

前 言

本书是为中等职业学校学生面向新世纪计算机教学的发展、适应信息化社会的需求而编写的。C语言是多年来国内外广泛应用的计算机语言,是程序开发者必须掌握的一种高级语言;而数据结构则是计算机专业的一门必修的基础理论课。本教材将“计算机应用”专业的两大骨干课程——“C语言程序设计”和“数据结构”的内容融为一体,力求学生在学习C语言程序设计的同时,理解数据结构的基本概念,并对C语言所描述的各种数据结构的基本存储方式、基本算法及简单应用进行学习,培养学生理解和分析的能力。

本教材共分十二章。前四章讲述C语言的基础知识、程序设计的基本结构及其应用;第五章讲述数组数据类型,主要介绍数组及其应用、线性表、栈、队列及常用数据查询算法;第六章讲述结构体与共用体数据类型;第七章讲述指针数据类型;第八章讲述树;第九章讲述图;第十章、第十一章讲述文件、位运算;第十二章为应用举例,主要讲述图形函数与图形设计方法、常用的窗口设计和菜单设计。

本书的学习重点定位在C语言应用和数据结构的基本知识上,讲述C语言基础知识、C语言编程方法及数据结构算法的实现。本书以《全国计算机等级考试大纲》C语言二级大纲为依据进行内容的组织和技能的介绍,并针对中职学生年轻、富有求知欲的特点,在给出基本语句格式的基础上各章中均给出应用例题,引导学生应用基础知识来解决实际问题,提高学习水平。在各应用例题中,给出较为详细的注释,以帮助学生更好地理解例题。通过本教材的学习,可有机地将数据结构知识和C语言知识结合在一起,有助于学生正确运用C语言编写出较为规范的C程序;读懂数据结构的算法程序,熟练掌握各类排序、查找算法。

“C语言程序设计”和“数据结构”两课程的内容合为一书是一种尝试,希望我们的尝试给教师的教和学生的学带来方便和学习进步,为教学改革带来活力。

全书需100学时,其中上机实验不少于30学时;若条件允许,可在课程结束后集中进行2周的课程设计或以大作业的形式上机进行程序调试。

参加本书编写的有孙经钰、白秀轩、张平、陈春燕、王宇昕、李东瀛,由孙经钰进行统稿。北京信息工程学院的卞孔武教授认真仔细地对全书进行了审定,并提出了许多宝贵的意见。对此特表示诚挚地感谢。在本书的编写出版过程中,得到各学校领导的大力支持,在此表示感谢。

由于编者的水平有限,若有不妥之处,恳请广大读者批评指正。

编 者

2001年5月

第一章 概 述

1.1 C 语言和数据结构简介

1.1.1 C 语言的发展

C 语言产生于 20 世纪 70 年代,它的根源可追溯到 ALGOL 60。ALGOL 60 是 1960 年出现的一种面向过程的高级语言。该语言不能直接对硬件进行操作,不宜用来编写系统程序。系统程序主要用汇编语言编写,但汇编语言是面向某一机型的,其程序的可读性和可移植性都比较差。因此,一种集高级语言和低级语言特性于一身的语言——C 语言便应运而生了。

最初版本的 C 语言是 1972 年由美国贝尔实验室的丹尼斯·里奇设计,在 PDP-11 机器上实现的一种小型语言。后来经过了多次改进,直到 1978 年,贝尔实验室的布莱恩·W·科宁汉和丹尼斯·里奇合著了《C 程序语言》(《The C Programming Language》)。从此 C 语言开始作为一种通用的计算机语言得到广泛应用。

1.1.2 C 语言的特点

C 语言自问世以来,之所以得到迅速普及,并表现出顽强的生命力,是由于 C 语言具有其它语言不能比拟的特点。其特点如下:

C 语言简洁,使用灵活、方便,共有 32 个关键字,10 种基本语句,压缩了一切不必要的成分,便于阅读和书写。

C 语言具有丰富的数据类型,不仅有基本数据类型,如整型、字符型、实型等;还具有构造类型,如数组、结构体、共用体;以及指针类型等丰富的数据类型。可用来实现各种复杂的数据结构。因此 C 语言具有很强的数据处理能力。

C 语言提供丰富的运算符,共有 24 种运算符。除算术运算、逻辑运算外,还可以实现位运算、指针运算、复合运算等各种运算和控制操作。

C 语言程序设计模块化,具有顺序、选择、循环三种基本控制结构,足以描述良好的程序。一个 C 程序是由若干个函数和外部说明构成,函数则成为程序的最小模块,可进行模块化程序设计。

C 语言可直接对某物理地址的单元及物理地址单元中的位进行操作,能实现汇编语言的部分功能,因此 C 语言是集高级语言与低级语言特性于一身的一种语言。

C 语言编写的程序,经过编译、连接,可产生与源文件同名的可执行文件,其运行速度较高。同时 C 程序本身并不依存于计算机的硬件系统,其编译系统几乎可以在所有的操作系统以及各种类型的机器上运行,从而在不同机型间实现程序的移植。

1.1.3 数据结构简介

C 语言在数据结构方面有着广泛的应用。数据结构和算法是当今计算机科学研究的基本

课题,两者之间有着非常密切的联系,一个好的程序离不开好的数据结构和算法。那么,什么是数据结构呢?数据结构就是数据的组织形式,包括:字符串、数组、队列、栈、表、树、有向图、向量、链等。

数据结构通常研究三方面内容:数据的逻辑结构、数据的存储结构、数据运算。

1. 数据的逻辑结构,指按某种逻辑关系组织起来的一组数据。数据的逻辑结构分为线性结构和非线性结构两类。本书将介绍一些常见的线性结构,如队列、栈、链表;非线性结构,如二叉树、图等。

2. 数据的存储结构,指按一定的存储方式将数据在计算机中进行存储。

3. 数据的运算,指在一组数据上定义一个运算的集合。

同一个问题的数据集合可以组织成不同的数据结构,数据结构的选择和评价是一复杂的、综合性的问题。

1.2 C 程序的基本结构

1.2.1 C 程序的组成

用 C 语言编写的程序称为 C 语言源程序,简称 C 程序。C 程序以文件形式存放,C 程序的扩展名为 .C。一个 C 程序可由一个或多个文件组成,每个文件可以包含一个或多个函数,当 C 程序进行编译、连接时,将自动生成可执行的目标文件,以供执行。

C 程序的基本单位是函数。每个程序中必须有一个且只能有一个主函数,主函数名称为 main()。每个 C 程序由一个主函数和多个函数组成,也可以由一个主函数组成。

C 程序总是从主函数 main() 开始执行,可以调用系统提供的库函数(如 scanf 和 printf 等函数)和用户自己定义的函数,各函数的存放位置任意。

1.2.2 C 程序的函数结构

C 程序由函数组成;函数由函数的说明部分和函数体组成。函数的说明部分包括函数名、函数类型、参数名、参数类型的说明;函数体包括变量的说明和执行部分。

函数结构如下:函数类型 函数名(参数名表)

 参数说明;

 { 变量说明;

 执行语句部分;

 }

一个函数名后必须跟一对花括号,花括号 { } 中的内容称为函数体。参数名表可以没有,如 main()。执行语句部分可由若干语句组成。

例 1-1 下面 C 程序由主函数 main() 和函数 max() 构成。

```
main()                /* 函数名 */
{int x,y,z;           /* 定义变量 x,y,z 为整型变量 */
```

```

printf("input data:");
scanf("%d,%d",&x,&y);          /* 执行语句,输入 x,y 的值 */
z=max(x,y);                      /* 调用 max 函数,函数返回值存入 z 变量中 */
printf("Max is %d",z);          /* 打印输出最大值 */
}

max(a,b)                          /* max 为函数名 */
int a,b;                          /* 参数说明 */
{int c;                           /* 变量定义 */
c=a>b? a:b;                      /* 执行语句,判断 a 大于 b 吗? 是,a 值送给 c;否,b 值给 c */
return(c)                        /* 返回主调程序中 */
}

```

运行结果:input data:4,5

Max is 5

本程序的功能是随机输入两个数,输出大数。

1.2.3 C 语言的标识符

标识符是 C 语言中用来表示变量名、数组名、函数名的符号。标识符可由 1~32 个字符组成,但必须由字母或下划线开始,由字母、数字和下划线组成。

例如:A1,AB,A_1 是合法的标识符,而 A+B,2A,A B 是错误的标识符。

1.2.4 C 程序的基本语法规则

1. C 程序书写格式自由,一行内可以写多个语句。
2. 每条语句后必须使用“;”作为结束符。
3. 可以用/*……*/对 C 程序中的任何部分做注释,以提高程序的可读性。
4. C 语言通常严格区分大小写英文字母。使用小写字母进行程序的编写,而大写字母通常作为常量的宏定义和其它特殊用途。

C 程序中的引号、分号、逗号应为英文状态,请读者在录入程序时加以注意。

例 1-2 编写一个加法运算程序。

```

#include "stdio.h"
main()
{int x,y,z;                      /* 定义三个整型变量 x,y,z */
x=3; y=5;                        /* 给变量赋值 */
z=x+y;                           /* 计算 */
printf("%d+%d=%d\n",x,y,z);     /* 输出计算结果 */
}

```

输出结果为:3+5=8

1.3 C 程序上机步骤

1.3.1 Turbo C 2.0 的集成环境

Turbo C 2.0 是 Borland 公司推出的。它为用户创造了编辑、编译、调试、跟踪、存储、调用文件等多种功能的集成开发环境(IDE)。

每种功能均有菜单选项供用户选择。

一、Turbo C 2.0 的启动

通常 Turbo C 2.0 安装在 C 盘根目录下的 TC 子目录中。要启动 Turbo C, 须先启动 DOS; 需要使用汉字时则再启动汉字 DOS(如 UC DOS); 然后按下面两种方法进入 Turbo C 集成环境。

1. 直接进入 TC 子目录启动 Turbo C。

```
C:\>CD TC
```

```
C:\TC>TC
```

2. 先进入用户子目录(设用户子目录为 C:\WANG), 然后启动 Turbo C。

```
C:\>CD WANG
```

```
C:\WANG>C:\TC\TC          或 C:\WANG>TC
```

Turbo C 启动后, 屏幕将出现如图 1-1 的画面, 即主屏幕。

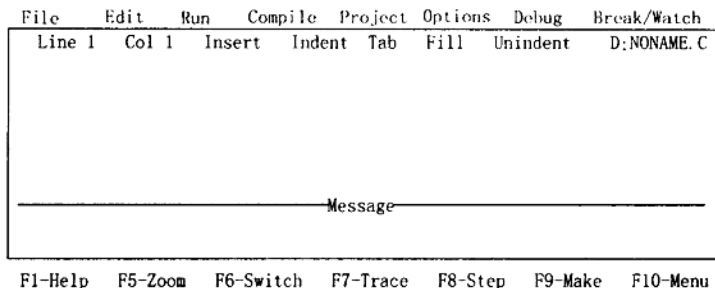


图 1-1 Turbo C 2.0 主屏幕

值得注意的是: 当我们把 Turbo C 安装在其它硬盘上, 如安装在 D 盘上, 则应该修改主菜单中 Options 菜单项下的 Directory 子菜单下文件的存放位置。如将 C:\TC\INCLUDE 改为 D:\TC\INCLUDE 等。

二、Turbo C 2.0 主屏幕

Turbo C 2.0 主屏幕由四部分组成: 主菜单、编辑窗口、编译信息窗口和功能键提示行。

1. 主菜单

主菜单是由横向菜单选项构成的菜单, 共有八项菜单项。

主菜单的选择方法有两种:

(1) 键入 Alt 及菜单选项上的大写字母。

(2) 用 F10 激活主菜单后,用←、→键将光标亮区移到所需选项上,按回车键。

2. 编辑窗口

此窗口是提供给用户的书写环境。窗口上部有一状态行,包括:

(1) Line 行号和 Col 列号,表示光标的当前位置。

(2) Insert 表示插入和改写状态,用 Ins 键进行两种状态的切换。

(3) Indent 和 Unindent 表示自动缩进格式转换,可实现设置或取消自动缩进排版,用 Ctrl+OI 完成切换。

(4) Tab 表示可用 Tab 键插入制表符或 Tab 无效,用 Ctrl+OT 完成切换。

(5) Fill 表示可用任意一组空格或制表符填充这一系列空格,如 Fill 打开,仅用空格填充,用 Ctrl+OF 完成切换。

(6) NONAME.C 是系统自动给出的、当前正在编辑的文件名。当文件保存后将允许用户重新为文件命名。

3. 编译信息窗口

编译信息窗口为 Message 下的空白区,用于显示编译链接和程序调试过程中出现的警告和错误信息。

用 F6 可在编辑窗口与编译信息窗口间切换。

4. 功能键提示行

功能键提示行位于屏幕底部,可给出当前操作中可以使用的主要功能键的功能说明。图 1-1 中显示的是主菜单下主要功能键的功能说明,如:F10 键的功能是激活主菜单。

三、Turbo C 2.0 子菜单

Turbo C 2.0 除 Edit 选项外,其它每个选项都有一个下拉式子菜单。下面将子菜单中常用选项的功能作一简单介绍,如图 1-2 所示。

File	Run	Compile
Load F3	Run Ctrl-F9	Compile to OBJ
Pick Alt-F3	Program reset Ctrl-F2	Make EXE File
New	Go to cursor F4	Link EXE File
Save F2	Trace into F7	Build all
Write to	Step over F8	Primary C File
Directory	User screen Alt-F5	Get into
Change Dir		
Os shell		
Quit Alt-X		

图 1-2 Turbo C 子菜单

1. 文件管理(File)

文件菜单的选择项如图 1-2(a)所示。

Load:用于装入指定的文件,装入时按屏幕提示输入路径及文件名。热键为 F3。

Pick:其功能是使用 Pick 表。Pick 表中存放着用户最近加载(操作)过的 8 个文件。在调用文件时,可直接用 Pick,以减少用户查找文件的时间。热键为 Alt-F3。

New:编辑新文件,删除编辑窗口中的原有程序(并清内存)。

Save:将正在编辑的文件存盘。热键为 F2。

Write to:将正在编辑的文件改名后存盘。

Directory:显示当前工作目录的内容。

Change dir:显示当前目录,并允许改变驱动器及目录。

Os shell:暂时退出 C 环境到 DOS 环境,键入 exit 后再返回到 C 环境。

Quit:退出 C 环境,返回 DOS 环境。热键为 Alt-X。

2. 编辑(Edit)

Edit 选项用于建立和修改源程序,该选项下不含子菜单。

3. 运行菜单(Run)

运行菜单的选择项如图 1-2(b)所示。

Run:将源文件进行编译、连接,然后运行。热键为 Ctrl-F9。

Program reset:中止当前调试,释放程序空间,关闭已打开文件。热键为 Ctrl-F2。

Go to cursor:使程序运行到光标所在行。热键为 F4。

Trace into:跟踪执行。热键为 F7。

Step over:单步执行。热键为 F8。

User screen:查看程序运行时产生的输出屏幕。热键为 Alt-F5。

4. 编译菜单(Compile)

编译菜单的选择项如图 1-2(c)所示。

Compile to OBJ:将当前正在编辑的源程序编译成 OBJ 文件。

Make EXE File:直接将源程序编译和连接成可执行文件。

Link EXE File:连接当前的 OBJ 文件和 LIB 文件,生成新的 EXE 文件。

Build all:重新编译和连接 Project 中的所有文件。

Primary C File:重新指定待编译的主文件,而不是当前带入到编辑程序中的文件。

Get into:显示当前对话的信息。

关于其他子菜单项及其使用方法,本章不再详述,请参阅有关 Turbo C 手册。

四、子菜单的四种选择方法

1. 先选中主菜单项,再用↑、↓(键将光标亮区移到所需的子菜单项,然后按回车键。例如,要退出 Turbo C,可在 File 子菜单将光标移到 Quit 选项,然后按回车键。

2. 先选中主菜单项,再键入所需子菜单项的首字母。例如,在 File 子菜单中按 Q 键,则可选中 Quit 选项。

3. 直接按 Alt 键加上所需菜单项的首字母键。例如,按 Alt-F 键可选中 File 子菜单,按 Alt-FQ 键可选中 File 子菜单下的 Quit 选项。

4. 热键选择。例如,要退出 Turbo C,按 Alt-X 键即可。

表 1-1 是 Turbo C 2.0 集成环境中提供的常用热键。

表 1-1 Turbo C 2.0 常用热键

热 键	功 能	热 键	功 能
F1	激活联机帮助系统	F2	将当前编辑的文件存盘,返回编辑状态
F3	装入指定文件	F4	运行程序,直到光标所在行
F5	放大、缩小当前窗口	F6	在编辑和编译信息窗口之间切换
F7	单步运行程序,跟踪函数调用	F8	单步运行程序,跳过函数调用
F9	编译和连接源程序	F10	激活主菜单
Ctrl-F9	运行当前程序(包括编译、连接和运行)	Alt-F3	选择文件装入
Alt-F5	用户屏幕和集成环境之间的切换		

1.3.2 在 Turbo C 集成环境下编写及运行 C 程序

C 语言是一种编译型的程序设计语言。一个 C 程序由用户构思、编写完成后,要在 Turbo C 上经过编辑、编译、连接和运行四个步骤才能得到运行结果。

一、编辑 C 程序

编辑一个 C 程序通常是指编辑一个新文件或装入旧文件进行修改。

1. 编辑新文件

按 F10 键激活主菜单,选择 Edit 选项或选择 File 子菜单下的 New 选项,则可进入文件编辑窗口。开始输入源程序,同时系统自动给出文件名 noname.c。

2. 装入老文件

若要装入磁盘上已有的源程序文件,则选择 File 子菜单下的 Load 选项(或按 F3 键),屏幕会提示用户输入文件名,如 exam1.c,则该程序被读入内存显示在屏幕上,并自动进入编辑状态。

无论是新文件还是老文件,编辑完成后都可直接按 F2 键存盘,也可选择 File 子菜单下 Save 或 Write to 选项进行存盘。

二、编译、连接和运行 C 程序

编译是指对源程序进行语法检查并翻译成可重定位的二进制目标程序,生成同名的“.obj”目标文件。连接是指将编译过的目标文件和程序中用到的库函数等连接在一起形成一个“.exe”的可执行文件。运行是可将执行文件投入运行,以获取程序的运行结果。

当 C 源程序编辑完成后,按 Ctrl-F9 键或选择 Run 选项可对 C 程序进行编译与连接,编译成功后会自动运行该程序,按 Alt-F5 可显示程序运行结果。

如果发现程序有错,编辑信息窗口中会显示错误信息,并将错误所在行反白显示,用户只须按一下回车键,就可对该程序进行编辑、修改。修改完成后,按 Ctrl-F9 重新进行编译,直至编译成功并返回集成环境。这时,用户可按 Alt-F5 键切换到用户屏幕查看运行结果。

三、保存源文件,退出 Turbo C

程序运行正确后或在编辑文件过程中,可随时按 F2 键将文件存盘;按 Alt-X 键可退出 Turbo C 回到 DOS 状态。

小 结

本章是课程的入门,新的概念较多,请同学们仔细阅读教材,着重掌握如下内容:

1. C 语言的特点和数据结构的定义及研究内容。
2. C 程序的基本单位为函数,函数通常由函数的说明部分和函数体组成;每个函数中必有一个主函数 `main()`,程序总是由 `main()` 函数开始执行的;存放函数的文件名必须符合标识符的规定,文件的扩展名为 `.c`。应注意 C 程序的语法规定,会为程序做注释。
3. 掌握 Turbo C 软件的基本使用方法,会使用热键。

习 题

一、选择题

1. 一个 C 源程序是由_____。
A) 函数组成 B) 子程序组成
C) 过程组成 D) 一个主程序和若干子程序组成
2. 以下叙述中不正确的是_____。
A) 一个 C 程序总是从 `main()` 函数开始执行的
B) C 程序中,`main()` 函数必须位于程序的最前面
C) C 源程序必须经过编译、连接成功后才能得到运行结果
D) 在对 C 程序进行编译过程中,不能发现注释中的拼写错误
3. 在 C 源程序中,`main()` 函数的位置_____。
A) 必须在程序开头 B) 必须在程序最后
C) 可以任意 D) 必须在系统调用的库函数的后面
4. 以下叙述中正确的是_____。
A) C 程序中每行只能写一条语句
B) C 语言不区分大小写英文字母
C) 一个 C 源程序必须包含一个 `main()` 函数
D) C 语言是一种高级语言,不能直接对硬件进行操作

二、填空题

1. C 源程序的基本单位是_____。
2. 一个函数名后必须跟_____。
3. 注释语句两侧的分界符分别为_____和_____。
4. C 语言的每条语句后必须使用_____作为结束符。
5. C 函数的组成为两部分:_____和_____。前者由_____四部分构成;后者由_____构成。
6. 数据结构研究的三个内容是_____、_____和_____。

三、编写程序

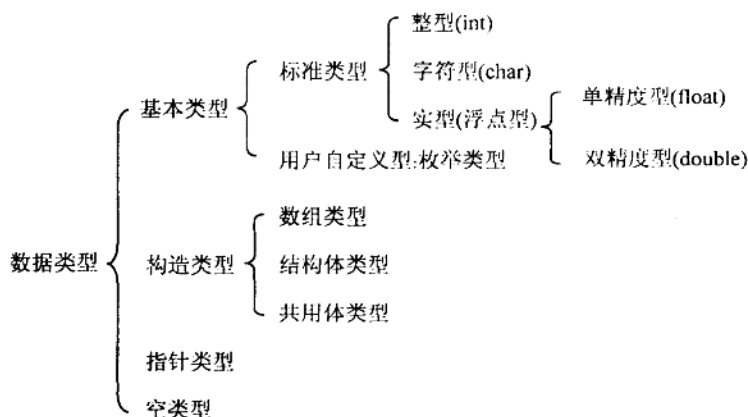
1. 编写表达式 $Y=3 \times A \div 5$ 的运算程序,并输出运算结果。已知 $A=20$ 。
2. 随机输入两个数,输出两者中最小的数。

第二章 数据类型与基本运算

2.1 C 语言的数据类型

数据是计算机处理的对象,通常分为常量和变量两种。数据类型是指简单数据的基本属性。C 语言提供的数据结构是以数据类型的形式出现的。

C 语言中的数据类型如下:



其中,标准类型是 C 语言本身提供的基本数据类型,结构比较简单。构造类型是按一定规则构造而成,结构比较复杂。指针类型是一种重要的数据类型,可以表示复杂的数据结构,普遍应用于动态运算处理。

本章中将讨论 C 语言中的基本数据类型及其基本运算。

2.2 常量和变量

2.2.1 常量

常量是指在程序运行过程中其值不能被改变的量。C 语言中,常量通常分为数值型常量、字符常量、字符串常量和符号常量。数值型常量和字符型常量在程序中不需预先说明就可直接使用。

一、数值型常量

数值型常量有整型常量与实型常量两种。

1. 整型常量

整型常量即整数。在 C 语言中,整型常量分为无符号整型常量和带符号整型常量两种,

并有十进制、八进制和十六进制三种表示形式。每种形式又可表示成基本整型、长整型和短整型三种。每一种数据类型都有与之对应的类型名保留字和取值范围,见表 2-1。

十进制:以非 0 开头的十进制数字表示,如:363,45,-3。

八进制:以数字 0 开头的八进制数字表示,如:034 等于 $(34)_8$,0248 等于 $(248)_8$ 。

十六进制:以 0x 开头的十六进制数字表示,如:0x17 代表 $(17)_{16}$,0x3d 代表 $(3d)_{16}$ 。

2. 实型常量

实型常量即实数,又称浮点数,有单精度和双精度两种类型。每种实型常量又有以下两种表示形式:

(1) 小数形式:它由整数部分、小数点和小数部分组成,如:3.14,31.4。

(2) 指数形式:它由整数部分、小数点、小数部分和指数部分组成。其中指数部分是在 e 或 E(相当于数学中幂底数 10)后跟整数阶码(可带符号的整数指数)。如:3.14E-2 表示 3.14×10^{-2} ,1e14 表示 1×10^{14} 。

表 2-1 数据类型的表示及取值范围

数据类型	类型名保留字	所占字节数	取值范围	
整型	int	2	-32 768~32 767	即 $-2^{15} \sim (2^{15}-1)$
短整型	short int	2	-32 768~32 767	即 $-2^{15} \sim (2^{15}-1)$
长整型	long int	4	$2^{31} \sim (2^{31}-1)$	
无符号型	unsigned int	2	0~65 535	即 $0 \sim (2^{16}-1)$
无符号短整型	unsigned short	2	0~65 535	即 $0 \sim (2^{16}-1)$
无符号长整型	unsigned long	4	$0 \sim (2^{32}-1)$	

二、字符常量

字符常量是用单引号(' ')括起来的单个字符,如:'a'和'l'。一个字符常量的值是由该字符的 ASCII 代码值来确定的。

在 C 语言中,还允许用一种特殊形式的字符常量,它是以一个“\”开头的字符序列,用于表示 ASCII 字符集内的控制代码和某些功能定义,如:换行、回车等,通常称为转义字符。表 2-2 为转义字符功能表。

表 2-2 转义字符功能表

字符形式	功 能
\n	换行
\t	横向跳格(即跳到下一个输出区)
\v	竖向跳格
\b	退格
\r	回车
\f	走纸换行
\\	反斜杠字符“\”
\'	单引号(撇号)字符
\ddd	1 到 3 位八进制数所代表的字符
\xhh	1 到 2 位十六进制数所代表的字符

```

例 2-1  main()
        {printf("ab\tcd\n");
         printf("xyz\n");
        }

```

运行结果:ab cd
 xyz

在例 2-1 中,用 printf 函数直接输出双引号中的字符串,其中\t 作用是跳到下一个输出区,\n 的作用是换到下一行接着输出。

三、字符串常量

字符串常量是由双引号(" ")括起来的一个字符序列,如:"name","How are you"。

字符常量和字符串常量是概念不同的两种常量。它们在存储方式上有着不同的组织方式,即'a'和"a"是不同的。

C 语言规定:在每一个字符串的结尾处系统自动加上一个字符串结束标志 \0,供系统判断该字符串是否结束。 \0 是一个 ASCII 码值为 0 的字符,表示空操作,因此存储每个字符串所需空间会比实际串长多 1 字节。如"a"占用 2 字节,而"name"占用 5 字节。如图 2-1 所示。



图 2-1 字符常量与字符串常量存储方式

四、符号常量

当某一常量在程序中多次出现时,可用符号来代替常量,该符号称为符号常量。C 语言规定符号常量用大写英文字母书写,以便与普通变量区别。使用符号常量可提高程序可读性。

符号常量的定义格式: # define 常量标识符 常量

```

例如: #define PI 3.1415926
       #define MAXNUM 100

```

这里定义 PI 和 MAXNUM 均为符号常量,其值分别为 3.1415926 和 100。在 C 程序中,符号常量必须先定义,后使用。其应用详见 4.3 节。

2.2.2 变 量

变量是指在程序运行过程中其值可以发生变化的量。C 语言的变量分为基本类型和构造类型。前者分为整型变量、实型变量和字符变量;后者是在基本类型的基础上扩展而得到的数据类型。本章中将介绍基本类型数据。

在 C 程序中,整型变量、实型变量和字符变量都必须用一个标识符作为变量名,通常用小写字母表示。变量名在程序中使用时必须先说明其类型。

一、变量的分类

1. 整型变量

整型变量的标识符是 int,分为以下四种类型。

(1) 基本型: int

- (2) 短整型: short 或 short int
- (3) 长整型: long 或 long int
- (4) 无符号型: 只存放无符号整数, 又分无符号整型(unsigned int)、无符号短整型(unsigned short)和无符号长整型(unsigned long)。

2. 实型变量

实型变量也称为浮点型变量, 分为两种类型。

- (1) 单精度型: float
- (2) 双精度型: double

两种类型的实型变量用来存放同一个实型常数, 只是截取的有效位数不同。单精度型实数据提供 7 位有效位, 双精度实数据提供 15~16 位有效位。因此 double 型变量中存放的数据要比 float 型变量中存放的数据精确的多。

```
有如下一段程序: float x;
                    double y;
                    x=123456.789;
                    y=123456.789;
```

其中 x 为单精度变量, 被赋值为 9 位, 但单精度变量只能接受 7 位, 因此最后 2 位小数不起作用; 而 y 为双精度量变量, 9 位数字均被有效存储起来。

3. 字符型变量

字符型变量用来存放单个字符, 但在存储时并不是存放字符本身, 而是将字符所对应的 ASCII 码值存放到存储单元中。因字符型数据与整型数据的存储形式相同, 所以字符型数据和整型数据之间可以通用。例如: $x = 'a' + 34$; 是合法的。

二、变量的定义和初始化

1. 变量的定义

C 语言程序中所用到的变量都必须在使用之前进行定义, 即应在使用前指定变量的名称和数据类型, 以便编译系统为它分配相应的存储单元。

变量定义的格式: 类型符 变量名表;

```
例如: int x,y,z;           /* 定义变量 x,y,z 为整型变量 */
      float aver;          /* 定义变量 aver 为单精度变量 */
      char ch;             /* 定义变量 ch 为字符型变量 */
```

2. 变量的初始化

变量的初始化是在定义变量的同时赋予相同类型的初值。

```
例 2-2 main()
{int x=3, y=4, z;
  z=x*y;
  printf("z=%d",z); }
```

运行结果: z=12

2.3 运算符和表达式

C 语言的运算符非常丰富, 除了控制语句和输入输出语句以外, 它把基本操作都视为运

算。C 语言的运算符不仅具有不同的优先级,在同一优先级运算符参与运算时还具有结合性,即确定是自左向右进行运算还是自右向左进行运算,前者称为“左结合性”,后者称为“右结合性”。

C 语言的运算符有以下几类:

1. 算术运算符: (+ - * / % ++ --)
2. 关系运算符: (> < == >= <= !=)
3. 逻辑运算符: (&& || !)
4. 位运算符: (& | ~ ^ << >>)
5. 赋值运算符: (= 及其扩展运算符)
6. 条件运算符: (? :)
7. 逗号运算符: (,)
8. 指针运算符: (* &)
9. 求字节数运算符: (sizeof)
10. 强制类型转换运算符: (类型)
11. 分量运算符: (. ->)
12. 特殊运算符: (如下标[],括号()等)

2.3.1 算术运算符和算术表达式

一、算术运算符

1. 基本的算术运算符

- + 加法运算符。如 $8+7$, $a+b$ 。
- 减法运算符。如 $5-4$, $a-b$ 。
- * 乘法运算符。如 $5*7$, $a*b$ 。
- / 除法运算符。如 $4/3$, a/b 。

值得注意的是:进行除法运算时两整数相除,其运算结果为整数(舍去小数部分)。如 $4/3$ 的结果为 1, $-5/3=-1$ 等。如果参加运算的两个数中有一个数为实数,则运算结果为 double 型数据,因为所有实数都按 double 型进行运算。如 $5.0/10=0.5$ 。

% 模运算符,或称求余运算符。要求参与运算的数据均为整型数据,其运算结果为两整型数据整除后所得余数。余数符号一般与被除数符号相同。如: $6\%4=2$, $-6\%4=-2$ 。

2. 自增、自减运算符

自增运算符“++”和自减运算符“--”是 C 语言中两个最有特色的单目运算符。它们要求操作对象必须是整型变量。按其使用方式分为前置运算和后置运算两种。

(1) 前置运算

前置运算格式: ++i 或 --i

功能:在使用变量 i 之前,先使 i 的值加(减)1。

例如: $i=3$;

$j=++i$;

/* 先执行 $i=i+1$,再执行 $j=i$, 即 $i=4$, $j=4$ */

(2) 后置运算

变量在同一个表达式中参与运算。

例如: $36.5 + 'a' - 75 * 12345.6789$ 是合法的。在进行运算时, C 编译系统自动的进行数据类型的转换工作, 然后再进行运算。转换规则如图 2-2 所示。

在图 2-2 中, 横向向左的箭头($\leftarrow$)表示运算时自动进行的转换。如 char 型和 short 型在运算前先自动转成 int 型; float 型在运行前自动转成 double 型; 即使是两个 float 型数据相加也先都转成 double 型再相加, 以提高其运算精度。

纵向向上的箭头($\uparrow$)表示当运算对象为不同类型数据时的转换方向, 转换原则是由级别低的向级别高的转换。如 int 型和 double 型数据进行混合运算时, 先将 int 型自动转换成 double 型再进行运算, 结果为 double 型。请读者分析表达式 $'a' + 4.5 * 3 - 100L$ 的运算次序和运算结果的数据类型。

(2) 强制类型转换

强制类型转换是指将一个变量或表达式的数据类型强制转换成所需的数据类型。

强制转换的格式: (类型名)表达式

例如: `(int) (a+b)` */* 将 a+b 的值转换成 int 型数据 */*
 `(float) (5%3)` */* 将 5%3 的值转换成 float 型数据 */*

说明:

- ① 表达式应使用括号括起来, 否则 `(int)a+b` 是将 a 转成 int 型数据再与 b 相加。
- ② 强制转换时, 得到一个所需类型的中间变量, 原来变量的类型未发生变化。

例 2-4 `main()`
 `{float a; int b;`
 `a=3.14; b=(int)a;`
 `printf("a=%f,b=%d",a,b);`
 `}`

程序运行结果: `a=3.140000,b=3`

由此可见, 变量 a 的类型仍为 float 型, 其值未变; 强制转换后的值送给变量 b, b 的值等于 a 的整数部分 3。

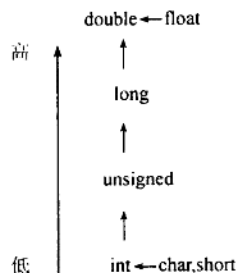


图 2-2 标准类型数据自动转换规则

2.3.2 赋值运算符和赋值表达式

一、赋值运算符和赋值表达式

赋值符号“=”就是赋值运算符, 其作用是将一个数据赋给一个变量。

赋值表达式的格式: 变量=表达式

说明:

1. 赋值运算符左边必须是变量而不能是表达式。例如: `i+1=i` 是错误的。
2. 如果赋值号两边的类型不一致, 则赋值时会自动进行类型转换。转换原则是: 将赋值号右边表达式的类型转换成赋值号左边变量的类型。如 i 为整型变量, 执行 `i=3.14` 的结果是使 i 的值为 3, 即实型数据赋给整型变量时, 舍弃实数的小数部分。
3. 赋值运算可连续进行, 赋值运算符的结合性是从右向左。如赋值表达式 `a=b=c=10`,

等价于 $a=(b=(c=10))$, 经过连续赋值后, a, b, c 的值都是 10, 但最后该表达式的值是 a 的值。

二、复合赋值运算符

在赋值运算符“=”之前加上其它运算符, 就构成了复合运算符, 以达到同化程序和提高运算效率的目标。C语言规定可以使用以下 10 种复合赋值运算符, 即:

$+= \quad -= \quad *= \quad /= \quad \% = \quad << = \quad >> = \quad \& = \quad \wedge = \quad ! =$

当赋值表达式中包含复合赋值运算符时就构成了复合赋值表达式。例如:

$a+=b$ 等价于 $a=a+b$

$x*=3$ 等价于 $x=x*3$

$y-=x+5$ 等价于 $y=y-(x+5)$

复合运算符在书写时, 两个运算符之间不能有空格, 否则是错误的。

例 2-5 运行下列语句并写出运行结果。

(1) `int m=2, n=2;`

(2) `int a=4;`

`n+=++m;`

`a+=a-=a*a;`

运行结果: n 的值为 5, m 值为 3。

运行结果: a 的值为 -24。

三、赋值语句

在赋值表达式后面加上一个分号就构成了赋值语句。

例如: $a=3; b=x*y+z; i=j=0; m*=n+1;$ 都是合法的赋值语句。

2.3.3 逗号运算符和逗号表达式

前面我们接触到的逗号都是作为分隔符号形式出现的, 它起着分隔符的作用。例如:

`int a, b, c;`

`printf("%d%d%d", a, b, c);`

C语言中, 逗号的另一个作用是作为运算符出现, 称为逗号运算符。由逗号运算符将若干个表达式连接起来, 就构成一个逗号表达式。

逗号表达式的格式: 表达式 1, 表达式 2, ..., 表达式 n

功能: 按照从左到右的顺序依次计算各表达式的值, 逗号表达式的值是表达式 n 的值。

例如: $a=5, a+3$ 逗号表达式的值为 8, 变量 a 值为 5。

$x=3*5, x*4$ 逗号表达式的值为 60, 变量 x 值为 15。

$a=2, b=3, a=a*b$ 逗号表达式的值为 6, 变量 a 值为 6。

说明:

1. 当逗号表达式前后含有其它意义的逗号, 如分隔变量的逗号, 逗号表达式要加括号。

例如: `printf("%d%d%d", (a=3, a*2), b, c);`

2. 一个逗号表达式可以与另一个表达式组成一个新的逗号表达式。

例如: $(a=3*5, a*2), a+3$ 逗号表达式的值为 18。

3. 逗号运算符是所有运算符中级别最低的。因此, 下面两个表达式的作用是不同的:

$x=(a=2, 3*a)$

$x=a=2, 3*a$

第一个表达式是一个赋值表达式, 是将括号内逗号表达式的值赋给 x , x 的值等于 6。

第二个表达式是一个逗号表达式, 它包含一个赋值表达式和一个算术表达式, x 的值为 2。