

计算机等级考试二级C语言常用的算法 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/135/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E7_c97_135295.htm

什么是程序？程序=数据结构 算法。对于面向对象程序设计，强调的是数据结构，而对于面向过程的程序设计语言如C、Pascal、FORTRAN等语言，主要关注的是算法。掌握算法，也是为面向对象程序设计打下一个扎实的基础。那么，什么是算法呢？人们使用计算机，就是要利用计算机处理各种不同的问题，而要做到这一点，人们就必须事先对各类问题进行分析，确定解决问题的具体方法和步骤，再编制好一组让计算机执行的指令即程序，交给计算机，让计算机按人们指定的步骤有效地工作。这些具体的方法和步骤，其实就是解决一个问题的算法。根据算法，依据某种规则编写计算机执行的命令序列，就是编制程序，而书写时所应遵守的规则，即为某种语言的语法。由此可见，程序设计的关键之一，是解题的方法与步骤，是算法。学习高级语言的重点，就是掌握分析问题、解决问题的方法，就是锻炼分析、分解，最终归纳整理出算法的能力。与之相对应，具体语言，如C语言的语法是工具，是算法的一个具体实现。所以在高级语言的学习中，一方面应熟练掌握该语言的语法，因为它是算法实现的基础，另一方面必须认识到算法的重要性，加强思维训练，以写出高质量的程序。下面通过例子来介绍如何设计一个算法：[例1-4] 输入三个数，然后输出其中最大的数。首先，得先有个地方装这三个数，我们定义三个变量A、B、C，将三个数依次输入到A、B、C中，另外，再准备一个MAX装最大数

。由于计算机一次只能比较两个数，我们首先把A与B比，大的数放入MAX中，再把MAX与C比，又把大的数放入MAX中。最后，把MAX输出，此时MAX中装的就是A、B、C三数中最大的一个数。算法可以表示如下：1) 输入A、B、C。2) A与B中大的一个放入MAX中。3) 把C与MAX中大的一个放入MAX中。4) 输出MAX，MAX即为最大数。其中的2)、3)两步仍不明确，无法直接转化为程序语句，可以继续细化：2) 把A与B中大的一个放入MAX中，若 $A > B$ ，则MAX = A；否则MAX = B。3) 把C与MAX中大的一个放入MAX中，若 $C > MAX$ ，则MAX = C。于是算法最后可以写成：1) 输入A, B, C。2) 若 $A > B$ ，则MAX = A；否则MAX = B。3) 若 $C > MAX$ ，则MAX = C。4) 输出MAX，MAX即为最大数。这样的算法已经可以很方便地转化为相应的程序语句了。

[例1-5] 猴子吃桃问题：有一堆桃子不知数目，猴子第一天吃掉一半，觉得不过瘾，又多吃了一只，第二天照此办理，吃掉剩下桃子的一半另加一个，天天如此，到第十天早上，猴子发现只剩一只桃子了，问这堆桃子原来有多少个？此题粗看起来有些无从着手的感觉，那么怎样开始呢？假设第一天开始时有 a_1 只桃子，第二天有 a_2 只，...，第9天有 a_9 只，第10天是 a_{10} 只，在 a_1, a_2, \dots, a_{10} 中，只有 $a_{10} = 1$ 是知道的，现要求 a_1 ，而我们可以看出， a_1, a_2, \dots, a_{10} 之间存在一个简单的关系： $a_9 = 2 * (a_{10} - 1)$ $a_8 = 2 * (a_9 - 1)$ $a_1 = 2 * (a_2 - 1)$ 也就是： $a_i = 2 * (a_{i+1} - 1)$ $i=9,8,7,6,\dots,1$ 这就是此题的数学模型。再考察上面从 a_9, a_8 直至 a_1 的计算过程，这其实是一个递推过程，这种递推的方法在计算机解题中经常用到。另一方面，这九步运算从形式上完全一样，不同的只是 a_i 的下标而已。由

此，我们引入循环的处理方法，并统一用 a_0 表示前一天的桃子数， a_1 表示后一天的桃子数，将算法改写如下：1) $a_1=1$ 。{ 第10天的桃子数， a_1 的初值 } $i=9$ 。{ 计数器初值为9 } 2) $a_0=2 * (a_1 - 1)$ 。{ 计算当天的桃子数 } 3) $a_1= a_0$ 。{ 将当天的桃子数作为下一次计算的初值 } 4) $i=i-1$ 。5) 若 $i >= 1$ ，转2)。6) 输出 a_0 的值。其中2) ~ 5)步为循环。这就是一个从具体到抽象的过程，具体方法是：1) 弄清如果由人来做，应该采取哪些步骤。2) 对这些步骤进行归纳整理，抽象出数学模型。3) 对其中的重复步骤，通过使用相同变量等方式求得形式的统一，然后简练地用循环解决。算法的描述方法有自然语言描述、伪代码、流程图、N-S图、PAD图等。

1.4.1 流程图与算法的结构化描述

1. 流程图

流程图是一种传统的算法表示法，它利用几何图形的框来代表各种不同性质的操作，用流程线来指示算法的执行方向。由于它简单直观，所以应用广泛，特别是在早期语言阶段，只有通过流程图才能简明地表述算法，流程图成为程序员们交流的重要手段，直到结构化的程序设计语言出现，对流程图的依赖才有所降低。下面介绍常见的流程图符号及流程图的例子。本章例1-1的算法的流程图如图1-2所示。本章例1-2的算法的流程图如图1-3所示。在流程图中，判断框左边的流程线表示判断条件为真时的流程，右边的流程线表示条件为假时的流程，有时就在其左、右流程线的上方分别标注“真”、“假”或“T”、“F”或“Y”、“N”。另外还规定，流程线是从下往上或从右向左时，必须带箭头，除此以外，都不画箭头，流程线的走向总是从上向下或从左向右。

2. 算法的结构化描述

早期的非结构化语言中都有goto语句，它允许程序从一个地方直接跳

转到另一个地方去。执行这样做的好处是程序设计十分方便灵活，减少了人工复杂度，但其缺点也是十分突出的，一大堆跳转语句使得程序的流程十分复杂紊乱，难以看懂也难以验证程序的正确性，如果有错，排起错来更是十分困难。这种转来转去的流程图所表达的混乱与复杂，正是软件危机中程序人员处境的一个生动写照。而结构化程序设计，就是要把这团乱麻理清。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