

21 世纪高等学校规划教材

C语言程序设计基础

马华 李玉娟 主编
王秀娟 张兰华 江震 副主编
韩忠东 主审

21st Century University
Planned Textbooks



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校规划教材

C语言程序设计基础

马华 李玉娟 主编
王秀娟 张兰华 江震 副主编
韩忠东 主审

21st Century University
Planned Textbooks

人民邮电出版社
北京

图书在版编目(CIP)数据

C语言程序设计基础 / 马华, 李玉娟主编. — 北京:
人民邮电出版社, 2012.9
21世纪高等学校规划教材
ISBN 978-7-115-28459-4

I. ①C… II. ①马… ②李… III. ①
C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(2012)第150890号

内 容 提 要

本书注重教材的可读性和适用性,共分9章,内容包括C语言概述、基本C语言程序设计、选择结构程序设计、循环结构程序设计、函数、数组、指针、结构体、文件等。

本书每章首先给出学习目标和重点难点,旨在给出该章的学习目标和着重需要掌握的知识点,然后引出案例及案例的运行结果,并给出案例涉及的知识点,接下来对关键知识点和要求掌握的知识点进行说明,最后给出案例解析以及案例源代码。本教材附有大量的图表和参考程序,使读者能正确、直观地理解问题;样例程序由浅入深,强化知识点、算法、编程方法与技巧,并给出了详细的解释。另外,本书还配套提供题型丰富的习题。

本书可作为高等学校大学本科、高职高专学生“C语言程序设计”课程的教学用书,也可作为全国计算机水平考试及各类短训班的培训教材。

21 世纪高等学校规划教材

C 语言程序设计基础

-
- ◆ 主 编 马 华 李玉娟
副 主 编 王秀娟 张兰华 江 震
主 审 韩忠东
责任编辑 董 楠
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮箱 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京铭成印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 11.25 2012 年 9 月第 1 版
字数: 292 千字 2012 年 9 月北京第 1 次印刷

ISBN 978-7-115-28459-4

定价: 25.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前言

C 语言功能丰富, 使用灵活, 目标程序效率高, 可移植性好, 程序结构性和可读性好, 体现了结构化程序设计的思想, 它一直超越众多的程序设计语言, 在编程语言排行榜上名列前茅。学生通过本课程的学习, 可以培养良好的编程风格, 掌握常见的算法思路, 真正提高运用 C 语言编程解决实际问题的综合能力, 为后续课程实践环节的教学打好基础。

在计算机专业程序设计类课程的教学中,往往会产生学生觉得难学,教师觉得难教的情况,如下所述。

(1) 程序设计类课程教学难度较大。学生学习程序设计, 感觉抽象, 难以理解, 这使得教学难度增加。这在授课进度和难易度控制等方面对教师提出了较高的要求。

(2) 传统教学中过分注重语言的语法、语义规则, 纠结于语言的细节问题, 而忽视了对编程思路的培养, 忽视了程序设计思想、算法的教学。这使得学生缺乏独立分析问题和解决问题的能力, 学生在上机实验时, 只能应付验证性实验, 在设计性实验中, 缺乏思路和解决问题的能力。

为了帮助广大学生更好地掌握C语言编程技术,帮助任课教师更好地组织教学,我们编写了本书。本书具有如下特点。

(1) 改变传统教材以语法为驱动的内容编排模式,有利于任课教师组织课堂教学,有利于培养学生解决实际问题的能力。

(2) 加强对学生分析问题及表达算法能力的培养, 拟通过训练画 N—S 图, 来提高学生的编程能力。传统教材对问题求解算法的表达不够重视, 对培养学生画流程图或 N—S 图的能力不够重视, 这导致大部分学生对新问题的分析及算法表达能力欠缺, 从而影响其编写程序的能力。

(3) 采用 Visual C++ 6.0 作为编程环境, 与全国计算机等级考试 C 语言考试的编程环境相一致, 与许多实际项目开发过程中要求使用的 Visual C++ 6.0 编程环境相一致。

本书共分9章,内容结构清晰。第1章对C语言进行概述,介绍了C语言的基本语法、Visual C++6.0的编程环境、编程解决问题的过程,其中特别讲述了VC6.0的程序调试方法及利用N—S图描述算法的过程;第2章利用3个案例分别介绍了基本输出函数、基本运算符和表达式、基本输入函数等知识点,以及程序的书写风格;第3章介绍了选择程序设计,用3个案例分别介绍了if语句、if嵌套语句、switch语句等知识点;第4章介绍循环结构程序设计,分别用蜡烛燃烧之谜、口令验证、阶乘问题以及打印杨辉三角形等案例讲解了3种循环语句和循环嵌套;第5章用案例讲解了自定义函数以及递归函数设计的思想和方法;第6章通过4个案例介绍了一维数组、二维数组、字符数组等知识点;第7章利用案例介绍了指针、指针变量及其与数组、字符串、函数等知识的关系;第8章利

用学生信息管理、学生成绩管理、猴子选大王等案例，介绍了结构体类型、结构体数组、指向结构体的指针等知识点；第 9 章利用 3 个案例介绍了文件、打开文件、对文件读写等操作。

本书在组织案例时采用启发式方法。对于较复杂的案例，首先提出问题，启发读者在用旧知识点不能或不便解决问题时该怎样做，从而引出新知识点；然后直观地给出程序的执行结果，并指出解决该问题涉及的知识点，从而激发读者对新知识点的探究兴趣；接下来对新知识点进行介绍；最后给出对案例的分析及源代码实现。为了提高学生分析问题以及描述算法的能力，每章后都有相应的练习题，并鼓励学生先画出 N—S 图，再编程并上机调试。

本书由马华负责全书的策划和组织，并对全书进行了统稿和校对，其中马华编写了第 1 章、第 9 章，李聪和孙静编写了第 2 章，江震和张峰编写了第 3 章，王玫和马晓艳编写了第 4 章，王秀娟编写了第 5 章，张兰华和许婷婷编写了第 6 章，彭磊和左风华编写了第 7 章，李玉娟编写了第 8 章，全书由韩忠东审稿。

本书的配套教材《C 语言程序设计实验指导和习题集》也同时出版，配套教材针对本书的知识点设计了验证性、改错性和设计性实验内容。此外，配套教材给出了几个综合性实验项目，它们可作为课程设计的题目。

本书在编写过程中得到许多老师的支持和帮助。泰山医学院张西学教授、张兆臣教授和张裕飞教授为本书提出了很多具体的指导意见，在此对他们表示衷心的感谢。同时对本书编写过程中参考的大量图书、资料和网站资料的作者，以及人民邮电出版社的各位老师的大力支持表示诚挚的谢意。

由于作者水平有限，书中难免会出现遗漏和不足，恳请各界同仁及读者朋友提出宝贵的意见和真诚的批评。

编 者

2012 年 4 月

目 录

第 1 章 C 语言概述..... 1

- 1.1 C 语言发展历史..... 1
- 1.2 C 语言的基本语法..... 1
 - 1.2.1 C 语言的字..... 2
 - 1.2.2 C 语言中的词..... 2
 - 1.2.3 C 语言的语句..... 3
- 1.3 Visual C++ 6.0 编程环境..... 4
 - 1.3.1 VC 的编辑环境..... 4
 - 1.3.2 运行..... 5
 - 1.3.3 调试..... 6
- 1.4 编程解决问题的过程..... 6
 - 1.4.1 编程解决问题的步骤..... 6
 - 1.4.2 算法及其描述工具..... 6
- 1.5 小结..... 8
- 习 题..... 8

第 2 章 基本 C 语言程序设计..... 9

- 2.1 基本输出函数..... 9
 - 2.1.1 案例描述..... 9
 - 2.1.2 各种类型数据的输出函数 printf()... 9
 - 2.1.3 主函数、库函数和头文件..... 11
 - 2.1.4 程序解析..... 12
- 2.2 基本的运算符和表达式..... 12
 - 2.2.1 案例描述..... 12
 - 2.2.2 常量、变量和基本数据类型..... 13
 - 2.2.3 基本运算符和表达式..... 15
 - 2.2.4 随机数产生函数和选择语句 if-else 17
 - 2.2.5 程序解析..... 18
- 2.3 基本输入函数..... 19
 - 2.3.1 案例描述..... 19
 - 2.3.2 各种类型数据的输入函数 scanf()... 19
 - 2.3.3 程序解析..... 20
- 2.4 特殊运算符和表达式..... 20
 - 2.4.1 逗号运算符和逗号表达式..... 20

- 2.4.2 条件运算符和条件表达式..... 21
- 2.4.3 位运算符..... 21
- 2.4.4 自增、自减运算符..... 22
- 2.5 类型转换..... 23
 - 2.5.1 自动类型转换..... 23
 - 2.5.2 强制类型转换..... 24
- 2.6 常用数学库函数..... 25
- 2.7 小结..... 26
- 习 题..... 26

第 3 章 选择结构程序设计..... 30

- 3.1 显示输入的字母、数字或其他字符的 ASCII 码值..... 30
 - 3.1.1 案例描述..... 30
 - 3.1.2 字符输入和输出函数 getchar() 和 putchar()..... 31
 - 3.1.3 逻辑运算符和逻辑表达式..... 32
 - 3.1.4 条件运算符..... 32
 - 3.1.5 if 语句..... 33
 - 3.1.6 程序解析..... 34
- 3.2 销售提成问题..... 34
 - 3.2.1 案例描述..... 34
 - 3.2.2 if 嵌套语句..... 35
 - 3.2.3 程序解析..... 35
- 3.3 学生成绩与等级..... 36
 - 3.3.1 案例描述..... 36
 - 3.3.2 switch 语句..... 36
 - 3.3.3 程序解析..... 39
- 3.4 小结..... 40
- 习 题..... 40

第 4 章 循环结构程序设计..... 44

- 4.1 蜡烛燃烧之谜..... 44
 - 4.1.1 案例描述..... 44
 - 4.1.2 while 语句..... 45

4.1.3 程序解析.....	47
4.2 口令程序.....	48
4.2.1 案例描述.....	48
4.2.2 do-while 语句.....	48
4.2.3 程序解析.....	48
4.3 阶乘问题.....	49
4.3.1 案例描述.....	49
4.3.2 for 语句.....	50
4.3.3 goto 语句和 if 语句构成循环.....	51
4.3.4 转移语句.....	52
4.3.5 程序解析.....	54
4.4 杨辉三角形问题.....	55
4.4.1 案例描述.....	55
4.4.2 循环的嵌套.....	56
4.4.3 程序解析.....	57
4.5 综合应用.....	58
4.6 小结.....	60
习 题.....	60

第 5 章 函数..... 62

5.1 自定义函数求 $E=1+1/1!+1/2!+...+1/10!62$	
5.1.1 案例描述.....	62
5.1.2 自定义函数.....	63
5.1.3 函数的调用.....	64
5.1.4 程序解析.....	66
5.2 变量的作用域与存储类别.....	67
5.2.1 局部变量.....	67
5.2.2 全局变量.....	68
5.2.3 存储类别.....	69
5.3 递归问题——求 $n!$	70
5.3.1 案例描述.....	70
5.3.2 递归函数.....	70
5.3.3 程序解析.....	71
5.4 编译预处理.....	72
5.4.1 宏定义.....	72
5.4.2 文件包含命令.....	74
5.4.3 条件编译.....	75
5.5 综合应用.....	75
5.6 小结.....	76
习 题.....	77

第 6 章 数组..... 78

6.1 一组数据的排序.....	78
6.1.1 案例描述.....	78
6.1.2 一维数组的定义.....	79
6.1.3 一维数组的引用.....	79
6.1.4 一维数组的初始化.....	80
6.1.5 程序解析.....	81
6.2 矩阵转置.....	82
6.2.1 案例描述.....	82
6.2.2 二维数组的定义.....	83
6.2.3 二维数组元素的引用.....	83
6.2.4 二维数组的初始化.....	84
6.2.5 程序解析.....	84
6.3 判断字符串是否是回文.....	85
6.3.1 案例描述.....	85
6.3.2 字符数组与字符串的区别.....	85
6.3.3 字符数组的定义及初始化.....	86
6.3.4 字符数组的输入/输出.....	86
6.3.5 字符串处理函数.....	87
6.3.6 程序解析.....	89
6.4 选择法排序.....	90
6.4.1 案例描述.....	90
6.4.2 数组元素和数组名作函数参数.....	91
6.4.3 程序解析.....	91
6.5 综合应用.....	92
6.6 小结.....	93
习 题.....	94

第 7 章 指针..... 95

7.1 通过地址找同学.....	95
7.1.1 案例描述.....	95
7.1.2 地址.....	95
7.1.3 指针类型和指针变量.....	96
7.1.4 案例解析.....	97
7.2 按正向和反向顺序打印字符串.....	98
7.2.1 案例描述.....	98
7.2.2 指针变量的定义.....	98
7.2.3 指针变量的访问.....	99
7.2.4 指针变量的引用.....	100

7.2.5 程序解析.....	103	8.5 小结.....	144
7.3 指向数组的指针.....	104	习 题.....	145
7.3.1 数组的指针和指向数组的指针 变量.....	104	第 9 章 文件.....	147
7.3.2 通过指针引用数组元素.....	105	9.1 将字符串写入文本文件.....	147
7.3.3 指向多维数组的指针和指针变量.....	106	9.1.1 案例描述.....	147
7.3.4 动态内存分配.....	108	9.1.2 文件概述.....	148
7.4 字符串的复制.....	109	9.1.3 文件的打开和关闭.....	149
7.4.1 案例描述.....	109	9.1.4 格式化文件写函数 fprintf().....	150
7.4.2 字符串的表示形式.....	110	9.1.5 程序解析.....	151
7.4.3 字符数组和字符指针.....	111	9.2 从文件中读取学生的信息.....	152
7.4.4 程序解析.....	111	9.2.1 案例描述.....	152
7.5 指针数组与指向指针的指针.....	112	9.2.2 格式化文件读取函数 fscanf().....	152
7.5.1 指针数组.....	112	9.2.3 程序解析.....	153
7.5.2 指针的指针.....	114	9.3 复制文件.....	154
7.6 指针与函数.....	115	9.3.1 案例描述.....	154
7.6.1 学生成绩分析——指针作为函数的 参数.....	115	9.3.2 字符方式的文件读写函数 fgetc()和 fputc().....	154
7.6.2 调用求两数中较大者的函数——用 指针调用函数.....	117	9.3.3 字符串方式的文件读写函数 fgets()和 fputs().....	155
7.6.3 输出对应月份的英文名称——指针 型函数.....	118	9.3.4 程序解析.....	156
习 题.....	119	9.3.5 其他文件操作相关函数.....	157
第 8 章 结构体.....	121	9.4 综合应用.....	159
8.1 学生信息管理.....	121	9.4.1 案例描述.....	159
8.1.1 案例描述.....	121	9.4.2 学生基本信息管理系统的分析和 设计.....	159
8.1.2 结构体与结构体变量.....	122	9.4.3 程序解析.....	160
8.1.3 程序解析.....	124	9.5 小结.....	161
8.2 学生成绩管理.....	126	习 题.....	161
8.2.1 案例描述.....	126	附录 1 ASCII 表.....	163
8.2.2 结构体数组与函数.....	127	附录 2 C 语言关键字.....	164
8.2.3 程序解析.....	128	附录 3 运算符及其优先级.....	166
8.3 单链表基本操作.....	131	附录 4 基本库函数.....	168
8.3.1 案例描述.....	131	参考文献.....	172
8.3.2 单链表及其基本操作.....	133		
8.3.3 程序解析.....	134		
8.3.4 利用循环链表解决猴子选大王 问题.....	138		
8.4 综合应用.....	141		

第 1 章

C 语言概述

学习目标

- 了解 C 语言发展历史;
- 掌握 C 语言的基本语法;
- 熟悉 VC++6.0 的编辑运行环境;
- 掌握编程解决问题的基本过程。

重点难点

- 重点: C 语言的语法, 包括关键字、词和句, 利用编程工具 VC++6.0 编辑运行一个程序。
- 难点: 理解并掌握用 C 语言解决问题的基本过程、算法的本质。

1.1 C 语言发展历史

C 语言最早由肯·汤普生 (Ken Thompson) 于 1973 年设计并实现。肯·汤普生毕业于著名的加利福尼亚大学伯克利分校, 并获得学士和硕士学位。1970 年, 他写出了 B 语言, 目的是改进当时的 BCL 语言。而 B 语言就是 C 语言的前身。

肯·汤普生和丹尼斯·里奇一起用 C 语言开发出 UNIX 操作系统后, 立刻引起程序员们的广泛关注。由于 UNIX 和 C 语言的巨大成功, 肯·汤普生和丹尼斯·里奇在 1983 年获得计算机界的最高奖——图灵奖。

C 语言具有灵活性、简单性、运行速度快等特点, 既具有一般高级语言的特性, 又具有低级语言的特性, 它既可以用来编写系统软件, 也可以用来编写应用程序, 倍受程序员的欢迎。它以简洁的语法和较高的执行效率, 将过程语言带到了最高峰, 可以说 C 语言是世界上使用最多的一种编程语言。后来的两个主流语言 C++ 和 Java, 都是建立在 C 语言的语法和基本结构基础上的。

1.2 C 语言的基本语法

与英语很相似, C 语言也是有字、词、句等单位, 并按一定的语法规则构成。下面, 通过一个简单的 C 语言源程序来体会一下 C 语言的语法。该程序完成了计算 1 到 100 之间所有整数求和的功能。

```
main()
```

```
{
    int i, sum=0;
    for (i = 1; i <= 100 ; i++)
        sum = sum + i;
    printf("The sum of 1 to 100 is: %d\n",sum);
}
```

通过上面的程序简单认识一下 C 语言的语法。

1.2.1 C 语言的字

C 语言中的字包括 26 个大写字母 (A~Z), 26 个小写字母 (a~z), 10 个数字 (0~9)。另外还有以下符号:

- (1) 算术运算符 + - * / % 加 减 乘 除 按模取余
- (2) 赋值运算符 = 赋值 (不是等号)
- (3) 关系运算符 < > 小于 大于 (另外还有组合符号 <= >= !=)
- (4) 标点符号 . , ; : ? 点 逗号 冒号 分号 问号
- (5) 逻辑符号 & | ! ~ ^ 与 或 非 取反 异或
- (6) 括号符号 " ' () [] { } 双引号 单引号 小括号 中括号 大括号
- (7) 特殊符号 空格 # \ _ 空格 #号 反斜杠 下划线

试着看一下前面的源程序用到 C 语言中的哪些字。

1.2.2 C 语言中的词

C 语言中的词由字来组成, 按词意和用途可分为 6 类词: 关键字、标识符、运算符、分隔符和注释符。

1. 关键字

关键字, 也叫保留字, 是 C 语言专用的词, 共 32 个, 分为 7 类, 见表 1-1。

表 1-1 C 语言的关键字

用 途	关 键 字	备 注
定义基本数据类型	int float double char void	
修饰数据类型	signed unsigned short long const volatile	
修饰存储类型	auto register static extern	
定义复合数据类型	struct union enum	
流程控制	if else switch case default for while do goto continue break return	
类型定义	typedef	
求数据类型长度	sizeof	如 sizeof (char)

2. 标识符

用户可以用它对变量、函数等对象命名。命名规则为: 由字母或下划线开头, 后面由字母、数字或下划线构成。标识符的名字要“见名知意”, 如存放年龄的变量定义为 age。

3. 运算符

C 语言拥有 43 种运算符, 这些运算符将数据连接起来就形成表达式, 用来进行数学或逻辑运算。这些运算符可分类如表 1-2 所示。

表 1-2 运算符

基本运算符	算术运算符	+ - * / % ++ --	单目运算符++、--以及符号- 结合性：自右向左
	关系运算符	< <= == > >= !=	
	逻辑运算符	&& !	
	赋值运算符	= += -= *= /= %=	结合性：自右向左
特殊运算符	位运算符	& ^ >> << &= = ^= >>= <<=	
	条件运算符	?:	结合性：自右向左
	指针运算符	->	
	逗号运算符	,	

表 1-2 中未说明结合性的运算符，其结合性均为自左向右。对于这些运算符的优先级可归纳成如下的口诀，其中的一二三四五六七为优先级。

括号一，单目二，乘除余三加减四；
移位五，关系六，等不等排第七；
位运算为第八，后跟与或及条件，赋值逗号级最低。

4. 分隔符

分隔符有 5 个，具体见表 1-3。

表 1-3 分隔符及其用途

符 号	举 例	用 途
逗号	int i,j,k;	将各对象分隔开
空格	static int s=0;	将句中的词分隔开
分号	int i; for(i=0;i<100;i++){	每条语句的结束， for 语句中多个表达式的分隔
冒号	switch(n) { case 1: y=0;break;}	多用于 case 语句中， 作为整数与后面语句的分隔
大括号	while(i<=100) {s=s+i;i++;}	用于构造程序结构、复合句的符号

5. 注释符

注释符以 // 开始，或用 /* 和 */ 将内容括起。它只是增加程序的可读性。一个好的程序应当有 1/3 的注释，以便于日后的阅读和维护。

1.2.3 C 语言的语句

程序员是通过 C 语言的语句与计算机进行交流的，为此看懂并会编写程序首先要认识 C 语言的语句。C 语言的语句分为两类：说明语句和执行语句。

1. 说明语句

说明类型语句即定义数据类型语句，其作用是定义变量，也就是声明一个存放数据的场所。变量声明后，就规定了变量名、变量的类型、变量的取值范围，以及变量能参与的运算。变量类型的关键词见表 1-4。

表 1-4 变量类型说明语句

基本类型	int（整型）、char（字符型）、float（单精度实数）、double（双精度型实数）、enum（枚举）
构造类型	数组类型：如 int a[10];
	结构体类型：struct
	共用体类型：union
指针类型	文件类型：FILE， 指针类型：*

2. 执行语句
各执行语句及格式见表 1-5。

表 1-5 执行语句

语 句	格 式	语 句	格 式
赋值	n = 10;	default	default: 语句;
if	if (x>0) y =1;	break	break;
while	while (i<=100) { s += i; i++; }	continue	continue;
do	do { s += i; i++; } While (i<=100) ;	return	return; 或 return (表达式);
for	for(i=0;i<=100;i++) s += i;	goto	goto 标号;
switch	switch (表达式) 语句;	函数调用	函数名 (实参表);
case	case 常数: 语句;	空语句	;

1.3 Visual C++ 6.0 编程环境

1.3.1 VC 的编辑环境

Visual C++ 6.0（简称 VC）是 Windows 下广泛使用的开发工具，简单易用，并且全国计算机等级考试大纲也把 VC 作为 C 语言的编程环境，故本书采用 VC 作为 C 语言的开发工具。下面简单说明一下 VC 的编辑环境。

启动 Visual C++ 6.0 建立 C 程序文件时，执行菜单“文件”(File)->“新建”(New)命令，并进入选项卡“文件”(Files)后，出现图 1-1 中的新建窗口，建立文本文件（Text File）hello.c，并选择文件存储位置，之后单击“确定”(OK)按钮。接下来就可以在图 1-2 所示的编辑区中编辑 C 程序文件 hello.c。

在编辑区中移动光标时，可以使用键盘上的上下左右光标键，以及 Home、End、PaUp 和 PaDo 等快速键。编辑时可以做选中、复制、移动、粘贴、删除等操作。

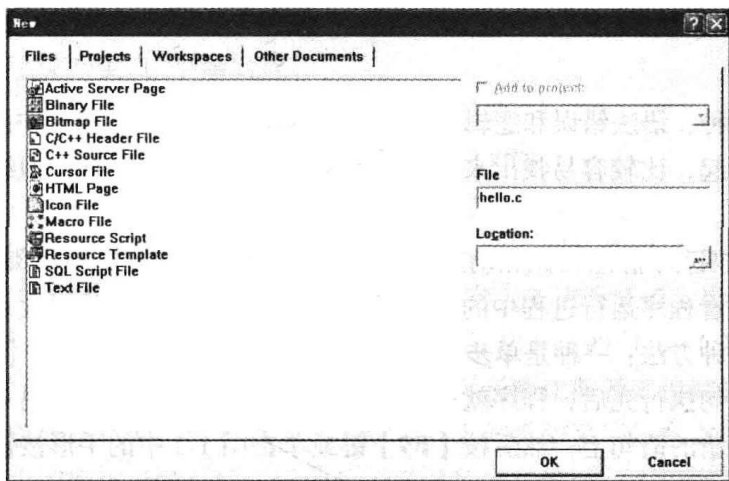


图 1-1 新建 C 程序文件

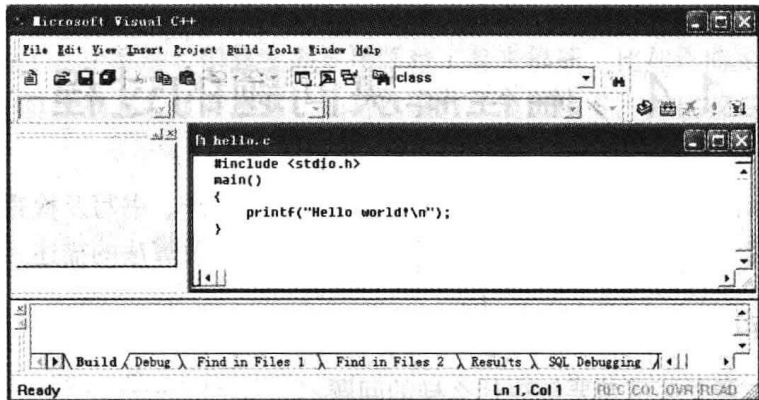


图 1-2 Visual C++ 6.0 编辑界面

1.3.2 运行

图 1-1 中的程序称为“源程序”，它是人们能直接理解的源代码。接下来需要用编译程序将源程序编译成二进制形式的目标文件，其文件扩展名为.obj，然后将目标程序与系统的函数库和其他目标程序连接起来，形成可执行文件，其扩展名为.exe。

编译源程序时，可单击图 1-3 中的第 1 个按钮；连接时单击图 1-3 中的第 2 个按钮；运行程序时单击图 1-3 中的第 4 个按钮。



图 1-3 常用按钮

编译、连接、运行程序时，若有错误或警告，会有提示信息出现在图 1-2 的下端窗口中，此时需要进行调试，并修改源代码继续运行；若没有错误，会显示最终结果（见图 1-4）。



图 1-4 程序运行结果

1.3.3 调试

程序的错误有两种：语法错误和逻辑错误。语法错误主要是由 C 语言中的字、词、句以及各种符号拼写不正确引起，比较容易找出来，而逻辑错误是指程序能运行，但结果不正确，此时则需要通过调试来解决。

程序调试方法有两种：静态调试和动态调试。静态调试需要程序员自己阅读程序来发现错误，而动态调试则通过查看程序运行过程中的中间结果来确定出错位置，然后改正。

动态调试又有两种方法：一种是**单步调试**，按【F10】(跳过子程序)或【F11】(进入子程序)键逐条执行，一条语句执行完后，程序就停下来并显示一些变量的结果；另一种是**设置断点**，先将光标定位到可能出错的语句上，然后按【F9】键或单击图 1-3 中的手形按钮，接下来按【F5】键或单击图 1-3 中的第 5 个按钮，程序会运行到断点处暂停，让你观察中间结果。通过查看中间结果就可以推定错误位置，然后改正。

1.4 编程解决问题的过程

利用计算机来解决问题，就需要程序设计。通常我们把设计、书写及检查调试程序的过程称为程序设计。下面简单说明设计程序解决问题的一般过程，以及算法的描述工具。

1.4.1 编程解决问题的步骤

(1) **分析问题**，需要搞清楚要解决什么样的问题。

(2) **确定数据结构**。根据任务提出的要求，指定变量，用来存放输入的数据和输出的结果，也就是要确定存放数据的数据结构。

(3) **确定算法**。针对存放数据的数据结构来确定解决问题、完成任务的步骤。

(4) **编码**。根据确定的数据结构和算法，使用选定的计算机语言来编写程序代码，输入到计算机，并保存在磁盘上，通常简称为“编程”。

(5) **调试程序**。消除由于疏忽而引起的语法错误或逻辑错误，用各种可能的输入数据对程序进行调试，使之对各种合理的数据都能得到正确的结果，对不合理的数据进行适当的处理。

(6) **整理并写出文档资料**。以上步骤中，算法是程序设计的灵魂，而语言只是一种形式。有了正确的算法，可以利用任何一种语言编写程序。因此，有必要简单介绍算法的概念以及算法的描述工具。

1.4.2 算法及其描述工具

1. 算法的概念

广义地讲，处理任何问题都有一个“算法”问题。例如，“太极拳动作图解”就是太极拳的算法，因为它详细地叙述了太极拳的动作和执行顺序。同样乐曲的乐谱也可以说是该乐曲的算法，因为指定了演奏该曲的的每一个步骤。同样在菜谱中也包含了算法，因为它除了列出做菜的原料以外，还列出了操作的每一个步骤。当然，我们在这里所讨论的算法指的还是计算机算法，也就是计算机能够执行的操作。

算法就是解题方法。严格地说，算法是由若干条指令组成的有穷序列，它必须满足下述准则。

- (1) 输入：应具有 0 个或多个输入。
- (2) 输出：至少产生 1 个输出量。
- (3) 有穷性：每一条指令的执行次数必须是有限的。
- (4) 确定性：每条指令的含义都必须明确，无二义性。
- (5) 可行性：每条指令的执行时间，都是有限的。

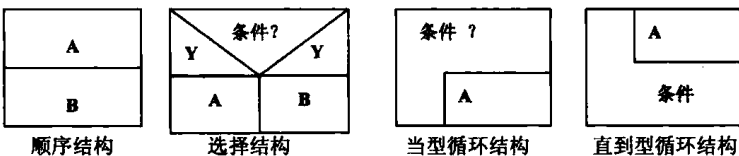
对于同一个问题，可有不同的解题方法和步骤，即不同的算法。

例如，求 $1+2+3+4+5+\cdots+100$ 就有不同的方法，有人先进行 $1+2$ ，再加 3，再加 4，一直加到 100，则得到结果 5050。而有的人则采用另外的方法， $100+(99+1)+(98+2)+\cdots+(51+49)+50$ 的方法，也得到同样的结果。显然，对心算来说，后者要比前者来得容易。当然还有其它方法，如，将奇数和偶数分别相加，再求和，如上所示。人们希望采用好的方法，即方法简单、运算步骤少、能迅速得出正确结果的算法。因此为了有效地进行解题，不仅需要保证算法正确，还要考虑算法的质量选择合适的算法。

对于完成一项工作来说，要设计算法，即指出应进行的操作和步骤。比如，作曲家创作了一首曲谱就是设计了一个算法。但算法需要用某种工具来描述，比如作曲家用五线谱或简谱这种工具来描述曲谱。算法描述工具有很多，如自然语言、流程图等。但比较规范的且现在用的比较多的是 N-S 图。

2. N-S 图

N-S 图又叫“盒图”，即由一个一个的盒子组成。它有如下几种基本的符号。



几个常用算法如下所述。

- (1) 输入 a 、 b 两个整数，然后将它们的值互换。
- (2) 输入 a 、 b 两个整数，输出它们当中较大的那一个。
- (3) 求 $1+2+3+4+5+\cdots+100$ 。

3 个题目算法的 N-S 图分别见图 1-5、图 1-6 和图 1-7。

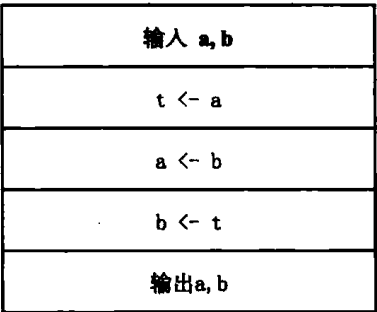


图 1-5 两数互换

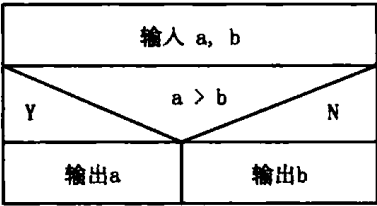


图 1-6 求两数中的较大数

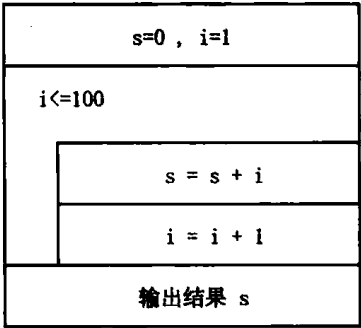


图 1-7 求前 100 个自然数之和

请用 N-S 图描述解决下列问题的算法。

练习 1-1 若输入 a, b, c 为 3 个整数，请输出它们当中最大的一个。

练习 1-2 从键盘输入若干个整数，若其中有负数，则输出该负数，并统计出负数的个数。

1.5 小结

本章重点介绍了 C 语言的基本语法知识，包括 C 语言的字、词、句，认识并掌握它们是编程的基础。了解编程解决问题的一般步骤，掌握用 N-S 图这一工具来表达算法。熟悉用 Visual C++ 6.0 来编辑、编译、连接和运行 C 语言程序，特别介绍了动态调试程序的单步执行和设置断点的方法。

习

题

1. C 语言源程序文件的扩展名是__，经过编译后，生成文件的扩展名是__，经过连接后，生成文件的扩展名是__。

2. 结构化程序由 3 种基本结构组成，分别是__、__、__。

3. 简述编程解决问题的步骤，设计求前 100 个自然数中奇数之和的算法，用 N-S 图描述。

4. 上机改错题（在 VC 中上机试一下，记录并熟悉出错信息）。

```
#include <stdio.h> ;
Main() / *main function * /
    float r , area ;
    r = 5.0 ;
    area = 3.14 ;
    printf("% f \n" , s );
```

5. 指出下面程序的错误。

```
main
{ float a , b , c , v ; /* a,b,c are sides , v is volume of cube */
  A = 2.0 ; b = 3.0 ; c = 4.0
  v = a * b * c ;
  Printf("%f \n" , .v )
}
```


第 2 章

基本 C 语言程序设计

学习目标

- C 语言中的常量、变量和基本数据类型;
- 实现各种数据类型的输入和输出;
- 掌握各种运算符和表达式, 特别是几个特殊运算符;
- 理解不同数据类型之间的运算和转换;
- 熟悉 C 语言中的数学库函数, 掌握几个常用库函数。

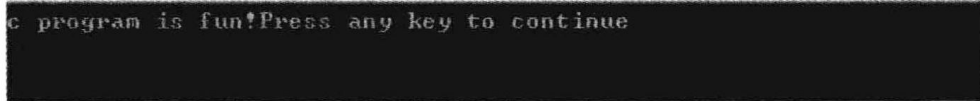
重点难点

- 重点: 表达式和语句, 各种数据类型: 整型、实型、字符型和字符串, 常量和变量。
- 难点: 基本运算符的运算规则和优先级。

2.1 基本输出函数

2.1.1 案例描述

- (1) 提出问题: 在屏幕上显示 c program is fun!
- (2) 该案例执行后的界面如图 2-1 所示。



```
c program is fun!Press any key to continue
```

图 2-1 c program is fun!

- (3) 涉及的知识点, 库函数和头文件, 主函数和自定义函数, 以及各种类型数据的输出函数。

2.1.2 各种类型数据的输出函数 printf()

在程序运行过程中, 需要在计算机与用户之间进行一些交互操作, 比如, 用户通过输入设备输入某种类型的数据供程序使用, 计算机也将运行结果通过输出设备反馈给用户。有些高级程序设计语言提供了专门的输入/输出语句, 但在 C 语言中, 输入/输出功能是由一些函数实现的。这些函数是由编译系统提供, 放在函数库 (.lib) 文件中, 用户可以直接调用这些函数完成相应的功能。但需要使用一个函数时, 应先通知系统中该函数所在的函数库, 这是通过包含头文件的方式来实现的。