第5章 循环控制

——条件控制的循环

选择循环的一般原则

- for，while，do-while
- Just a general rule.

循环次数已知	计数控制的循环 for语句
循环次数未知 由一个给定的条件来控制	条件控制的循环 while语句
循环体至少要执行一次	直到型循环 do-while语句

A Shopping street Game

- 购物街“看商品猜价格”游戏

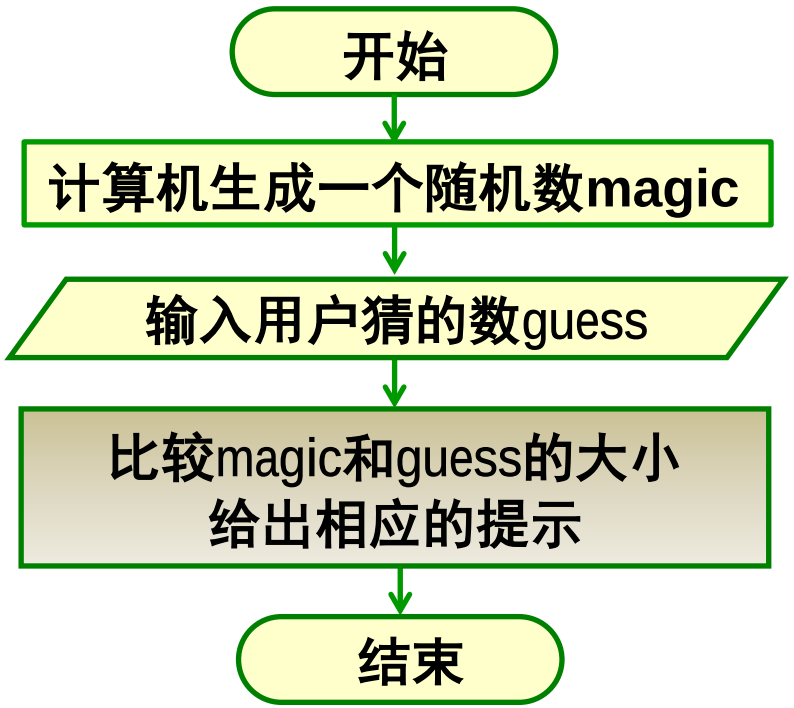
Guess the price of a piece of goods（an integer, [1,100]）

Right: Congratulations!
商品归你了！

Wrong: greater or less?
太大了！ or 太小了！

Shopping street Game—Guess a number

- 猜数游戏
  - 核心操作就是比较两个数的大小
  - 两个数比较大小会有几种情况呢？



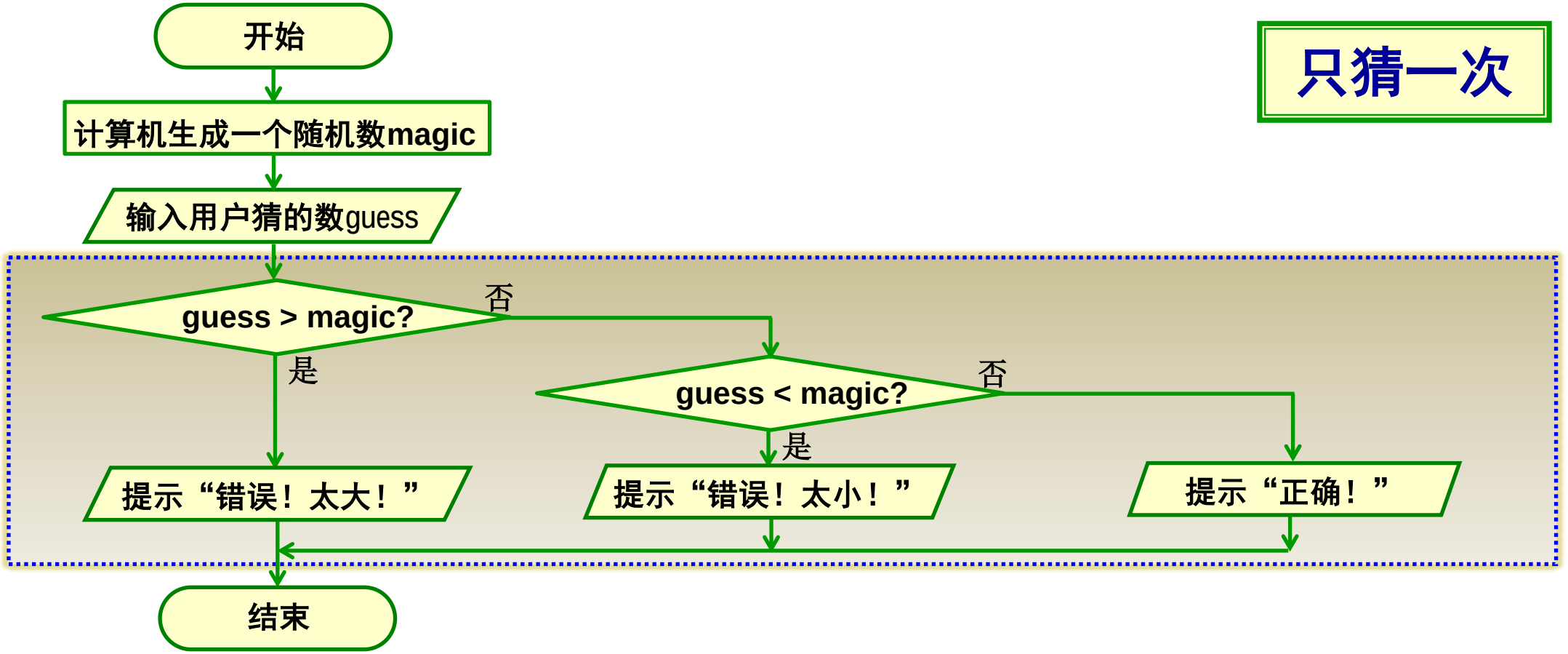
猜多个数，10次猜不对就猜下一个数

最多猜10次

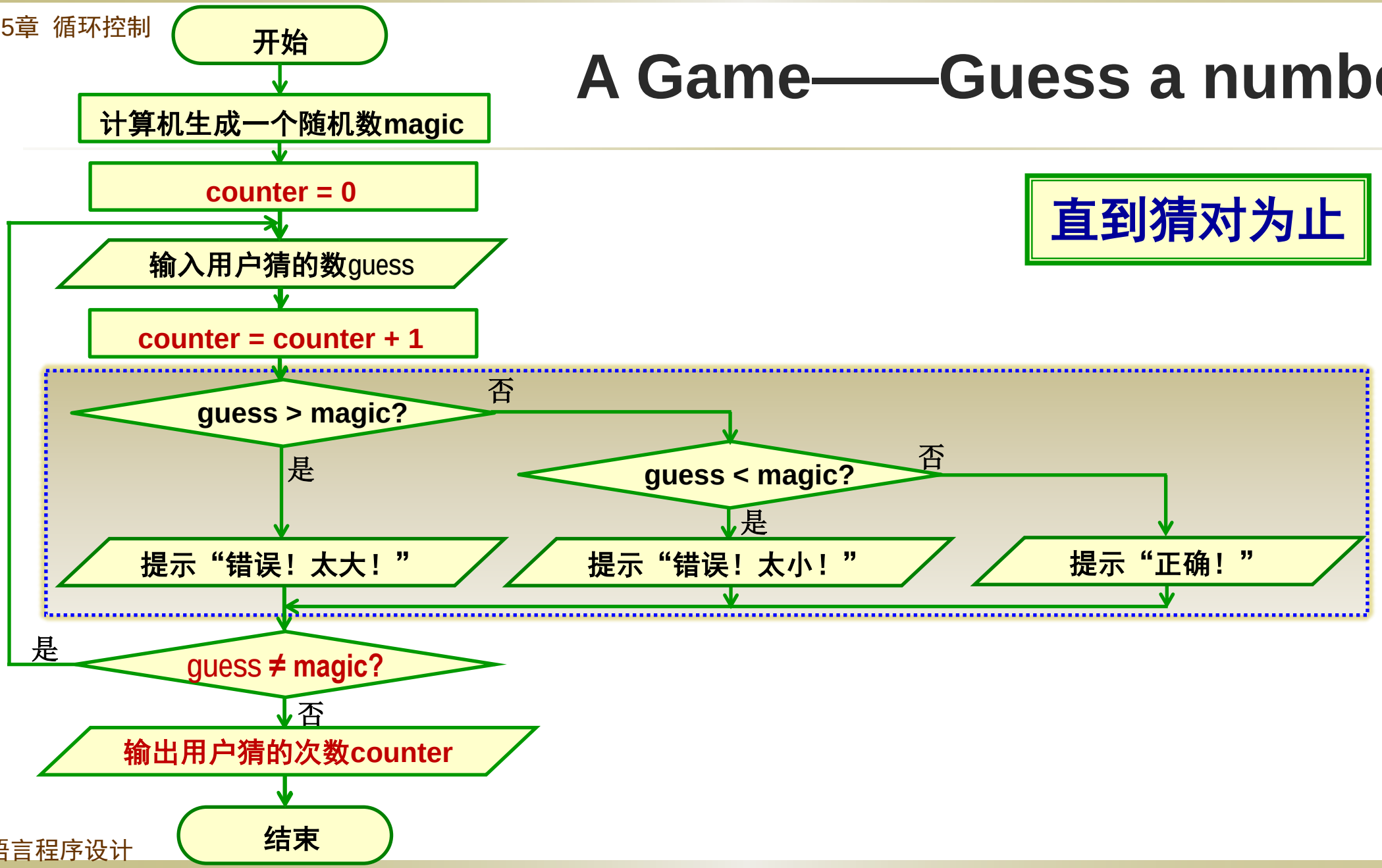
直到猜对为止

只猜一次

A Game——Guess a number

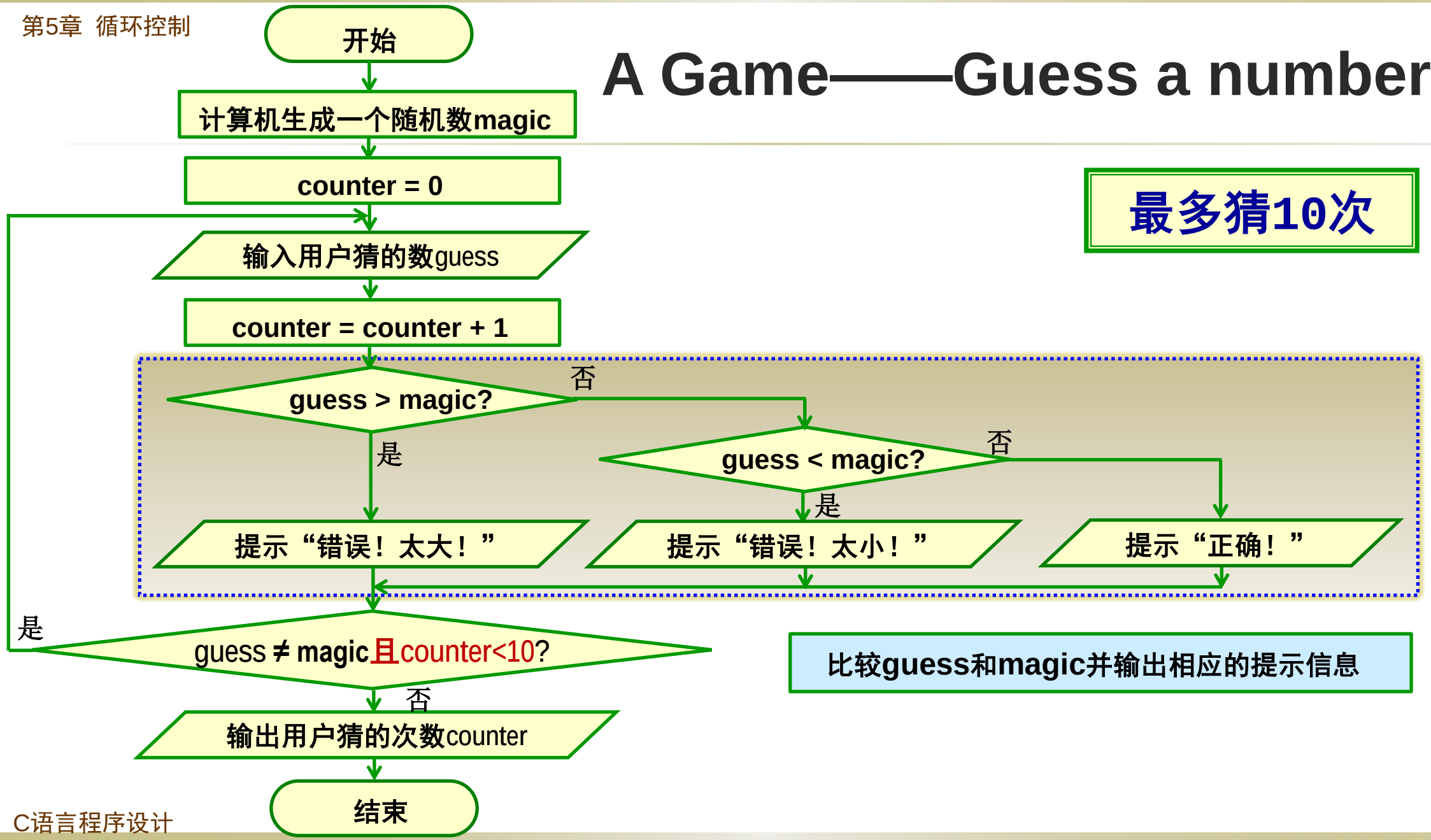


A Game——Guess a number



直到猜对为止

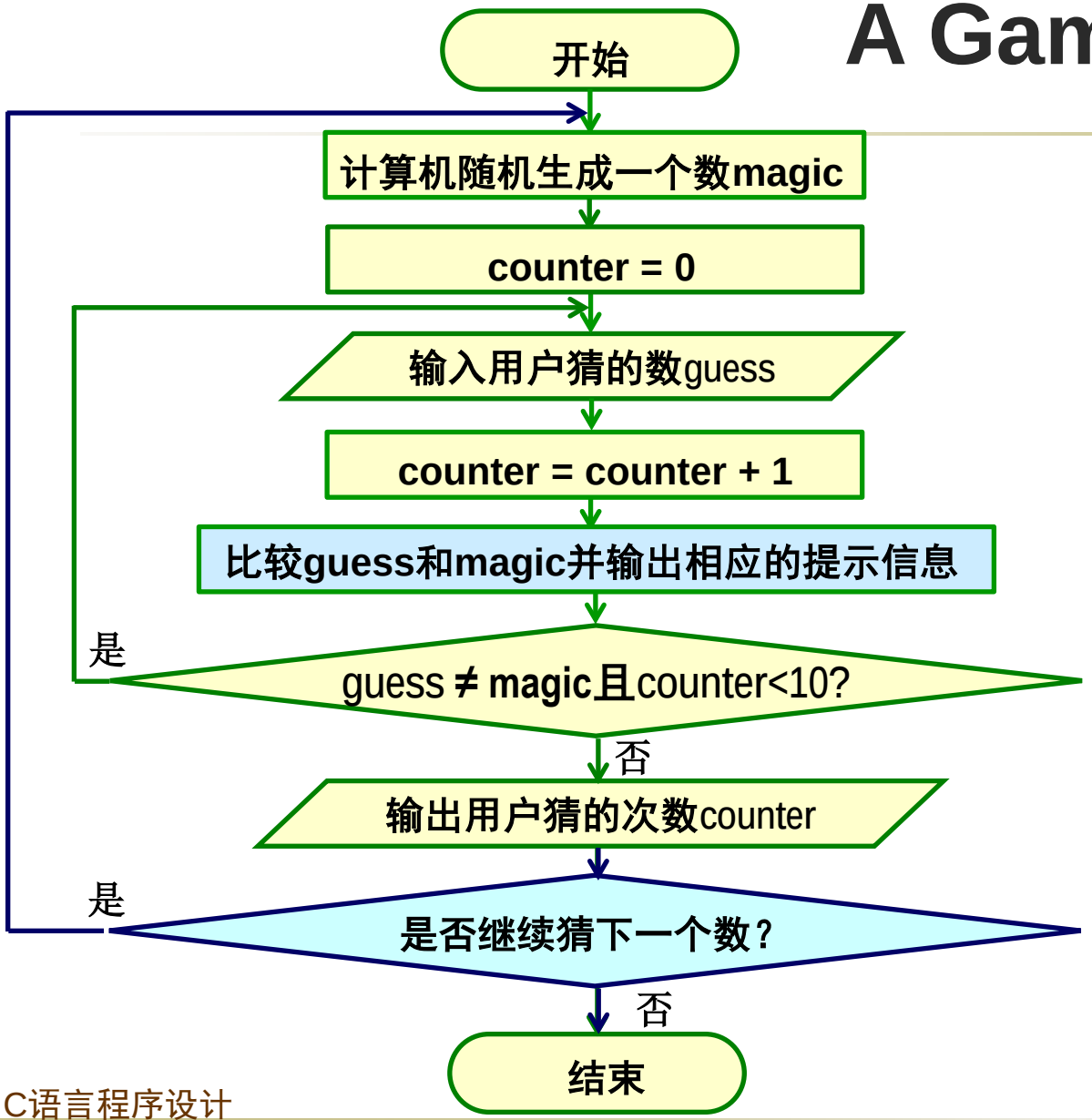
A Game——Guess a number



最多猜10次

比较guess和magic并输出相应的提示信息

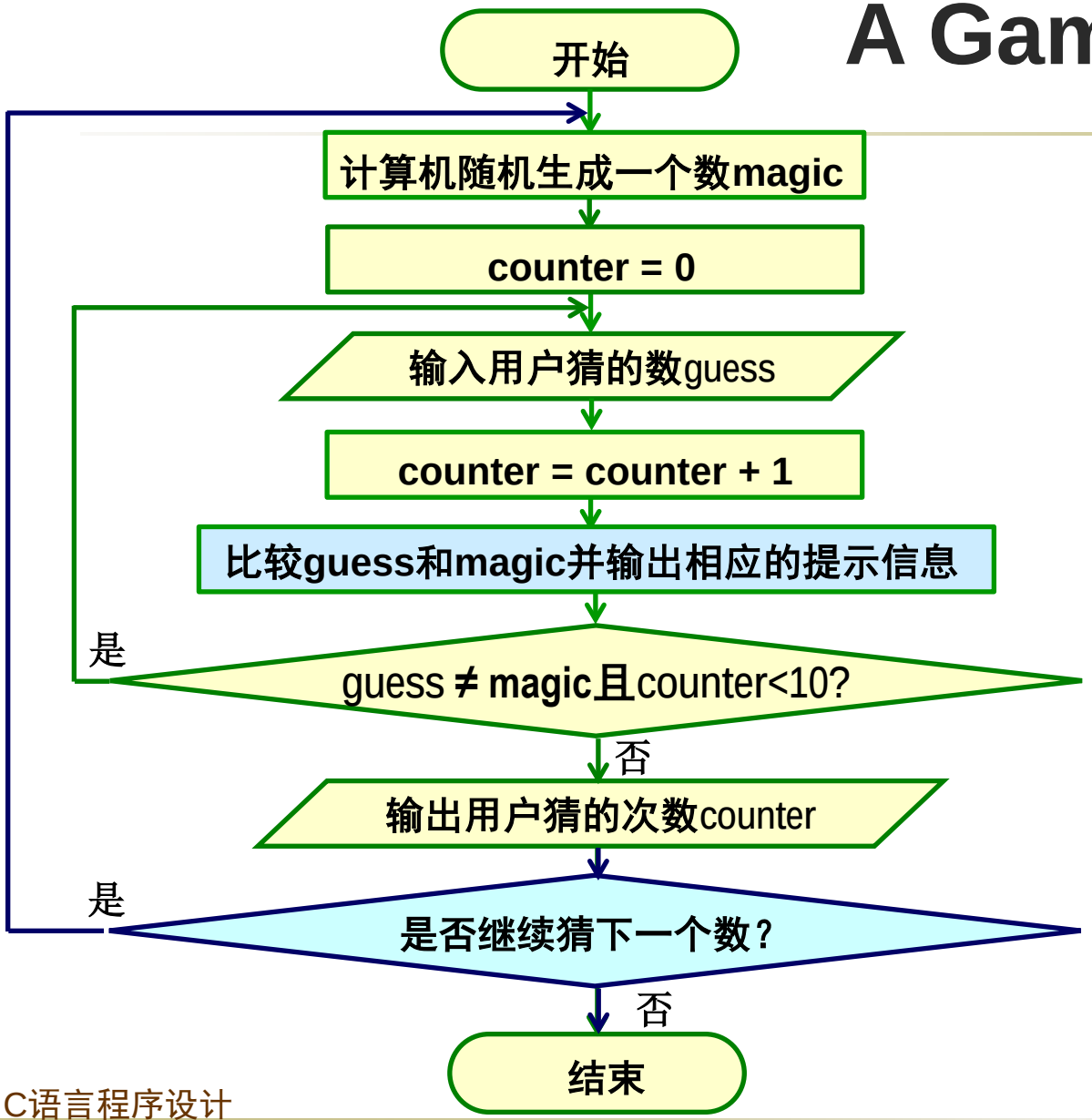
A Game——Guess a number

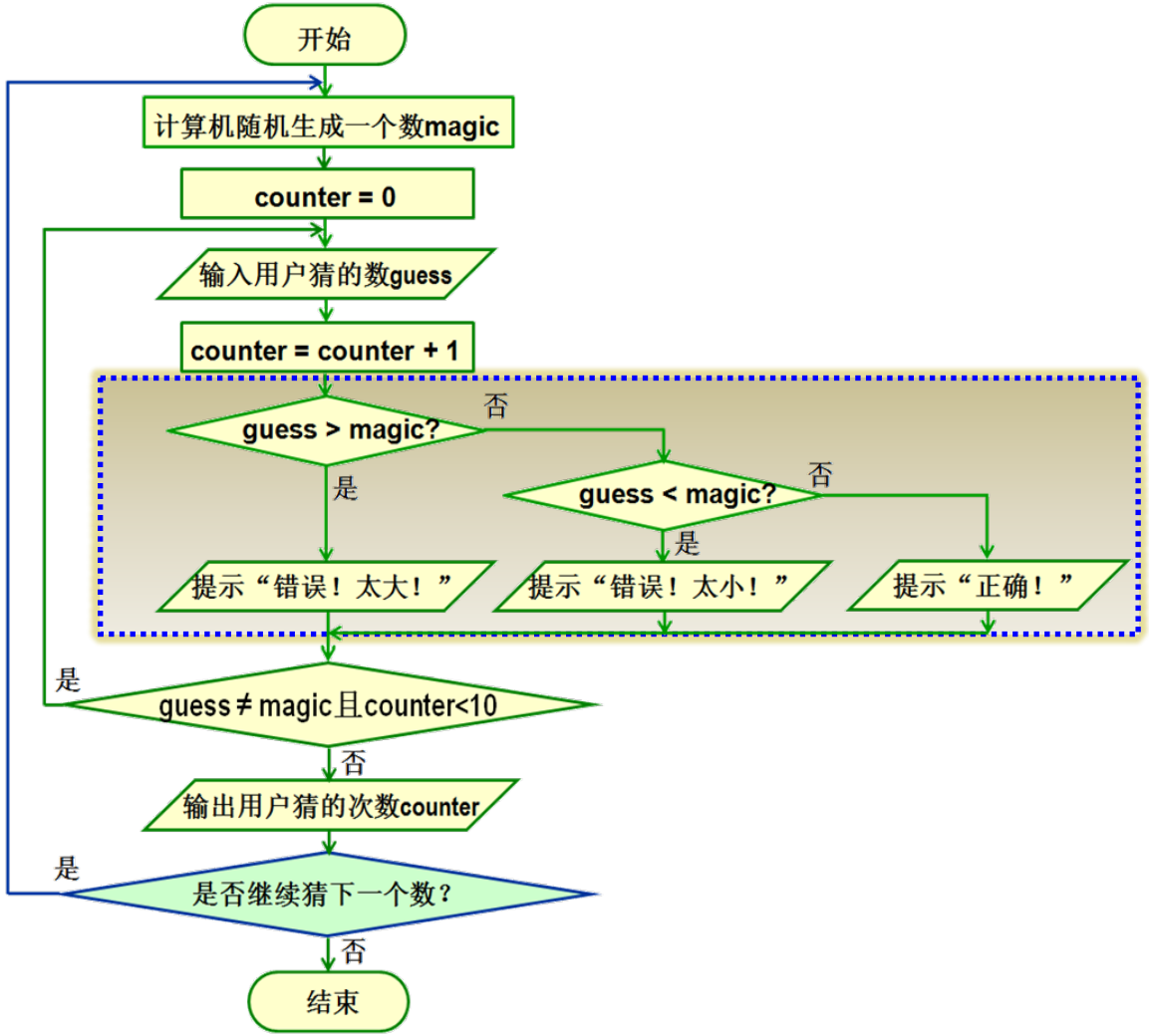


猜多个数，10次猜不对就猜下一个数

嵌套循环

A Game——Guess a number



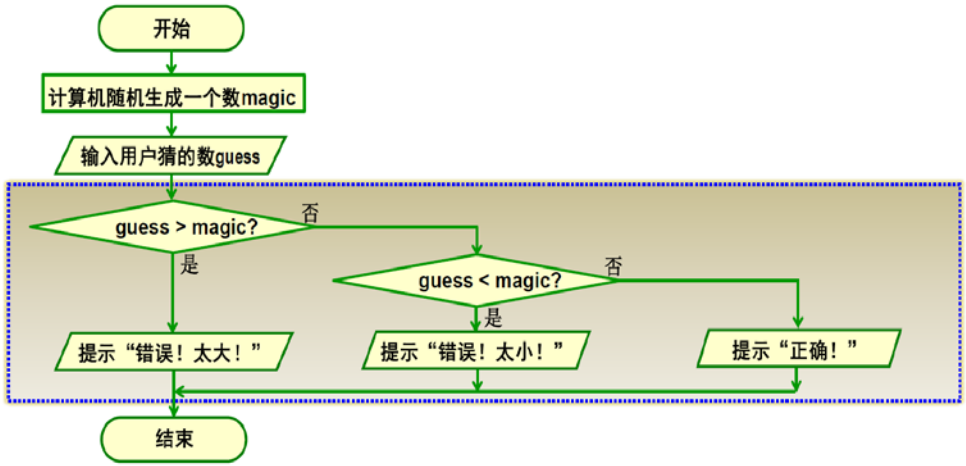


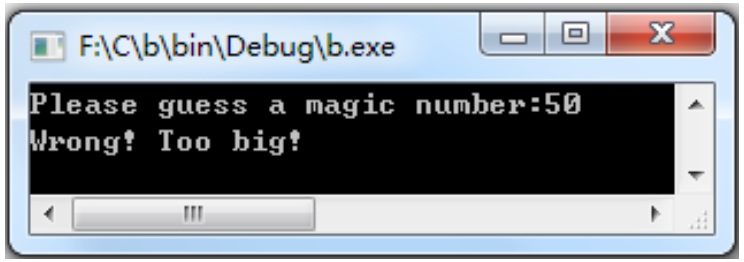
程序实现猜数游戏的关键

- 随机函数rand()
  - magic = rand();
  - 产生[0,RAND_MAX]间的随机数
  - RAND_MAX在stdlib.h中定义，不大于双字节整数的最大值32767
  - #include <stdlib.h>
- 产生[0,99] 之间的随机数
  - magic = rand()%100;
- 产生[1,100] 之间的随机数
  - magic = rand()%100 + 1;

只猜一次

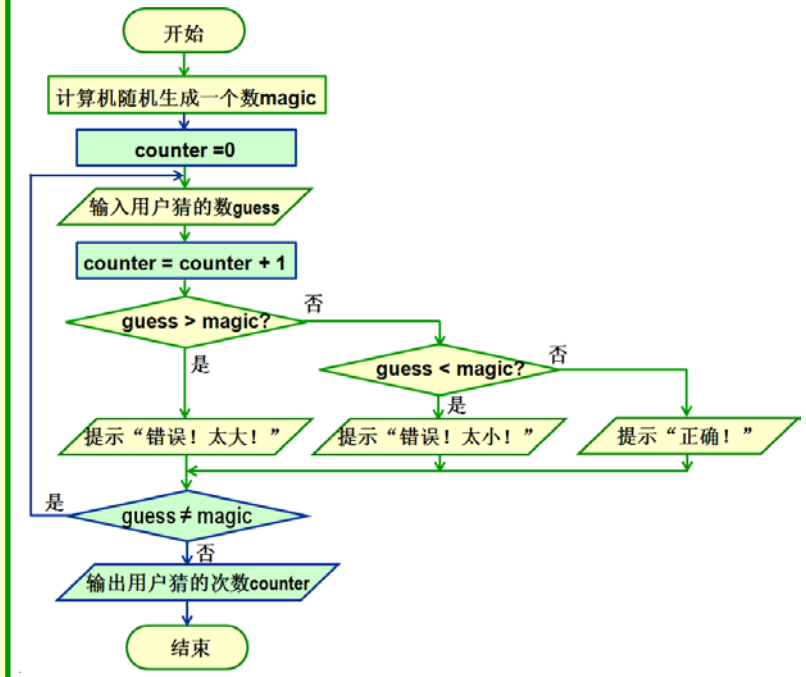
```
#include  <stdlib.h>
#include  <stdio.h>
int main()
{
   int  magic;         /*计算机"想"的数*/
   int  guess;         /*人猜的数*/
   magic = rand()%100 + 1;
   printf("Please guess a magic number:");
   scanf("%d", &guess);

   if (guess > magic)
   {
        printf("Wrong! Too big!\n");
   }
   else if (guess < magic)
   {
        printf("Wrong! Too small!\n");
   }
   else
   {
       printf("Right! \n");
       printf("The number is:%d \n", magic);
   }
    return 0;
}
```

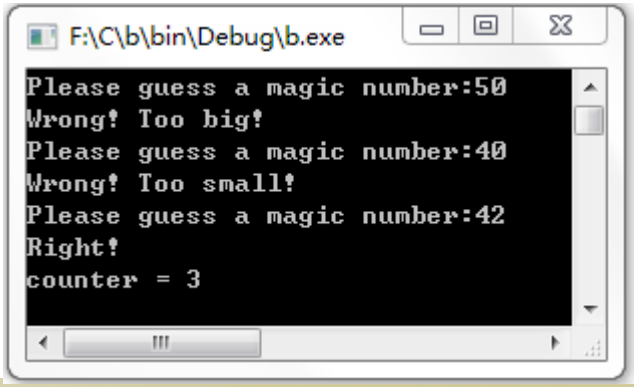




```c
#include  <stdlib.h>
#include  <stdio.h>
int main()
{
   int  magic;
   int  guess;
   int  counter;    /*记录人猜次数的计数器变量*/
   magic = rand() % 100 + 1;
   counter = 0;     /*计数器变量count初始化为0*/
   do{
     printf("Please guess a magic number:");
     scanf("%d", &guess);
     counter++;     /*计数器变量count加1*/
     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;
}
```

直到猜对为止

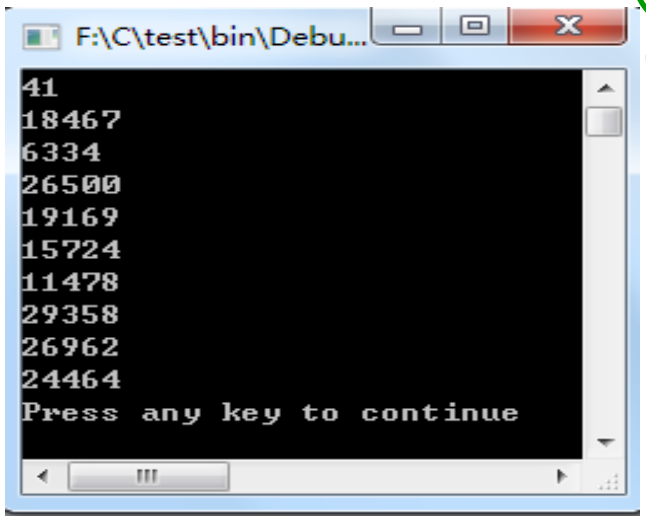


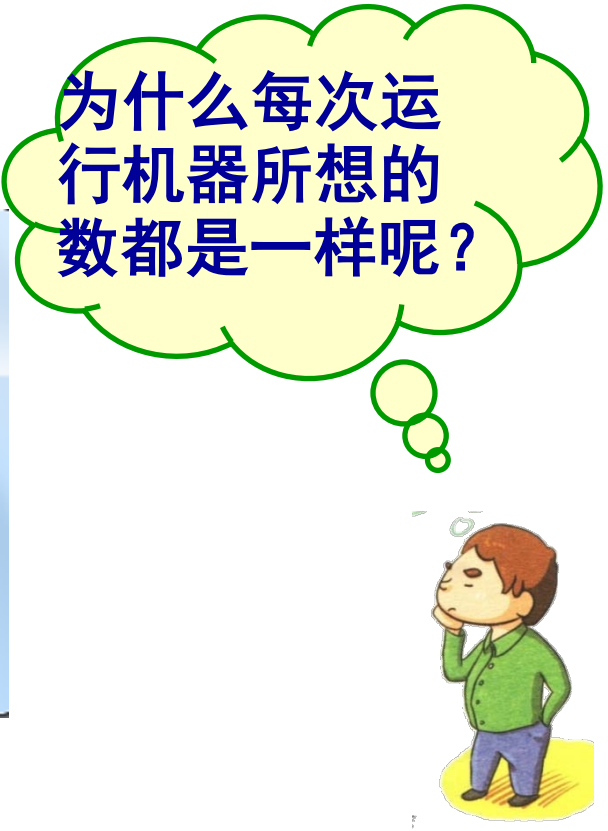


猜数游戏用到的库函数

- 函数rand()产生的是伪随机数

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int i;
   for (i=0; i<10; i++)
   {
      printf("%d\n", rand());
   }
   return 0;
}
```

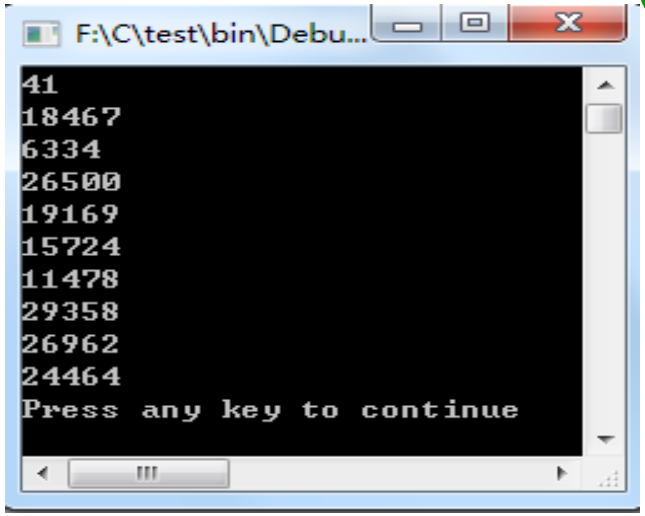


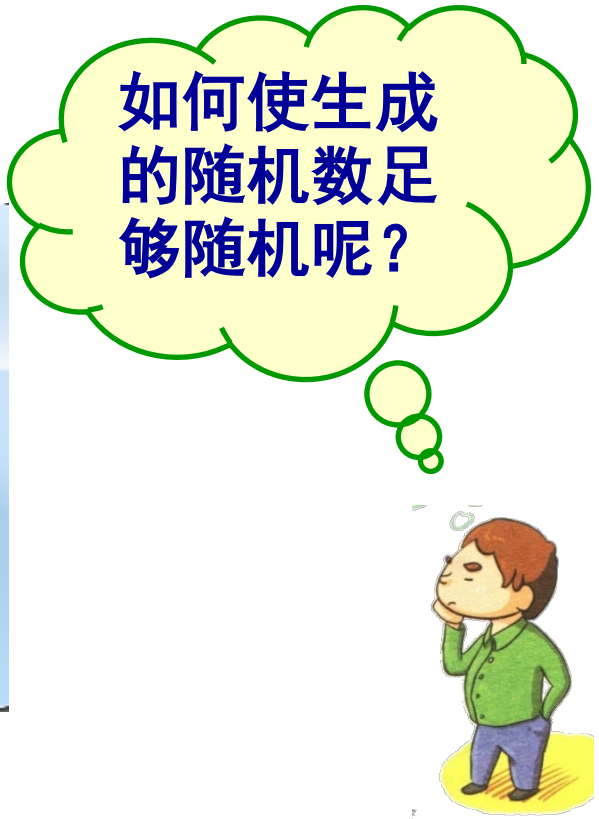


猜数游戏用到的库函数

- 函数rand()产生的是伪随机数

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int i;
   for (i=0; i<10; i++)
   {
      printf("%d\n", rand());
   }
   return 0;
}
```

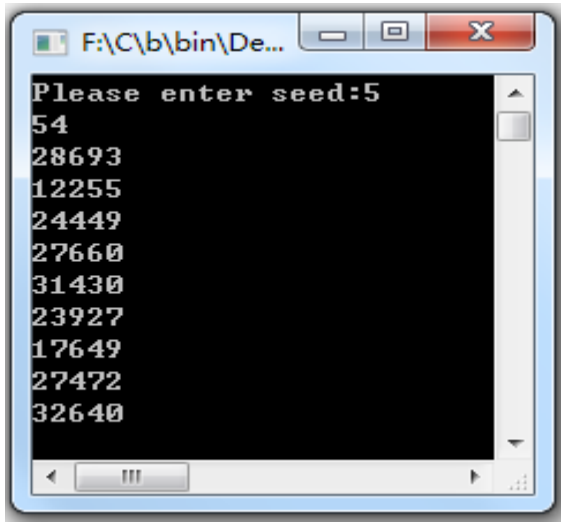




猜数游戏用到的库函数

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int i;
   unsigned int seed;
   printf("Please enter seed:");
   scanf("%u", &seed);
   srand(seed);
   for (i=0; i<10; i++)
   {
     printf("%d\n", rand());
   }
   return 0;
}
```

为rand()设置随机数种子，
使产生的随机数“随机化”



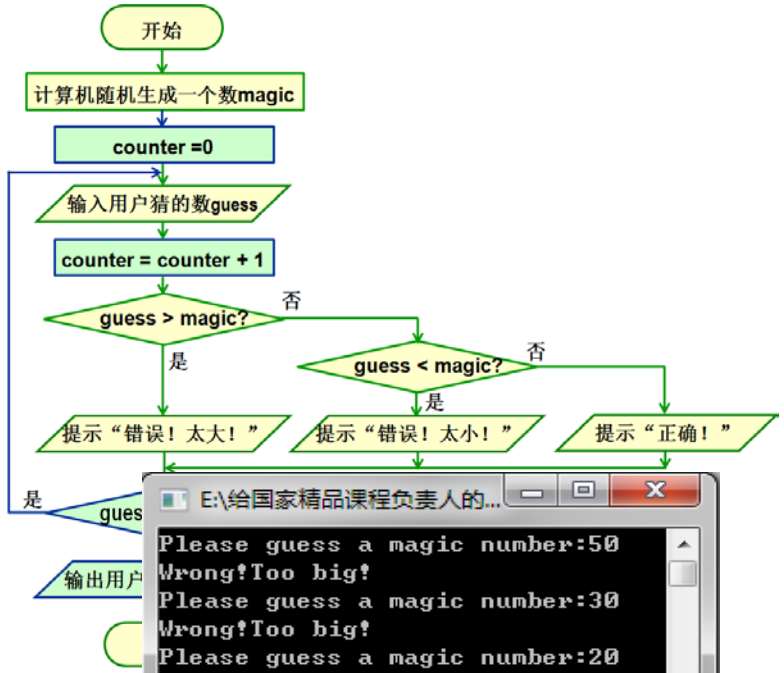
猜数游戏用到的库函数

```
#include <stdio.h>
#include <stdlib.h>
#include  <time.h>
int main()
{
   int i;
  srand(time(NULL));
   for (i=0; i<10; i++)
   {
    printf("%d\n", rand());
   }
   return 0;
}
```

- 改用系统时间作为随机数种子更好
- 用函数time()获得系统时间
  - 两种方法——函数参数，函数返回值
  - 用NULL作为函数参数，使其仅能从返回值取得系统时间，便于将函数写到表达式中

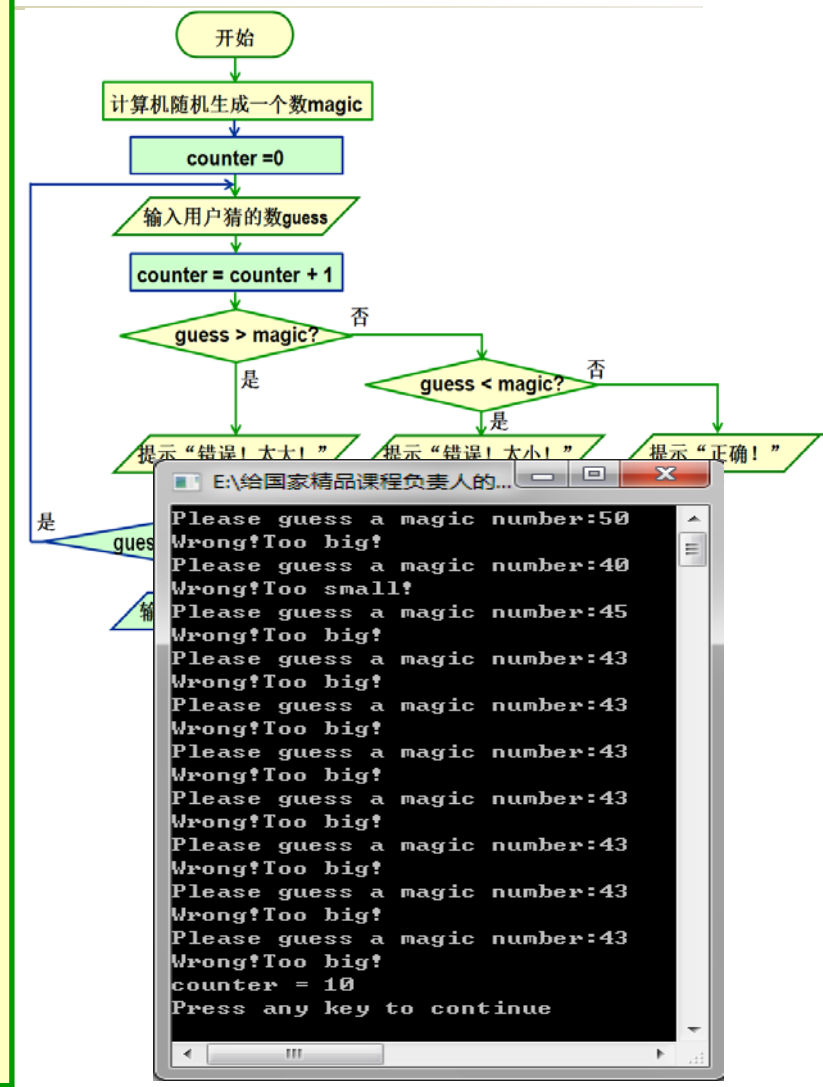
直到猜对为止

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



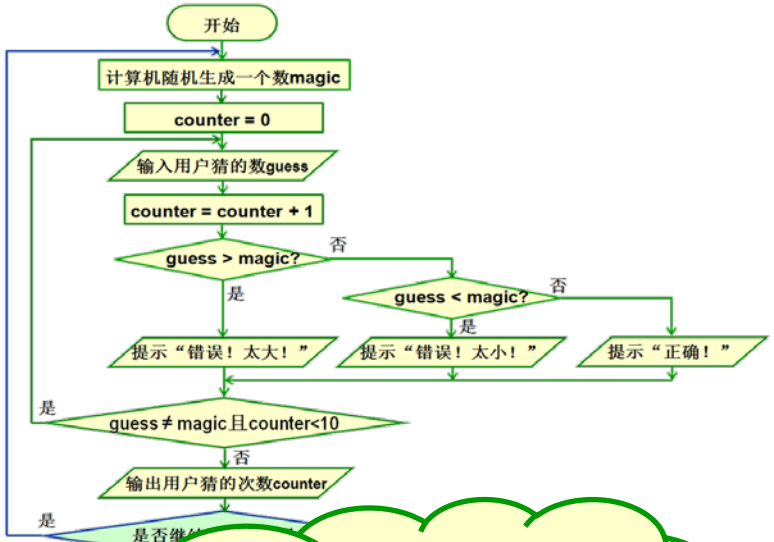
```c
#include  <stdlib.h>
#include  <stdio.h>
#include  <time.h>
int main()
{
   int  magic;
   int  guess;
   int  counter;
   srand(time(NULL));
   magic = rand() % 100 + 1;
   counter = 0;
   do{
     printf("Please guess a magic number:");
     scanf("%d", &guess);
     counter++;
     if (guess > 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

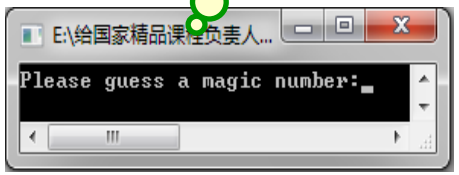


```
……
srand(time(NULL));
do{
        magic = rand() % 100 + 1;
        counter = 0;
        do{
                printf("Please guess a magic number:");
                scanf("%d", &guess);
                counter ++;
                if (guess > magic)
                    printf("Wrong! Too big!\n");
                else if (guess < magic)
                    printf("Wrong! Too small!\n");
                else
                    printf("Right!\n");
        }while (guess != magic && counter < 10);
        printf("counter = %d\n", counter);
        printf("Do you want to continue(Y/N or y/n)?\n");
        scanf(" %c", &reply);
}while ((reply == 'Y') || (reply == 'y'));
```

猜多个数

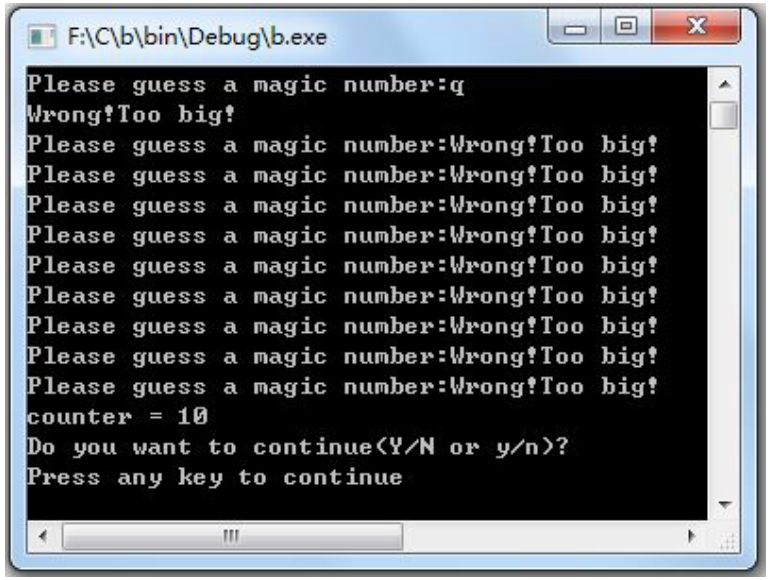


若格式不匹配，输入了非数字字符？



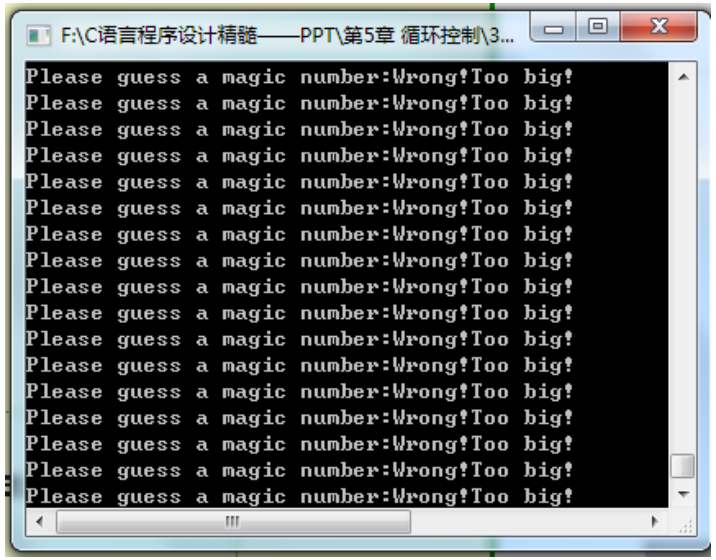
猜多个数

```
……
srand(time(NULL));
do{
        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);
        printf("Do you want to continue(Y/N or y/n)?\n");
        scanf(" %c", &reply);
}while ((reply == 'Y') || (reply == 'y'));
```



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

直到猜对为止

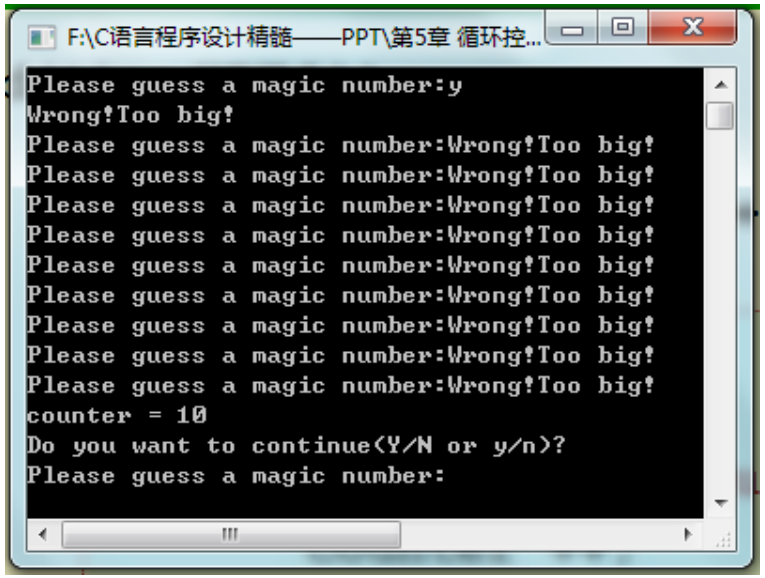


scanf()按指定格式读取缓冲区中的数据，若读取失败，则缓冲区中的非数字字符不会被读走，因不等而一直处于判断、读取、判断、读取、…（死机）

```
……
srand(time(NULL));
do{
        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);
        printf("Do you want to continue(Y/N or y/n)?\n");
        scanf(" %c", &reply);
}while ((reply == 'Y') || (reply == 'y'));
```

如何判断处理输入的非数字字符?

猜多个数



这个为什么没有死机?
缓冲区中的非数字字符被谁读走了?

猜多个数

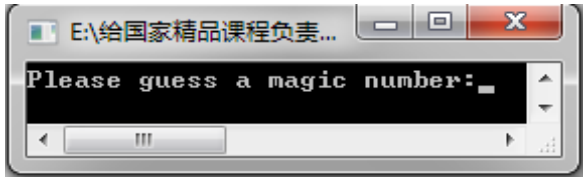
```
……
srand(time(NULL));
do{
        magic = rand() % 100 + 1;
        counter = 0;
        do{
                printf("Please guess a magic number:");
                scanf("%d", &guess);
                counter ++;
                if
                el
                                                    ")
                else
                    printf("Right!\n");
        }while (guess != magic && counter < 10);
        printf("counter = %d\n", counter);
        printf("Do you want to continue(Y/N or y/n)?\n");
        scanf(" %c", &reply);
}while ((reply == 'Y') || (reply == 'y'));
```

```
while (getchar() != '\n')
{
    ;//空语句
}
```

scanf()返回值为正确读入的数据项数

```
ret = scanf("%d", &guess);
while (ret != 1)
{
    while (getchar() != '\n');
    printf("Please guess a magic number:");
    ret = scanf("%d", &guess);
}
counter ++;
```

清除输入缓冲区中的残留数据，然后提示重新输入



```
……
srand(time(NULL));
do{
        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);
        printf("Do you want to continue(Y/N or y/n)?\n");
        scanf(" %c", &reply);
}while ((reply == 'Y') || (reply == 'y'));
```

讨论

改成(reply != 'N') && (reply != 'n')后的执行效果是否完全一样?