input == 72) x--; //上
        if (input == 80) x++; //下
    }
}
```



课后作业

- 飞机游戏V2.0版
- 设游戏画面尺寸为high*width
- 按空格键时，让飞机发射移动的激光子弹（'|'）

x



y