

# 第5章 循环控制

—while和do-while语句及其在递推求解中的应用

---

# 递推法的基本思想

## ■ 递推法

- \* 利用问题本身所具有的一种递推关系来求解问题的一种方法

## ■ 递推

- \* 从已知的初始条件出发，依据某种递推关系，逐次推出所要计算的中间结果和最终结果
  - \* 初始条件要么在问题本身中已经给定
  - \* 要么需要通过对问题的分析和化简后来确定

# 递推的基本方法

## ■ 正向顺推

- \* 从已知条件出发，向着所求问题的解前进，一步一步推出答案



## ■ 反向逆推

- \* 从问题的结果出发，一步一步还原出答案




# 用递推法求解问题的实例

## ■ 猴子吃桃问题

- \* 猴子第一天摘下若干个桃子，吃了一半，还不过瘾，又多吃了一个。第二天早上又将剩下的桃子吃掉一半，并且又多吃了一个。以后每天早上都吃掉前一天剩下的一半零一个。到第10天早上再想吃时，发现只剩下一个桃子。问第一天共摘了多少桃子。
- \* 若猴子每天少吃1个（多剩1个），则每天剩下的桃子数将是前一天的一半
- \* 每天剩下的桃子数加1之后，刚好是前一天的一半

天数：      1            2            3            4            5            6            7            8            9            10

 1534   ← 766   ← 382   ← 190   ← 94   ← 46   ← 22   ← 10   ← 4   ← 

$$x_n = 1$$

$$n = 10$$

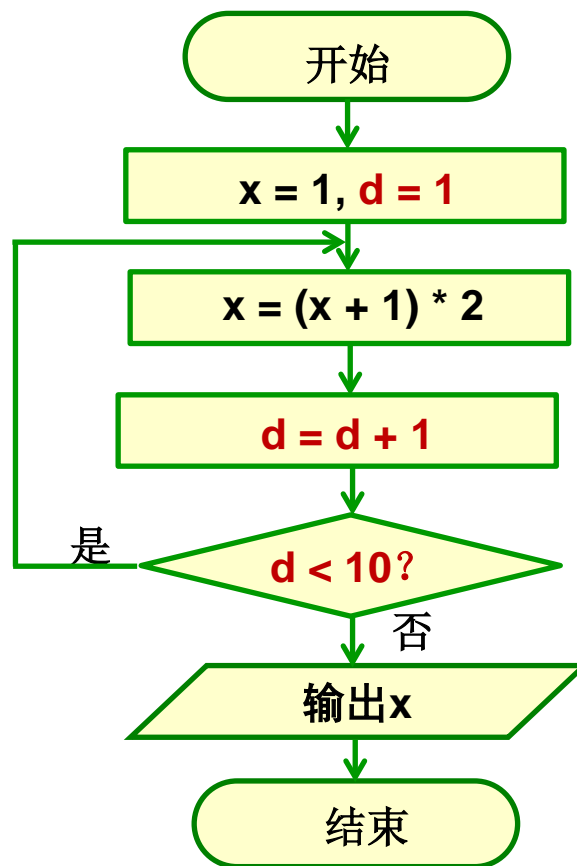
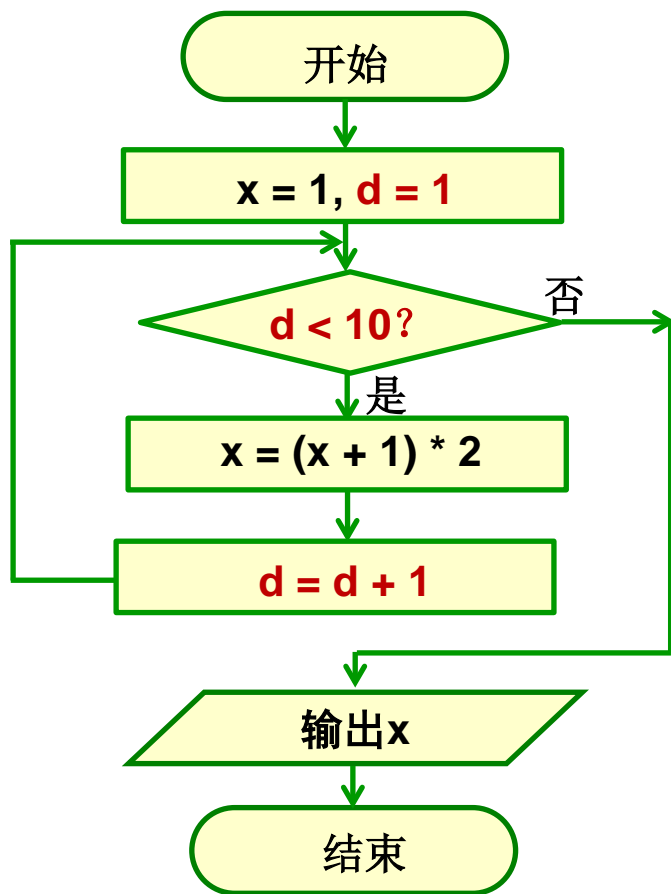
$$x_n = 2 \times (x_{n+1} + 1)$$

$$1 \leq n < 10$$

递推9次



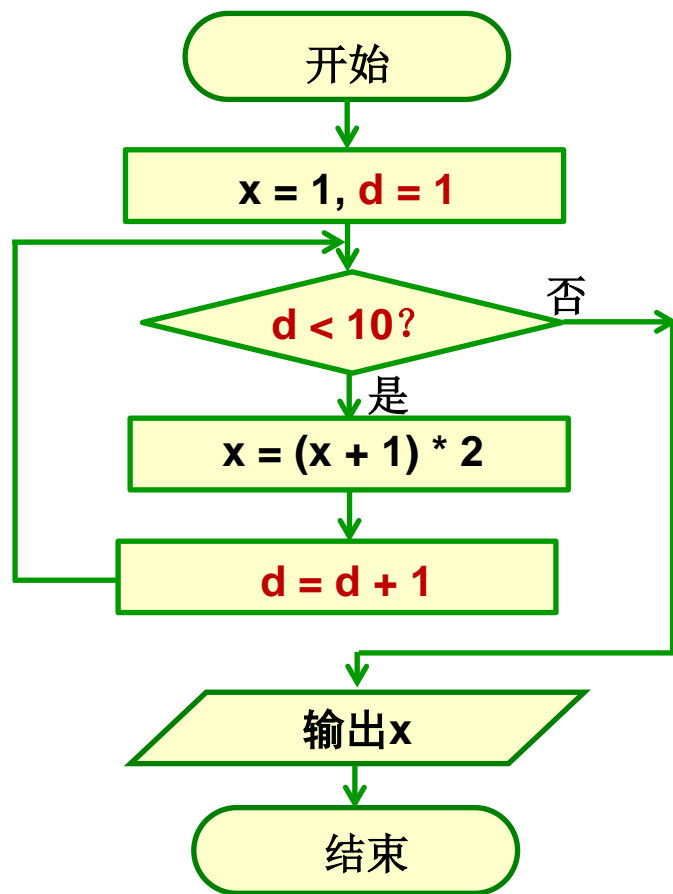
# 用递推法求解问题的实例



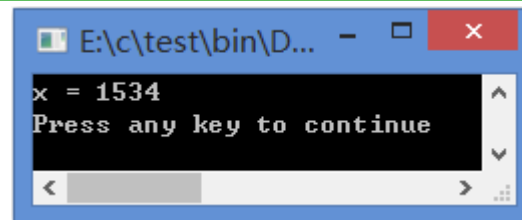
$d$ : 记录递推次数, 用于递推的循环控制

$x$ : 保存递推的结果, 即桃子的数量

# 用递推法求解问题的实例

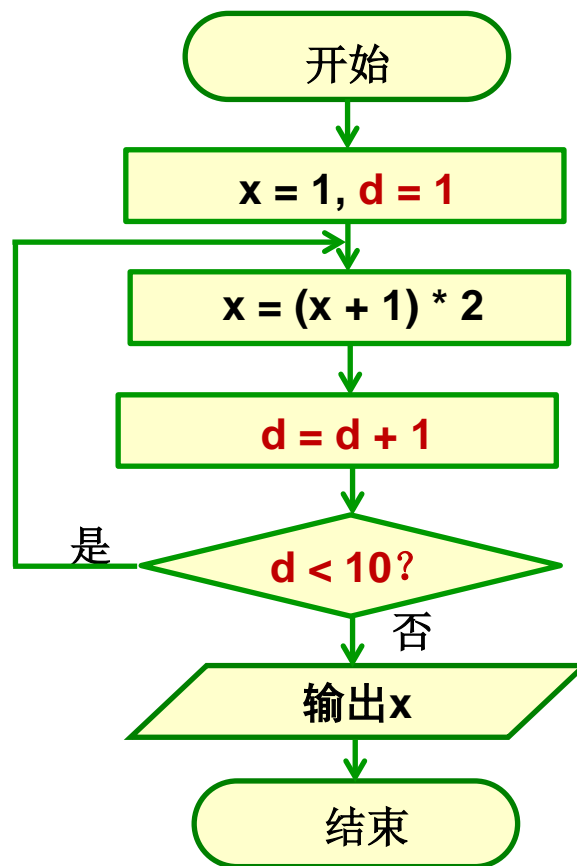


```
#include <stdio.h>
int main()
{
    int x = 1, d = 1;
    while (d < 10)
    {
        x = (x + 1) * 2;
        d++;
    }
    printf("x = %d\n", x);
    return 0;
}
```

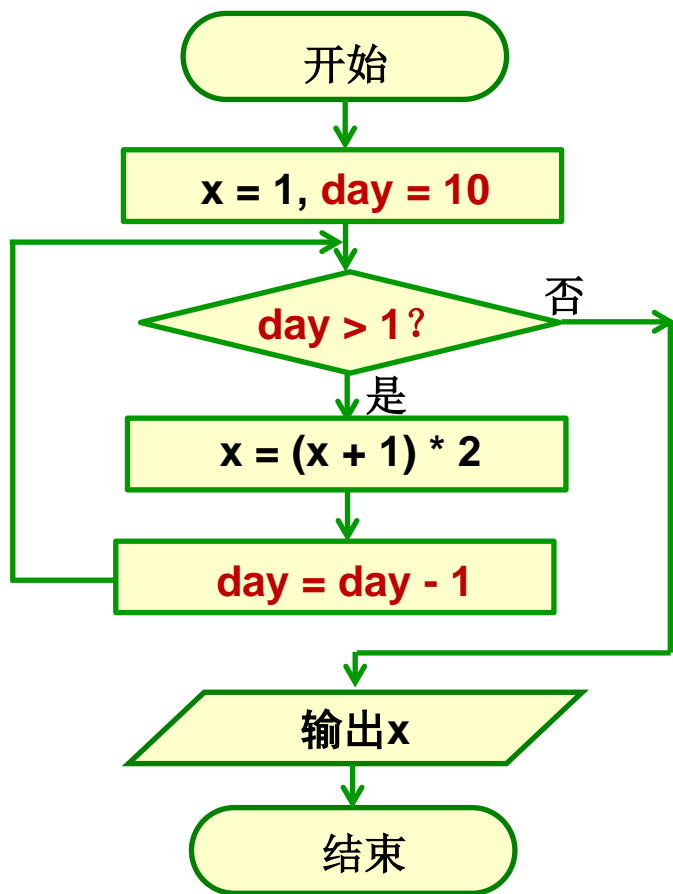


# 用递推法求解问题的实例

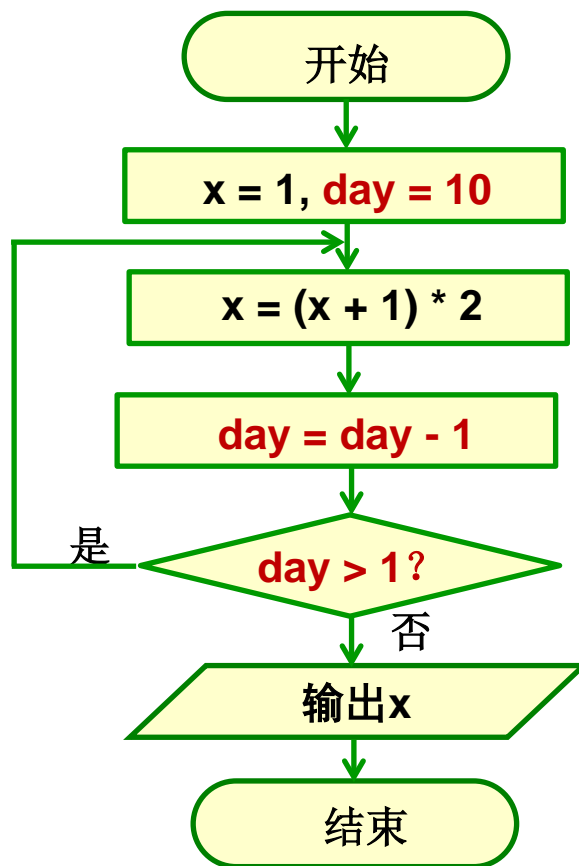
```
#include <stdio.h>
int main()
{
    int x = 1, d = 1;
    do{
        x = (x + 1) * 2;
        d++;
    }while (d < 10);
    printf("x = %d\n", x);
    return 0;
}
```



# 用递推法求解问题的实例



由第day天推出  
第day-1天

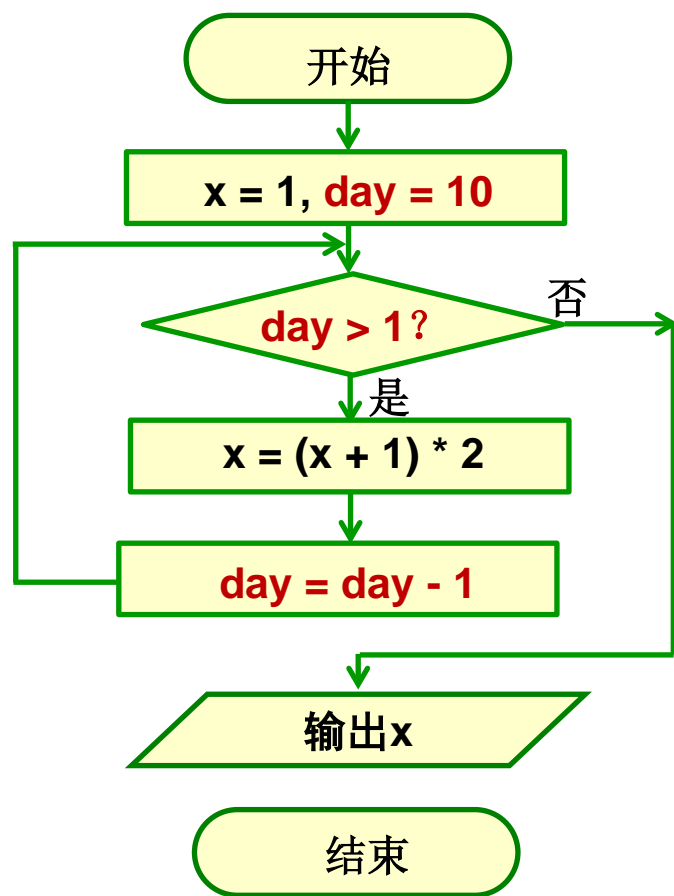


day: 表示是第几天, 用于递推的循环控制

x: 保存递推的结果, 即桃子的数量



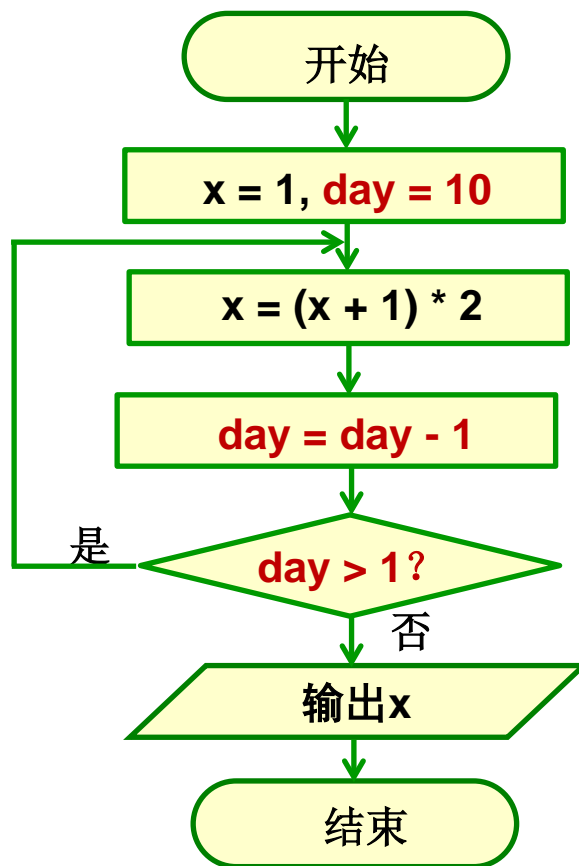
# 用递推法求解问题的实例



```
#include <stdio.h>
int main()
{
    int x = 1, day = 10;
    while (day > 1)
    {
        x = (x + 1) * 2;
        day--;
    }
    printf("x = %d\n", x);
    return 0;
}
```

# 用递推法求解问题的实例

```
#include <stdio.h>
int main()
{
    int x = 1, day = 10;
    do{
        x = (x + 1) * 2;
        day--;
    }while (day > 1);
    printf("x = %d\n", x);
    return 0;
}
```



# 选择循环语句的一般原则


- **Just a general rule.**

循环次数已知	for语句
循环次数未知，循环的次数由一个给定的条件来控制	while和do-while语句
循环体至少要执行一次	do-while语句

# 讨论

- 1) 用do-while语句，从键盘输入两个异号数（符号相反的数），然后打印出来
- 2) 修改猴子吃桃问题的程序，使其能够显示出从第10天到第1天的递推过程中计算得到的每一天的桃子数。

天数：      1            2            3            4            5            6            7            8            9            10

 1534   ← 766   ← 382   ← 190   ← 94   ← 46   ← 22   ← 10   ← 4   ← 

$$x_n = 1$$

$$n = 10$$

$$x_n = 2 \times (x_{n+1} + 1)$$

$$1 \leq n < 10$$

