



工业和信息化“十三五”
人才培养规划教材

NUTE

国家软件与集成电路公共服务平台
信息技术紧缺人才培养工程 指定教材



有问题，就找问答精灵！



C 语言开发 基础教程 (Dev-C++ | 第2版)

黑马程序员 • 编著

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

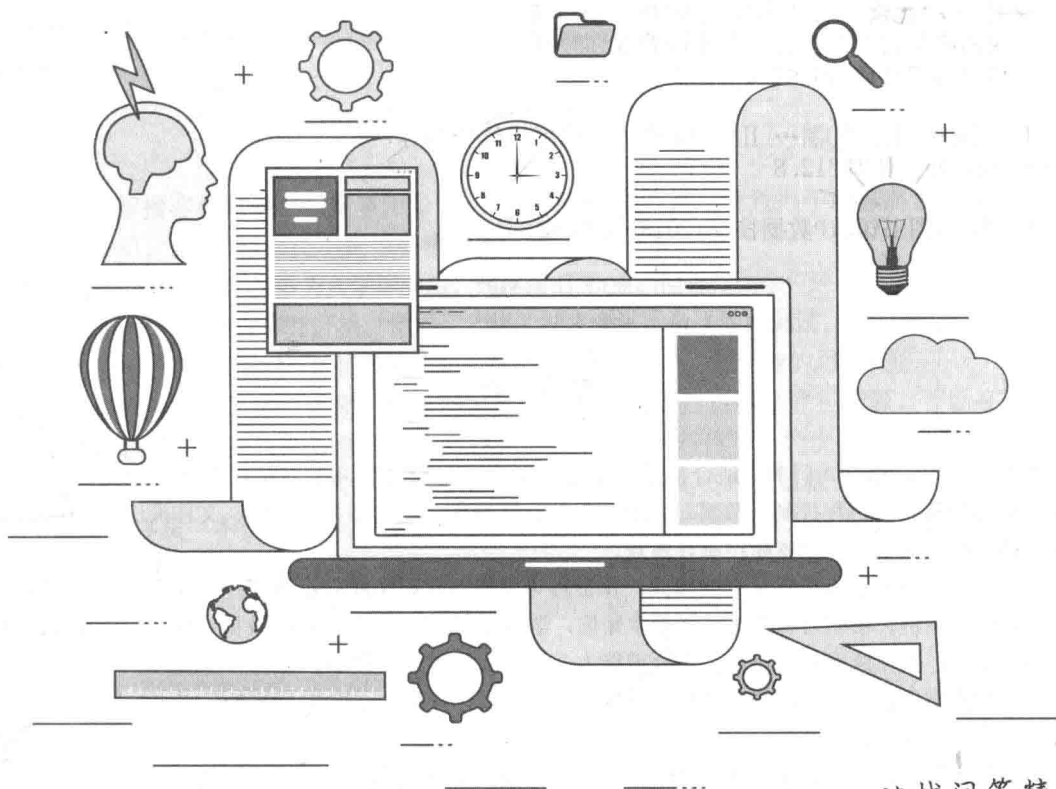


工业和信息化“十三五”
人才培养规划教材

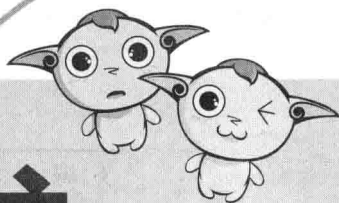
NITE

国家软件与集成电路公共服务平台
信息技术紧缺人才培养工程

指定教材



有问题，就找问答精灵！



C语言开发 基础教程 (Dev-C++ | 第2版)

黑马程序员 ● 编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

C语言开发基础教程 (Dev-C++) / 黑马程序员编著

— 2版. — 北京: 人民邮电出版社, 2019. 5

工业和信息化“十三五”人才培养规划教材

ISBN 978-7-115-50202-5

I. ①C… II. ①黑… 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288912号

内 容 提 要

本书是一本C语言入门书,适合初学者使用。全书共分12章,内容包括程序设计与C语言、数据类型与运算符、流程控制、数组、函数、指针、字符串、结构体、预处理、文件操作、常见的数据结构、综合项目——贪吃蛇控制台游戏。

全书采用理论与示例代码结合的模式,保证理论学习与代码实践同步进行;根据学习进度在章节中穿插阶段案例,帮助读者系统掌握所学知识,学以致用;第11章介绍的数据结构,旨在拓展读者的编程思维,强化读者的数据组织与处理能力,为后续学习做好铺垫。

本书的大纲以前一版《C语言开发入门教程》为基础,并参考了市面上多本C语言教程,力求构造完整的知识体系。

本书附有配套的教学PPT、题库(600道)、教学视频、源代码、教学设计等资源。同时,为了帮助初学者及时地解决学习过程中遇到的问题,传智播客还专门提供了免费的在线答疑平台,并承诺3小时内给予解答。

本书可作为高等院校本、专科计算机相关专业,以及其他理工科专业的程序设计入门教材。

◆ 编 著 黑马程序员

责任编辑 范博涛

责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京圣夫亚美印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 15

2019年5月第2版

字数: 368千字

2019年5月北京第1次印刷

定价: 49.80元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147号

江苏传智播客教育科技股份有限公司（简称传智播客）是一家致力于培养高素质软件开发人才的科技公司，“黑马程序员”是传智播客旗下高端IT教育品牌。

“黑马程序员”的学员多为大学毕业后，想从事IT行业，但各方面条件还不成熟的年轻人。“黑马程序员”的学员筛选制度非常严格，包括了严格的技术测试、自学能力测试，还包括性格测试、压力测试、品德测试等。百里挑一的残酷筛选制度确保学员质量，并降低企业的用人风险。

自“黑马程序员”成立以来，教学研发团队一直致力于打造精品课程资源，不断在产、学、研3个层面创新自己的执教理念与教学方针，并集中“黑马程序员”的优势力量，针对性地出版了计算机系列教材80多册，制作教学视频数十套，发表各类技术文章数百篇。

“黑马程序员”不仅斥资研发IT系列教材，还为高校师生提供以下配套学习资源与服务。

为大学生提供的配套服务

1. 请同学们登录在线平台 <http://yx.ityxb.com>，进入“高校学习平台”，免费获取海量学习资源。帮助高校学生解决学习问题。
2. 针对高校学生在学习过程中存在的压力等问题，我们还面向大学生量身打造了IT技术女神——“播妞学姐”，可提供教材配套源码、习题答案以及更多学习资源。同学们快来关注“播妞学姐”的微信公众号 [boniu1024](https://www.weixin.com/boniu1024)。



“播妞学姐”微信公众号

为教师提供的配套服务

针对高校教学，“黑马程序员”为IT系列教材精心设计了“教案+授课资源+考试系统+题库+教学辅助案例”的系列教学资源，高校老师请登录在线平台 <http://yx.ityxb.com> 进入“高校教辅平台”或关注码大牛老师微信/QQ：2011168841，获取配套资源，也可以扫描下方二维码，加入专为IT教师打造的师资服务平台——“教学好助手”，获取最新的教学辅助资源。



“教学好助手”微信公众号

随着互联网的发展，人类生活方式中的重要部分，包括衣、食、住、行、教育、娱乐甚至医疗等都在线上得以应用，这一切的便利离不开互联网的发展，也与形形色色的应用软件密不可分。开发应用软件需要使用编程语言，作为最古老的编程语言之一，C 语言因具有简洁、紧凑、高效灵活、可直接访问硬件、可移植等特性，而被应用于程序开发的众多领域。

为什么要学习本书

C 语言是众多院校计算机专业学习的第一门编程语言；为了一窥程序的编写原理，了解互联网时代发展的基础，许多工科专业也将 C 语言作为学生的必修课程。作为一本入门教程，本书站在初学者的角度，先对 C 语言的基础知识进行了详细的讲解，将复杂问题简单化，之后以代码形式实践基础知识，最后结合讲解对代码进行分析，真正做到了由浅入深、由易到难。

本书是《C 语言开发入门教程》的全新改版。本次改版吸取了广大读者 4 年来的真实反馈，与第 1 版教程相比，本书具有以下亮点。

1. 采用更加便捷、小巧的开发工具 Dev-C++ 作为教学环境，降低下载、安装和使用工具的难度。
2. 对原书中的代码进行精简，并添加了大量进阶案例，既能保证读者充分理解、吸收所讲内容，又能帮助读者巩固所学知识、提高编程能力。
3. 语言描述更加精炼、合乎逻辑，通俗易懂；内容安排更加合理，体系结构更加完善。

如何使用本书

本书是一本 C 语言入门书，内容包含 12 章，其中第 1~10 章介绍 C 语言基础语法，第 11 章介绍数据结构，第 12 章为综合项目，具体介绍如下。

第 1 章首先简单介绍了计算机语言、算法等与程序设计相关的知识，其次介绍了 C 语言的发展史、标准和应用领域，之后介绍了几种 C 语言开发工具，讲解了 Dev-C++ 的安装流程，并结合案例展示了该工具的基础用法与 C 语言的编写流程，最后讲解了编译过程。通过本章的学习，读者会对计算机语言、程序设计、算法、C 语言等概念有所了解，并能自主安装 Dev-C++ 工具，熟悉程序的编写流程，了解程序的编译过程。

第 2 章主要讲解 C 语言中的数据类型与运算符。其中数据类型包括基本数据类型、构造类型、指针类型；运算符包括算术运算符、关系运算符、逻辑运算符、赋值运算符、条件运算符、位运算符和 sizeof 运算符。除此之外，本章还介绍了与数据类型相关的关键字、标识符、常量、变量，以及类型转换和运算符优先级等知识。通过本章的学习，读者可以掌握 C 语言中数据类型及其运算的相关知识。熟练掌握本章的内容，可以为后面的学习打下坚实的基础。

第 3 章首先讲解程序的运行流程图，然后讲解 C 语言中最基本的 3 种流程控制语句，包括

顺序结构语句、选择结构语句和循环语句，之后介绍了循环嵌套和跳转语句。通过本章的学习，读者能够熟练地运用 if 判断语句、switch 判断语句、while 循环语句、do...while 循环语句及 for 循环语句。

第 4 章首先讲解什么是数组，其次讲解一维数组的定义、初始化、引用，以及数组的常见操作，之后讲解二维数组的相关知识，最后简单介绍多维数组的定义方式。掌握好本章的内容有助于后面课程的学习。

第 5 章主要讲解 C 语言中的函数，包括函数的定义和声明、函数的调用；其次介绍局部变量、全局变量及变量的作用域等知识。通过本章的学习，读者能掌握模块化思想，熟练封装功能代码，并以函数名实参列表的形式进行调用，从而简化代码，提高代码的可读性。

第 6 章首先讲解指针的概念与指针的运算，然后讲解指针与数组、指针与函数、指针数组的相关知识，最后讲解了二级指针、指针与 const 的相关知识。通过本章的学习，读者能掌握指针的定义与使用方法，学会使用指针优化代码，提高代码的灵活性。

第 7 章首先讲解 C 语言中字符数组、字符串的概念，以及字符串与指针的关系，然后讲解字符串的输入/输出，之后讲解字符串常用的操作函数，最后讲解数字与字符串之间的转换。通过本章的学习，读者能熟练掌握字符串的常用操作。

第 8 章首先介绍构造类型中的结构体类型，包括结构体类型的声明、定义、初始化、访问、大小等知识，其次介绍结构体数组、结构体与指针、结构体与函数等进阶内容，之后介绍 typedef 关键字的应用，最后通过阶段案例帮助读者巩固本章内容。通过本章的学习，读者可以掌握结构体的存储结构，并能熟练应用结构体。

第 9 章主要讲解预处理与断言。常用的预处理方式有 3 种，分别是宏定义、文件包含和条件编译。其中，宏定义是最常用的一种预处理方式，文件包含对于程序功能的扩充很有帮助；条件编译可以优化程序代码。断言用于检测假设的条件是否成立，对程序调试非常有帮助。熟练掌握程序预处理方式和断言，对于以后的程序设计工作至关重要。

第 10 章首先讲解文件的基本概念，包括流、文件、文件指针与文件位置指针；然后讲解文件的基本操作，包括文件的打开与关闭、单字符读写文件、单行读写文件、二进制形式读写文件；之后讲解文件的随机读写；最后通过一个案例来加深读者对文件读写的理解。通过本章的学习，读者将学会对文件进行读写操作，从而站在更高的层面来理解和使用文件。

第 11 章主要讲解 C 语言中的 3 种数据结构，分别是链表、栈和队列。通过本章的学习，读者能够掌握这 3 种数据结构的存储原理、定义及常用操作，并熟练运用这些数据结构优化程序中的数据存贮，提高程序的运行效率。

第 12 章运用前面各章所讲知识实现一个综合项目，并分别介绍项目开发过程中的需求分析、模块设计、代码实现、代码调试等环节，不仅帮助读者温习所学知识，更能引领读者了解程序开发流程，巩固程序设计思想。

在学习的过程中，读者应勤思考、勤总结，并自主实践中提供的案例。

读者若不能完全理解书中所讲知识，可登录博学谷平台，配合平台中的教学视频进行学习。此外读者在学习的过程中，务必要勤于练习，确保真正掌握所学知识。在学习过程中，读者如果遇到困难，不要纠结，继续往后学习，或许会豁然开朗。

致谢

本书的编写和整理工作由传智播客公司完成，主要参与人员有吕春林、高美云、薛蒙蒙、郑

瑶瑶、李卓等，全体人员在近一年的编写过程中付出了辛勤的汗水，在此一并表示衷心的感谢。

意见反馈

尽管我们付出了很大的努力，但书中难免会有不妥之处，欢迎读者来信给予宝贵意见，我们将不胜感激。电子邮件：itcast_book@vip.sina.com。

黑马程序员

2018年11月于北京

专属于老师及学生的在线教育平台
<http://yx.ityxb.com/>

让 IT 教学更简单

教师获取教材配套资源

教案

搜索讲义

考试系统

在线题库

教学辅助
案例

添加微信/QQ

2011168841

让 IT 学习更有效

学生获取配套源码

关注微信公众号“播妞学姐”
 获取教材配套源码



专属大学生的圈子

第 1 章 程序设计与 C 语言..... 1

- 1.1 计算机语言 1
- 1.2 程序设计与算法 2
 - 1.2.1 算法——程序的灵魂..... 2
 - 1.2.2 算法的表示 3
- 1.3 C 语言概述 4
 - 1.3.1 C 语言的发展史..... 5
 - 1.3.2 C 语言的标准..... 5
 - 1.3.3 C 语言的应用领域..... 5
- 1.4 开发环境搭建 7
 - 1.4.1 主流开发工具介绍..... 7
 - 1.4.2 安装 Dev-C++ 8
- 1.5 编写 Hello World 10
 - 1.5.1 编写第一个程序 Hello World 10
 - 1.5.2 C 程序编译运行原理..... 13
- 1.6 阶段案例——我的名片 14
- 1.7 本章小结..... 14
- 1.8 习题..... 14

第 2 章 数据类型与运算符..... 16

- 2.1 关键字和标识符 16
 - 2.1.1 关键字 16
 - 2.1.2 标识符 17
- 2.2 常量与变量 18
 - 2.2.1 常量 18
 - 2.2.2 变量 18
- 2.3 数据类型..... 19
 - 2.3.1 基本类型 20
 - 2.3.2 构造类型 24
 - 2.3.3 指针类型 26
 - 2.3.4 类型转换 27
- 2.4 运算符与表达式..... 28

2.4.1	运算符与表达式的概念.....	28	4.3	阶段案例——双色球.....	63
2.4.2	算术运算符与算术表达式.....	29	4.4	二维数组.....	64
2.4.3	关系运算符与关系表达式.....	31	4.4.1	二维数组定义与初始化.....	64
2.4.4	逻辑运算符与逻辑表达式.....	31	4.4.2	二维数组的访问.....	66
2.4.5	赋值运算符与赋值表达式.....	33	4.5	阶段案例——杨辉三角.....	67
2.4.6	条件运算符与条件表达式.....	33	4.6	多维数组.....	68
2.4.7	位运算符.....	34	4.7	本章小结.....	69
2.4.8	sizeof 运算符.....	36	4.8	习题.....	69
2.5	运算优先级.....	36			
2.6	阶段案例——加密.....	38	第5章 函数.....	71	
2.7	阶段案例——数字反转.....	38	5.1	初识函数.....	71
2.8	本章小结.....	39	5.1.1	函数的定义.....	71
2.9	习题.....	39	5.1.2	格式化输入/输出.....	75
			5.2	函数调用.....	76
第3章 流程控制.....	41		5.2.1	函数调用方式.....	76
3.1	程序流程图.....	41	5.2.2	嵌套调用.....	77
3.2	顺序结构.....	42	5.2.3	递归调用.....	78
3.3	选择结构.....	42	5.3	阶段案例——汉诺塔.....	79
3.3.1	if 条件语句.....	42	5.4	外部函数与内部函数.....	80
3.3.2	switch 条件语句.....	46	5.4.1	外部函数.....	80
3.4	阶段案例——自动贩卖机.....	48	5.4.2	内部函数.....	81
3.5	循环结构.....	48	5.5	局部变量与全局变量.....	82
3.5.1	while 循环.....	48	5.5.1	局部变量.....	82
3.5.2	do...while 循环.....	49	5.5.2	全局变量.....	83
3.5.3	for 循环.....	50	5.6	认识静态库与动态库.....	84
3.5.4	循环嵌套.....	51	5.7	阶段案例——体测成绩判定.....	85
3.5.5	跳转语句.....	52	5.8	本章小结.....	87
3.6	阶段案例——薪水问题.....	53	5.9	习题.....	87
3.7	本章小结.....	54			
3.8	习题.....	54	第6章 指针.....	89	
第4章 数组.....	57		6.1	指针的概念.....	89
4.1	什么是数组.....	57	6.2	指针运算.....	91
4.2	一维数组.....	58	6.2.1	取址运算符.....	91
4.2.1	一维数组的定义与初始化.....	58	6.2.2	取值运算符.....	91
4.2.2	一维数组的访问.....	59	6.2.3	常用指针运算.....	92
4.2.3	数组元素排序.....	60	6.3	指针与数组.....	93

6.3.1 指针与一维数组	94	8.1.2 结构体变量定义.....	129
6.3.2 指针与二维数组	95	8.1.3 结构体变量的大小.....	130
6.4 阶段案例——幻方	97	8.1.4 结构体变量初始化.....	132
6.5 指针与函数	99	8.1.5 结构体变量访问.....	132
6.5.1 指针变量作为函数参数	99	8.2 结构体数组	133
6.5.2 函数指针	101	8.2.1 结构体数组的定义.....	133
6.5.3 回调函数	103	8.2.2 结构体数组的初始化.....	133
6.6 指针数组.....	104	8.2.3 结构体数组的访问.....	134
6.6.1 定义指针数组	104	8.3 结构体与指针	134
6.6.2 指针数组的应用	105	8.3.1 结构体指针	135
6.7 二级指针	107	8.3.2 结构体数组指针.....	135
6.8 阶段案例——天生棋局	108	8.4 结构体与函数	136
6.9 指针与 const.....	109	8.4.1 结构体变量作为函数参数.....	136
6.10 本章小结.....	110	8.4.2 结构体数组作为函数参数.....	137
6.11 习题	110	8.4.3 结构体指针作为函数参数.....	138
第 7 章 字符串.....	113	8.5 typedef 的使用	138
7.1 字符数组和字符串	113	8.6 阶段案例——学生成绩 管理系统.....	139
7.1.1 字符数组	113	8.7 本章小结.....	141
7.1.2 字符串	114	8.8 习题	141
7.1.3 字符串与指针	116	第 9 章 预处理	144
7.2 字符串的输入/输出	118	9.1 宏定义	144
7.2.1 常见的输入函数	118	9.1.1 不带参数的宏定义.....	144
7.2.2 常见的输出函数	119	9.1.2 带参数的宏定义.....	146
7.3 字符串操作函数	120	9.2 文件包含.....	147
7.3.1 字符串比较	120	9.2.1 文件包含命令的格式.....	147
7.3.2 字符串查找	121	9.2.2 多文件包含实例.....	148
7.3.3 字符串连接	122	9.3 条件编译.....	149
7.3.4 字符串复制	123	9.3.1 #if/#else/#endif 指令	149
7.4 数字与字符串转换	124	9.3.2 #ifdef 指令	150
7.5 阶段案例——回文字符串	125	9.3.3 #ifndef 指令	150
7.6 本章小结.....	126	9.4 断言	154
7.7 习题	126	9.4.1 断言的作用	154
第 8 章 结构体.....	128	9.4.2 断言与 debug.....	155
8.1 结构体类型	128	9.5 本章小结.....	155
8.1.1 结构体类型声明	128	9.6 习题	155

第 10 章 文件操作	157		
10.1 文件概述	157		
10.1.1 流	157		
10.1.2 文件	158		
10.1.3 文件指针	159		
10.1.4 文件位置指针	160		
10.2 文件的打开和关闭	161		
10.3 文件的读写	162		
10.3.1 单字符读写文件	162		
10.3.2 单行读写文件	164		
10.3.3 二进制形式读写文件	165		
10.4 阶段案例——文件加密	166		
10.5 文件随机读写	167		
10.6 阶段案例——个人日记本	168		
10.7 本章小结	169		
10.8 习题	169		
第 11 章 常见的数据结构	172		
11.1 链表	172		
11.1.1 链表概述	172		
11.1.2 链表的结构	173		
11.1.3 链表的实现	173		
11.2 栈	177		
11.2.1 什么是栈	178		
11.2.2 栈的链式存储与实现	178		
11.3 队列	181		
11.3.1 什么是队列	182		
11.3.2 链式队列的存储与实现	182		
11.4 阶段案例——机器运算	186		
11.5 本章小结	188		
11.6 习题	188		
第 12 章 综合项目——贪吃蛇 控制台游戏	190		
12.1 项目分析	190		
		12.1.1 项目需求分析	190
		12.1.2 效果显示	193
		12.1.3 项目设计	195
		12.2 项目实施	197
		12.2.1 项目创建	197
		12.2.2 snake.h 文件定义	199
		12.2.3 界面管理模块的实现	200
		12.2.4 贪吃蛇初始化模块的实现	203
		12.2.5 食物模块的实现	204
		12.2.6 游戏规则设计模块的实现	205
		12.2.7 贪吃蛇移动控制模块的实现	207
		12.2.8 信息管理模块的实现	212
		12.2.9 main()函数实现	214
		12.3 程序调试	215
		12.3.1 设置断点	215
		12.3.2 单步调试	216
		12.3.3 观察变量	217
		12.3.4 条件判断	217
		12.3.5 项目调试	218
		12.4 项目心得	220
		12.5 本章小结	220
		附录 I 二进制与十进制对应 关系表	221
		附录 II 八进制与十进制对应 关系表	222
		附录 III 十六进制与十进制对应 关系表	223
		附录 IV 二进制与八进制对应 关系表	224
		附录 V 二进制与十六进制对应 关系表	225
		附录 VI ASCII 码表	226
		附录 VII C 语言常用的字符串操作 函数	227

1

Chapter

第 1 章

程序设计与 C 语言

学习目标

- 了解计算机语言的特点
- 了解算法在程序设计中的重要性
- 了解 C 语言的发展史、标准及应用领域
- 了解主流的开发工具，能够独立安装 Dev-C++ 工具
- 会编写 Hello World 程序，了解程序编译的过程

C 语言是一种通用的、过程式的编程语言，它具有高效、灵活、可移植等优点。在最近 20 多年里，它被运用在各种系统软件与应用软件的开发中，是使用最广泛的编程语言之一。本章作为整本书的第 1 章，将对 C 语言的发展历史、开发环境搭建以及 C 语言程序编写方法等内容进行详细的讲解。

1.1 计算机语言

计算机语言 (Computer Language) 是人与计算机之间通信的语言，它主要由一些指令组成，这些指令包括数字、符号和语法等内容，编程人员可以通过这些指令来指挥计算机进行各种工作。

计算机语言有很多，根据不同的功能和实现方式可分为 3 类，即机器语言、汇编语言和高级语言，下面分别介绍这 3 类语言的特点。

1. 机器语言

机器语言是能够被计算机直接识别的语言，由二进制数 0 或 1 组成的一串指令集合，是计算机处理器可直接解读的数据。对于编程人员来说，机器语言不便于记忆和识别。

2. 汇编语言

人们很早就认识到这样的一个事实，尽管机器语言对计算机来说很好懂也很好用，但是对于编程人员来说，记住 0 和 1 组成的指令简直就是煎熬。为了解决这个问题，汇编语言诞生了。汇编语言用英文字母或符号串来替代机器语言，把不易理解和记忆的机器语言按照对应关系转换成汇编指令，因此汇编语言比机器语言更易于阅读和理解。

3. 高级语言

由于汇编语言依赖于硬件,因此汇编程序的可移植性极差,而且编程人员在使用计算机时需要学习新的汇编指令,大大增加了编程人员的工作量,为此计算机高级语言诞生了。高级语言不是一门语言,而是一类语言的统称,它比汇编语言更贴近于人类使用的语言,也更易于理解、记忆和使用。此外高级语言和计算机的架构、指令集无关,因此它具有良好的可移植性。

高级语言应用非常广泛,世界上绝大多数的编程人员都在使用高级语言进行程序开发。常见的高级语言包括 C、C++、Java、C#、Python、Ruby 等。本书讲解的 C 语言从诞生到现在一直都是最流行、应用场景最丰富的高级语言之一。

1.2 程序设计与算法

算法在计算机科学领域是非常重要的概念,有着举足轻重的地位。算法将要解决的实际问题和解决实际问题的计算机程序联系起来。在编写程序的过程中,不可避免地要考虑到算法设计方案。本节内容介绍算法的基本概念和特征,引领读者了解算法在程序设计中的重要性。

1.2.1 算法——程序的灵魂

生活中算法随处可见,同样计算机程序设计也离不开算法,在计算机学科中算法是独立课程,开始编程之前了解算法的基本知识,对后续学习编程有着重要的意义。

1. 什么是算法

从广义上讲算法就是解决问题的方法和过程,在计算机领域,算法是从输入到输出的有穷序列,是一系列解决问题的清晰指令。

计算机程序通常具备两个方面的描述:一是对数据的描述;二是对程序中操作数据流程的描述。对数据的描述指的是数据类型和数据组织形式。数据类型有整型、浮点型、字符型、组合类型等,数据的组织形式有链表、队列等;对程序操作流程的描述即算法,是程序执行的步骤,类比我们生活中要解决一个问题的具体流程。

算法在程序中是不可缺少的一部分,比如使用百度搜索资料时,用到排序算法;淘宝购物时使用推荐算法等。若把一个运行的程序比喻成有生命的人,数据的结构就是人的躯体,算法就是这个人的灵魂。正如计算机科学家尼基劳斯·沃思(Nikiklaus Wirth)将程序描述为:

$$\text{程序} = \text{数据结构} + \text{算法}$$

举一个简单的例子,比如数学中关于素数的定义:素数是在大于 1 的自然数中,除了 1 和它自身外,不能被其他自然数整除的数。这是素数的求解思路,也是一个算法的描述。

2. 算法特征

算法应具有以下 5 项特征。

(1) 有穷性。算法的有穷性是指算法必须能在执行有限个步骤之后结束,在具体的算法中指的是在当前解决问题的合理范围之内,如果一个算法解决问题历时一年,算法尽管有穷也不会被考虑使用。

(2) 确定性。算法的每一个步骤必须有确切的定义,计算机处理问题的步骤是确定的,算法设计过程中不能出现二义性、选择不确定的情况。

(3) 输入项。一个算法有0个或多个输入,以获取程序处理的必要信息。输入项可以由程序中其他功能模块传递,也可以从键盘输入获取。

(4) 输出项。一个算法有1个或多个输出,输出是对输入数据加工后产生的结果,没有输出结果的算法是没有意义的。

(5) 可行性。算法中执行的计算步骤都应可被分解为基本的、可执行的操作步骤。

1.2.2 算法的表示

遇到需要解决的问题通过思考得出解决的办法,结合编程思想,将解决问题时的方案用算法表示出来,称之为算法的表示。算法一般有4种表示方法:自然语言、流程图、N-S流程图和伪代码,下面我们分别对这4种表示方法进行介绍。

(1) 自然语言描述

使用自然语言描述法表示算法,就是使用自然语言描述问题的求解思路与过程。可以用这样的自然语言描述法判断一个数是否为素数的算法:大于1的自然数中,除了1和它本身不再有其他因数的数就是素数。

在一些大型的开源项目中,说明文档会用自然语言粗略地表述算法,但因为自然语言容易产生二义性,一般不使用自然语言描述算法的具体实现。

(2) 流程图表示

流程图简单直观并且易于理解,是使用最广泛的算法表示方法。流程图的表示方式如图1-1所示。



图1-1 常见的流程图结构

图1-1中所示的各个框图结构表示的含义如下。

- 开始或结束。使用圆角矩形表示,用于标识流程的开始或结束。
- 输入/输出。使用平行四边形表示,其中可以写明输入或输出的内容。
- 条件判断。使用菱形表示,它的作用是对条件进行判断,根据条件是否成立来决定如何执行后续的操作。
 - 程序处理。使用矩形表示,它代表程序中的处理功能,如算术运算和赋值等。
 - 流程线。使用实心单向箭头表示,可以连接不同位置的图框。
 - 连接点。使用圆形表示,用于流程图的延续。

以判断一个数是否为素数的算法为例,使用流程图来表示算法的具体方法如图1-2所示。

(3) N-S流程图表示

1973年美国学者艾克·纳西(Ike Nassi)和本·施奈德曼(Ben Shneiderman)提出了一种新的流程图形式,这种流程图完全去掉了流程线,算法的每一步都用一个矩形框来描述,把一个个矩形框按执行的次序连接起来就是一个完整的算法描述。这种流程图用两位学者名字的第一个字母来命名,称为N-S流程图,如图1-3所示。

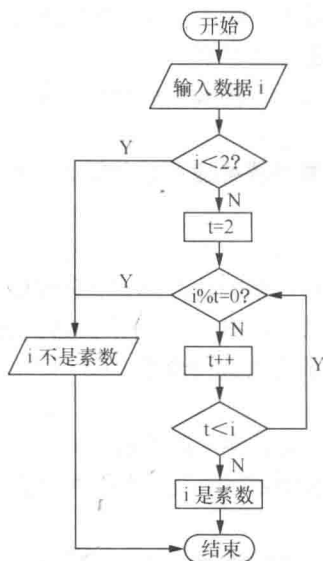
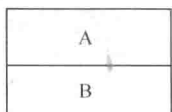
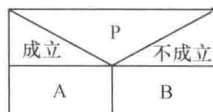


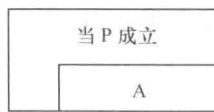
图1-2 素数判断流程图



(1) 顺序结构



(2) 分支结构



(3) 循环条件

图1-3 N-S流程图

(4) 伪代码表示法

使用伪代码的目的是使被描述的算法可以容易地以任何一种编程语言实现,此种表示类似自然语言,但结构清晰、可读性好。伪代码没有固定格式,不用拘泥于具体实现,使用接近自然语言的形式将整个算法运行过程的结构表述出来即可。判断素数的伪代码如下所示:

```

If i < 2
Then i 不是素数, 程序结束
Else
t=2
Repeat:
    If i mod t=0
        Then i 不是素数, 程序结束
    Else t=t+1
Until t>=i
i 是素数, 程序结束

```

1.3 C语言概述

早期的编程工程师只能借助查表的方法和机器语言编写程序,但随着计算机技术的发展,这

种低级语言编程已不能满足需求,随之出现了不同类型的编程语言,其中,C语言就是出现最早的编程语言之一。本节我们将介绍C语言,让初学者对C语言有清晰的认识。

1.3.1 C语言的发展史

C语言特性源自一种称为“B”的早期语言(基本组合编程语言,BCPL),其设计是为了实现UNIX操作系统。

1963年,剑桥大学将ALGOL 60语言发展成为CPL(Combined Programming Language)语言。

1967年,剑桥大学的马丁·理查兹(Martin Richards)对CPL语言进行了简化,BCPL语言诞生。

1970年,美国贝尔实验室的肯·汤普森(Ken Thompson)对BCPL进行了修改,并将其命名为“B语言”,其含义是将BCPL语言“煮干”,提炼出它的精华,之后他用B语言编写了第一个基于非汇编语言的UNIX操作系统。

1972年,美国贝尔实验室的丹尼斯·里奇(Dennis M.Ritchie)在B语言的基础上设计出了一种新的语言,他取了BCPL的第2个字母作为这种语言的名字,即C语言。之后丹尼斯·里奇与肯·汤普森成功地使用C语言重写了UNIX的第3版内核,该版内核具有良好的可移植性且易于扩展,为UNIX日后的普及打下了坚实基础。

1978年,布赖恩·凯尼汉(Brian W.Kernighan)和丹尼斯·里奇(Dennis M.Ritchie)出版了名著《The C Programming Language》,从而使C语言成为目前世界上使用最广泛的高级程序设计语言。

1.3.2 C语言的标准

随着微型计算机的普及,C语言衍生出了诸多版本,这些版本之间存在差异,为了使C语言得到统一,美国国家标准学会(ANSI)制定了一套标准,称为ANSI C。ANSI C标准自1989年诞生以来,又历经了下述几次修改。

- 1989年,美国国家标准学会(ANSI)通过的C语言标准ANSI X3.159-1989被称为C89。
- 1990年,国际标准化组织(ISO)接受C89作为国际标准ISO 9899-1990,该标准被称为C90。这两个标准只有细微的差别,因此,通常来讲C89和C90指的是同一个版本。
- 1999年,ANSI通过了C99标准。C99标准相对C89做了很多修改,例如变量声明可以不放在函数开头,支持变长数组,初始化结构体允许对特定的元素赋值等。本书将以C99标准为主进行讲解。
- 2011年,ISO和IEC(国际电工委员会)正式发布C语言标准第3版草案(N1570),提高了C语言对C++的兼容性,并增加了一些新的特性。这些新特性包括泛型宏、多线程、带边界检查的函数等。

1.3.3 C语言的应用领域

根据TIOBE网站公布的编程语言热门程度排行榜,C语言热门程度稳居前三。历年语言热门程度排名如图1-4所示。

C语言也获得了远高于大多数编程语言的评分,TIOBE于2018年10月公布的编程语言评分如图1-5所示。

C语言之所以稳居前三,获得高度评价,与其良好的性能及广泛的应用领域密不可分,C语

言常被应用在以下领域。

Programming Language	2018	2013	2008	2003	1998	1993	1988
Java	1	2	1	1	17	-	-
C	2	1	2	2	1	1	1
C++	3	4	3	3	2	2	4
Python	4	7	6	11	24	13	-
C#	5	5	7	8	-	-	-
Visual Basic .NET	6	11	-	-	-	-	-
PHP	7	6	4	5	-	-	-
JavaScript	8	9	8	7	21	-	-
Ruby	9	10	9	18	-	-	-
R	10	23	48	-	-	-	-
Objective-C	14	3	40	50	-	-	-
Perl	16	8	5	4	3	9	22
Ada	29	19	18	16	12	5	3
Lisp	30	12	16	13	8	6	2
Fortran	31	24	21	12	6	3	15

图1-4 编程语言排行

Oct 2018	Oct 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.801%	+5.37%
2	2		C	15.376%	+7.00%
3	3		C++	7.593%	+2.59%
4	5	^	Python	7.156%	+3.35%
5	8	^	Visual Basic .NET	5.864%	+3.15%
6	4	v	C#	3.485%	-0.37%
7	7		PHP	2.794%	+0.00%
8	6	v	JavaScript	2.280%	-0.73%
9	-	≈	SQL	2.038%	+2.04%
10	16	≈	Swift	1.500%	-0.17%

图1-5 2018年10月语言评分排行

1. 操作系统

C 语言可以开发操作系统，主要应用在个人桌面领域的 Windows 系统内核、服务器领域的 Linux 系统内核、FreeBSD、苹果公司研发的 Mac 系统。

2. 应用软件

C 语言可以开发应用软件。在企业数据管理中，需要可靠的软件处理有价值的信息，C 语言具有高效、稳定等特性，企业数据管理中使用的数据库如 Oracle、MySQL、MS SQL Server 和 SQLite 等都由 C 语言开发。此外金山办公软件 WPS 及微软的 Office 办公软件、功能强大的数学软件 MatLab 等都使用 C 语言开发。

3. 嵌入式底层开发

当今时代，生活的各个方面都在智能化，智能城市、智能家庭等概念已不再是设想。这些智能领域离不开嵌入式开发，熟知的智能手环、智能扫地机器人、轿车电子系统等都离不开嵌入式开发。

组成这些智能系统的东西，如底层的微处理器控制的传感器、蓝牙、WiFi 网络传输模块等使用的硬件驱动库、嵌入式实时操作系统 FreeRtos、UCOS 和 VxWorks 等，都主要由 C 语言开发。

4. 游戏开发

C 语言具有强大的图像处理能力、可移植性、高效性等特点。一些大型的游戏，游戏环境渲染、图像处理等使用 C 语言来处理，成熟的跨平台游戏库 OpenGL、SDL 等也由 C 语言编写而成。