HZ BOOKS
华章教育

高等院校精品课程系列教材·省级

C语言程序设计教程

第2版

朱鸣华 刘旭麟 杨 微 ◎等编著

精品课程

C Programming

Second Edition

机械工业出版社
China Machine Press

高等院校精品课程系列教材·省级

C语言程序设计教程

第2版

朱鸣华 刘旭麟 杨 微
李 慧 孙大为 罗晓芳 赵 晶 ◎编著

C Programming
Second Edition

机械工业出版社
China Machine Press

本书介绍利用C语言进行程序设计的基本知识。全书共13章，详细介绍了C语言的基本概念、算法与程序设计基础、数据的输入和输出、选择结构、循环结构、数组、函数、指针、结构体与共用体、文件、面向对象程序设计与C++基础以及数据结构基础等内容。每章配有大量的习题，便于读者巩固所学知识，掌握程序设计的基本方法和编程技巧。

本书力求概念叙述准确、严谨，语言通俗易懂，适合作为高等院校理工科非计算机专业的“C语言程序设计”课程的教材，也可供工程技术人员参考。

图书在版编目（CIP）数据

C语言程序设计教程/朱鸣华等编著. —2版. —北京：机械工业出版社，2011.1
（高等院校精品课程系列教材）

ISBN 978-7-111-32866-7

Ⅰ.C…　Ⅱ.朱…　Ⅲ.C语言-程序设计-高等学校-教材　Ⅳ.TP312

中国版本图书馆CIP数据核字（2010）第256806号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：迟振春
北京市荣盛彩色印刷有限公司印刷
2011年3月第2版第2次印刷
185mm×260mm · 17印张
标准书号：ISBN 978-7-111-32866-7
定价：29.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

第2版前言

《C语言程序设计教程》教材于2007年2月出版发行第1版，出版后受到广大读者的喜爱，截止到目前共印刷了6次，累计发行了近两万册，被全国多所院校选作C语言程序设计课程的教材。另外，该教材2009年被评为辽宁省精品教材。

为了配合本教材使用，2009年3月作者组织编写并出版了《C语言程序设计习题解析与上机指导》实验配套教材。该配套教材从教学实际需要出发，兼顾不同学生的计算机基础，对主教材中的主要知识点进行了归纳和总结，对各章节的练习题中的疑难问题和常见问题进行了详细解析，使学生能自行进行有目的的实验训练。

为了适应计算机科学技术的发展，更好地满足教学的实际需要，作者征求广大读者的意见，在第1版的基础上对全书进行了如下的修订和调整：

1）叙述适应C语言的发展，兼顾 C99标准，以Visual C++ 6.0系统环境为主，规范程序的书写格式。

2）增加了第2章“算法与程序设计基础”，以加强学生程序设计中算法设计的训练。

3）增加了程序示例的注释，更换和充实了部分章节中的例题，增加了综合程序设计实例，强调综合程序设计的能力，突出程序设计能力的培养，更加方便学生学习和理解。

4）重新组织内容，淡化不重要的章节和语法，将编译预处理压缩放在第8章“函数”中进行简要介绍。

5）对第10章结构体部分增加了链表的简单叙述和操作，以满足不同学习者和教学的需求。

6）所有程序均在Visual C++ 6.0环境下调试通过。

第2版由朱鸣华、刘旭麟、杨微等编写。其中，第1、3、4章由罗晓芳编写；第5、8章由李慧编写；第6、9章和附录由刘旭麟编写；第7、11章由孙大为编写；第10章和编译预处理内容由杨微编写；第12章由赵晶编写；第2、13章由朱鸣华编写。全书由朱鸣华、刘旭麟统稿。

在本书的修订过程中，得到了大连理工大学计算机基础教研室很多老师的帮助，许多使用本教材的高校老师都提出了很好的建议，在此表示衷心的感谢。由于作者水平有限，书中难免有疏漏和不当之处，敬请读者和专家指正。

编　者

2010年11月

第1版前言

掌握一门高级程序设计语言是高等院校理工科学生利用计算机解决实际问题的必要条件之一。目前，国内各高等院校都把计算机程序设计课程作为重要的计算机基础课程。

C语言是国内外广泛使用的计算机程序设计语言之一，是国内外大学普遍开设的程序设计课程。C语言功能丰富，表达能力强，使用灵活方便，程序执行效率高，它不但具有高级语言的功能，而且还可以实现汇编语言的许多功能。C语言程序具有完善的模块程序结构，可移植性好，而且可以直接实现对系统硬件及外部设备接口的控制，具有较强的系统处理能力。

本书是根据教育部《关于进一步加强高等学校计算机基础教学的意见》的教学基本要求和高等院校计算机基础教学改革的需要，结合作者多年讲授C语言程序设计课程的教学经验编写而成的。全书共分13章，详细介绍了C语言的基本语法结构，同时介绍了面向对象程序设计语言C++的基本概念和数据结构的基本内容。另外，每章均配有精心安排的例题和习题，书后还给出了习题参考答案。

本书内容丰富，实用性强，侧重结合一般数值计算介绍利用C语言进行计算机程序设计的基本知识，使读者掌握C语言的基本内容及程序设计的基本方法和编程技巧，了解进行科学计算的一般思路与方法，为进一步学习和应用计算机打下基础。本书具有如下特点：

1) 内容全面。教材内容体系、前后内容的衔接、知识点的安排由浅入深，循序渐进，全书围绕结构化程序设计的方法，详细介绍C语言的语法规则和程序设计的基本方法。另外，C++和数据结构的内容有利于学生扩充知识及后续课程的学习。

2) 通俗易懂。本书在编写中力求做到概念叙述准确、严谨，语言简练通俗。为便于学生更好地学习和理解C语言，在一些较难理解的部分尽量使用图解的方法，在例题中加入了部分代码注释，使读者低起点、高效率、高目标地掌握C语言。

3) 实践性强。书中示例丰富，习题难易适中，覆盖面广，既包括理解语法的内容，又有增加兴趣和提高编程能力的例子。例题和习题强调了编程能力训练的重要性，淡化了部分语法细节。

本书的叙述以ANSI C为基础，书中的程序是在Turbo C环境下调试通过的，使用其他C语言编译系统的读者请注意参考有关手册。

本书适合作为高等院校理工科非计算机专业的“C语言程序设计”课程的教材和工程技术人员学习C语言的参考书。另外，本书还可作为自学用书。

本书适合于学时较少，并与多媒体教学方法相匹配的教学方式。对于本书的学时数，建议课堂教学32～48学时，上机实践24～32学时。为了克服学时少、内容多的矛盾，建议在教学中注重学生程序设计能力的培养，采用任务驱动、案例教学相结合的教学模式，以提高学生学习的兴趣和主动性，语法问题则留给学生课后自学。

本书第1、2、3章由罗晓芳编写，第4、7章由李慧编写，第5、9章和附录由刘旭麟编写，第6、11章由孙大为编写，第8、10章由杨微编写，第12章由赵晶编写，第13章由朱鸣华和赵晶编写。全书由朱鸣华、刘旭麟统稿。

在本书的编写过程中得到了许多老师的帮助，在此表示诚挚的谢意。由于作者水平有限，书中难免有疏漏和不当之处，敬请读者和专家指正。

编　者

2006年12月

教学建议

章节	教学要求	课时
第1章 C语言概述	了解C语言的特点及发展 了解程序设计的一般步骤 掌握C程序的基本结构与书写格式 掌握头文件、数据说明、函数的开始和结束标志	1
第2章 算法与程序设计基础	了解算法的基本概念 掌握算法的常用表示方法 掌握结构化程序设计的思想和方法	1
第3章 数据类型、运算符与表达式	掌握基本数据类型的定义方法 掌握运算符的种类、运算优先级、结合性 掌握不同类型数据间的转换与运算 掌握各表达式求值规则 了解位运算的规则	4
第4章 数据的输入和输出	掌握各种类型数据的输入和输出	2
第5章 选择结构	掌握用关系运算符和逻辑运算符设置表达式 掌握用if语句实现选择结构 掌握用switch语句实现多分支选择结构 掌握条件表达式的设置	2
第6章 循环结构	掌握for循环结构 掌握while和do-while循环结构 掌握continue、break、return语句 掌握 goto语句与标号的配合使用 掌握利用循环进行程序设计的方法及各种循环结构的正确嵌套	3
第7章 数组	掌握一维数组和多维数组的定义、初始化和引用 掌握字符串与字符数组的使用 掌握常用的字符串处理函数 掌握利用数组进行程序设计的方法	3
第8章 函数	掌握函数的定义方法 掌握函数的类型和返回值 掌握形式参数与实际参数的传递 掌握函数的调用方式和递归调用 掌握局部变量和全局变量的使用 掌握变量的存储类型（自动、静态、寄存器、外部）、变量的作用域和生存期 了解内部函数和外部函数 掌握带参数的宏定义及其使用 掌握不带参数的宏定义及其使用 掌握文件包含的概念和使用	4

（续）

章节	教学要求	课时
第9章 指针	掌握指针的概念及指针的定义 掌握指针的运算 掌握指向变量、数组、字符串、函数的指针变量的定义和使用 掌握用指针作函数参数 了解指针数组和二级指针的概念及其定义方法 了解main函数的命令行参数	5
第10章 结构体与共用体	掌握结构体和共用体类型数据的定义方法 掌握结构体和共用体类型数据的引用方法 掌握指向结构体的指针变量 了解用typedef定义类型的方法 了解链表的建立和使用	3
第11章 文件	掌握文件的基本概念 掌握文件类型指针的定义 掌握文件的打开与关闭 掌握文件的常用读写方式	2
第12章（选讲） 面向对象程序设计与C++基础	了解面向对象程序设计的基本特点 了解C++程序设计语言的基本思想	2
第13章（自学） 数据结构基础	掌握栈、队列、二叉树、图的数据存储结构及访问方法	
总课时（第1~12章）		32

说明：

1）建议采用课堂讲授与自学相结合的教学方法，教师在课堂上重点介绍编程思路以及如何使用C语言实现算法，讲授关键的知识点，细节的语法规则主要靠学生自己学习以及在实践中理解和掌握。可以采用任务驱动、互动式教学、案例教学相结合的教学模式，调动学生学习的兴趣和主动性。

2）为了达到更好的教学效果，使学生更容易入门并掌握利用C语言编写一般程序的方法，建议充分利用本教材的配套实验指导书，要求学生按照单元完成各章节的习题，并仔细阅读实验指导书的解析，加深对各知识点的理解。

3）除了本书各章节的编程题目外，建议教师结合学生的学习情况补充其他题目，加强实践课程的指导和要求，对于共性的问题及时进行讲解。在课程最后，给出一个有一定规模的综合性的程序设计题目，要求学生独立或分组完成，强调程序的调试能力以及对常见错误的处理能力。

4）对于本书的学时数，可根据具体情况确定，有些学校的学时安排较少，教材中的所有内容不能完全讲解，建议在教学中突出重点，重点强调学生程序设计能力的培养，而对于位运算、第12章及第13章不做介绍，感兴趣的学生可以自己阅读，了解相关的知识。

目　录

第1章 C语言概述

1.1 C语言发展简史

在计算机科学应用领域，人与计算机之间交流最通用的手段是程序设计语言。当人们想利用计算机解决某个问题时，必须用程序设计语言安排好处理步骤，并存入计算机内供计算机执行，这些用程序设计语言安排好的处理步骤称为*计算机程序*。程序是计算机操作指令的集合。用程序设计语言编制计算机程序的过程叫做*程序设计*。

目前有很多程序设计语言，而且新的语言不断涌现。常见的有Fortran、BASIC、COBOL、Pascal 、C、C++、Java、Delphi等。各类语言都有其各自的特点和适用的领域。

C语言是一种通用的程序设计语言，由于它很适合用来编写编译器和操作系统，因此被称为“系统编程语言”，但它同样适用于编写不同领域中的大多数程序。

1. C语言的产生

C语言是一种被广泛应用的计算机高级程序设计语言，是在B语言的基础上发展起来的，它经历了不同的发展阶段。

早期的系统软件设计均采用汇编语言，例如，大家熟知的UNIX操作系统。尽管汇编语言在可移植性、可维护性和描述问题的效率等方面远远不及高级程序设计语言，但是一般的高级语言有时难以实现汇编语言的某些功能。

那么，能否设计出一种集汇编语言与高级语言的优点于一身的语言呢？这种思路促成了UNIX系统的开发者（美国贝尔实验室的Ken Thompson）于1970年设计出了既简单又便于硬件操作的B语言，并用B语言写了第一个UNIX操作系统，这个操作系统先在PDP-7上实现，1971年又在PDP-11/20上实现。

B语言的前身是BCPL（Basic Combined Programming Language），它是英国剑桥大学的Matin Richards在1967年基于CPL语言设计的，而CPL语言又是在1963年基于ALGOL 60产生的。

1972~1973年，贝尔实验室的D. M. Ritchie在B语言的基础上设计出C语言，该语言弥补了B语言过于简单、功能有限的不足。

1973年，Ken Thompson和D. M. Ritchie合作将90%以上的UNIX代码用C改写。随着改写UNIX操作系统的成功，C语言也逐渐被人们接受。

1987年以后，C语言已先后被移植到大、中、小、微型机上，并独立于UNIX和PDP，从而得到了广泛应用。

2. C 语言的发展和应用

1978年，B. M. Kernighan 和D. M. Ritchie合写了一本经典著作——《C程序设计语言》(The C Programming Language，中文版、影印版均已由机械工业出版社引进出版)，它奠定了C语言的基础，被称为标准 C。

1983年，美国国家标准学会（ANSI）根据C语言问世以来的各种版本对C的发展和扩充，制订了新的标准，称为ANSI C。1987年又公布了新标准，称为87 ANSI C。目前流行的多种版本的C语言编译系统都是以此为基础的。例如，在微机上广泛使用的Microsoft C、Turbo C、Quick C等。

3. C语言和C++语言交融发展

由于C语言是面向过程的结构化和模块化的程序设计语言，当处理的问题比较复杂、规模庞大时，就显现出一些不足，由此面向对象的程序设计语言C++应运而生。C++的基础是C，它保留了C的所有优点，增加了面向对象机制，并且与之完全兼容。绝大多数C语言程序可以不经修改直接在C++环境中运行。

1.2 C语言的特点

C语言产生的历史并不长，它之所以后来居上，成为深受欢迎的高级语言之一，是因为和其他早期的高级语言相比，它具有以下优点：

1）兼具高级、低级语言的双重能力。C语言允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作，所以又被称为中级语言。

2）生成的目标代码质量好，程序执行效率高。C语言具有汇编语言的许多特性，一般只比汇编程序生成的目标代码效率低10%～20%，可以开发出执行速度很快的程序。

3）语言简洁，结构清晰。C程序通常是由若干函数组成的，强大的函数功能为程序的模块化和结构化提供了保证，因此程序简洁清晰，可读性强。

4）语言表达能力强。C语言运算符丰富，例如，在C语言中，把括号、赋值、强制类型转换等都作为运算符处理。C语言具有现代化语言的各种数据结构，如整型、字符型、数组型、指针型、结构体和共用体等，而且具有结构化的控制语句。

5）程序通用性、可移植性好。C语言没有依赖于硬件的输入/输出语句，而采用系统的库函数进行输入/输出操作，因此C语言不依赖于任何硬件系统，这种特性使得用C语言编写的程序很容易移植到其他环境中。

当然，C语言也有自身的不足，和其他高级语言相比，其语法限制不太严格，例如，对变量的类型约束不严格，影响程序的安全性，对数组下标越界不作检查等。从应用的角度来看，C语言比其他高级语言较难掌握。

总之，C语言既具有高级语言的特点，又具有汇编语言的特点；既是一个成功的系统设计语言，又是一个实用的程序设计语言；既能用来编写不依赖计算机硬件的应用程序，又能用来编写各种系统程序。它是一种深受欢迎、应用广泛的程序设计语言。

1.3 简单C语言程序举例

在这一节中，我们通过两个简单的C语言程序例子来介绍C语言的程序结构，并对C语言的基本语法成分进行相应的说明，以便使读者对C语言程序有一个大致了解。

【例1-1】 计算矩形的面积。

```
#include <stdio.h>                  /*1:编译预处理*/
int main( )                         /*2:主函数*/
{                                   /*3:函数体开始*/
  float  h, w, area;                /*4:声明部分，定义变量*/
  h=10.5;                           /*5:以下4条C语句为执行部分,给变量h和w赋值*/
  w=20.5;
  area=h*w;                         /*7:计算矩形的面积*/
  printf(" area=%6.2f", area);      /*8:输出area的值*/
  return 0;                         /*9:返回值为0 */
}                                   /*10:函数体结束*/
```

运行结果：

```
area=215.25
```

每行中的 /*……*/内包括表示程序注释的内容。

第1行：是一个编译预处理，在程序编译前执行，指示编译程序如何对源程序进行处理。它以“#”开头，结尾不加分号，以示和C语句的区别。

第2行：main表示主函数，每一个C程序都必须有一个主函数，int表示主函数为整型。函数体由第3行和第9行的一对花括号括起来。

第4行：是变量声明部分，定义变量h、w和area为实型变量。

第5、6行：是两条赋值语句，给变量h赋值10.5，w赋值20.5。

第7行：将算术表达式h*w的值赋予变量area。

第8行：调用函数printf输出矩形面积值。

第9行：向操作系统返回一个零值，如果程序不能正常执行，则会自动向操作系统返回一个非零值，一般为−1。

【例1-2】　计算两个矩形的面积之和。

```
#include <stdio.h>                          /*1:编译预处理*/
float area(float h, float w )               /*2:定义函数area*/
{
  double s;
  s=h*w;
  return s;                                 /*6:返回s的值, return是关键字*/
}
int main( )                                 //8:主函数
{
  double  h1, h2, w1, w2, s1, s2;           //10:声明部分, 定义变量
  h1=10.5; w1=20.5;
  h2=1.5*h1; w2=1.5*w1;                     //12:计算变量h2, w2的值
  s1=area(h1, w1);                          //13:调用area函数, 将得到的返回值赋给变量s1
  s2=area(h2, w2);
  printf("area=%6.2f ", s1+s2);             //15:输出两个矩形的面积之和
  return 0;
}
```

运行结果：

```
area=699.56
```

本程序包括主函数、被调用函数 area和一个编译预处理指令。

第2行：从该行开始到第6行定义函数area，包括函数类型、函数名和函数体等部分。

第13、14行：调用函数area，将两次调用的返回值分别赋给变量s1和s2。

第15行：计算并输出两个矩形的面积之和。其中每行中“//”后面的所有字符也是注释的内容。

上述两个函数构成了一个完整的程序，称为*源程序*。它以文件的方式存在，文件中包含函数的源程序代码。可以把这两个函数放在一个文件中，当程序语句多的时候也可以分别以函数为单位放在两个以上的文件中，C语言规定保存C源程序文件的扩展名为“.c”。

1.4　C语言程序的组成与结构

通过以上两个例子，我们对C语言程序的组成和结构有了初步和直观的了解，总结如下：

1）一个C语言程序的主体结构是由一个或若干个函数构成的。这些函数的代码以一个或若干个文件的形式保存。这些函数中必须有且只能有一个名为main的主函数。

2）主函数main是程序的入口，它可以出现在程序的任何位置。一个C程序总是从主函数main开始执行。

3）C程序中的函数包括：主函数main，用户自定义函数（例如，例1-2中的area），系统提供的库函数（例如，输出函数 printf）。

4）函数由函数头和函数体两部分组成，函数头由函数类型的定义、函数名和参数表组成，函数体由声明部分（所使用变量和函数的说明）和若干执行语句组成。

5）语句由关键字和表达式组成，每个语句和声明部分的结尾都必须加分号。复合语句的开头和结尾使用左、右花括号{ }。

关键字是C语言系统为了清晰地表达程序的功能而使用的一些英文单词或单词缩写。例如，例1-2中的 return就是关键字。

用运算符将操作对象连接起来、符合C语言语法的式子称为*表达式*。表达式的组成元素有：变量、常量、函数调用、运算符。这些组成元素是以标识符和关键字的形式存在的。例如，例1-2中的h2=1.5*h1和 w1=20.5 都是表达式。

6) 程序中“/* */”内的文字是程序的注释部分，是便于阅读理解程序的解释性附加文本，程序编译器完全忽略注释部分的内容。此外，在程序调试时，也可以将一部分代码转换为注释保留，而不必删除，以提高程序调试的效率。

另外，一些C语言开发工具还支持用“//”标识注释部分，如果某行程序代码前面插入符号“//”，该符号后面的部分就变为注释行，并且本行有效，不能跨行。一般情况下，如果注释内容在程序中占用多行，习惯用“/* */”，而单行注释内容用“//”标识即可。

从以上的分析可以发现，C程序的组织和构造与我们日常的文章的结构很类似，如表1-1所示。

表1-1 文章和 C语言对应的层次结构

文章的层次结构	对应C程序的层次结构	文章的层次结构	对应C程序的层次结构
文章	程序	句子	语句
章节	文件	词组	表达式
段落	函数	字	常量、变量、关键字、运算符等
开头第一段	主函数		

在一般语言的学习过程中，我们是先学字、词组，然后造句，阅读范文，最后写作文。现在学习计算机语言，我们也同样遵循这个规律，即先学习常量、变量的类型和定义方法，然后依次学习表达式、语句和函数等，同时阅读一些程序范例，最后编写程序。当然二者也有本质上的区别，一般语言的学习以形象思维为主，而计算机语言的学习是以逻辑思维为主。C语言的程序结构如图1-1所示。

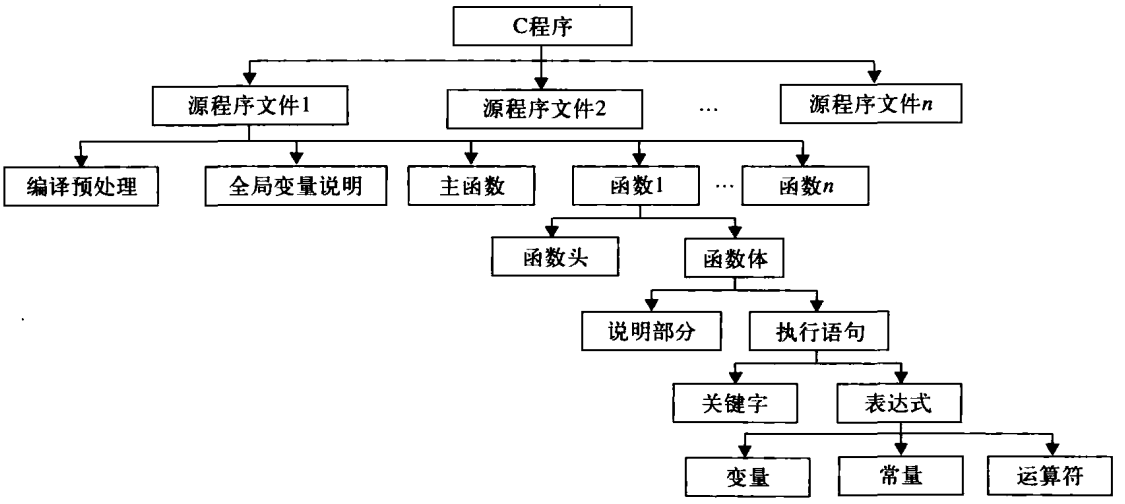

图1-1 C语言程序的层次结构图

1.5　C语言程序的开发步骤

一个C语言程序从最初编写到得到最终结果，大致经过以下几个步骤：

1）编辑源程序。选择一种C语言开发工具，输入编写好的程序代码，称之为源程序，它以文件的方式存在，文件的扩展名为“.c”。

2）编译源程序。为了使计算机能执行高级语言源程序，必须把源程序转换为二进制形式的目标程序，这个过程称为*编译源程序*。

编译是以源程序文件为单位分别进行的，每一个源程序文件对应生成一个目标文件，目标文件的扩展名为“.obj”。

编译过程中对源程序的全部内容进行检查，检查程序中关键字的拼写是否正确，根据程序的上下文检查语法是否有错，编译结束后系统显示所有的编译出错信息。

一般编译系统的出错信息有两种：一种是*错误*（error）信息，这类错误出现后，系统不生成目标文件，必须改正后重新编译；另一种是*警告*（warning）信息，是指一些不影响程序运行的不合理现象或轻微错误。例如，程序中定义了一个变量，却一直没有使用，这类警告信息出现后，系统可以生成目标文件。

3）连接目标文件。编译结束，得到一个或多个目标文件，此时要用系统提供的“连接程序”（linker）将一个程序的所有目标文件和系统的库文件以及系统提供的其他信息连接起来，最终形成一个可执行的二进制文件，可执行文件的扩展名为“.exe”。

4）运行程序。运行最终形成的可执行文件，得到运行的结果。

5）结果分析。分析程序的运行结果，如果发现结果不对，应检查程序或算法是否有问题，修改程序后再重复上面的步骤。其过程如图1-2所示。

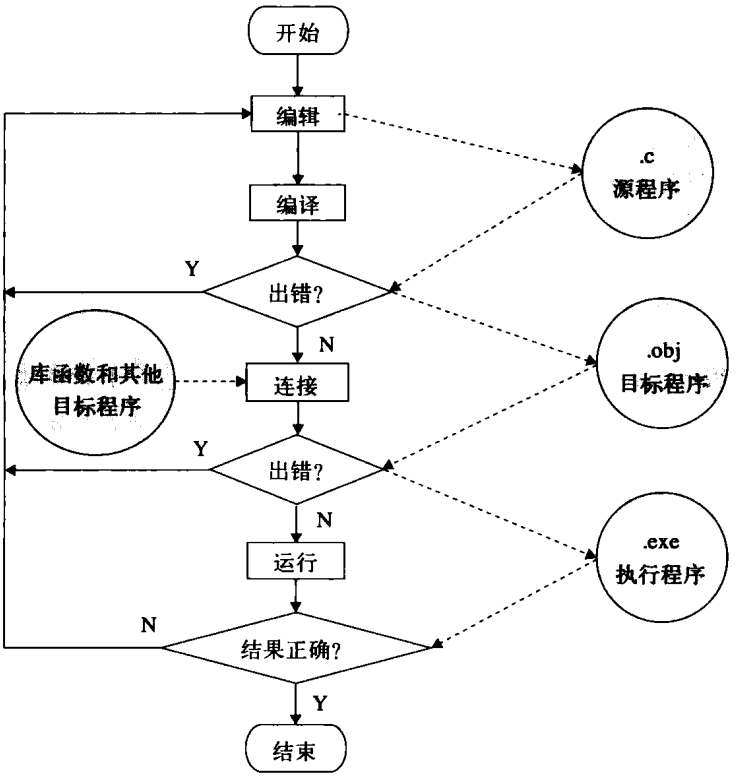

图1-2　C语言程序的开发步骤