



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

编程实现枚举算法策略

- ◆ 2、迭代算法
- ◆ 3、递推算法

3.5.7 循环结构程序举例

- ▶ 2. 迭代算法
- ▶ 迭代法是一种不断用变量的旧值递推新值的求解方法。

3.5.7 循环结构程序举例

- ▶ 采用迭代算法求解问题的基本思路为：
- ▶ （1）确定迭代变量。在可以用迭代算法解决的问题中，至少存在一个直接或间接地不断由旧值递推出新值的变量，这个变量就是迭代变量。
- ▶ （2）建立迭代关系式。所谓迭代关系式，指如何从变量的前一个值推出其下一个值的公式（或关系）。迭代关系式的建立是解决迭代问题的关键，通常可以使用递推或倒推的方法来完成。

3.5.7 循环结构程序举例

- ▶ (3) 对迭代过程进行控制。迭代过程的控制通常可分为两种情况：一种是所需的迭代次数是个确定的值，可以计算出来，使用计数型循环；另一种是所需的迭代次数无法确定，使用条件型循环。

3.5.7 循环结构程序举例



【例3.15】

求斐波那契(Fibonacci)数列前40个数。斐波那契数列公式为：

$$f(1) = 1 \quad (n = 1)$$

$$f(2) = 1 \quad (n = 2)$$

$$f(n) = f(n-1) + f(n-2) \quad (n > 2)$$

3.5.7 循环结构程序举例

例3.15

```
1 #include <stdio.h>
2 int main()
3 {
4     int i, f1=0, f2=1, fn; //迭代变量
5     for(i=1; i<=40; i++) { //迭代次数
6         fn = f1 + f2; //迭代关系式
7         f1 = f2, f2 = fn; //f1和f2迭代前进
8         printf("%d\n", f1);
9     }
10    return 0;
11 }
```

第6行将 $f1+f2$ （即 $f3$ ）保存到 fn 。第7行将 $f2$ 赋值给 $f1$ ，则此后 $f1$ 即为 $f2$ ，将 fn 赋值给 $f2$ ，则此后 $f2$ 即为 $f3$ ，以此类推，迭代前进

3.5.7 循环结构程序举例

- ▶ 3. 递推算法
- ▶ 递推法实际上是一种递推关系，就是为了得到问题的解，把它推到比原问题简单的问题求解，可分为顺推法和倒推法。
- ▶ （1）顺推法，就是先找到递推关系式，然后从初始条件出发，一步步地按递推关系式递推，直至求出最终结果
- ▶ （2）倒推法，就是在不知道初始条件的情况下，经某种递推关系而获知问题的解，再倒过来，推知它的初始条件。

3.5.7 循环结构程序举例



【循环程序举例】

8除不尽的自然数

一个自然数被8除余1，所得的商被8除也余1，再将第二次的商被8除后余7，最后得到一个商为 a ，又知这个自然数被17除余4，所得的商被17除余15，最后得到一个商是 a 的2倍，求这个自然数。

3.5.7 循环结构程序举例



例题分析

递推法求解的关键在于找出递推的初值、公式及次数。根据题意，设最后的商为*i*（*i*从0开始取值），用递推法可以列出关系式：

$$(((i*8+7)+1)*8+1) = ((i*2*17)+15)*17+4$$

再用试探法求出商*i*的值。

3.5.7 循环结构程序举例

例3.64

```
1 #include <stdio.h>
2 int main()
3 {
4     int a,x1,x2;
5     for(a=1;a<8;a++) { //试探商的值
6         x1=((a*8+7)*8+1)*8+1;
7         x2=(2*a*17+15)*17+4;
8         if(x1==x2)
9             printf("%d\n",x1);
10        else if(a==8)
11            printf("不存在这样的数! ");
12    }
13    return 0;
14 }
```

3.5.7 循环结构程序举例



【循环程序举例】

统计从1到n的正整数中1出现的次数。

给定一个十进制正整数n，写下从1开始到n的所有整数，然后数一下其中出现的所有“1”的个数。

例如：

N= 2，写下1， 2。出现了1个“1”。

N= 12，写下1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12。“1”的个数是5。

3.5.7 循环结构程序举例



例题分析

枚举法。

遍历 $1 \sim n$ ，统计每个数1出现的个数，相加便得到所有1的个数。

3.5.7 循环结构程序举例

例3.70

```
1  #include <stdio.h>
2  int main ()
3  {
4      int n,i,t,count;
5      scanf("%d",&n);
6      count = 0;
7      for(i = 1;i <= n;i++) {
8          t=i;
9          while(t) { //统计i含1的个数
10             count += (t % 10 == 1) ? 1:0;
11             t = t / 10;
12         }
13     }
14     printf("%d\n",count);
15     return 0;
```

3.5.7 循环结构程序举例



例题分析

递推法。

把问题变成：写一个函数 $f(N)$,返回1到 n 之间出现1的个数，比如 $f(12)=5$ 。

(1) n 是1位数情况

如果 $n=3$ ，那么从1到3的所有数字：1,2,3，只有个位数出现1，而且只出现一次。可以发现， n 是个位数时：

$n \geq 1$, 那么 $f(n)=1$;

$n=0$, $f(n)=0$;

3.5.7 循环结构程序举例



例题分析

(2) n 是2位数情况

如果 $n=13$ ，那么从1到13的所有数字：1,2,3,4,5,6,7,8,9,10,11,12,13，个位和十位数都可能出现1，可以分开考虑。个位出现1的2次：1,11，十位出现1的4次：10,11,12,13。所以 $f(n)=2+4=6$ 。

要注意的是11在十位和个位都出现1了，但是11恰好在个位和十位被计算了两次，因此不用作特殊处理。

再考虑 $n=23$ ，它和13不同，十位出现1的次数为10次，即11到19。个位出现1的数1,11,21。所以 $f(n)=3+10=13$ 。

3.5.7 循环结构程序举例



例题分析

(2) n 是2位数情况

个位数出现1的次数不仅和个位数字有关，还和十位数有关：如果 n 的个位数大于等于1，则个位数出现1的次数为十位数的数字加1；如果 n 的个位数为0，则个位数出现1的次数等于十位数的数字。

十位数上出现1的次数不仅和十位数有关，还和个位数有关。如果十位数等于1，则十位数上出现1的次数为个位数的数字加1；如果十位数大于1，则十位数上出现1的次数为10。

3.5.7 循环结构程序举例



例题分析

(2) n是2位数情况

$f(13) = \text{个位数出现1的个数} + \text{十位数出现1的个数} = 2 + 4 = 6$

$f(23) = \text{个位数出现1的个数} + \text{十位数出现1的个数} = 3 + 10 = 13$

$f(33) = \text{个位数出现1的个数} + \text{十位数出现1的个数} = 4 + 10 = 14$

.....

$f(93) = \text{个位数出现1的个数} + \text{十位数出现1的个数} = 10 + 10 = 20$

3.5.7 循环结构程序举例



例题分析

(3) n 是3位数情况

如果 $f(n)=123$

个位出现1个个数为13: 1,11,21,...,91,101,111,121

十位出现1个个数为20: 10~19,110~119

百位出现1个个数为24: 100~123

3.5.7 循环结构程序举例



例题分析

(3) n 是3位数情况

$$\begin{aligned} f(123) &= \text{个位数出现1的个数} + \text{十位数出现1的个数} + \text{百位数出现1的个数} \\ &= 13 + 20 + 24 = 57 \end{aligned}$$

同理分析4位数，5位数.....

3.5.7 循环结构程序举例



例题分析

(3) n 是任意位数

如果要计算百位上1出现的次数，它要受到三方面的影响：百位上的数字，百位以下（低位）上的数字，百位以上（高位）上的数字。

如果百位上数字为0，百位上可能出现1的次数由更高位决定。

比如：12013，则可以知道百位出现1的情况可能是：100~199，1100~1199, 2100~2199, ,, 11100~11199，一共1200个。可以看出是由更高位数字（12）决定，并且等于更高位数字（12）乘以当前位数（100）。

3.5.7 循环结构程序举例



例题分析

(3) n 是任意位数

如果百位上数字为1，百位上可能出现1的次数不仅受更高位影响还受低位影响。

比如：12113，则可以知道百位受高位影响出现的情况是：100~199，1100~1199, 2100~2199，，.....，11100~11199，一共1200个。和上面情况一样，并且等于更高位数字（12）乘以当前位数（100）。但同时它还受低位影响，百位出现1的情况是：12100~12113, 一共114个，等于低位数字（113）+1。

3.5.7 循环结构程序举例



例题分析

(3) n 是任意位数

如果百位上数字大于1 (2~9)，则百位上出现1的情况仅由更高位决定。

比如12213，则百位出现1的情况是：100~199, 1100~1199, 2100~2199,, 11100~11199, 12100~12199, 一共有1300个，并且等于更高位数字+1 (12+1) 乘以当前位数 (100)。

3.5.7 循环结构程序举例

假定 $n=abcde$ ，那么

```
//n=abcde 百位上数字是c，求百位上出现1的方法是
if(c==0) { //百位上数字为0，百位上可能出现1的次数由更高位决定
    //等于更高位数字(ab) * 当前位数(100)
    count += ab*100;
}
//百位上数字为1，百位上可能出现1的次数不仅受更高位影响还受低位影响
else if(c == 1){
    //更高位数字(ab) * 当前位数(100) + 低位数字(de) + 1
    count += ab*100 + de + 1;
}
else { //百位上数字大于1(2~9)，百位上出现1的情况仅由更高位决定
    //(更高位数字+1(ab+1)) * 当前位数(100)
    count += (ab + 1) * 100;
}
```


3.5.7 循环结构程序举例

例3.71

```
1  #include <stdio.h>
2  int main()
3  {
4      int n, count=0, f=1; //1的个数, 当前位,
5      int low=0, num=0, high=0; //低位数字, 当前位数字, 高位数字
6      scanf("%d", &n);
7      if (n>0) {
8          while(n/f!=0){
9              low = n-(n/f)*f; //低位数字
10             num = (n/f)%10; //当前位数字
11             high = n/(f*10); //高位数字
12             if(num == 0) { //如果为0, 出现1的次数由高位决定
13                 count += high*f; //等于高位数字 * 当前位数
14             }
15             //如果为1, 出现1的次数由高位和低位决定
```

3.5.7 循环结构程序举例

例3.71

```
16     else if(num == 1){
17         //高位数字 * 当前位数 + 低位数字 + 1
18         count += high*f+low+1;
19     }
20     else{ //如果大于1,出现1的次数由高位决定
21         // (高位数字+1) * 当前位数
22         count += (high+1)*f;
23     }
24     f *= 10; //上移一位
25 }
26 printf("%d\n",count);
27 }
28 return 0;
29 }
```

CP 程序设计