

二级C 函数：递归函数 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/133/2021_2022__E4_BA_8C_E7_BA_A7C___E5_c97_133526.htm

引言 在调用一个函数的过程中有出现直接或间接地调用该函数本身,称为函数的递归调用.对于初学者来说,递归函数是比较难以理解的.现在的程序设计的教材对于递归都比较简单,例子从一开始较难理解.加之不了解函数调用在计算机中的实现过程就更难理解递归调用了.这篇文章从初学者的角度对递归给予了一定的解释.从一个简单的数学题开始 有一个等差数列,首项是3,公差是2,求第5项 a_5 .我想这是所有高中生都会的题.再简单不过,只要知道通项公式 $a_n=a_1+(n-1)k$ (这里 k 是公差) 代入即可. 大家都知道通项是由递推公式推导来的,递推公式直接用数学语言反映了等差数列的性质, $a_n=a_{(n-1)}+k$.如果大家不知道通项公式,那么在这道题中,计算的方法是: $a_2=a_1+2$. $a_3=a_2+2$. $a_4=a_3+2$. $a_5=a_4+2$.知道 $a_5=11$,经过5次计算便可以得到 a_5 的值.如果计算机也不知道通项公式,在如果我们要计算第一百项,亦或是第一万项,我们难道要 $a_2=a_1+2$. $a_3=a_2+2$ $a_{10000}=a_{9999}+2$ 这样来计算吗.当然不是,我们可以用递归函数实现,而 $a_2=a_1+2$. $a_3=a_2+2$ $a_{10000}=a_{9999}+2$ 的过程便是递归的回推过程,可是计算机确实没有我们广大高中生聪明,它不知道直接这样求解,呵呵. 在计算机中的实现 要说清楚递归函数的实现就要了解程序里函数嵌套调用在计算机中的实现过程.我们来看下面一个函数的调用,这里我们用类似C语言的函数形式: $f() \{ \dots //其他代码 g(a). //调用函数g(),参数是a return \dots //函数的返回值 \}$

//----- main() //

主函数 { //其他代码 f(b). //调用函数f(),参数是b } 主函数在
执行的过程中调用了函数f(),而f()其自身又调用了另一个函
数g().那么这种嵌套调用在计算机中又是怎么实现的呢?在函
数的执行过程中,系统用到了栈内存.栈是一种数据结构,就是
有特定数据存取方式的容器.栈使用的存取方式称为后进先
出LIFO(late in first out),也就是后面放入的数据先离开栈,最先
放入的数据最后离开.大家可以把栈看成一个和盘子一样大小
放的水槽,放入其中的盘子只能从最上面依次取走,不下面的盘
子只能等它上面的取走之后才能取出.栈是一种在计算机中非
常重要的数据结构,它后进先出的性质也直接关系到函数嵌
套调用在计算机中的实现. 当一个函数的执行过程中需要调用
另一个函数时,需要保留现场,意思是系统保存现有的所有数
据,把他们放入栈内存,称为压入栈.之后就转到要调用的那个
函数处执行了,并把参数传给它.等到调用的那个函数执行完
了,系统就把刚才压入栈内存的数据都取出,称为弹出栈.并且
把他调用的函数的返回值给原函数.如果被调用的函数中还要
调用其他函数,系统也按照上述的方法保存和取回数据.栈的性
质能很好得适用于这一过程. 在来看递归函数,按引言里的意
思,在上述的过程中被调用的函数就是调用函数自己.其实现的
过程是一样的.是不是很简单? 再看一些例子 文章开始等差数
列的例子写成函数就是(方便起见,值都设为整数): int getan(int
n, int a1) // 要求第n项,a1为已知的首项 { if(n==1) return a1. //如
果n=1,返回a1的值,这是递归得以结束的关键 else return
getan(n-1,a1) 2. //不等于1,则再次调用自身求出第n-1项,返回结
果加上公差2 }
//----- main() //主

函数 { int a1=3. //设首项 int a5=getan(5,3). //递归求解第5项a5
printf(d%,a5). //输出 } 递归函数调用自身是一个不断循环过程,
就像循环语句需要有条件让它停止循环,否则就陷入了无限循
环之中,那后果.....天啊!所以if(n=1) return a1.就是停止递归的
条件. 我们来仔细地看一下a5是如何求出的,不要嫌罗嗦,来: 首
先当主函数执行到int a5=getan(5,3). 开始求 a5的值. n=5>1 所
以gatan()函数又调用自身,并把参数5-1=4和3传给下一个函数,
所以执行getan(4,3). n=4>1 gatan()函数再次调用自身,并把参
数3和3传给下一个函数,所以执行getan(3,3). n=3>1 执
行getan(2,3). n=2>1 执行getan(1,3). 终于,n=1了,现在getan(1,3)
的返回值是3. getan(2,3). 返回值是3 2=5. getan(3,3).返回7.
getan(4,3).返回9. 最后getan(5,3).返回11,变量a5=11,结果也就出
来了. 从理论上我们把函数反复调用自身的过程称为递推的过程,
而每个函数的返回称之为回推.递推的过程把复杂的问题一
步步分解开,直到出现最简单的情况,就像已知a1=3.得到了最
简单情况的解,就可以回推回去得出原来复杂问题的答案了.要
说明的一点是,在这里我们认为函数的使用者总是正确的,不会
把诸如-1,0,5.2等之类不合理项数作为参数输入,所以没有做出
错检查. 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