

二级C语言考试辅导教程第五章:函数[6] PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/134/2021_2022__E4_BA_8C_E7_BA_A7C_E8_AF_AD_c97_134590.htm

函数的递归调用 一个函数在它的函数体内调用它自身称为递归调用。这种函数称为递归函数。C语言允许函数的递归调用。在递归调用中，主调函数又是被调函数。执行递归函数将反复调用其自身。每调用一次就进入新的一层。例如有函数f如下：`int f(int x) { int y. z=f(y). return z. }`这个函数是一个递归函数。但是运行该函数将无休止地调用其自身，这当然是不正确的。为了防止递归调用无终止地进行，必须在函数内有终止递归调用的手段。常用的办法是加条件判断，满足某种条件后就不再作递归调用，然后逐层返回。下面举例说明递归调用的执行过程。

[例5.9]用递归法计算n!用递归法计算n!可用下述公式表示： $n! = 1 \quad (n=0,1) \quad n \times (n-1)! \quad (n>1)$ 按公式可编程如下：

```
long ff(int n) { long f. if(n else if(n==0||n==1) f=1. else f=ff(n-1)*n. return(f). }
main() { int n. long y. printf("\ninput a inteager number:\n").
scanf("%d",&n). y=ff(n). printf("%d!=%ld",n,y). }
long ff(int n) { ..... else f=ff(n-1)*n. .... }
main() { ..... y=ff(n). .... }
```

来源：www.examda.com 程序中给出的函数ff是一个递归函数。主函数调用ff后即进入函数ff执行，如果n回值为 $6 \times 4 = 24$ ，最后返回值ff(5)为 $24 \times 5 = 120$ 。例5.9也可以不用递归的方法来完成。如可以用递推法，即从1开始乘以2，再乘以3...直到n。递推法比递归法更容易理解和实现。但是有些问题则只能用递归算法才能实现。典型的问题是Hanoi塔问题。 [

例5.10]Hanoi塔问题 一块板上有三根针，A，B，C。A针上套

有64个大小不等的圆盘，大的在下，小的在上。如图5.4所示。要把这64个圆盘从A针移动到C针上，每次只能移动一个圆盘，移动可以借助B针进行。但在任何时候，任何针上的圆盘都必须保持大盘在下，小盘在上。求移动的步骤。本题算法分析如下，设A上有 n 个盘子。如果 $n=1$ ，则将圆盘从A直接移动到C。如果 $n=2$ ，则：1.将A上的 $n-1$ (等于1)个圆盘移到B上；2.再将A上的一个圆盘移到C上；3.最后将B上的 $n-1$ (等于1)个圆盘移到C上。如果 $n=3$ ，则：A. 将A上的 $n-1$ (等于2，令其为 n')个圆盘移到B(借助于C)，步骤如下：(1)将A上的 $n'-1$ (等于1)个圆盘移到C上，见图5.5(b)。(2)将A上的一个圆盘移到B，见图5.5(c)。(3)将C上的 $n'-1$ (等于1)个圆盘移到B，见图5.5(d) B. 将A上的一个圆盘移到C，见图5.5(e) C. 将B上的 $n-1$ (等于2，令其为 n')个圆盘移到C(借助A)，步骤如下：(1)将B上的 $n'-1$ (等于1)个圆盘移到A，见图5.5(f)。(2)将B上的一个盘子移到C，见图5.5(g)。(3)将A上的 $n'-1$ (等于1)个圆盘移到C，见图5.5(h)。到此，完成了三个圆盘的移动过程。

100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com