



高等院校计算机应用技术规划教材

# 研究式学习— C语言程序设计

李向阳 主编 张启富 副主编



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

高等院校计算机应用技术规划教材

# 研究式学习——C 语言程序设计

主 编：李向阳

副主编：张启富

参 编：楼 静 张燕涛 方娇莉 耿植林

**中国铁道出版社**  
CHINA RAILWAY PUBLISHING HOUSE

---

## 内 容 简 介

本教材用提出问题—分析问题—给出相关知识—举一反三的研究式学习方法组织“C 程序设计”这门课程,在注重语言知识培养的同时,更注重逻辑思维能力、程序设计能力及创新能力的培养。

本书共分 9 章,有 3 个层次:第 1、2、3 章是基础篇,第 4、5 章是提高篇,第 6、7、8、9 章是深入篇。第 1 章是程序入门,介绍 C 语言的基本语法、基本数据类型、基本操作和顺序结构;第 2、3 章介绍选择结构和循环结构的程序及基本算法;第 4 章介绍数组及相关算法;第 5 章介绍函数和变量的存储类型;第 6 章介绍用户定制数据类型;第 7 章介绍指针;第 8 章介绍文件;第 9 章介绍面向对象的程序设计。

本书可作为高等院校学生学习 C 语言的教材,与本书配套的还有《研究式学习——C 语言程序设计习题与实验》。

### 图书在版编目(CIP)数据

研究式学习: C 语言程序设计/李向阳主编;楼静等  
编. —北京:中国铁道出版社,2006.1 (2008.1 重印)  
高等院校计算机应用技术规划教材  
ISBN 978-7-113-06884-4

I. 研... II. ①李...②楼... 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(2006)第 003225 号

书 名: 研究式学习——C 语言程序设计

作 者: 李向阳 张启富 等

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 邸秋罗

责任编辑: 严 力 翟玉峰 熊严飞

封面设计: 薛 为

封面制作: 白 雪

责任校对: 李 旸

印 刷: 北京鑫正大印刷有限公司

开 本: 787×1092 1/16 印张: 14 字数: 334 千

版 本: 2006 年 1 月第 1 版 2008 年 1 月第 2 次印刷

印 数: 5 001~7 000 册

书 号: ISBN 978-7-113-06884-4/TP·1717

定 价: 20.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

本书尝试用研究式、启发式的学习方法来组织“C 语言程序设计”课程的教学,在注重程序设计能力,特别是创新能力培养的同时,也兼顾语言知识的培养,努力将本课程打造成精品课程。本书的参考学时为 70 学时,其中上课 35 学时,上机 35 学时,体现了“精讲多练”的教学方法。与教材配套的还有适合多媒体教学的电子课件及《研究式学习——C 语言程序设计习题与实验》一书。

本教材每章都通过 4 个步骤来完成学习。

第 1 步,通过提出问题、分析问题,吸引学生的兴趣,引起学生的思考。

第 2 步,介绍解决问题的方法、相关知识点的跟进,使学生能逐步提升程序设计的能力,同时也逐步掌握 C 语言的知识点,引导学生深入思考。

第 3 步,用一定数量的语法练习题使学生能对所学的知识举一反三;用一定数量的编程练习题,培养学生的逻辑思维能力和编程能力,挖掘学生的创新能力。

第 4 步,对每部分的知识做小结,以便学生进一步巩固应掌握的知识。

本书主要针对初学者,选用基本的、实用的习题和算法,所有例题均通过调试。

本书共分 9 章,有 3 个层次:第 1、2、3 章是基础篇,第 4、5 章是提高篇,第 6、7、8、9 章是深入篇,每篇的讲授大约需要 10 学时。

第 1 章是需要 4 学时的程序入门,介绍 C 语言的基本语法、基本数据类型、运算符等,并介绍了 C 程序的顺序结构。

第 2、3 章用 6 学时左右的时间介绍选择结构和循环结构的程序及基本算法。

第 4 章用 4 学时左右的时间介绍数组及相关算法。

第 5 章用 6 学时左右的时间介绍函数和变量的存储类型。

第 6 章用 4 学时左右的时间介绍用户定制数据类型。

第 7 章用 4 学时左右的时间介绍指针。

第 8 章用 2 学时左右的时间介绍文件。

第 9 章用 4 学时左右的时间介绍面向对象的程序设计。

本书第 1、5 章由李向阳编写,第 2、3 章由楼静编写,第 4 章由张燕涛编写,第 6、8 章由张启富编写,第 7 章由方娇莉编写,第 9 章由耿植林编写。全书由李向阳主编,并负责全书的统稿,张启富担任副主编,秦卫平审稿。

本书得到昆明理工大学精品课程的资助,得到教务处、教材科及计算中心领导和同志们的大力支持和帮助,在此,对他们表示感谢!同时对本书所有参考书籍、资料的作者们表示感谢!

由于时间仓促,水平有限,不足之处在所难免,敬请批评指正。

编 者

2005 年 11 月

|                           |    |
|---------------------------|----|
| 第1章 C语言程序入门 .....         | 1  |
| 1.1 程序 .....              | 1  |
| 1.1.1 程序的概念 .....         | 1  |
| 1.1.2 简单C程序 .....         | 2  |
| 1.2 程序操作的对象——数据 .....     | 3  |
| 1.2.1 变量和常量 .....         | 4  |
| 1.2.2 数据类型 .....          | 5  |
| 1.3 C程序的基本操作 .....        | 8  |
| 1.3.1 运算符与表达式 .....       | 8  |
| 1.3.2 算术运算符和算术表达式 .....   | 10 |
| 1.3.3 赋值运算符 .....         | 10 |
| 1.3.4 复合的赋值运算符 .....      | 11 |
| 1.3.5 自增运算符和自减运算符 .....   | 11 |
| 1.3.6 逗号运算符 .....         | 12 |
| 1.3.7 强制转换类型运算符 .....     | 12 |
| 1.3.8 容量运算符 .....         | 13 |
| 1.4 顺序结构的程序 .....         | 13 |
| 1.4.1 C程序语句 .....         | 14 |
| 1.4.2 输出/输入函数 .....       | 15 |
| 1.5 语法练习 .....            | 19 |
| 1.6 编程练习 .....            | 21 |
| 1.7 小结 .....              | 22 |
| 第2章 选择结构 .....            | 23 |
| 2.1 问题的提出 .....           | 23 |
| 2.1.1 问题 .....            | 23 |
| 2.1.2 问题分析 .....          | 23 |
| 2.1.3 程序 .....            | 24 |
| 2.1.4 程序测试 .....          | 24 |
| 2.2 与选择结构有关的运算符和表达式 ..... | 25 |
| 2.2.1 关系运算符和关系表达式 .....   | 25 |
| 2.2.2 逻辑运算符和逻辑表达式 .....   | 26 |
| 2.2.3 条件运算符和条件表达式 .....   | 27 |
| 2.3 选择语句 .....            | 28 |
| 2.3.1 if语句 .....          | 28 |
| 2.3.2 switch语句 .....      | 34 |

|              |                             |           |
|--------------|-----------------------------|-----------|
| 2.4          | 语法练习 .....                  | 36        |
| 2.5          | 编程练习 .....                  | 38        |
| 2.6          | 小结 .....                    | 43        |
| <b>第 3 章</b> | <b>循环结构 .....</b>           | <b>44</b> |
| 3.1          | 问题的提出 .....                 | 44        |
| 3.1.1        | 问题 .....                    | 44        |
| 3.1.2        | 问题分析 .....                  | 44        |
| 3.1.3        | 程序 .....                    | 45        |
| 3.1.4        | 程序测试 .....                  | 46        |
| 3.2          | 循环语句 .....                  | 47        |
| 3.2.1        | for 语句 .....                | 47        |
| 3.2.2        | while 语句 .....              | 51        |
| 3.2.3        | do...while 语句 .....         | 53        |
| 3.2.4        | 循环嵌套 .....                  | 55        |
| 3.2.5        | break 语句和 continue 语句 ..... | 58        |
| 3.3          | 语法练习 .....                  | 61        |
| 3.4          | 编程练习 .....                  | 64        |
| 3.5          | 小结 .....                    | 69        |
| <b>第 4 章</b> | <b>数组 .....</b>             | <b>70</b> |
| 4.1          | 一维数组 .....                  | 70        |
| 4.1.1        | 问题的提出 .....                 | 70        |
| 4.1.2        | 问题分析 .....                  | 70        |
| 4.1.3        | 程序 .....                    | 71        |
| 4.1.4        | 一维数组的定义及引用 .....            | 72        |
| 4.1.5        | 一维数组程序举例 .....              | 74        |
| 4.2          | 二维数组 .....                  | 78        |
| 4.2.1        | 问题的提出 .....                 | 78        |
| 4.2.2        | 问题分析 .....                  | 78        |
| 4.2.3        | 程序 .....                    | 78        |
| 4.2.4        | 二维数组的定义及引用 .....            | 79        |
| 4.3          | 字符数组 .....                  | 83        |
| 4.3.1        | 问题的提出 .....                 | 83        |
| 4.3.2        | 问题分析 .....                  | 83        |
| 4.3.3        | 程序 .....                    | 83        |
| 4.3.4        | 字符数组的定义及引用 .....            | 84        |
| 4.3.5        | 字符数组程序举例 .....              | 87        |
| 4.3.6        | 用于字符串处理的函数 .....            | 90        |

|              |                      |           |
|--------------|----------------------|-----------|
| 4.4          | 语法练习 .....           | 92        |
| 4.5          | 编程练习 .....           | 94        |
| 4.6          | 小结 .....             | 96        |
| <b>第 5 章</b> | <b>函数 .....</b>      | <b>98</b> |
| 5.1          | 问题的提出 .....          | 98        |
| 5.1.1        | 问题 .....             | 98        |
| 5.1.2        | 问题分析 .....           | 98        |
| 5.1.3        | 程序 .....             | 99        |
| 5.2          | 函数的定义 .....          | 99        |
| 5.2.1        | 函数头 .....            | 99        |
| 5.2.2        | 函数体 .....            | 100       |
| 5.3          | 函数原型、函数声明与函数调用 ..... | 101       |
| 5.3.1        | 函数原型 .....           | 101       |
| 5.3.2        | 函数的声明 .....          | 101       |
| 5.3.3        | 函数的调用 .....          | 101       |
| 5.4          | 数组名作函数的参数 .....      | 103       |
| 5.4.1        | 问题的提出 .....          | 103       |
| 5.4.2        | 问题分析 .....           | 103       |
| 5.4.3        | 程序 .....             | 104       |
| 5.5          | 函数的嵌套调用和递归调用 .....   | 104       |
| 5.6          | 库函数 .....            | 106       |
| 5.7          | 变量的作用域 .....         | 107       |
| 5.7.1        | 问题的提出 .....          | 107       |
| 5.7.2        | 问题分析 .....           | 107       |
| 5.7.3        | 程序 .....             | 107       |
| 5.7.4        | 局部变量和全局变量 .....      | 108       |
| 5.8          | 变量的存储类型 .....        | 108       |
| 5.8.1        | 问题的提出 .....          | 108       |
| 5.8.2        | 问题分析 .....           | 109       |
| 5.8.3        | 程序 .....             | 109       |
| 5.8.4        | 局部变量的存储类型 .....      | 110       |
| 5.8.5        | 全局变量的存储类型 .....      | 110       |
| 5.9          | 编译预处理 .....          | 112       |
| 5.9.1        | 宏定义命令 .....          | 112       |
| 5.9.2        | 文件包含 .....           | 113       |
| 5.10         | 语法练习 .....           | 114       |
| 5.11         | 编程练习 .....           | 118       |
| 5.12         | 小结 .....             | 119       |

|                        |     |
|------------------------|-----|
| <b>第 6 章 用户定制数据类型</b>  | 120 |
| 6.1 枚举                 | 120 |
| 6.1.1 问题的提出            | 120 |
| 6.1.2 问题的分析            | 120 |
| 6.1.3 程序               | 121 |
| 6.1.4 枚举类型的定义及使用规则     | 122 |
| 6.2 结构体                | 122 |
| 6.2.1 问题的提出            | 122 |
| 6.2.2 定义一个结构体类型        | 123 |
| 6.2.3 结构体变量的定义、初始化与引用  | 124 |
| 6.2.4 用结构体数组处理一组学生的学籍  | 126 |
| 6.2.5 函数间结构体变量的数据传递    | 128 |
| 6.3 共用体                | 129 |
| 6.3.1 问题的提出            | 130 |
| 6.3.2 问题分析             | 130 |
| 6.3.3 程序               | 130 |
| 6.3.4 共用体类型的定义与引用      | 132 |
| 6.4 语法练习               | 133 |
| 6.5 编程练习               | 134 |
| 6.6 小结                 | 135 |
| <b>第 7 章 指针</b>        | 137 |
| 7.1 指针作函数的参数           | 137 |
| 7.1.1 问题的提出            | 137 |
| 7.1.2 问题分析             | 137 |
| 7.1.3 程序               | 138 |
| 7.1.4 指针的定义及运算         | 139 |
| 7.2 数组与指针              | 140 |
| 7.2.1 指向一维数组的指针        | 140 |
| 7.2.2 指针指向数组时的运算       | 142 |
| 7.2.3 指向二维数组的指针        | 142 |
| 7.3 用指针实现字符串的操作        | 143 |
| 7.4 指针和函数              | 145 |
| 7.4.1 用指向函数的指针实现函数调用   | 145 |
| 7.4.2 返回指针值的函数         | 147 |
| 7.5 指针数组和指向指针的指针       | 148 |
| 7.5.1 指针数组的概念          | 148 |
| 7.5.2 指向指针的指针          | 150 |
| 7.5.3 指针数组作 main 函数的形参 | 151 |



|              |                          |            |
|--------------|--------------------------|------------|
| 7.6          | 用指针实现数据的动态管理 .....       | 152        |
| 7.6.1        | 问题的提出 .....              | 152        |
| 7.6.2        | 问题分析 .....               | 152        |
| 7.6.3        | 程序 .....                 | 153        |
| 7.6.4        | 动态数据管理在插入、删除操作中的优点 ..... | 157        |
| 7.7          | 学以致用：综合举例 .....          | 157        |
| 7.8          | 语法练习 .....               | 159        |
| 7.9          | 编程练习 .....               | 161        |
| 7.10         | 小结 .....                 | 161        |
| <b>第 8 章</b> | <b>文件 .....</b>          | <b>164</b> |
| 8.1          | 保存若干个字符到磁盘中 .....        | 164        |
| 8.1.1        | 问题的提出 .....              | 164        |
| 8.1.2        | 问题分析 .....               | 164        |
| 8.1.3        | 程序 .....                 | 164        |
| 8.2          | 把若干个字符串保存到文本文件 .....     | 166        |
| 8.2.1        | 问题的提出 .....              | 166        |
| 8.2.2        | 问题分析 .....               | 166        |
| 8.2.3        | 程序 .....                 | 166        |
| 8.3          | 文件的概念、定义及使用规则 .....      | 168        |
| 8.3.1        | C 语言文件的概念 .....          | 168        |
| 8.3.2        | 文件的打开与关闭 .....           | 171        |
| 8.3.3        | 文件的读写操作标准函数 .....        | 172        |
| 8.3.4        | 文件的随机读写 .....            | 173        |
| 8.4          | 语法练习 .....               | 175        |
| 8.5          | 编程练习 .....               | 178        |
| 8.6          | 小结 .....                 | 178        |
| <b>第 9 章</b> | <b>面向对象的程序设计 .....</b>   | <b>180</b> |
| 9.1          | 问题的提出 .....              | 180        |
| 9.1.1        | 问题 .....                 | 180        |
| 9.1.2        | 问题分析 .....               | 180        |
| 9.1.3        | 程序 .....                 | 182        |
| 9.1.4        | 程序调试与运行 .....            | 183        |
| 9.1.5        | 对该问题的进一步讨论 .....         | 184        |
| 9.2          | 类和对象 .....               | 186        |
| 9.2.1        | 定义类 .....                | 187        |
| 9.2.2        | 创建和引用对象 .....            | 187        |
| 9.2.3        | 构造函数与析构函数 .....          | 189        |
| 9.3          | 继承与派生 .....              | 191        |

|                                       |     |
|---------------------------------------|-----|
| 9.3.1 单一继承 .....                      | 191 |
| 9.3.2 多重继承 .....                      | 193 |
| 9.3.3 友元函数和友元类 .....                  | 194 |
| 9.4 多态性 .....                         | 195 |
| 9.4.1 函数重载和运算符重载 .....                | 196 |
| 9.4.2 虚函数 .....                       | 197 |
| 9.5 面向对象的程序设计方法 .....                 | 198 |
| 9.5.1 C 语言的发展 .....                   | 198 |
| 9.5.2 面向对象的设计方法 .....                 | 199 |
| 9.5.3 用 Microsoft Visual C++ 编程 ..... | 199 |
| 9.6 语法练习 .....                        | 201 |
| 9.7 用面向对象方法编程序 .....                  | 204 |
| 9.8 小结 .....                          | 204 |
| 附录 A ASCII 码表 .....                   | 205 |
| 附录 B C 语言中的关键字 .....                  | 207 |
| 附录 C C 标准库函数 .....                    | 208 |
| 参考文献 .....                            | 214 |

# 第 1 章

## C 语言程序入门

什么叫程序？怎样进行程序设计？C 语言程序是什么样的？编写简单 C 语言程序应具备哪些基本知识？这是本章要回答的问题。

### 1.1 程 序

#### 1.1.1 程序的概念

简单地讲，程序就是做一件事情的先后步骤或称为过程，例如，植树这件事情，要经过以下步骤：

挖坑 → 在坑中放入树苗 → 添土到坑中 → 浇水

这些步骤就构成植树程序，程序中的每一步一般不能颠倒，例如不先挖坑，树苗就无法植入土中。人们说某人对做某事已“胸有成竹”，就是说他已经知道了做事的“程序”。同样自从计算机诞生以来，要计算机完成的每件工作都必须用计算机语言编出程序，将“程序”输入计算机内存后，计算机才能按程序完成任务。在信息社会，不仅应该会用别人设计好的软件，还应该会按自己的需要设计一些小程序，以完成某些特殊的需要。

要学会编计算机程序，首先要有编程的思路（算法），并能用流程图等方法把思路表达出来。如例 1.1 所示。

【例 1.1】任意输入两个数，输出其中的大数的算法如图 1-1 所示。

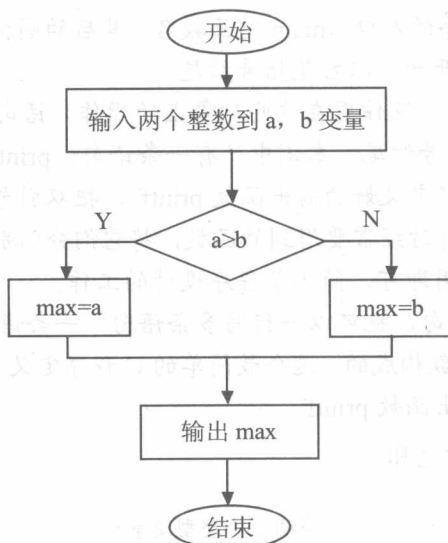


图 1-1 两数中取大者

在这个例子中，a 和 b 分别用来保存两个整数，称为变量。max 称为中间变量，用来存放两个数中的大者。在流程图中矩形框为处理框，菱形框为判断框，箭头为流程线，流程图

描述了完成某一任务对数据的操作过程，表达了编程的算法，较复杂的编程题要经过自顶向下、逐步求精的分析过程，才能得到结果。本书中只介绍一些简单的、常用的算法。有解题的算法，还必须用计算机可以理解的语言来表达，使用语言不同，其表达的形式就有所不同。计算机语言是随着计算技术的发展和應用不断发展的，本书介绍的 C 语言是一种既能写应用软件又能写系统软件的高级语言。著名的 UNIX 操作系统就是用 C 语言写成的，C 语言是 C++ 语言和 Java 语言的“前辈”。下面从最简单的 C 程序入手，来学习 C 语言程序设计。

### 1.1.2 简单 C 程序

**【例 1.2】**在显示器上输出：hello world。

首先要编辑该程序（可用任何编辑文本的软件）：

```
main()
{
    printf("hello world");
}
```

程序经过编译、连接后，运行该程序就会在显示器上显示：

hello world

**说明：**

(1) C 程序的基本形式为

```
main()
{
    <语句系列>
}
```

“main()”是 C 程序的主函数，任何 C 程序都有且仅有一个主函数。任何程序都从 main 函数开始执行程序，它是程序的入口，main 是函数名，其后的圆括号表示这是一个没有参数的函数，函数体以左花括号开始，以右花括号结尾。

(2) 语句用于描述对象以及作用在这些对象上的操作，语句写在函数体中（函数名后的一对花括号中），语句以分号结尾。本例中只有一条语句：printf("hello world");。

这条语句是调用系统已经定义好的输出函数 printf，把双引号中的内容输出到显示器。系统预先定义了很多程序设计时经常要用到的函数，将它们分门别类地保存在函数库中，称为库函数，需要用的时候调用即可，简化了程序设计的工作。

(3) 可以一行写一条语句，也可以一行写多条语句，一条语句写在多行。

(4) C 语言程序是由函数构成的，这个最简单的 C 程序定义了一个 main 函数，在 main 函数中调用了系统提供的输出函数 printf。

**【例 1.3】**输出两个整数之和。

```
main()
{
    int a,b,sum;           /*声明 3 个整型变量*/
    a=123;b=456;           /*使用已定义的 a,b 变量(给 a,b 变量赋值)*/
    sum=a+b;
    printf("sum =%d\n",sum);
```

程序运行结果

```
sum=579
```

说明:

(1) `int a,b,sum;`是声明语句,它定义了3个整型(`int`)变量 `a`, `b`, `sum`。变量具有以下特点:

- 变量是用来存储数据的内存空间,空间的大小(字节数)由变量的类型决定,类型有整型(`int`),实型(`float`),字符型(`char`)等。
- 变量的值就是该变量的存储空间中存放的数据,通过赋值操作可改变变量的值(`a=123;b=456;`)。
- 每个变量必须有名称,以便存取值,该例中变量名有 `a`、`b`、`sum` 三个,变量的命名有一定的规则,必须按规则命名。

(2) 本例中的 `printf` 函数比上一个例子复杂,双引号中的“`sum=`”与上例一样会按原样输出到显示器上,“`%d`”表示要输出一个整型值,这个值在双引号后的 `sum` 变量中。该例的输出是: `sum=579`。“`\n`”表示换行,它使下一个输出在下一行出现。

(3) `/*.....*/` 称为注释,通常是给程序阅读者的说明,用于增加程序的可读性。此外在调试程序时,怀疑某语句有问题,可将该语句变为注释语句,如 `/* printf("sum=%d",sum); */`,则该 `printf` 语句将不被执行,以后需要恢复该语句功能时,只需删去“`/*`”、“`*/`”即可。

**【例 1.4】**从键盘输入任意两个整数,求它们的积。

```
main()
{
    int a,b,product;
    printf("Please input two integers:\n");
    scanf("%d,%d",&a,&b);
    product=a*b;
    printf("product=%d\n",sum);
}
```

说明:

与例 1.3 不同, `a`, `b` 变量的值不是编程时就给定的,而是运行程序时,从键盘输入的。这个功能由 `scanf("%d,%d",&a,&b);` 语句完成,程序运行到该句时,将等待从键盘输入两个整型数据,系统将这两个数存到 `a`, `b` 两个变量中。输入两个数时,用逗号分隔两个数,用回车键结束输入。`scanf` 是系统已经写好的格式输入函数,双引号中的两个“`%d`”指明双引号后的两个变量接受的值是整型,变量名前的“`&`”是取地址运算符,`scanf` 函数要求需要输入值的变量前必须用该运算符。“`*`”是乘运算符。`printf("Please input two integers:\n");` 语句是在输入数据时给用户的提示,使用户知道应该输入两个整型数据。程序运行后,屏幕显示

```
Please input two integers:      /* 程序输出 */
5,7   <CR>                    /* 用户输入 (<CR>为回车键) */
sum=35                          /* 程序输出 */
```

## 1.2 程序操作的对象——数据

从例 1.4 可知,程序操作的对象是数据(例 1.4 中的 `a`, `b`, `sum`)。学习程序设计的第一步,应该学会用该程序设计语言描述的数据来模拟处理对象,然后再学习对数据的操作,这

一节介绍基本数据类型和基本操作。

就像商店出售的衣服有不同的尺寸,以供不同身材的人享用一样,C 语言也根据数据存在的不同形式,将其分为不同的类型,各种数据类型具有不同的存储长度、取值范围和允许的操作,不同版本的 C 语言提供的数据类型的长度、取值范围不同,本书按照标准 C (ANSI) 的规定给出数据类型的长度、取值范围。C 语言提供的数据类型如图 1-2 所示。

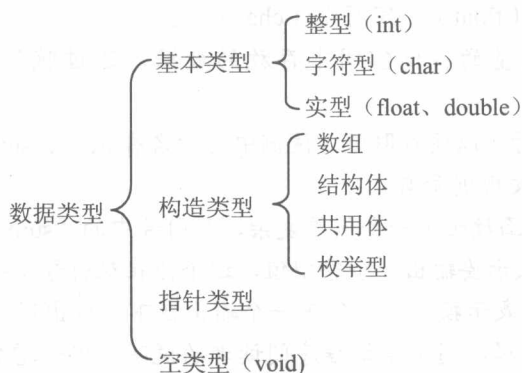


图 1-2 数据类型

本节只介绍基本类型中的整型、字符型和实型。

### 1.2.1 变量和常量

程序中使用的数据有两种形式:变量和常量。常量具有固定值,不能改变;变量的值可以改变。

#### 1. 变量

程序通过变量来使用内存空间,存储数据。变量是“容器”,变量中存放的数据就是变量的值,每个变量都有一个名称,这个名称和该变量的内存空间相联系,程序通过变量名对相应的存储空间进行存取操作。

使用变量首先要正确地给变量命名,变量的命名规则遵循标识符的命名规则。标识符就是用来标识变量名、符号常量名、函数名、数组名、文件名等由程序设计者命名的名称。

C 语言的标识符规定:

标识符只能由英文字母、下划线和数字组成,且只能以字母和下划线开头。由于系统定义的变量大多以下划线开头,建议给标识符命名时不要以下划线开头,以免冲突。

变量名区分大小写,a 和 A,p 和 P 是不同的变量。通常变量都用小写,以增加程序的可读性。

标识符的可辨认长度在不同的系统下有不同的规定,最小为 8 个,最长为 32 个,因此建议不要超过 8 个。

不能使用系统保留字(关键字)作为标识符。这些保留字系统已有专门的含义。如:auto,break,int,short 等。命名时,最好能“见名思义”,如 sum、name、age 等。

变量要先声明(定义),后使用。声明的内容包括:变量的类型(int,float),变量的名称以及初值。如在函数体内有“int a,sum=0;”则定义 a 和 sum 变量为整型变量,可以存放整型数据,sum 的初值为 0,a 的初值不确定。在该定义语句后就可以使用 a 和 sum 变量。

对变量的基本操作是赋值，通过赋值运算符(=)可以改变变量的值，如：

```
int x;
x=3;      /* 3 存入 x 变量，x 变量的值为 3 */
x=5;      /* 5 存入 x 变量，x 变量的值为 5 */
x=x+1;    /* x 变量的值加 1 后存入 x 变量，x 变量的值为 6 */
x=x*x;    /* 该语句执行后，x 的值为多少？ */
```

2. 常量

其值不能改变的量称为常量。从字面形式就可以判别的常量称为字面常量，如：3，8，-5 为整型常量，5.3，-0.2 为实型常量，用标识符代表的常量称为符号常量。

【例 1.5】已知圆的半径，求圆的面积。

```
#define PI 3.14
main()
{
    float areal,area2;
    areal=2*2*PI;
    area2=5*5*PI;
    printf("areal=%f,area2=%f\n",areal,area2);
}
```

程序的执行结果如下：

```
areal=12.560000,area2=78.500000
```

main 函数前的“#define PI 3.14”称为宏定义命令，它定义 PI 代表常量 3.14，符号常量不能改变，如 PI=3.1415;将是错误的。宏定义命令在编译前进行字符替换，属于编译预处理命令的一种。编译预处理命令写在程序开头，以“#”开始，命令后面不加分号，一行只能写一句。此外符号常量还可以用 const 和类型说明符来声明，如 const float PI=3.14;这种形式声明的变量 PI 也不能任意改变其值，通常符号常量用大写字母表示。

1.2.2 数据类型

1. 整型数据

有 3 种形式的整型常量：

- (1) 与习惯用法相同的十进制整数。如：46，-23，0。
- (2) 以数字 0 开头的八进制整数。如：0456 表示八进制数 456，即 $(456)_8=4\times 8^2+5\times 8^1+6\times 8^0$ ，运算结果等于十进制的 302。
- (3) 以数字 0 和 x 开头的十六进制整数。如：0x456 表示十六进制整数 456，即 $(456)_{16}=4\times 16^2+5\times 16^1+6\times 16^0$ ，运行结果等于十进制的 1 110。

当要存储整型数据到变量时，可根据数据的大小选择表 1-1 所示的定义形式定义整型变量。

表 1-1 ANSI 标准定义的整数类型

| 类 型          | 字 节 数 | 取 值 范 围                                  |
|--------------|-------|------------------------------------------|
| int          | 2     | -32 768~32 767 即 $-2^{15}\sim(2^{15}-1)$ |
| unsigned int | 2     | 0~65 535 即 $0\sim(2^{16}-1)$             |
| short [int]  | 2     | -32 768~32 767 即 $-2^{15}\sim(2^{15}-1)$ |

续上表

| 类 型                  | 字 节 数 | 取 值 范 围                                                |
|----------------------|-------|--------------------------------------------------------|
| unsigned short [int] | 2     | 0~65 535 即 $0\sim(2^{16}-1)$                           |
| long [int]           | 4     | -2 147 483 648~2 147 483 647 即 $-2^{31}\sim(2^{31}-1)$ |
| unsigned long [int]  | 4     | 0~4 294 967 295 即 $0\sim(2^{32}-1)$                    |

方括号中的部分可以省略，如：long x;和 long int x;都表示 x 是 long int 型，当不使用 unsigned（无符号）定义变量时，该变量的存储单元最高位为符号位，0 正 1 负，这样表示数的位数少 1 位，如 int a;，则 a 的取值范围为-32 768~32 767 即 $-2^{15}\sim(2^{15}-1)$ 。当使用 unsigned 定义变量时，该变量的存储单元全部用来表示数，无符号位，只有正数，则正数的取值范围就增大，如 unsigned int a;，则 a 的取值范围为 0~65 535 即  $0\sim(2^{16}-1)$ 。当整型数据要存入整型变量时，请注意所定义的整型变量应能容纳该数。

【例 1.6】按照 ANSI 标准，以下程序有什么错误？请改正。

```
main()
{
    int j;
    unsigned x;
    j=40000;
    x=70000 ;
    printf("%d,%d\n",j,x);
}
```

2. 实型数据

(1) 实型常量

实型常量只有十进制，无其他进制，但有如下两种表示形式：

- 小数形式，它由数字和小数点组成（必须有小数点）。如 12.3、.12、456.、124.0、0.0 等是合法的小数形式，而 1/2 是表达式，不是实数。
- 指数形式（科学计数法），如 123e3、123E3、123E+3 都表示  $123\times10^3$ ，注意 e（E）前必须有数，e（E）后的指数必须为整数，e2、2.1e3.5、.e3、e 等都是非法的实数。同一个实数有多种表示形式，如 456.123 可以表示为 4.561 23e2、45.612 3e1、4 561.23e-1 等，所以指数表示法也称为浮点数（小数点位置浮动）。用指数形式输出一个实数时，按规范化的形式输出，即小数点左边有且只有一位非零的数字。如：2.478e2、3.099e5。

(2) 实型变量

实型变量可定义为单精度型（float）、双精度型（double）和长双精度型（long double），有关规定如表 1-2 所示。

表 1-2 实型数据（ANSI 标准）

| 类 型         | 字 节 数 | 有 效 数 字 | 取 值 范 围                   |
|-------------|-------|---------|---------------------------|
| float       | 4     | 6~7     | $10^{-37}\sim10^{38}$     |
| double      | 8     | 15~16   | $10^{-307}\sim10^{308}$   |
| long double | 16    | 18~19   | $10^{-4931}\sim10^{4932}$ |

使用实型变量前可根据存入数据的大小定义，如：



```
float x,y;
double z;
long double t;
```

实型变量是用有限的存储单元存储数据的，因此有效数字总是有限的，而且小数部分在十进制转换为二进制时会有误差，所以在程序设计时要注意以下几点。

- 很大的数和很小的数相加减是无意义的。

【例 1.7】求变量 b 的值。

```
main()
{ float a,b;
  a=123456.789e5;
  b=a+20;
  printf("%f\n",b);
}
```

程序运行结果如下：

```
12345678848.000000
```

- 应避免两个实数作数值是否相等的比较，如：

```
float x,y;
...
if(x==y) printf("%f,%f",x,y);
```

由于存在误差，x、y 两个实数可能永远不相等，这样永远不会执行 printf("%f,%f",x,y); 语句。

3. 字符型数据

(1) 字符型常量

字符型常量的形式如下：

- 用单引号括起来的一个字符，如：'a'、'+'、'2'、' '。
- 以“\”开头的特殊字符（转义字符），这些字符有的转为控制码，如'\n'、'\t'，有的转为字符自身('\\')，总之用转义字符可输出任何用 ASCII 码表示的字符，如'\374'代表'Π'。如表 1-3 所示是转义字符及其含义。

表 1-3 转义字符及其含义

| 转 义 字 符 | ASCII 码 值 | 功 能    |
|---------|-----------|--------|
| \a      | 0X07      | 警告响铃   |
| \b      | 0X08      | 退格     |
| \f      | 0X0C      | 走纸     |
| \n      | 0X0A      | 换行     |
| \r      | 0X0D      | 回车，不换行 |
| \t      | 0X09      | 水平制表   |
| \v      | 0X0B      | 垂直制表   |
| \\      | 0X5C      | 反斜杠    |
| \'      | 0X27      | 单引号    |
| \"      | 0X22      | 双引号    |