

21世纪应用型本科人才培养规划教材

C YUYAN CHENGXU SHEJI
YINGYONG JIAOCHENG

C 语言程序设计 应用教程

主 编 吴欣茹



西北大学出版社
NORTHWEST UNIVERSITY PRESS

21世纪应用型本科人才培养规划教材

C YUYAN CHENGXU SHEJI
YINGYONG JIAOCHENG

C 语言程序设计 应用教程

主 编 吴欣茹
副 编 牛同壮
参 编 王 飞 王 楠
刘彩红



西北大学出版社
NORTHWEST UNIVERSITY PRESS

内容简介

本书通过大量的例题,深入浅出、循序渐进地讲解了C语言的基本概念和基本语法。详细介绍了C语言的基本数据类型与表达式、3种基本结构、数组、函数、编译预处理、指针、结构体、共用体、枚举、位运算、文件和综合实例等内容。用1~3个实例体现各章应掌握的知识点,且每个实例都给出了实例效果、要点解析及源程序,形成了面向应用技能的特色。本书语言简洁、通俗易懂,适于初学者。可作为各类应用型本科、专科或培训机构的教材,也可作为初学者的自学用书。

图书在版编目(CIP)数据

C语言程序设计应用教程/吴欣茹主编. —西安: 西北大学出版社, 2018. 1

ISBN 978 - 7 - 5604 - 4131 - 3

I. ①C… II. ①吴… 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019232号

C语言程序设计应用教程

主 编: 吴欣茹

出版发行: 西北大学出版社

地 址: 西安市太白北路229号

邮 编: 710069

电 话: 029-88302825

经 销: 全国新华书店

印 装: 陕西向阳印务有限公司

开 本: 787毫米×1092毫米 1/16

印 张: 22.25

字 数: 510千字

版 次: 2018年1月第1版 2018年1月第1次印刷

书 号: ISBN 978 - 7 - 5604 - 4131 - 3

定 价: 58.00元

本版图书如有印装质量问题, 请拨打电话029 - 88302966 予以调换

前 言

C 语言是当今使用最广泛的高级语言之一,是操作系统、编译系统等大型复杂系统的首选语言。C 语言以其灵活方便的特点,成为培养学生计算思维、了解计算机编程思想的最佳语言,已被当今国内外众多高校列为理工科各专业程序设计知识的入门课程。

本教材系陕西省教育厅教学改革研究项目——“独立学院信息类专业学生编程能力培养研究”(项目编号:13BY118)系列成果之一。作者结合多年的教学经验,以实例组织教学的思路编写了本书。本教材具有以下特色。

(1) 语言简洁,图文并茂

本书各章在概念表述、语法讲解和例题讲解等方面力求简洁,难点部分用图表进行讲解。

(2) 注重编码规范

本书详细介绍了程序的排版格式及变量的命名规范。各章实例的源程序编写规范,以此引导读者编写规范的源程序。

(3) 分析引导,流程图规范

本书采用计算思维的方法设计程序,并给出规范的流程图,以此培养读者解决问题的能力。

(4) 无编程基础的读者可自学

本书详细介绍了如何在 Microsoft Visual C++2010 学习版环境下运行 C 语言程序,重点介绍了 C 语言编程基础以及 C 语言编程进阶知识。每章的基本概念和基本语法在各章都以 1~3 个实例体现出来,以此使读者掌握各章的知识点。

全书共分为 3 篇 13 章,第 1~8 章为第 1 篇 C 语言编程基础,主要内容为 C 语言的基本数据类型与表达式、3 种基本结构、数组、函数、编译预处理;第 9~12 章为第 2 篇 C 语言编程进阶,主要内容为指针、结构体、共用体、枚举、位运算、文件;第 13 章为第 3 篇综合实例,演示了一个学生成绩管理系统的详细开发过程。

本书由西安工业大学北方信息工程学院吴欣茹担任主编并负责全书的统稿。编写人员分工如下:第 1 章、第 13 章和附录由西安工业大学牛同壮编写,第 2 章和第 10 章由西安工业大学北方信息工程学院吴欣茹编写,第 3 章、第 6 章和第 12 章由西安工业大学北方信息工程学院王楠编写,第 4 章、第 7 章和第 8 章由西安工业大学北方信息工程学院刘彩虹编写;第 5 章、第 9 章和第 11 章由西安工业大学北方信息工程学院王飞编写。

在本书的编写过程中我们直接或间接地参阅了大量的论著和教材,书中引用的地方没有进行标注,而是采用了书后列出参考文献的方式,在此向有关编者表示衷心地感谢。

担任本书主审的是西安工业大学王建国教授,他在百忙之中对该书进行了审阅,提出了很多宝贵意见,在此我们表示衷心地感谢。

由于时间仓促,加上编者水平所限,书中难免有错误和不足之处,敬请广大师生和读者批评指正。

编 者

2018 年 1 月

目 录

第 1 篇 C 语言编程基础		第 2 章 C 语言的基本数据类型与表达式 /23	
第 1 章 C 语言程序设计概述 /3		2.1 C 语言的基本数据类型 /23	
1.1 程序与程序设计语言 /3		2.1.1 数据类型概述 /23	
1.1.1 程序的基本概念 /3		2.1.2 整型数据 /24	
1.1.2 程序设计语言 /3		2.1.3 实型数据 /25	
1.2 C 语言的发展及特点 /4		2.1.4 字符型数据 /27	
1.2.1 C 语言的发展概况 /4		2.2 变量与常量 /28	
1.2.2 C 语言的特点 /4		2.2.1 变量 /28	
1.3 简单的 C 语言程序 /5		2.2.2 常量 /31	
1.4 C 语言字符集、标识符与关键字 /7		2.3 运算符与表达式 /33	
1.4.1 C 语言字符集 /7		2.3.1 C 语言运算符简介 /33	
1.4.2 系统定义标识符 /7		2.3.2 算术运算符与算术表达式 /34	
1.4.3 用户定义标识符 /8		2.3.3 赋值运算符与赋值表达式 /36	
1.5 Microsoft Visual C ++ 2010 学习版简介 /8		2.3.4 逗号运算符和求字节运算符 /38	
1.5.1 概述 /8		2.4 数据类型转换 /39	
1.5.2 运行 C 语言程序的步骤 /8		2.4.1 类型自动转换 /39	
1.5.3 Microsoft Visual C ++ 2010 学习版的开发环境布局 /15		2.4.2 赋值转换 /40	
1.6 算法及其描述 /16		2.4.3 强制类型转换 /42	
1.6.1 算法的概念 /16		2.5 实例 /43	
1.6.2 算法的描述方法 /17		实例 计算总分和平均分 /43	
1.6.3 3 种基本结构的流程图 /18		2.6 本章小结及易错问题 /44	
1.7 程序的排版格式 /19		2.6.1 本章知识点小结 /44	
1.8 本章小结及易错问题 /21		2.6.2 本章常见错误小结 /45	
1.8.1 本章知识点小结 /21		习题 2 /45	
1.8.2 本章常见错误小结 /21		第 3 章 顺序结构程序设计 /48	
习题 1 /21		3.1 C 语言基本语句 /48	
		3.2 赋值语句 /51	
		3.3 数据的输入/输出语句 /53	

3.3.1 格式的输入与输出	/53	/102
3.3.2 字符的输入与输出	/63		
3.4 实例	/65		
实例 1 求直角三角形的斜边长 及面积	/65		
实例 2 鸡兔同笼	/66		
3.5 本章小结及易错问题	/67		
3.5.1 本章知识点小结	/67		
3.5.2 本章常见错误小结	/67		
习题 3	/68		
第 4 章 选择结构程序设计	/70		
4.1 选择结构程序举例	/70		
4.2 关系运算和逻辑运算	/71		
4.2.1 关系运算符和关系表达式	/71		
4.2.2 逻辑运算符和逻辑表达式	/73		
4.3 if 语句	/74		
4.3.1 单分支 if 语句	/75		
4.3.2 双分支 if 语句	/76		
4.3.3 多分支 if 语句	/77		
4.3.4 if 语句的嵌套	/80		
4.4 switch 语句	/80		
4.5 条件运算符及其表达式	/84		
4.6 实例	/85		
实例 1 求最大值	/85		
实例 2 四则运算器	/89		
4.7 本章小结及易错问题	/91		
4.7.1 本章知识点小结	/91		
4.7.2 本章常见错误小结	/91		
习题 4	/93		
第 5 章 循环结构程序设计	/95		
5.1 循环结构概述	/95		
5.2 while 语句	/95		
5.3 do-while 语句	/99		
5.4 for 语句	/100		
5.5 循环嵌套	/101		
5.5.1 循环嵌套	/101		
5.5.2 循环嵌套的组合形式	/102		
5.6 break 语句与 continue 语句	/104		
5.6.1 break 语句	/105		
5.6.2 continue 语句	/106		
5.7 goto 语句	/107		
5.8 循环结构类型的选择及转换	/108		
5.8.1 循环结构类型的选择 ..	/109		
5.8.2 循环结构的相互转化 ..	/109		
5.9 实例	/112		
实例 1 百钱百鸡问题	/112		
实例 2 九九乘法表的输出	/113		
实例 3 余弦函数 cos(x) 曲线	/114		
5.10 本章小结及易错问题	/116		
5.10.1 本章知识点小结	/116		
5.10.2 本章常见错误小结	/116		
习题 5	/117		
第 6 章 数组	/123		
6.1 一维数组	/123		
6.1.1 一维数组的定义	/123		
6.1.2 一维数组的存储结构 ..	/124		
6.1.3 一维数组的初始化	/124		
6.1.4 一维数组元素的引用 ..	/125		
6.2 二维数组	/126		
6.2.1 二维数组的定义	/126		
6.2.2 二维数组的存储结构 ..	/127		
6.2.3 二维数组的初始化	/127		
6.2.4 二维数组的引用	/128		
6.3 字符数组	/129		
6.3.1 字符数组的定义	/129		
6.3.2 字符数组的初始化	/130		
6.3.3 字符数组的引用	/130		
6.3.4 字符串和字符串结束标志	/132		
6.3.5 字符数组的输入/输出	/133		
6.3.6 字符串处理函数	/135		

6.4 实例	/140	7.10 实例	/173
实例 1 斐波那契数列	/140	实例 小型运算器	/173
实例 2 杨辉三角形	/141	7.11 本章小结及易错问题	/175
6.5 本章小结及易错问题	/142	7.11.1 本章知识点小结	/175
6.5.1 本章知识点小结	/142	7.11.2 本章常见错误小结	/175
6.5.2 本章常见错误小结	/143	习题 7	/176
习题 6	/143	第 8 章 编译预处理	/177
第 7 章 函 数	/147	8.1 编译预处理的特点	/177
7.1 函数的概述	/147	8.2 宏定义	/177
7.2 函数的定义	/149	8.2.1 不带参数的宏定义	/177
7.2.1 无参函数定义的一般形式	/150	8.2.2 带参数的宏定义	/179
7.2.2 有参函数定义的一般形式	/150	8.3 文件包含	/182
7.3 函数的参数和函数的返回值	/151	8.4 本章小结及易错问题	/182
7.3.1 形参和实参	/151	8.4.1 本章知识点小结	/182
7.3.2 函数的返回值	/152	8.4.2 本章常见错误小结	/183
7.4 函数的调用	/153	习题 8	/183
7.4.1 函数调用的一般格式	/153		
7.4.2 函数调用的 3 种表示形式	/157		
7.4.3 被调函数的声明和函数原型	/157		
7.5 函数的嵌套调用	/158		
7.6 函数的递归调用	/160		
7.7 数组作为函数参数	/162		
7.7.1 数组元素作函数实参	/162		
7.7.2 数组名作函数参数	/163		
7.8 局部变量和全局变量	/168		
7.8.1 局部变量	/168		
7.8.2 全局变量	/169		
7.9 变量的存储类别	/170		
7.9.1 动态存储方式与静态存储方式	/170		
7.9.2 auto 变量	/171		
7.9.3 static 声明的局部变量	/171		
7.9.4 register 变量	/172		
		第 2 篇 C 语言编程进阶	
		第 9 章 指 针	/187
		9.1 计算机中的内存	/187
		9.1.1 内存地址	/187
		9.1.2 变量的地址	/187
		9.2 指针与指针变量	/188
		9.3 指针变量的定义和引用	/188
		9.3.1 访问变量的方法	/188
		9.3.2 指针变量的定义	/188
		9.3.3 指针变量的赋值	/189
		9.3.4 零指针与空类型指针	/192
		9.3.5 指针变量的引用	/192
		9.3.6 指针运算	/193
		9.4 指针与数组	/194
		9.4.1 数组的指针	/194
		9.4.2 指向数组的指针	/196
		9.4.3 指向多维数组的指针	/199
		9.5 指针与字符串	/203
		9.5.1 字符串的表示	/203
		9.5.2 字符串的引用	/203
		9.6 多级指针	/205
		9.7 指针与函数	/207

9.7.1 指向函数的指针变量 ... /207 /254
9.7.2 指针函数 /209	10.9 枚举类型 /257
9.7.3 动态内存管理 /210	10.9.1 枚举类型的定义 /258
9.8 实例 /212	10.9.2 枚举类型变量的定义 /258
实例 1 求二维数组中的最大值 /212	10.9.3 枚举类型变量的引用方式 /259
实例 2 梁山好汉排座次 /214	10.10 用 typedef 定义类型 /260
9.9 本章小结及易错问题 /215	10.11 实例 /261
9.9.1 本章知识点小结 /215	实例 1 求平面上两点的距离 /261
9.9.2 本章常见错误小结 /216	实例 2 设计通讯录 /262
习题 9 /216	10.12 本章小结及易错问题 /264
第 10 章 结构体、共用体和枚举 /225	10.12.1 本章知识点小结 /264
10.1 结构体类型的定义 /225	10.12.2 本章常见错误小结 ... /265
10.2 定义结构体类型变量的方法 /226	习题 10 /265
10.3 结构体变量的引用 /228	第 11 章 位运算 /270
10.4 结构体变量的初始化 /229	11.1 位运算概述 /270
10.5 结构体数组 /231	11.2 位逻辑运算符 /270
10.5.1 结构体数组的定义 /231	11.2.1 按位取反运算 /271
10.5.2 结构体数组的初始化 /233	11.2.2 按位与运算 /271
10.6 结构体指针 /236	11.2.3 按位或运算 /271
10.6.1 结构体指针的定义 /236	11.2.4 按位异或运算 /272
10.6.2 结构体指针引用结构体 变量中的成员 /236	11.3 移位运算 /273
10.6.3 结构体指针指向结构体 数组 /238	11.3.1 按位左移运算 /273
10.6.4 结构体指针作函数参数 /240	11.3.2 按位右移运算 /274
10.7 用指针处理链表 /241	11.4 位运算的复合赋值 /275
10.7.1 链表概述 /241	11.5 实例 /275
10.7.2 建立静态链表 /242	实例 1 奇偶判断 /275
10.7.3 建立动态链表 /243	实例 2 两个整数的交换 /276
10.7.4 链表结点的删除 /247	11.6 本章小结及易错问题 /277
10.7.5 链表结点的插入 /250	11.6.1 本章知识点小结 /277
10.8 共用体 /253	11.6.2 本章常见错误小结 /277
10.8.1 共用体的概念 /253	习题 11 /277
10.8.2 共用体变量的定义 /253	第 12 章 文件 /279
10.8.3 共用体变量的引用方式	12.1 文件概述 /279
	12.1.1 文件的分类 /279
	12.1.2 文件缓冲区 /280
	12.2 文件的基本操作 /280

12.2.1 文件指针	/280	13.4.1 主函数	/306
12.2.2 文件的打开与关闭	/280	13.4.2 输入学生记录	/309
12.2.3 文件的读/写	/283	13.4.3 插入学生记录	/312
12.2.4 文件的随机读/写	/292	13.4.4 修改学生记录	/315
12.3 实例	/294	13.4.5 删除学生记录	/316
实例 存储并修改学生成绩	/294	13.4.6 查询学生记录	/318
12.4 本章小结及易错问题	/297	13.4.7 排序	/320
12.4.1 本章知识点小结	/297	13.4.8 显示学生记录	/322
12.4.2 本章常见错误小结	/297	13.4.9 统计学生记录	/323
习题 12	/298	13.4.10 保存学生记录	/325
		13.4.11 其他函数	/326
		13.5 测试	/331
		13.6 小结	/332
第 3 篇 综合实例		附 录	/333
第 13 章 学生成绩管理系统	/303	附录 A 在 Dev - C ++ 环境下运行	
13.1 概述	/303	C 语言程序的方法	/333
13.2 需求分析	/303	附录 B 运算符和结合性	/336
13.3 系统设计	/304	附录 C ASCII 字符表	/337
13.3.1 系统目标	/304	附录 D 常用的 C 语言库函数	/339
13.3.2 开发及运行环境	/304	参考文献	/344
13.3.3 界面设计	/304		
13.3.4 数据结构设计	/305		
13.4 编码实现	/306		

第1篇

C 语言编程基础

第1章 C语言程序设计概述

C语言是当今使用最广泛的高级语言之一,是操作系统、编译系统等大型复杂系统的首选语言。C语言以其灵活方便的特点,成为培养学生计算思维、了解计算机编程思想的最佳语言,已被当今国内外众多高校列为理工科各专业的必修课程。

1.1 程序与程序设计语言

1.1.1 程序的基本概念

所谓程序,实际上是用计算机语言描述的某一问题的解决步骤,是符合一定语法规则的符号序列。打个比方说,计算机程序就像是自然语言的一段话或一篇文章,可以让他人理解并执行。

人们常把编写程序的过程称为程序设计。按照不同种类计算机语言的语法编写的程序,称之为某一种语言的程序。例如,用C语言编写的程序,称为C语言程序。

为解决某一个具体问题所编写的程序并不是唯一的,不同的用户所开发的程序也都不完全相同。不同的程序有不同的效率,这涉及程序的优化、程序所采用的数据结构以及算法等多方面的因素。

1.1.2 程序设计语言

当今程序设计语言的发展非常迅速,新的程序设计语言层出不穷,其功能越来越强大。从其发展历史以及功能看,可以大致划分为以下几个阶段。

(1) 机器语言。机器语言(Machine language)是一种面向机器的程序设计语言,全部由二进制代码0、1组成。用机器语言所设计的程序是一系列的指令,计算机可以直接识别和执行。这种程序称为目标程序(Object program)。手工编写目标程序非常困难,需要编程者熟悉中央处理器(Central Processing Unit,简称CPU)指令系统和它的内部结构。机器语言作为面向机器的语言,在不同类型的处理器之间差别很大,即机器语言程序的可移植性差。

(2) 汇编语言。汇编语言(Assembly language)是一种接近机器语言的符号语言。它将机器语言的指令用便于人们记忆的符号来表示,通过这种语言系统所带的翻译程序翻译成目标程序后再执行。汇编语言程序执行效率很高,目前在实时控制等方面的编程中仍有不少应用。

(3) 高级语言。机器语言和汇编语言都是面向机器的语言,一般称为低级语言。它们对机器的依赖性很大,用它们开发出来的程序通用性差,且要求程序开发者必须熟悉和了解计算机硬件的每个细节,因此它们面向的用户是计算机专业人员。

高级语言(High-level language)是一种完全符号化的语言,通常采用自然语言(英语)中的词汇和语法习惯,容易被人们理解和掌握;它完全独立于具体的计算机,具有很强的可移

植性。用高级语言编写的程序称为源程序(Source program),源程序不能在计算机上直接执行,必须将它翻译或解释成目标程序后,才能被计算机所理解和执行。将源程序翻译成目标程序,其翻译过程有两种方式:一种是解释方式,另一种是编译方式。把源程序翻译成目标程序的软件称为编译程序。每一种高级语言都有与其相对应的编译程序,C语言也不例外。与低级语言相比,用高级语言编写的程序,其执行的时间和空间效率要差一些。

随着 Windows 操作系统的普遍应用,程序设计语言已经发生了很大的变化,除逐步采用可视化、图形化的编程环境,大量采用各种程序设计工具外,重要的是引入了“面向对象的程序设计”思想。从程序设计的理念、编程的思维方式直到程序设计的具体方法都发生了变化,使编程者面对的不再是繁杂的语句和众多的过程,而是层次关系明确的各种对象,只要适当地描述对象的某些属性,指定对象的某些事件方法就可以完成原来需要使用大量代码才能完成的工作。例如,目前流行的 Visual C++、Java 等都采用了面向对象的程序设计思想和方法。

1.2 C 语言的发展及特点

1.2.1 C 语言的发展概况

C 语言是国际上广泛流行的计算机高级语言之一。它既可作为系统软件描述语言,也可用来开发应用软件。

C 语言产生于 20 世纪 70 年代初。最初版本的 C 语言是 1972 年由美国贝尔实验室的 Dennis M. Ritchie 设计的,其目的只是为描述和实现 UNIX 操作系统而提供的一种工作语言。后来经过多次改进,直到 1978 年,贝尔实验室的 Brian W. Kernighan 和 Dennis M. Ritchie(合称 K&R)合著了影响深远的名著《The C Programming Language》,这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础,它被称为标准 C。

1983 年,美国国家标准化协会(ANSI)根据 C 语言问世以来的各种版本,对 C 语言的发展和扩充制定了新的标准,称为 ANSI C。1989 年,ANSI 又公布了新标准——89ANSI C。1990 年,国际标准化组织 ISO(International Organization for Standards)接受了 89 ANSI C 为 ISO C 的标准,俗称 C89 标准。1999 年,ISO 对 C 语言标准进行修订,在基本保留原来 C 语言特征的基础上,增加了一些功能,命名为 ISO/IEC 9899:1999,俗称 C99 标准。2011 年,ISO 正式公布 C 语言新的国际标准,命名为 ISO/IEC 9899:2011,俗称 C11 标准。目前,流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的,但不同版本的 C 语言编译系统所实现的语言功能和语法规则略有差别。

1.2.2 C 语言的特点

C 语言是一种极具生命力的、通用的结构化语言,具有多方面的特点。其主要特点如下。

(1)语言简洁、紧凑,使用方便、灵活。它压缩了一切不必要的成分,便于阅读和书写,因此输入程序时工作量较小。

(2)运算符丰富。C 语言把赋值、括号、强制类型转换等都作为运算符处理,从而使 C 语言的运算类型极其丰富,表达式类型多样化。因此,C 语言具有较强的数据处理能力,不仅能完成一般高级语言的运算功能,而且还能实现许多汇编语言的功能。

(3)数据类型丰富。数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据结构(如链表、树等)的运算。

(4)程序层次结构清晰。C 语言具有结构化的流程控制语句(如 if-else 语句、switch 语句、while 语句、do-while 语句、for 语句),支持若干种循环结构,允许编程者采用缩进的书写形式编程。因此,用 C 语言编写的程序层次结构清晰。

(5)主要结构成分为函数。函数可以在程序中被定义完成独立的任务,独立地编译成代码,以实现程序的模块化。

(6)允许直接访问物理地址,能进行位(bit)操作。C 语言可直接对硬件进行操作,实现汇编语言的大部分功能。因此,C 语言成为编写系统软件的基本语言(UNIX 的绝大部分程序就是由 C 语言编写的)。

(7)生成的目标代码质量高,程序执行效率高。

(8)与汇编语言相比,用 C 语言编写的程序可移植性好,基本上不做修改就能用于各种型号的计算机和各种操作系统。

(9)语法限制不太严格,程序设计自由度高,允许程序编写者有较大的自由度,放宽了语法检查。

1.3 简单的 C 语言程序

在学习 C 语言的具体语法之前,先通过两个简单的 C 语言程序示例,初步了解 C 语言程序的基本结构。

例 1.1 要求在屏幕上输出以下一行信息。

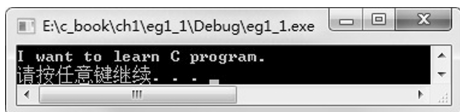
I want to learn C program.

编写程序:

创建一个名为“eg1_1.c”的新文件,在编辑窗口中输入下面的程序代码。

```
/* C 语言程序的典型结构 */
#include <stdio.h >           // 这是编译预处理命令
#include <stdlib.h >         // 这是编译预处理命令
int main()                   // 定义主函数
{                             // 函数开始的标志
    printf("I want to learn C program. \n"); // 调用 printf()函数打印一行字符
    system("pause");         // 调用 system()函数,实现屏幕暂停
    return 0;                // 返回函数值 0
}                             // 函数结束的标志
```

运行结果:



程序说明:

(1)“#include <stdio.h >”是编译预处理命令,其含义是将“stdio.h”头文件包含在本文件中。源文件中需要用到输入和输出函数时,只需将对应的标准的输入/输出头文件

“stdio. h”用“#include <stdio. h >”命令包含进来并放在程序的首部就可直接使用,头文件的扩展名一般为“. h”。

(2)“#include <stdlib. h >”也是编译预处理命令,其含义是将“stdlib. h”头文件包含在本文件中。

(3)main()是主函数。main 是函数的名字,表示主函数,main 前面的 int 表示此函数的返回值是 int 类型(整型)。每一个 C 语言程序都必须有一个 main()函数,main 是 C 语言编译系统使用的专用名字。main()后面由大括号“{}”括起来的部分是函数体,程序从main()函数的第一条可执行语句开始执行。

(4)注释方式。

1)以“/*”开始,以“*/”结束的块式注释。“/*”与“*/”之间的内容可以是一行,也可以是多行。

2)以“//”开始的单行注释。这种注释可以单独占一行,也可以出现在一行中代码的右侧,以换行符结束。

注释部分不参与程序的编译和执行,只起说明作用,增加程序的可读性。

(5)“printf(“I want to learn C program. \n”);”是一条语句。每条语句的最后必须有一个分号“;”。分号是 C 语言语句的必要组成部分。“\n”是换行符,即在输出“I want to learn C program.”后,显示屏上的光标位置移到下一行的开头。C 语言本身没有输入/输出语句。输入和输出的操作是由库函数 scanf()和 printf()等函数来完成的。

(6)“system(“pause”);”语句。该语句可以实现屏幕暂停,便于观察程序的执行结果。函数 system()的定义包含在“stdlib. h”头文件中,因此程序的首部需有“#include <stdlib. h >”命令。

(7)“return 0;”语句。当主函数正常结束时,得到的函数值为 0。当执行 main()函数过程中出现异常或错误时,函数值为一个非 0 的整数。这个函数值返回给调用 main()函数的操作系统。程序员通过检查 main()函数的返回值,判断 main()函数是否正常执行。

例 1.2 求两个整数之和。

编写程序:

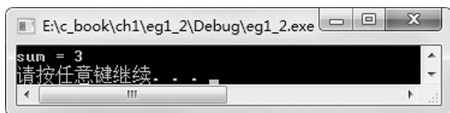
创建一个名为“eg1_2. c”的新文件,在编辑窗口中输入下面的程序代码。

```

/* 求两个整数之和 */
#include <stdio. h >
#include <stdlib. h >
int main()
{
    int a, b, sum;           // 定义 3 个整型变量
    a = 1;                  // 给变量 a 赋值
    b = 2;                  // 给变量 b 赋值
    sum = a + b;           // a 与 b 相加后的结果赋值给变量 sum
    printf(“sum = %d\n”, sum); // 输出结果
    system(“pause”);
    return 0;
}

```

运行结果:



程序说明:

(1) `main()` 后面由大括号“{}”括起来的部分是函数体,函数体一般包括:

1) 变量定义。例如,本例中的“`int a, b, sum;`”。

2) 执行部分。由若干个语句组成。例如,本例中大括号“{}”括起来的部分除去“`int a, b, sum;`”后的其余语句。

(2) “`a = 1;`”“`b = 2;`”是两个赋值语句。

(3) “`sum = a + b;`”是一条语句。功能是将 `a` 与 `b` 相加后的结果赋值给变量 `sum`。

(4) “`printf("sum = %d\n", sum);`”是一条语句。输出操作是由库函数 `printf()` 函数来完成的。

1.4 C语言字符集、标识符与关键字

标识符是给程序中的实体——变量、常量、函数、数组、结构体以及文件等所起的名字。C语言中的标识符分为系统定义标识符和用户定义标识符。

1.4.1 C语言字符集

任何一个计算机系统所能使用的字符都是固定的、有限的,它受硬件设备的限制。使用某种计算机语言编写程序,就必须使用符合该语言规定的且计算机系统能够使用的字符。

C语言的基本字符包括英文字母、阿拉伯数字以及其他一些符号,具体归纳如下。

(1) 英文字母大小写各 26 个,共计 52 个。

(2) 阿拉伯数字 0~9,共计 10 个。

(3) 下划线“_”。

(4) 其他特殊符号主要指运算符,通常由 1~2 个特殊符号组成,特殊符号集如下。

+	-	*	/	%	++	--
<	>	=	>=	<=	==	!=
!		&&	^	~		&
<<	>>	()	[]	{}	\	"
?	:	.	,	;	'	

1.4.2 系统定义标识符

系统定义标识符是指具有固定名字和特定含义的符号。系统定义标识符分为关键字和预定义标识符两种类型。

1. 关键字

关键字是 C 语言中规定具有特定含义的标识符。根据 C99 标准,C语言可使用 37 个关键字,分为以下几类。

(1) 数据类型。`int`, `long`, `short`, `char`, `float`, `double`, `signed`, `unsigned`, `struct`, `union`, `enum`, `void`, `volatile`, `const`, `typedef`, `restrict`, `inline`, `_Bool`, `_Complex`, `_Imaginary`。