



21世纪普通高等院校规划教材 • 信息技术类

普通高等教育“十二五”规划教材

C YUYAN CHENGXU SHEJI

C语言程序设计

主 编 谢 妙 肖志军

副主编 张茂胜 孙小雁 谢春萍 莫燕斌



西南交通大学出版社
[Http://press.swjtu.edu.cn](http://press.swjtu.edu.cn)



第2版 C语言程序设计(第2版)

ISBN 7-309-05400-3

定价: 24.00元

中国轻工业出版社

C语言程序设计

主编: 陈国平
副主编: 陈国平 陈国平 陈国平 陈国平

中国轻工业出版社

21 世纪普通高等院校规划教材—— 信息技术类

普通高等教育“十二五”规划教材

C 语言程序设计

主 编	谢 妙	肖志军		
副主编	张茂胜	孙小雁	谢春萍	莫燕斌
编 委	闭吕庆	洪月华	刘 恒	覃 娜
	熊春荣	肖租秀	余旺新	杨夏妮

西南交通大学出版社

· 成 都 ·

内容简介

本书是一本为培养学生程序设计的基本能力,将C语言作为入门语言的程序设计课程而编写的教材。

全书以程序设计为主线,针对初学者的认知规律,以编程应用为驱动,各章节以案例和问题入手,重点讲解程序设计的思想和方法,同时介绍了相关的C语言语法知识。全书的主要内容包括数据表达、数据控制和流程控制、常用的基本算法、编译预处理、命令行参数和程序设计风格等。涉及数据类型、表达式、顺序、分支、循环、函数、数组、指针、结构体、文件、指针和各种构造类型的混合应用、基本算法等。另外,还有《C语言程序设计实验指导》教材与本书配套使用。

本书可作为高等院校教学用书和计算机等级考试的培训教材,也可作为对C程序设计感兴趣的读者自学用书。

图书在版编目(CIP)数据

C语言程序设计 / 谢妙, 肖志军主编. —成都: 西南交通大学出版社, 2013.1

21世纪普通高等院校规划教材. 信息技术类 普通高等教育“十二五”规划教材

ISBN 978-7-5643-2168-0

I. ①C… II. ①谢… ②肖… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第020570号

21世纪普通高等院校规划教材——信息技术类
普通高等教育“十二五”规划教材
C语言程序设计

主编 谢 妙 肖志军

责任编辑	黄淑文
特邀编辑	黄庆斌
封面设计	本格设计
出版发行	西南交通大学出版社 (成都二环路北一段111号)
发行部电话	028-87600564 87600533
邮政编码	610031
网 址	http://press.swjtu.edu.cn
印 刷	成都中铁二局永经堂印务有限责任公司
成品尺寸	185 mm × 260 mm
印 张	16.625
字 数	415千字
版 次	2013年1月第1版
印 次	2013年1月第1次
书 号	ISBN 978-7-5643-2168-0
定 价	29.80元

图书如有印装质量问题 本社负责退换
版权所有 盗版必究 举报电话: 028-87600562

前 言

C 语言是一门以培养学生程序设计基本方法和技能为目标,以培养实践能力为重点的特色鲜明的课程。目前市面上大多数 C 语言教材都采取传统编排模式,先是入门总体介绍,然后是基础知识,再是程序结构体系等。此种方法以知识点为主线,这样很容易让学生过多地关注语法细节,而忽略了程序本身的重要性,不利于培养学生的程序设计能力和语言应用能力,但 C 语言程序设计课程的教学重点应该是培养学生的实践编程能力,因此,相应地,教材也要以程序设计为中心来组织内容。

本书作者基于多年教学过程的实践和思考,为了培养学生的编程能力,特撰写了《C 语言程序设计》教材,该教材打破了传统教材的讲授模式,先以案例入手,分析问题需要的知识点,再提出解决问题的方法和思路。然后根据需要讲解语法知识点。最后举一反三,并以应用实例来加以提升和巩固知识点,使得学生不仅要掌握高级程序设计语言的知识,更重要的是在实践中逐步掌握程序设计的思想和方法,以培养求解问题和语言应用的能力。

全书以程序设计为主线,以编程应用为驱动,着重让学生理解和掌握编程的基本思想和方法。为了能让读者尽快地进行编程应用,各章节通过问题和实例引入,重点分析程序设计的思想和方法,根据本章的应用需求,对于基础语言知识以用到什么介绍什么为原则,对相应的语言知识进行介绍,以便让读者通过实际的应用来学习语言知识。全书共分为 11 章,对于 C 语言的基本内容(数据类型、数据处理和流程控制)、常用算法和结构化程序设计思想都做了全面详细地介绍。其中,第 1~6 章侧重于讲解 C 语言的基本知识和基本编程方法,内容包括基本数据类型、数据处理中的表达式,以及流程控制中的顺序、分支、循环三种控制结构和编程应用;第 7~9 章介绍构造类型(数组、结构体类型、共用体类型和枚举类型)和指针在编程中的应用;第 10~11 章介绍文件的使用、编译预处理和用结构化程序设计思想实现复杂问题的编程等内容。

为了能让读者更好地学习掌握各个知识要点,对于各章节中的知识要点、程序设计的技巧、方法以及编程中的常见错误,均以“★”(注意)的形式给出。为了鼓励学生多思考、多应用,提高综合能力,本教材设计了多种类型和形式的练习题。根据需要在各节中安排了针对本节涉及的概念和编程实例的练习,题型多样,难度较低,学生可以即学即练,加深理解,巩固知识,提高学习兴趣;另外,每一章结束后还安排有习题,主要是综合性的题目,以程序设计题为主。

本书的第 1 章介绍了程序设计语言的基础知识、C 语言的主要特点和知识要点,以及利用计算机求解问题的程序设计过程;第 2 章通过几个顺序结构的实例,介绍顺序控制结构,以及在实例程序中所用到的语言知识;第 3、4 章通过对大量实例的分析与实现,分别详细讲解了分支结构和循环结构,以及在实例程序中用到的语言知识;第 5 章通过实例的引入,详细介绍了函数的基础知识、函数程序设计的基本思想和方法,以及变量的作用域和存储类型;第 6 章介绍了 C 语言的基本数据类型和表达式的作用;第 7 章通过典型的范例介绍了一维数组、二维数组和字符串的应用;第 8 章详细介绍了指针的基本概念,以及指针和数组、字符

串、函数的关系及其在编程中的应用；第 9 章用案例详细说明了结构体类型在编程中的应用，以及指向结构体类型数据的指针在编程中的应用，同时介绍了共用体类型、枚举类型和 typedef 重定义类型；第 10 章讲解了程序的结构化设计思想，进一步介绍了函数和程序的结构方面具有一定深度的内容；第 11 章介绍了文件的基本概念和编程应用；附录 I 将分布在本书各章节中的数据类型、表达式和控制结构等内容做了归纳性汇总，使读者对 C 语言的数据表达、数据处理和流程控制有一种更清楚的认识。

本书的编写小组成员由多年来一直从事 C 语言程序设计教育的具有丰富教学经验的一线教师组成，教材内容力求基于理论、注重实践、符合教学规律，希望借此培养和提高大学生的程序设计基本能力。全书由谢妙、肖志军担任主编。具体编写分工为：谢春萍编写第 1 章，孙小雁编写第 2 章，覃娜编写第 3 章，刘恒编写第 4 章，肖志军、杨夏妮编写第 5 章，张茂胜编写第 6 章，熊春荣编写第 7 章，洪月华、闭吕庆编写第 8 章，余旺新编写第 9 章，谢妙编写第 10 章，莫燕斌编写第 11 章，肖租秀编写附录。谢妙、肖志军完成了本书的统稿和审读工作。

为方便教学，本书配套了实验指导书，每个实验包括精心设计的编程示例和调试示例，以及实验题（编程题与改错题）。读者可以先模仿示例操作，然后再做实验题，通过“模仿-改写-编写”的上机实践过程，在循序渐进的引导中逐步熟悉编程环境，理解和掌握程序设计的思想、方法和技巧以及程序调试方法。

全书在组织和编写过程中，得到了各级相关领导和部门的关心和支持，得到了很多计算机科学教师的热心支持与帮助，才使我们顺利地完成了本教材的编写，在此表示衷心感谢！同时，还要特别感谢在编写过程中使用的所有文献的作者和出版社。

由于编者水平有限，书中难免会有不足之处，敬请各位专家和读者批评指正，以便我们在以后修正。

编 者

2012 年 10 月

目 录

第 1 章 C 语言概述	1
1.1 一个简单的 C 语言程序	1
1.2 程序与程序设计语言	2
1.3 C 语言的发展历史与特点	2
1.4 求解问题的程序设计过程	5
习 题	8
第 2 章 简单例子与数据表示	9
2.1 在屏幕上显示学生个人信息	9
2.2 计算学生成绩	10
2.3 圆球表面积与体积的计算	15
习 题	18
第 3 章 分支结构	19
3.1 市民用电量分段收费	19
3.2 判断英文字母并进行大小写转换	23
3.3 学生管理系统目录选择	34
习 题	40
第 4 章 循环结构	41
4.1 求 1 累加到 100 的和	41
4.2 统计输入的一批整数中正数和负数的个数	46
4.3 统计一个整数的位数	50
4.4 素数的判断	52
4.5 求 $1! + 2! + \dots + 20!$	56
4.6 循环结构程序设计举例	59
习 题	65
第 5 章 函 数	67
5.1 计算长方形的面积并按指定的样式输出	67
5.2 变量的作用域和生命周期	77
习 题	83

第 6 章 基本数据类型与表达式	85
6.1 数据的存储	85
6.2 基本数据类型	86
6.3 运算符与表达式	100
习 题	108
第 7 章 数 组	111
7.1 某门课程多名学生的成绩统计	111
7.2 多门课程多名学生的成绩统计	119
7.3 统计字符串中数字字符的个数	124
习 题	127
第 8 章 指 针	129
8.1 寻找考试座位号	129
8.2 利用指针输出存放成绩的数组	135
8.3 交换管理密码	147
8.4 统计句子中关键字符的个数	153
8.5 中英文星期名称的转换	160
8.6 找最长字符串	169
习 题	173
第 9 章 结构体与共用体	176
9.1 设计学生成绩记录系统	176
9.2 结构体变量	181
9.3 结构体数组	185
9.4 指向结构体类型数据的指针	187
9.5 单向链表	189
9.6 共用体	198
9.7 枚举类型	199
9.8 typedef 重定义类型	200
习 题	201
第 10 章 函数与程序结构	203
10.1 C 程序函数的组织	203
10.2 递归函数	205
10.3 全局变量与程序文件模块	211
10.4 外部函数和内部函数	214
10.5 编译预处理	215
习 题	222
第 11 章 文 件	225

11.1 将学生成绩写入文件	225
11.2 录入学生成绩	229
11.3 常用文件读写函数	230
11.4 其他相关函数	234
11.5 文件综合应用：网店交易记录系统	236
习 题	239
附录 I C 语言中常用关键字	240
附录 II 运算符和结合性	242
附录 III C 语言使用中常见错误	244
附录 IV 常用字符与 ASCII 代码对照表	248
附录 V C 语言常用语法提要	253
附录 VI C 语言中常用库函数	255
参考文献	258

第1章 C语言概述

程序设计语言的初学者，应该尽快掌握程序设计的流程，克服可能的畏难想法。首先要对程序设计语言，即本教材的C语言有所了解，再通过不断地编程练习，逐步领会和掌握程序设计的基本思想和方法。本书建议读者从一开始学习C语言就要试着编写程序，可以先模仿教材中的例题，试着修改体会，循序渐进，再到独立编写程序。熟练的编程技能是在反复练习的基础上日积月累而来的，希望读者通过本章的学习，能初步体会到C语言程序设计的基本思想和方法，领略程序设计的魅力所在。

1.1 一个简单的C语言程序

为了让读者感性认识C语言，首先来看一个简单的C语言程序。

【例 1.1】 求两数乘积。

源代码

```
#include <stdio.h>          /*编译预处理命令，包含标准输入输出头文件*/
void main()                /*主函数*/
{
    int x,y,z;              /*定义三个整型变量*/
    int function(int x,int y) ; /*声明函数*/
    scanf("%d%d",&x,&y);    /*从键盘输入两个整数，先后赋值给 x、y*/
    z=function(x,y);        /*调用 function 函数完成计算功能*/
    printf("%d\n",z);       /*在屏幕上输出两数乘积*/
}

int function(int x,int y)   /*定义一个函数 function*/
{
    int z;                  /*定义一个整型变量*/
    z=x*y;                  /*x、y 值相乘赋给 z*/
    return z;               /*返回运算结果值 z 给主调函数*/
}
```

运行情况：

运行程序，从键盘输入“18 5”，屏幕上输出“90”。

以上C程序读者不一定要能完全理解，所涉及的各个知识点会在后面章节逐步学习，但是希望能对C程序有一个初步认识。

C 程序是由函数构成的。C 语言是面向过程的语言，函数是组成 C 程序的基本单元。在上面例子中，程序由 4 个函数组成：main 函数、function 函数、scanf 函数和 printf 函数。main 函数是程序的主函数；function 函数是程序中定义的功能函数，其功能是求两数乘积；scanf 函数和 printf 函数则是系统自定义的库函数，用于输入和输出数据。

C 语言规定所有的 C 程序都有且只有一个 main 函数。任何一个 C 程序都是从 main 函数开始执行的，最后整个程序结束也是在 main 函数结束。在例 1.1 程序中，程序先执行 main 函数中的 scanf 函数调用，完成数据的输入。然后再调用自定义函数 function 完成对两个数据的乘积运算。最后通过调用 printf 函数将结果按十进制（%d）输出。

程序的最基本功能就是数据运算和运算过程控制，因此，以上程序中定义了 3 个整型变量 x、y、z 来表示参加运算的整数。同时通过调用 function 函数完成运算功能，以及使用 return 语句将运算结果回送到主函数 main。

1.2 程序与程序设计语言

计算机的出现改变了人类的生活方式，计算机不仅计算能力超强，还能监控，做出判断，但它不是自动工作的，而是由程序控制的。计算机程序是人们为解决某问题，用计算机可以识别的代码编写的一系列操作步骤。计算机能严格按照这些步骤执行。

人怎么指挥计算机按照人们的意愿工作，通过什么语言沟通呢？因为计算机只能识别二进制的信息。因此，在计算机产生的初期，人们必须编写由 0 和 1 所组成的一系列指令，即机器指令，通过它来指挥计算机的工作。某种型号的机器语言的指令集称为该计算机的机器语言。机器语言紧密依赖计算机的硬件，而且难学、难用，非常难以推广。

20 世纪 50 年代出现了用于程序设计的高级语言，它比较接近于人们习惯用的自然语言和数学语言。如 if 表示如果，printf 表示打印。用高级语言编写的程序直观易学，容易理解、修改和维护，通用性强，移植性好。全世界先后出现了 2 500 种以上的高级语言，其中，应用广泛，影响较大的有：FORTRAN、BASIC、PASCAL、Visual Basic、C、C++、JAVA 等。

用高级语言编写的程序，计算机不能直接识别和执行（计算机只能直接识别二进制指令），必须用编译系统把高级语言编写的程序翻译成机器语言程序。

1.3 C 语言的发展历史与特点

1. C 语言的发展历史过程

C 语言是在 20 世纪 70 年代初问世的。1978 年由美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言。同时由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的“The C programming Language”一书（称之为《K&R》标准）。在《K&R》中并没有定义一门完整的标准 C 语言，后来由美国国家标准学会在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称之为 ANSI C。

早期的C语言主要用于UNIX系统。人们逐渐认识到C语言的强大功能和各方面的优点之后,20世纪80年代,C语言开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到广泛使用。C语言成为当代最优秀的程序设计语言之一。

在C语言的基础上,1983年又由贝尔实验室的Bjarne Stroustrup推出了C++。C++进一步扩充和完善了C语言,成为一种面向对象的程序设计语言。C++提出了一些更为深入的概念,它所支持的面向对象的概念容易将问题空间直接映射到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法,因而也增加了整个语言的复杂性,掌握起来有一定难度。

C语言是C++语言的基础,C++语言和C语言在很多方面是兼容的。因此,掌握了C语言,再进一步学习C++语言就能以一种熟悉的语法来学习面向对象的语言,从而取得事半功倍的效果。

2. C语言的特点

C语言是一种结构化语言。它层次清晰,便于按模块化方式组织程序,易于调试和维护。C语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,而且还可以直接访问内存的物理地址,如进行位(bit)操作。C语言由于实现了对硬件的编程操作,集高级语言和低级语言的功能于一体,因此既可用于系统软件的开发,也可用于应用软件的开发。此外,C语言还具有效率高,可移植性强等特点,因此,广泛地移植到了各类型计算机上,从而形成了多种版本的C语言。

目前最流行的C语言版本有以下几种:Microsoft C或称MS C;Borland Turbo C或称Turbo C;AT&T C。这些C语言版本不仅实现了ANSI C标准,而且在此基础上各自做了一些扩充,使之更加方便、完美。

C语言的特点如下:

(1) C语言简洁紧凑,使用方便而灵活。ANSI C一共只有32个关键字,9种控制语句。程序书写自由,主要用小写字母表示。

(2) 运算符丰富,共有34种。C语言把括号、赋值、逗号等都作为运算符处理。数据类型丰富,能充分有效地表达各种各样的数据。

(3) 具有结构化的控制语句,可以有效地表达数据处理的过程,即程序的控制过程。

(4) 语法限制不太严格,程序设计自由度大。

(5) C语言允许直接访问物理地址,能进行位(bit)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。因此有人把它称为中级语言。

(6) 生成的目标代码质量高,程序执行效率高。

(7) 与汇编语言相比,用C语言写的程序可移植性好。

在C语言的学习过程中,读者会逐步地体会到C语言的灵活性和强大的功能。

3. C程序的结构特点

(1) 一个C语言源程序可以由一个或多个源文件组成。

(2) 每个源文件可由一个或多个函数组成。函数是C程序的基本组成单位。

(3) 一个源程序不论由多少个源文件组成,都有一个且只能有一个main函数,即主函数。

(4) 源程序中可以有预处理命令, 预处理命令通常应放在源文件或源程序的最前面。

(5) 每一个说明, 每一个语句都必须以分号结尾。但预处理命令、函数头和花括号“{”之后不能加分号。

(6) 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符, 也可不再加空格来间隔。

4. C 语言词汇

在 C 语言中使用的词汇分为六类: 标识符、关键字、运算符、分隔符、常量、注释符。

(1) 标识符。

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外, 其余都由用户自定义。C 规定, 标识符只能由字母、数字和下划线组成。其中, 第一个字符必须是字母或下划线。

例如, 以下标识符是合法的:

a, x, x3, BOOK_1, sum5

以下标识符是非法的:

3s 以数字开头

s*T 出现非法字符*

- 3x 以减号开头

bowy - 1 出现非法字符 - (减号)

在使用标识符时, 还必须注意以下几点:

① 标准 C 不限制标识符的长度, 但它受各种版本的 C 语言编译系统和具体机器的限制。如在某版本 C 中规定标识符前八位有效, 当两个标识符前八位相同时, 则被认为是同一个标识符。

② 在标识符中, 大小写是有区别的。例如, BOOK 和 book 是两个不同的标识符。

③ 标识符虽然可由程序员随意定义, 但标识符是用于标识某个量的符号, 因此, 命名应尽量有相应的意义, 以便于阅读理解, 做到“见名知义”。

(2) 关键字。

关键字是由 C 语言规定的具有特定意义的字符串, 通常也称为保留字。用户定义的标识符不应与关键字相同。C 语言的关键字详见附录 I。

★注意: 在 C 语言中, 关键字都是小写的。

(3) 运算符。

C 语言中含有相当丰富的运算符。运算符、变量和函数一起组成了表达式, 表示各种运算功能。运算符由一个或多个字符组成。

(4) 分隔符。

在 C 语言中采用的分隔符有逗号和空格等。逗号主要用在类型说明和函数参数表中, 分隔各个变量。空格多用于语句各单词之间, 作间隔符。在关键字、标识符之间必须要有一个以上的空格符作间隔, 否则, 将会出现语法错误, 例如, 把“int a;”写成“inta;”C 语言编译器会把“inta”当成一个标识符处理, 其结果必然出错。

(5) 常量。

C语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量等多种。在后面章节中会做专门介绍。

(6) 注释符。

C语言的注释符是以“/*”开头，以“*/”结尾的串。在“/*”和“*/”之间的即为注释。程序编译时，不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，使翻译跳过不作处理，待调试结束后再去掉注释符。注释符也可以用“//”，但是只能对当前行进行注释。

1.4 求解问题的程序设计过程

本节通过一个例子来说明程序设计的过程。

【例 1.2】求 5 的阶乘，即 $1 \times 2 \times 3 \times 4 \times 5$ 。

1. 问题分析与算法设计

最原始最直接的方法：设一个变量 t 表示阶乘，直接计算

$$t = 1 \times 2 \times 3 \times 4 \times 5$$

这样的设计虽然正确，但不具有广泛性和通用性，程序扩展性较差。本问题是求在一定次数（或范围）的若干整数乘积，是一个求多个乘数的乘积问题。

解决这类问题的基本方法如下：设置一个保存乘积的变量（比如 t ），再将指定范围内的整数逐一乘以 t 并保存到 t 中。假设用 i 表示乘数。算法可以这样设计：

S1：使 $t = 1$

S2：使 $i = 2$

S3：使 $t \times i$ ，乘积仍然放在变量 t 中，即 $t = t \times i$

S4：使 i 的值加 1，即 $i = i + 1$

S5：如果 $i \leq 5$ ，返回重新执行步骤 S3 以及其后的 S4 和 S5；否则，算法结束。

如果计算 $100!$ ，只需将 S5 中的 $i \leq 5$ ，改成 $i \leq 100$ 即可。

如果改求 $1 \times 3 \times 5 \times 7 \times 9 \times 11$ ，算法也只需做很少的改动：

S1： $1 \rightarrow t$

S2： $3 \rightarrow i$

S3： $t \times i \rightarrow t$

S4： $i + 2 \rightarrow i$

S5：若 $i \leq 11$ ，返回 S3，否则，结束。

例题中乘法多次执行，就可以用程序设计语言的循环控制语句来实现。该算法是较好的算法，因为计算机具有高速运算能力，所以实现循环轻而易举。这样设计的程序稍加修改，就可以用来求解其他类似问题，例如有规律的多个数据求和问题。

明确了解决问题的步骤，就可以用流程图来表示。用流程图来表示算法，直观形象，易于理解。流程图常用元素如图 1.1 所示，例 1.2 的流程图如图 1.2 所示。

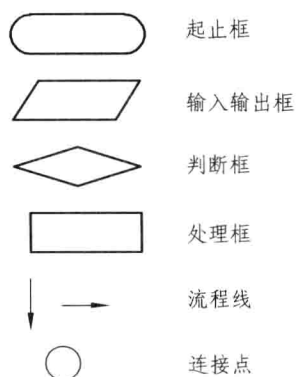


图 1.1 流程图常用元素

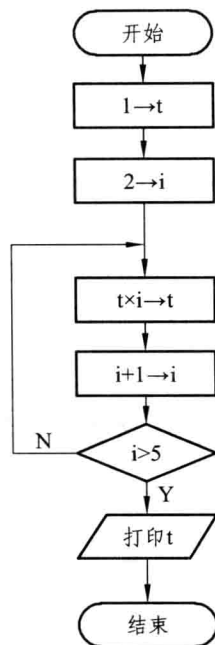


图 1.2 例 1.2 流程图

按照结构化程序设计的观点，任何程序都可以由三种基本控制结构组成，即顺序结构、分支结构、循环结构。这三种结构顺序组成的算法结构，可以解决任何复杂的问题。

(1) 顺序结构。如图 1.3 所示，程序在执行过程中，按照模块的顺序，执行完模块 A 后，接着执行模块 B。

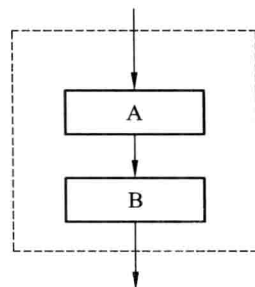
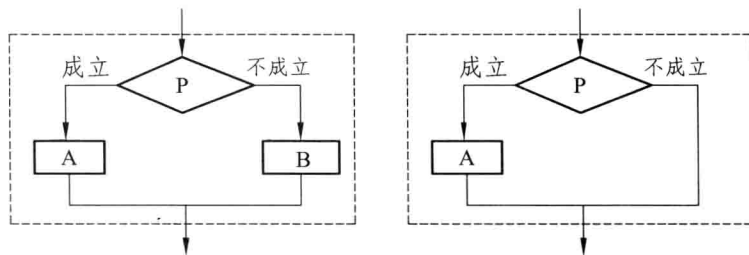


图 1.3 顺序结构

(2) 分支结构又称为选择结构。如图 1.4 (a) 所示，程序在执行过程中，先对条件 P 是否成立进行判断，如果条件 P 成立，执行模块 A。否则，执行模块 B。在分支结构中，只能根据条件 P，选择执行模块 A 和模块 B 中的其中一个模块，完成模块的执行后结束选择。

模块 A 或模块 B 也可以是一个空模块，即不执行任何语句，如图 1.4 (b) 所示。



(a) 分支结构一

(b) 分支结构二

图 1.4 分支结构

(3) 循环结构。如图 1.5 所示，程序在执行过程中，会反复执行某个模块。循环结构有

两种形式。

① 当 (while) 型循环结构。当型循环结构如图 1.5 (a) 所示。其执行过程是：先对条件 P1 是否成立进行判断，如果 P1 成立，执行模块 A，再判断条件 P1 是否成立，如果仍成立，继续执行模块 A，如此反复，直到条件 P1 不成立，立即结束循环。

② 直到 (until) 型循环结构。直到循环结构如图 1.5 (b) 所示。先执行模块 A，再对条件 P2 是否成立进行判断，如果 P2 不成立，继续执行模块 A，再判断条件 P2 是否成立，如果仍不成立，继续执行模块 A，如此反复，直到条件 P2 成立，立即结束循环。

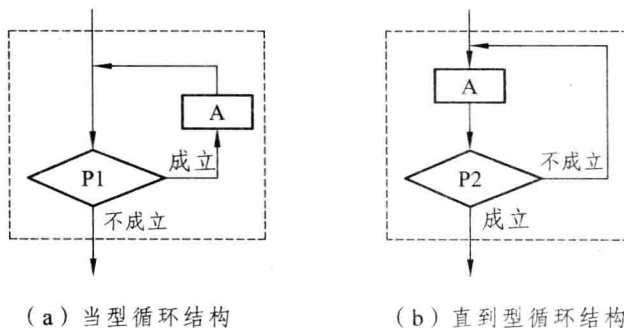


图 1.5 循环结构

三种基本控制结构的共同特点：

- (1) 只有一个入口；
- (2) 只有一个出口；
- (3) 结构内的每一部分都有机会被执行到；
- (4) 结构内不存在“死循环”(不能结束的循环)。

2. 编辑程序

在确定了解决问题的步骤后，就可以编写程序了。一般是在编程环境下直接利用其中的编辑功能编写程序，然后以.c 或.cpp 为扩展名保存源文件，相应程序如下：

```
#include <stdio.h>
void main()
{
    int i,t;                /*定义了两个整型 (int) 变量*/

    t=1;                    /*给变量赋初值 1*/
    i=2;                    /*给变量赋初值 2*/
    while(i<=5)              /*循环语句，表示 i<=5 时，执行下面 {} 所包含的循环语句，直到
                             条件不成立时循环停止*/
    {
        t=t*i;              /*求乘积并用 t 来保存*/
        i=i+1;              /*乘数加 1*/
    }
    printf("5!=%d",t);      /*把乘积即 5!以十进制整数形式输出到显示器屏幕上*/
}
```

3. 程序编译

程序编写完毕后, 接下来就是使用该语言的编译程序编译, 以生成二进制代码表示的目标程序, 一般扩展名为.obj。编译器只对源程序进行语法分析, 如有语法错误, 则在编译结果窗口显示出出错的指令行及出错原因。——修改错误后再编译, 直到编译成功为止。

编译成功, 还需要与编译环境提供的库函数连接 (link), 形成可执行的程序, 扩展名为.exe。

4. 运行与调试

当程序通过了语法检查、编译生成可执行文件后, 就可以在编程环境或操作系统环境中运行 (Run) 该程序。

如果程序中存在语义错误 (编译器只做语法分析, 不做逻辑分析), 程序运行结果有可能不是我们预期的。这时就需要对程序进行调试。调试是在程序中查找错误并修改错误的过程。调试最主要的工作是找出错误所在的地方。一般程序的编程环境都提供有相应的调试手段。调试主要的方法有: 设置断点、单步运行、观察变量等。

(1) 设置断点 (Break Point Setting): 可以在程序的任一个语句设置断点, 程序运行到断点所在语句会停下来。

(2) 单步运行 (Step): 程序单步运行时执行一个语句就停下, 此时我们可以一步一步跟踪程序执行过程, 查看运行的中间结果, 判断程序错误所在。

(3) 观察变量 (Variable Watching): 当程序在某一个语句上暂停下来, 我们可以观察各种变量的值, 判断此时的变量值是否是我们所希望的。如果不是, 说明该语句之前肯定有错误发生。这样, 就可以把找错范围集中在此语句之前的程序段上。

C 语言程序设计需要我们逻辑清晰, 周全地考虑问题, 细心关注每一个细节。调试是一个需要耐心和经验的工作, 也是程序设计最基本的技能。

习 题

1. 对 C 语言来说, 下列标识符中哪些是合法的, 哪些是不合法的?
int, 32_num, a&b, num2, _num, grade, INT, total
2. 画出求 1 ~ 100 的累加和的流程图。
3. 画出求 1 ~ 100 的整数中能被 5 整除的所有整数的和的流程图。