```
#include"stdio.h"
main( )
{
   long fact(int n);
   int n;
   printf("\nplease enter n: ");
   scanf("%d",&n);
   printf("\nn!=%ld",fact(n));
}
long fact(int n)
{
   if (n<=1) return(1);
   else return(n*fact(n-1));
}
```

高等学校应用型特色规划教材

C语言程序设计

C Programming Language

主　编 ◎ 张仁忠　曾昭江
副主编 ◎ 马莉莉　廖慎勤　杨　昊

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等学校应用型特色规划教材

C 语言程序设计

张仁忠　曾昭江　主编

马莉莉　廖慎勤　杨　昊　副主编

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

C 语言是一门基础性的程序设计语言，学习 C 语言有助于计算机专业的学生更好地学习其他程序设计语言。本书的主要内容包括：C语言程序设计基础、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、指针、结构体、文件、位运算等内容。书中每章都提供了丰富的案例和精心挑选的课后练习题，重点章节还提供了相关实训项目。本书体系结构完整，内容介绍深入浅出，注重理论与实践相结合，每个案例都经过精心调试并配有源代码和运行结果，方便读者学习。

本书既可作为高等院校“C 语言程序设计”课程的教材，又可作为广大计算机程序设计人员和计算机程序设计爱好者的参考书，同时可供参加相关考试的读者参考。

图书在版编目（CIP）数据

C 语言程序设计/张仁忠，曾昭江主编. —北京：电子工业出版社，2018.8

ISBN 978-7-121-34592-0

Ⅰ. ①C… Ⅱ. ①张… ②曾… Ⅲ. ①C 语言－程序设计－高等学校－教材 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2018）第 137273 号

策划编辑：章海涛
责任编辑：章海涛　　文字编辑：刘　瑀
印　　刷：三河市华成印务有限公司
装　　订：三河市华成印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×1 092　1/16　　印张：13.25　字数：318 千字
版　　次：2018 年 8 月第 1 版
印　　次：2018 年 8 月第 1 次印刷
定　　价：38.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：liuy01@phei.com.cn。

前　言

自诞生以来，C 语言以其灵活和实用的特点得到了广大用户的喜爱，迅速发展成一门应用广泛的编程语言。从网站后台到底层操作系统，从多媒体应用到大型网络游戏，均可使用 C 语言开发。在工业领域，C 语言也是首选的系统语言，特别是在图形处理和工业控制方面，其使用更为广泛。此外，C 语言是一门结构化程序设计语言，有利于学生掌握程序设计的思想，培养认真、严谨的编程态度。因此，C 语言已成为学习程序设计的一门基础性语言。

本书是作者在多年教学的基础上，融入职业教育的特点而编写的。不仅注重概念理解，力求使学生建立起对程序设计和 C 语言的清晰认识，更注重学以致用，使学生在较短的时间内初步学会使用 C 语言编写程序，掌握相关的知识和技能。本书遵循“提出问题—解决问题—进一步提出问题—再解决问题”的循序渐进的讲解过程，使学生养成由简到繁、逐步求精的编程习惯。

本书共分为 10 章，第 1 章介绍程序设计的基础知识、C 语言的发展、特点、开发环境等；第 2 章介绍 C 语言中常用的数据类型、运算符与表达式、输入/输出函数和顺序结构程序设计等；第 3 章介绍选择结构程序设计，包括 if 语句、switch 语句等；第 4 章介绍循环结构程序设计，包括 while 循环、for 循环等；第 5 章介绍一维数组、二维数组和字符数组的定义、引用和初始化等；第 6 章介绍函数的概念、声明、调用，以及变量的作用域等；第 7 章介绍指针的概念、指针变量的定义、指针与数组、指针与函数等；第 8 章介绍结构体类型、结构体变量、结构体数组等；第 9 章介绍文件的概念、文件的打开与关闭、文件的读写操作等；第 10 章介绍位运算符并进行案例实践。带星号*的章节，读者在学习过程中可以根据需要进行取舍。

本书由广东创新科技职业学院张仁忠、曾昭江担任主编。第 1 章由张仁忠、曾昭江编写；第 2、3、4 章由廖慎勤编写；第 5、6、8 章由马莉莉编写；第 7 章由曾昭江编写；第 9 章、附录由曾昭江、杨昊编写；第 10 章由马莉莉、张仁忠编写；实训部分内容由对应章节的作者编写。全书由张仁忠教授统稿。在本书的编写过程中，广州市靖凯科技有限公司杨昊总监提供了部分精选案例。C 语言程序设计课程的任课教师巫思敏、雷少玲、汤怀、李杏清等，为本书提出了宝贵的意见和建议。在此一并表示感谢。

本书提供电子课件、源代码等配套资源，读者可登录华信教育资源网（www.hxedu.com.cn）注册并免费下载。

尽管我们做出了种种努力，付出了许多辛勤劳动，但由于水平有限、时间仓促，书中难免有错误之处，恳请广大读者批评指正。

编　者

目　录

第1章　C语言程序设计基础

本章主要内容

- 程序与程序设计语言
- 算法及其描述
- C语言的发展及其特点
- C语言的基本结构
- C语言程序的开发环境

1.1　程序与程序设计语言

一个完整的计算机系统包括硬件系统和软件系统，硬件是物质基础，软件是计算机的灵魂。没有软件的计算机是一台“裸机”，什么操作都无法进行。有了软件，计算机才有了生命，成为一台真正的“电脑”。软件都是用计算机语言编写的，是包含程序的有机集合体，程序是软件的必要元素。软件可以用以下公式来表示：

软件=程序+文档=（数据结构+算法）+文档

任何软件都有可运行的程序。例如，操作系统提供的工具软件，很多都只有一个可运行的程序，而有些也包含多个可运行的程序，如Office办公软件包中包含了很多可运行的程序。软件是程序与开发、使用和维护它的文档的总称，程序是软件的一部分。

1.1.1　程序

程序是人们为了解决某种问题用计算机可以识别的代码编排的一系列加工步骤。计算机程序是软件开发人员根据用户需求开发的、用程序设计语言描述的、适合计算机执行的指令序列。计算机本身不会做任何工作，它按照程序中的有序指令完成相应的任务。

由于计算机不能理解人类的自然语言，所以不能用自然语言编写计算机程序，只能用专门的程序设计语言来编写。人们借助计算机能够处理的语言，告诉计算机要处理哪些数据，以及按什么步骤来处理，这便是程序设计。

为解决某一问题而编写的程序不是唯一的，不同的用户编写程序的思路也不会完全一样，因此，不同程序的执行效率不同，这涉及程序的优化、程序所采用的数据结构和算法等多方面的因素。

1.1.2 程序设计语言

自 1946 年世界上第一台电子计算机问世以来，计算机科学的发展十分迅猛，计算机被广泛地应用于人们生产、生活的各个领域，推动了人类社会的进步与发展。特别是互联网（Internet）的发展，使传统的信息收集、传输及交换方式正在被革命性地改变，人们已经难以摆脱对计算机的依赖，计算机已将人类带入一个新的时代——信息时代。掌握计算机的基本知识和基本技能已经成为人们应该具备的基本素质，缺乏计算机知识，就是信息时代的“文盲”。

对于理工科学生而言，掌握一门高级语言及基本的编程技能是必需的。学习计算机语言，是因为它是与计算机进行交互的有力工具，同时也能够培养认真、严谨的做事习惯。

计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

机器语言是第一代计算机语言，属于低级语言的范畴。机器语言是由 0 和 1 这两个二进制代码组成的，是计算机能直接识别和执行的一种机器指令的集合。由于机器语言使用的是针对特定型号计算机的语言，因此其运算效率是所有语言中最高的。机器语言具有直接执行和速度快等特点。

但是机器语言的学习和使用复杂、烦琐、费时、易出差错，特别是在程序出错时，更是如此。由于每台计算机的指令系统往往各不相同，所以，在一台计算机上执行的程序，要想在另一台计算机上执行，必须重新编写，造成了重复工作。

2. 汇编语言

汇编语言是面向机器的程序设计语言。在汇编语言中，用助记符代替操作码，用地址符号（Symbol）或标号（Label）代替地址码。例如，用 ADD 代表加法，MOV 代表数据传递。这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得比较方便。

使用汇编语言编写的程序，机器不能直接识别，要由一段程序将汇编语言翻译成机器语言，这种起到翻译作用的程序称为汇编程序。汇编程序是语言处理系统软件，将汇编语言翻译成机器语言的过程称为汇编。

汇编语言同样十分依赖于机器硬件，移植性不好，但效率仍然很高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精练、质量高，至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

由于汇编语言依赖于硬件体系，且助记符量大又难记，于是人们又发明了更加易用的高级语言。这种语言的语法和结构更类似普通英文，表示方法要比低级语言更接近于待解问题，其特点是在一定程度上与具体机器无关，易学、易用、易维护。

1954 年，第一种完全脱离机器硬件的高级语言 FORTRAN 问世了，60 多年来，共有几百种高级语言出现，有重要意义的有几十种，影响较大、使用较普遍的有 FORTRAN、

ALGOL、COBOL、BASIC、LISP、SNOBOL、Pascal、C、C++、VB、Dephi、Java 等。

从早期语言到结构化程序设计语言，从面向过程的程序设计语言到非过程化的程序设计语言，高级语言的发展经历了一个漫长的进化过程。相应地，软件的开发也由最初个体手工作坊式的封闭式生产，发展为产业化、流水线式的工业化生产。

20 世纪 60 年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是各自为战，缺乏科学规范的系统规划、测试和评估标准，其结果是大量耗费巨资建立起来的软件系统，由于存在错误而无法使用，甚至带来巨大损失，软件给人的感觉是越来越不可靠，以至于几乎没有不出错的软件。这一切，极大地影响了计算机界，史称“软件危机”。人们认识到，大型程序的编制不同于小型程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性，也便于验证正确性。因此，人们提出了结构化程序设计方法，第一个结构化程序设计语言——Pascal 语言的出现，标志着结构化程序设计时期的开端。

20 世纪 80 年代初，在软件设计思想上，又产生了一次革命，其成果就是面向对象的程序设计。在此之前的高级语言，几乎都是面向过程的，程序的执行是流水线式的，在一个模块被执行完成前，不能执行其他操作，也无法动态地改变程序的执行方向，这和人们日常处理事物的方式是不一致的。我们希望发生一件事就处理一件事，也就是说，不能面向过程，而应是面向具体的应用功能，即对象（Object）。其方法就是软件的集成化，如同硬件的集成电路一样，生产一些通用的、封装紧密的功能模块，称为软件集成块，它与具体应用无关，但能相互组合，完成具体的应用功能，又能被重复使用。对使用者来说，只需关心它的接口（输入、输出）及它能实现的功能，至于它是如何实现的，那是它内部的事，使用者无须关心。C++、Java 就是面向对象程序设计语言的典型代表。

高级语言的下一个发展目标是面向应用，也就是说，只需要告诉程序要干什么，程序就能自动生成算法，自动处理，也就是非过程化的程序语言。

1.2　算法及其描述

一个计算机程序应该包括以下两方面的内容。

（1）对数据的描述，在程序中指定数据的类型和数据的组织形式，即数据结构（Data Structure）。

（2）对操作的描述，也就是算法（Algorithm）。

著名计算机学家沃斯提出了一个公式：数据结构+算法=程序。实际上，一个程序除以上两个主要的要素外，还应当采用程序设计方法进行设计，并且用一种计算机语言来表示。因此，算法、数据结构、程序设计方法和计算机语言这 4 个方面是一名程序员所应具备的基本知识。可见，算法在计算机科学界与计算机应用界的重要地位。

1.2.1　算法的概念

算法就是为了解决一个具体问题而采取的方法和有限步骤，或者是指对解题方法准确

而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略。如果一个算法有缺陷，或不适用于某个问题，执行这个算法将不会解决这个问题。

一个算法应该具有以下7个重要的特征。

（1）有穷性（Finiteness）：算法必须能在执行有限个步骤之后终止。

（2）确切性（Definiteness）：算法的每个步骤必须有确切的定义。

（3）输入项（Input）：一个算法有零个或多个输入，以表示运算对象的初始情况，所谓零个输入是指算法本身给出了初始条件。

（4）输出项（Output）：一个算法有一个或多个输出，以反映对输入数据加工后的结果，没有输出的算法是毫无意义的。

（5）可行性（Effectiveness）：算法中执行的任何计算步骤都可以被分解为基本的可执行操作，即每个计算步骤都可以在有限时间内完成（也称有效性）。

（6）高效性（High Efficiency）：执行速度快，占用资源少。

（7）健壮性（Robustness）：对数据响应正确。

一个算法质量的优劣将影响算法乃至整个程序的效率，不同的算法可能会以不同的时间复杂度、空间复杂度或效率完成同样的任务，算法分析的目的在于选择合适的算法并进行改进。对一个算法的评价主要从时间复杂度和空间复杂度两方面来考虑。

1. 时间复杂度

算法的时间复杂度是指执行算法所需要的时间。一般来说，计算机算法是问题规模 n 的函数 $f(n)$，算法的时间复杂度因此可记为：

$$T(n)=O(f(n))$$

由此可知，算法执行时间的增长率与 $f(n)$ 的增长率正相关，称为渐进时间复杂度（Asymptotic Time Complexity）。

2. 空间复杂度

算法的空间复杂度是指算法需要消耗的内存空间。其计算和表示方法与时间复杂度类似，一般都是用复杂度的渐进性来表示。同时间复杂度相比，空间复杂度的分析要简单得多。

1.2.2 算法的描述

算法可以使用自然语言、伪代码、流程图等多种不同的方法来描述，它们的优势和不足可以简单地归纳如下。

1. 自然语言

优势：通俗易懂，不用专门训练。

不足：自然语言的歧义性容易导致算法执行的不确定性；自然语言的语句一般较长，导致描述的算法太长；当一个算法中循环和分支较多时，自然语言就很难将其清晰地表示

出来；自然语言表示的算法不便翻译成计算机语言。

2. 伪代码

伪代码方式用介于自然语言与计算机语言之间的文字及符号来描述算法。

优势：回避了程序设计语言严格、烦琐的书写格式，书写方便；同时具备格式紧凑、易于理解、便于向计算机语言过渡的优点。

不足：伪代码的种类繁多，语句不容易规范，有时会产生误读。

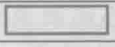
【例 1.1】 用伪代码描述“输出 x 的绝对值”的算法。

```
若 x 为正
    输出 x
若 x 为负
    输出-x
```

3. 流程图

流程图是一种描述算法控制流程和指令执行情况的有向图，是一种比较直观的描述方式。几种常用的流程图符号如表 1-1 所示。

表 1-1　流程图中的常用符号

图形符号	符号名称	说　明
	起止框	表示算法的开始或结束
	处理框	表示算法的某个处理步骤
	判断框	表示对给定条件进行判断，根据条件是否成立来决定如何执行
	输入/输出框	表示输入/输出操作
	流线	表示程序的流向
	连接圈	表示算法流向出口和入口连接点

优势：清晰简洁，容易表达选择结构，不依赖于任何具体的计算机语言，有利于不同环境下的程序设计。

不足：不易书写，不易修改，可借助专用的流程图制作软件进行绘制和修改。

【例 1.2】 用流程图描述以下算法：从键盘输入圆的半径 r，输出圆的周长 l 和面积 s。

算法步骤如下：

```
输入半径 r
计算圆的周长 cl=2*PI*r
计算圆的面积 cs=PI*r^2
输出结果
```

说明：该算法中，计算面积所需的初始数据半径 r 待定，要求在程序运行后从键盘输入。算法流程图如图 1-1 所示。

1.2.3 常用算法举例

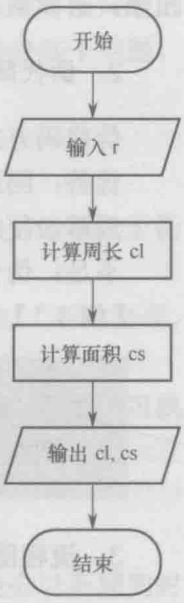


图 1-1 流程图

1. 穷举法

穷举法又称枚举法，是一种简单而直接的解决问题的方法。其基本思想是，逐一列举问题所涉及的所有情形，并根据问题提出的条件检验哪些是问题的解，哪些应该排除。

2. 递归法

递归是设计和描述算法的一种有力的工具，在复杂算法的描述中经常被采用。能采用递归描述的算法通常有这样的特征：为求解规模为 N 的问题，设法将它分解成规模较小的问题，然后从这些小问题的解中方便地构造出大问题的解，并且这些规模较小的问题也能采用同样方法，分解成规模更小的问题。特别地，当规模 N=1 时，能直接得到解。

递归法是一种效率比较低的算法，但是其优点也很明显，它能够大幅降低程序设计的复杂性，非常容易理解。但对时间复杂度要求较高的程序不适合用递归法。

3. 回溯法

回溯法是一种选优搜索法，按选优条件向前搜索，以达到目标。当探索到某一步时，发现原来的选择不满足条件或达不到目标，就退回一步重新选择，这种走不通就退回再重新走的方法称为回溯法，而满足回溯条件的某个状态的点称为回溯点。

4. 贪心法

贪心法（又称贪婪算法）是指，在求解问题时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，所求出的仅是某种意义上的局部最优解。贪心法不能对所有问题都得到整体最优解，但对大部分问题能得到整体最优解或整体最优解的近似解。

5. 分治法

在计算机科学中，分治法是一种很重要的算法。分治法字面上的解释是“分而治之”，就是把一个复杂的问题分成两个或更多个相同或相似的子问题，再把子问题分成更小的子问题，直到最后的子问题可以简单地直接求解，原问题的解即子问题的解的合并。这个技巧是很多高效算法的基础，如排序算法（快速排序、归并排序）、傅里叶变换（快速傅里叶变换）等。

1.3 C 语言的发展及其特点

1.3.1 C 语言的发展历史

C 语言的发展颇为有趣，它的原型是 1960 年出现的 ALGOL 60 语言（也称 A 语言）。

和汇编语言相比，A 语言的可读性、可移植性较好，但离硬件远，不适合编写系统程序。1963 年，剑桥大学将 ALGOL 60 语言发展成 CPL（Combined Programming Language）语言。CPL 更接近硬件一些，但规模较大。1967 年，剑桥大学的 Matin Richards 对 CPL 语言进行了简化，于是产生了 BCPL 语言。1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改，并为它起了一个有趣的名字——B 语言。意思是将 CPL“煮干”，提炼出它的精华，并且他用 B 语言编写了第一个 UNIX 操作系统。但是，B 语言过于简单，功能有限，并且没有数据类型。而在 1972 年，B 语言也被人“煮”了一下，美国贝尔实验室的 Dennis M.Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。C 语言非常精练，接近硬件，又克服了没有数据类型的缺点。

为了推广 UNIX 操作系统，1977 年，Dennis M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年，Brian W.Kernighian 和 Dennis M.Ritchie 出版了著名的 *The C Programming Language*，从而使 C 语言成为目前世界上最流行的高级程序设计语言。1988 年，随着微型计算机的日益普及，出现了许多 C 语言版本。由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准学会（ANSI）为 C 语言制定了一套 ANSI 标准，成为了现行的 C 语言标准。

C 语言发展迅速，而且成为最受欢迎的语言之一，主要因为它具有强大的功能。许多著名的系统软件，如 DBASE Ⅲ PLUS、DBASE Ⅳ都是由 C 语言编写的。若用 C 语言加上一些汇编语言子程序，就更能显示出 C 语言的优势了，例如，PC-DOS、WORDSTAR 等就是用这种方法编写的。

1.3.2　C 语言的特点

一门语言之所以能存在和发展并具有生命力，总是有其不同于其他语言的特点。C 语言的主要特点如下。

（1）C 语言简洁、紧凑，使用方便、灵活。C 语言一共只有 32 个保留字、9 种控制语句，程序书写形式自由，主要用小写字母表示，压缩了一切不必要的成分，相对其他计算机语言而言，源程序较短，因此输入程序时工作量较少。

（2）C 语言既具有高级语言的特点，又具有低级语言的一些功能。它允许直接访问地址，能进行位（bit）运算，可以直接对硬件进行操作。

（3）C 语言是一种结构化程序设计语言，它具有结构化控制语句（if-else、while、do-while、switch、for 等语句）。C 语言用函数作为程序模块，以实现程序的模块化。因此，C 语言十分有利于实现结构化、模块化的程序设计。

（4）C 语言的运算符丰富。C 语言运算符包含的范围很广，共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理，从而使它的运算符类型极其丰富，表达式类型多样化。灵活使用各种 C 语言的运算符可以实现在其他高级语言中难以实现的运算。

（5）C 语言的数据类型丰富，具有现代化语言的各种数据类型。C 语言的数据类型有：整型、实型、字符型、数组型、指针型、结构型、联合型和枚举型等。它们能用来实现各种复杂的数据结构。因此，C 语言具有很强的数据处理能力。

（6）C 语言程序中可以使用如#define、#include 等编译预处理命令，能进行字符串或特定参数的宏定义，以及实现对外部文本文件的读取和合并，同时还具有#if、#else 等条件编译预处理功能。这些功能有利于提高程序质量和软件开发的效率。

（7）C 语言生成的代码质量高。高级语言能否用来描述系统软件，特别是像操作系统、编译程序等，除取决于语言表达能力以外，还有一个很重要的因素，就是该语言的代码质量。实验表明，C 语言代码效率只比汇编语言低 10%～20%，C 语言是描述系统软件和应用软件比较理想的工具。

（8）C语言程序的可移植性好。C语言程序本身不依赖于机器硬件系统，从而便于在硬件结构不同的机种和操作系统之间实现程序的移植。

1.4 C 语言的基本结构

程序设计方法对程序设计的质量有非常重要的影响。程序设计方法主要有结构化程序设计方法和面向对象的程序设计方法。

1.4.1 结构化程序设计

C 语言程序是一种结构化程序设计语言，它提供了实现 3 种基本结构的语句，只要确定了算法，就可以很容易实现结构化程序的编写。

结构化程序设计的思路是，自顶向下、逐步求精；其程序结构是，按功能划分为若干个基本模块，各模块之间的关系尽可能简单，在功能上相对独立；每模块内部均由顺序、选择和循环 3 种基本结构组成；其模块化实现的具体方法是，使用子程序。结构化程序设计由于采用了模块分解与功能抽象、自顶向下、分而治之的方法，从而有效地将一个较复杂的系统程序设计任务分解成许多易于控制和处理的子任务，便于开发和维护。

虽然结构化程序设计方法具有很多优点，但它仍是一种面向过程的程序设计方法，它把数据和处理数据的过程分离为相互独立的实体。当数据结构改变时，所有相关的处理过程都要进行相应的修改，每种相对于老问题的新方法都要带来额外的开销，程序的重用性差。由于图形用户界面的应用，程序运行由顺序运行演变为事件驱动，使软件使用起来越来越方便，但开发起来却越来越困难，这种软件的功能，很难用过程来描述和实现，用面向过程的方法来开发和维护都将非常困难。因此，产生了面向对象的程序设计方法。

1.4.2 一个 C 语言程序的结构

下面来看一个用 C 语言编写的 C 程序，并从这个例子中说明 C 程序的基本结构。

```
#include <stdio.h>
#include <stdlib.h>                    //预编译：包含头文件
int main( ) {                          //主函数名，是程序的执行入口
    int a,b,sum;                       //定义了三个整型变量 a、b、sum
```