

教学情境 1 一个简单的 C 语言程序

学习目标

C 语言是一种结构化语言。它层次清晰，便于按模块化方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，还可以直接访问内存的物理地址，进行位（bit）一级的操作。由于 C 语言实现了对硬件的编程操作，因此，C 语言集高级语言和低级语言的功能于一体，即既可用于系统软件的开发，也适合于应用软件的开发。本情境讲解内容如下：

- C 语言的发展及特点；
- 初步认识 C 语言程序；
- 软件开发方法。

情境导入

问 题

请输出结果 “happy new year!”。

实 现

```
*/  
#include<stdio.h>          /* 预处理，头文件对 printf 函数的声明 */  
main ()  
{  
printf (" happy new year! \n"); /* 直接用 printf 函数输出结果 */  
}
```

运行结果：

happy new year!

相关知识

一、C 语言的发展及特点

（一）C 语言的发展

C 语言的发展颇为有趣。它的原型是 ALGOL 60 语言。

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 Martin Richards 对 CPL 语言进行了简化, 于是产生了 BCPL 语言。

1970 年, 美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改, 并为它起了一个有趣的名字“B 语言”。意思是将 CPL 语言煮干, 提炼出它的精华。并且他用 B 语言写了第一个 UNIX 操作系统。

1973 年, B 语言也被人“煮”了一下, 美国贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言, 他取了 BCPL 的第二字母作为这种语言的名字, 这就是 C 语言。

为了使 UNIX 操作系统推广, 1977 年, Dennis M. Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。

1978 年, Brian W. Kernighan 和 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最流行的高级程序设计语言。

1988 年, 随着微型计算机的日益普及, 出现了许多 C 语言版本。由于没有统一的标准, 使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况, 美国国家标准研究所 (ANSI) 为 C 语言制定了一套 ANSI 标准, 成为现行的 C 语言标准 3。C 语言发展迅速, 而且成为最受欢迎的语言之一, 主要是因为它具有强大的功能。许多著名的系统软件, 如 DBASE III PLUS、DBASE IV 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序, 就更能显出 C 语言的优势了, 如 PC-DOS、WORDSTAR 等就是用这种方法编写的。

(二) C 语言的特点

1. 简洁紧凑、灵活方便

C 语言一共只有 32 个关键字, 9 种控制语句, 程序书写自由, 主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合了起来。

2. 运算符丰富

C 语言的运算符包含的范围很广泛, 共有 34 个运算符。C 语言把括号、赋值、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, 灵活使用各种运算符, 可以实现在其他高级语言中难以实现的运算。

3. 数据结构丰富

C 语言的数据类型有整型、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类型的运算, 并引入了指针的概念, 使程序效率更高。另外, C 语言具有强大的图形功能, 支持多种显示器和驱动器, 且计算功能、逻辑判断功能强大。

4. C 语言是结构式语言

结构式语言的显著特点是代码及数据的分隔化, 即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰, 便于使用、维护及调试。C 语言是以函数形式提供给用户的, 这些函数可方便地调用, 并具有多种循环、条件语句控制程序流向, 从而使程序完全结构化。

5. C 语言语法限制不太严格, 程序设计自由度大

一般的高级语言语法检查比较严, 能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度。

6. C 语言允许直接访问物理地址, 可以直接对硬件进行操作

C 语言既具有高级语言的功能, 又具有低级语言的许多功能, 能够像汇编语言一样对位、字节和地址进行操作, 而这三者是计算机最基本的工作单元, 可以用来写系统软件。

7. C 语言程序生成代码质量高, 程序执行效率高

一般只比汇编程序生成的目标代码效率低 10%~20%。

8. C 语言适用范围大, 可移植性好

C 语言有一个突出的优点就是适合于多种操作系统, 如 DOS、UNIX, 也适用于多种机型。

二、初步认识 C 语言程序

为了帮助同学们对 C 语言程序形成完整的认识, 这里准备了一个较为完整的程序。下面的程序用来演示在 C 语言程序设计中使用的程序设计方法和基本要素。编写程序, 用来计算输入的两个数的和, 并把和输出。

```
#include <stdio.h> /* 预处理, 头文件对 scanf、printf 函数的声明 */
void main ()
{
    int a, b, sum; /* 定义变量 */
    printf ("Qing shu ru:"); /* 使用函数输出 Qing shu ru: */
    scanf ("%d %d", &a, &b); /* 手动输入变量的值 */
    sum=a+b; /* 运算得出变量 sum 的值 */
    printf ("%d+%d=%d", a, b, sum);
}
```

程序输出结果为:

Qing shu ru:

3 15

3+15=18

(一) C 语言程序的格式

C 语言中格式书写比较自由, 一行可以写多条语句, 一条语句也可以书写在多行上。C 语言识别大小写字母, 如“A”和“a”是两个不同的字母。C 语言语句后必须有分号, 只有分号的语句为空语句。为了增强程序的可读性, 应该避免在一行中书写多条语句, 并使用锯齿形书写程序代码, 还可以通过空行来增强可读性。

(二) C 语言程序的构成

C 程序的基本结构是函数, 一个或多个 C 函数组成一个 C 程序, 若干个 C 语句构成 C 语言函数, 若干个基本单词形成 C 语句。C 语言中使用的函数有两类, 一类是系统定义的函数, 如 printf 和 scanf 等, 称为标准库函数, 可以直接在程序中使用。另一类是用户自己定义的函数, 必须由用户自己编写源程序代码。

函数的基本格式如下：

```
[函数类型] 函数名 ([函数形参表]) /* 函数首部 */  
{  
    /* 函数体 */  
    [变量定义和声明语句;]  
    可执行语句部分;  
}
```

(三) C 语言程序的基本要求

(1) 在整个程序文件中，函数可以出现在任意位置。主函数不一定出现在程序的开始处，但不管主函数位于程序何处，程序总是在主函数中开始，也在主函数中结束。

(2) 每个程序行中的语句数量任意，既允许一行内写多条语句，也允许多条语句写在同一行上。

(3) 为了对程序进行必要的描述，可以给程序进行说明，说明必须写在 /* 和 */ 之内。

三、软件开发方法

(一) 软件开发方法

20 世纪 60 年代出现了软件危机，其现象表现为软件开发费用和精度失控，软件的可靠性差，生产出来的软件难以维护。为了解决软件危机，在 60 年代末期提出了软件工程的概念，并在以后不断发展、完善。与此同时，软件研究人员也在不断探索新的软件开发方法，至今已形成八类软件开发方法，主要有 Parnas 方法、SASA 方法、面向数据结构的软件开发方法、问题分析法、面向对象的软件开发方法、可视化开发方法等。

(二) 算法

算法定义：算法是规则的有限集合，是为解决特定问题而规定的一系列操作。

算法的特性：算法必须具备 5 个基本的特性。

(1) 有限性：有限步骤之内正常结束，不能形成无限循环。

(2) 确定性：算法中间的每个代码行必须有确定含义。

(3) 输入：可以有 0 个或者多个输入。

(4) 输出：至少有一个输出。

(5) 可行性：原则上能够精确运行，操作可以经过已经实现的基本运算通过有限次执行完成。

算法的表示方法：算法的表示方法有多种，下面介绍算法的流程图表示方法。传统的流程图由下面的几种基本框组成，如图 1-1 所示。使用这些框和流程线组成的流程图表示算法，形象直观，简单方便。在设计算法时对于整理设计思路很有帮助。

(三) 结构化程序设计

结构化程序设计的思想最早是由著名计算机科学家 E. W. Dijkstra 提出的。1966 年，Bohm 和 Jacopin 证明了只用三种基本结构就能实现任何一个入口、一个出口的程序；1977 年，IBM 公司的 Mills 又进一步提出：“程序应该只有一个入口和一个出口。”在长期程序

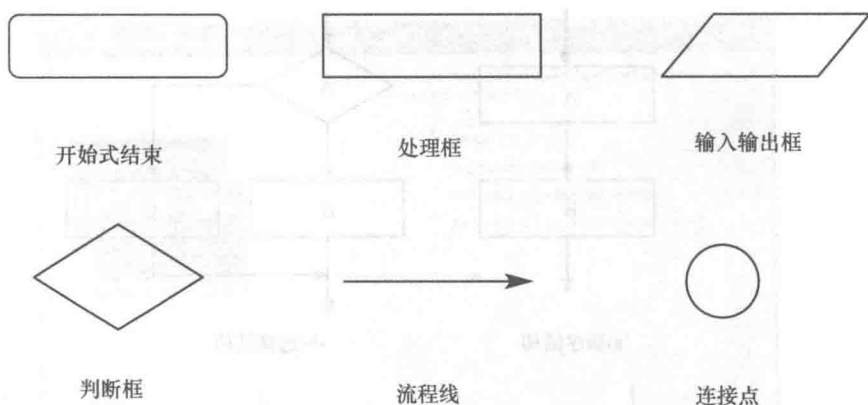


图 1-1 流程图的基本框图

设计的实践中，结构化程序设计方法不断得以完善，使之成为开发传统应用领域应用系统的主要方法之一。

结构化程序设计由三种基本结构组成，分别是顺序结构、选择结构和循环结构。

1. 顺序结构

顺序结构是最简单的一种结构，可以由赋值语句和输入、输出语句构成。当执行由这些语句构成的程序时，将按照这些语句在程序中的先后顺序逐条执行。流程图如图 1-2 (a) 所示。

2. 选择结构

选择结构也称分支结构。当执行该结构中的语句时，程序根据不同的条件执行不同分支中的语句。如图 1-2 (b) 所示。

3. 循环结构

循环结构是指根据各自的条件，使同一组语句重复执行多次或者一次也不执行。循环结构有两种形式：当型循环和直到型循环。当型循环中的循环体可能一次都不执行，而直到型循环中的循环体至少执行一次。分别如图 1-2 (c) 和 (d) 所示。

四、C 语言程序上机调试步骤和方法

本教材使用 Microsoft Visual C++ 6.0 作为开发工具，下面对于如何创建应用程序作个说明，步骤如下：

(1) 启动 Microsoft Visual C++ 6.0。

(2) 点击菜单上的【File】|【New】，出现如图 1-3 所示的新建对话框，选择【Projects】下的【Win32 Console Application】选项，在【Project name】中输入项目名称，这里输入 Demo001，点【Location】右边的浏览按钮选择文件存放的位置，然后点 OK 按钮。

(3) 接下来 Finish 按钮，在接着出现的对话框上点 OK 按钮，就创建好了一个空项目，如图 1-4。

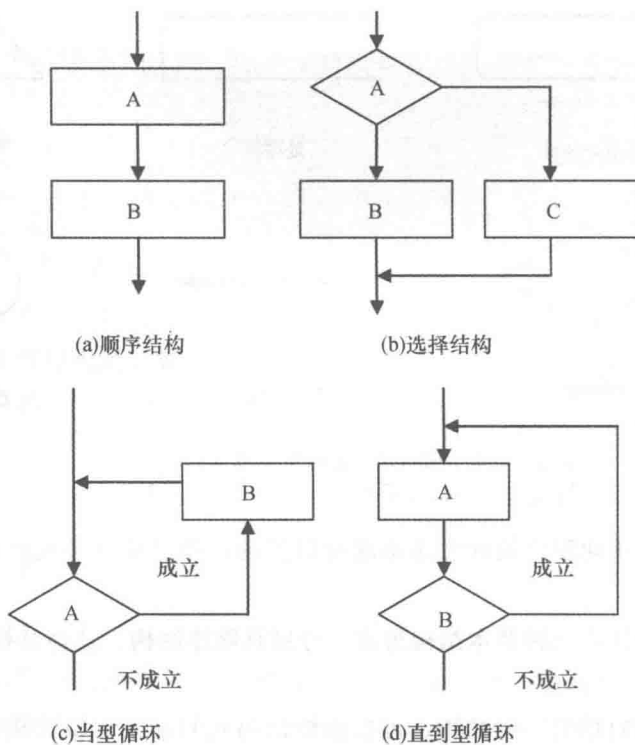


图 1-2 程序的基本结构框图

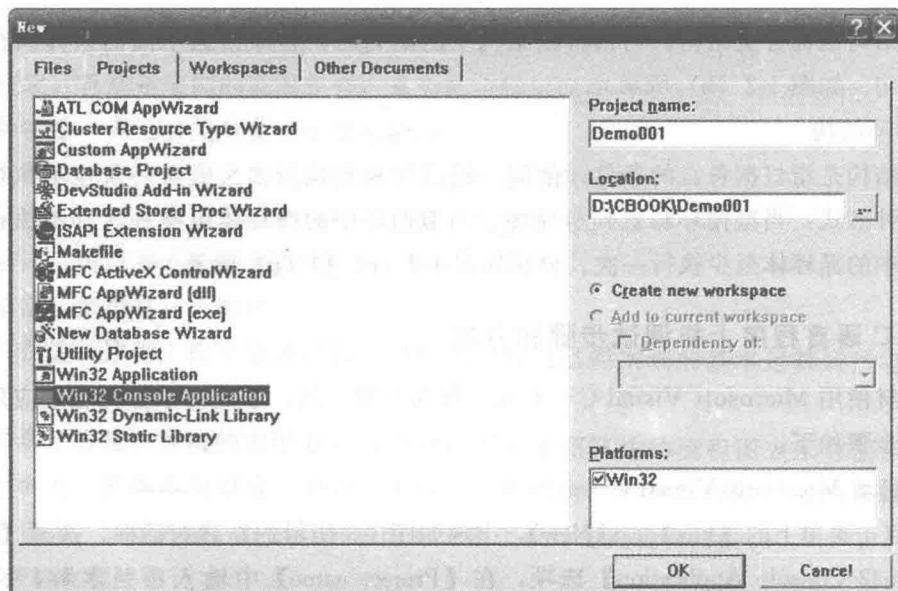


图 1-3 新建对话框

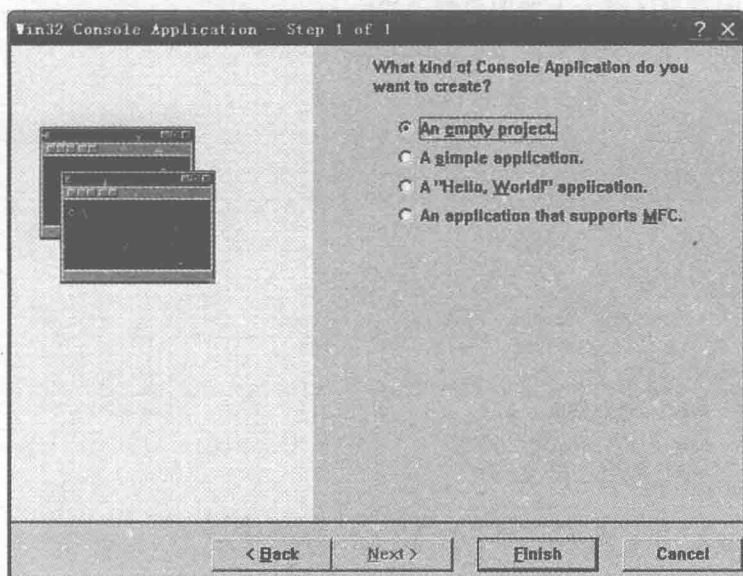


图 1-4 新建对话框结束

(4) 然后添加源代码文件。打开【New】对话框，在【Files】选项卡下选择 C++ Source File 选项，在【File】文本框中输入源文件的名称，这里输入 Code01.C，然后点 OK 按钮。切记，这里输入 Code01.C 文件名时必须输入文件扩展名，如图 1-5 所示。

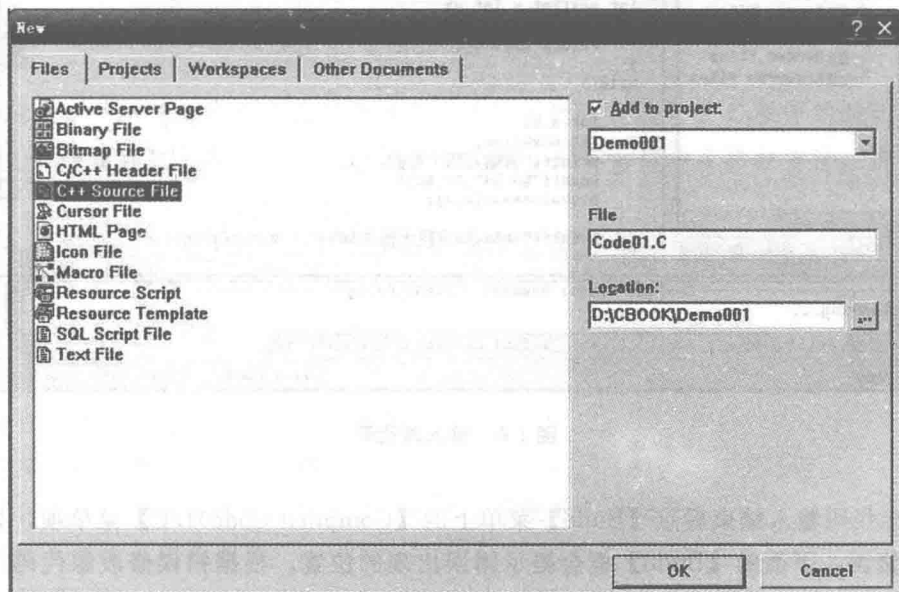


图 1-5 新建 C 源代码文件

(5) 在右边的空白区输入下面的源程序代码。

```
#include "stdio.h"

int max (int x, int y)
{
    return x>y? x: y;
}

main ()
{
    int a, b;
    int maxValue;
    printf (" 请输入两个整数:");
    scanf ("%d %d", &a, &b);
    maxValue=max (a, b);
    printf (" \n%d,%d中较大值为%d\n", a, b, maxValue);
}
```

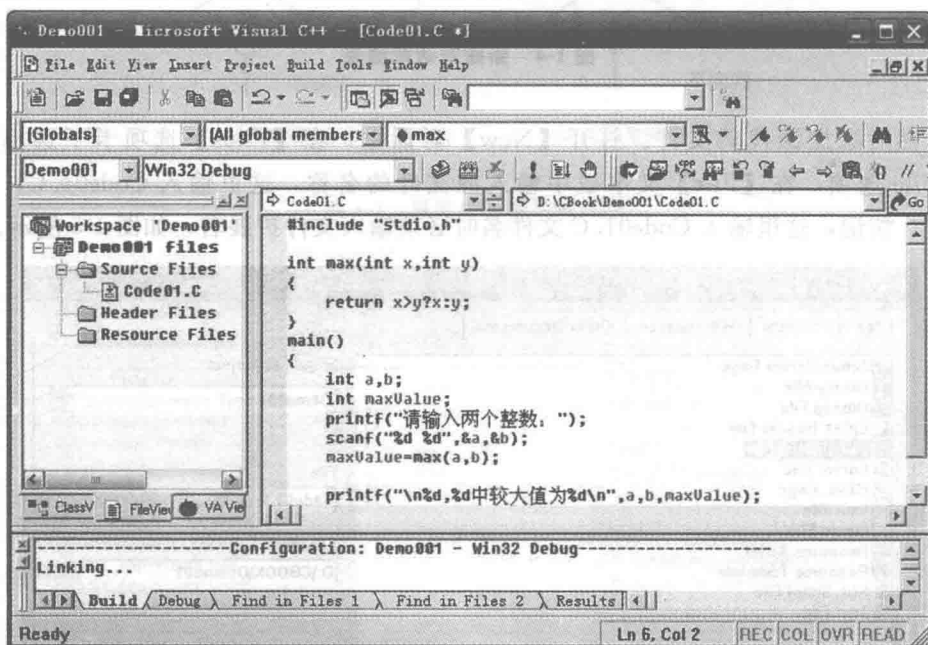


图 1-6 输入源代码

(6) 源代码输入结束后点【Build】菜单上的【Compiled Code01.C】菜单项，如果源代码有语法错误，下面的【Build】框会提示错误出现的位置。根据错误修改源代码，直到错误数为 0，如图 1-7 所示。

(7) 最后点【Build】菜单上的【Build Demo001.exe】菜单项，生成应用程序。最后点击【Execute Demo001.exe】菜单项运行程序。根据屏幕提示运行程序，得到程序运行结果如下：

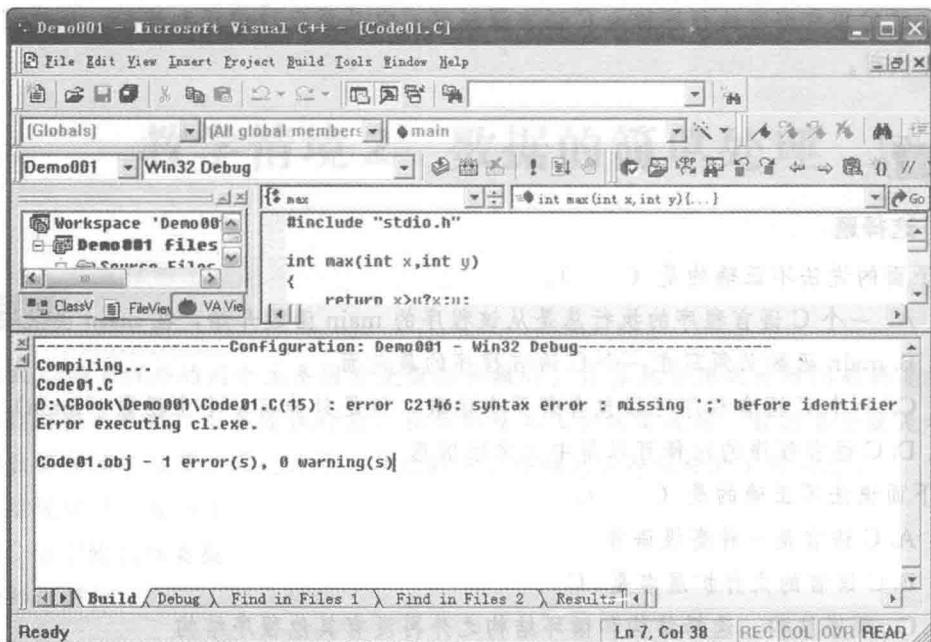


图 1-7 Build 窗口

请输入两个整数: 3 5 <回车>

3, 5 中较大值为 5

知识总结

本情境概括介绍了 C 语言的发展及特点, 通过实例分析讨论了 C 程序的格式、构成及基本要求、程序设计的基本知识, 最后介绍了 C 语言上机调试的步骤和方法。需要学习的要点如下:

(1) C 语言是一种兼有汇编语言和高级语言特点的语言, 于 20 世纪 70 年代初期由贝尔实验室研制开发。

(2) C 语言是一种理想的结构化语言。

① 具有结构化的控制语句。

② 语言简洁, 结构紧凑, 使用方便灵活。

③ 运算丰富, 数据处理能力强。

④ 可以直接访问物理地址, 实现硬件和底层系统软件的访问。

⑤ 语言生成的代码质量高。

⑥ 可移植性好。

(3) 函数是 C 语言程序的基本单位。一个 C 语言源程序可以由多个函数组成, 其中有且只有一个名为 main 的主函数, 无论 main 函数在程序的什么位置, C 语言程序总是从 main 函数开始执行。

(4) 用 C 语言编写的程序称为 C 语言源程序，它必须经过编译和连接，生成可执行程序后才能执行。

知识巩固

一、选择题

1. 下面的说法不正确的是 ()。
 - A. 一个 C 语言程序的执行总是从该程序的 main 函数开始，在 main 函数中结束
 - B. main 函数必须写在一个 C 语言程序的最前面
 - C. 一个 C 语言程序可以包含若干个函数，但是只能有一个主函数 (main)
 - D. C 语言程序的注释可以是中文文字信息
2. 下面说法不正确的是 ()。
 - A. C 语言是一种高级语言
 - B. C 语言的文件扩展名是 .C
 - C. 顺序结构、选择结构和循环结构之外再没有其他程序结构
 - D. 算法可以没有输出

二、填空题

1. C 语言程序是由 _____ 构成的，一个 C 程序中至少包含一个 _____。因此，_____ 是 C 语言程序的基本单位。
2. C 语言程序注释是由 _____ 和 _____ 所界定的文字信息组成的。
3. 函数体一般包括 _____ 和 _____。
4. C 语言是一种 _____ 化程序设计语言。

三、程序题

1. 编写程序在控制台输出下面的图案。

```
*****
*      Hello China      *
*****
```

教学情境 2 数据的简单处理

学习目标

构成 C 语言程序的两个主要因素是数据和操作。计算机中用来处理问题的是程序，其主要操作对象是数据。当程序执行后，操作的结果又会改变数据。数据类型就是对程序所处理的数据按照其性质、表达方式、构造特点、存储范围等划分的不同种类。

本情境学习目标如下：

- C 语言的数据类型；
- 标识符；
- 常量；
- 变量；
- 运算符与表达式；
- 自增（自减）运算符及 C 语言运算符的优先级别；
- 不同类型数据间的混合运算（包含数据类型转换）。

情境导入

问 题

输入两个数，输出两数的乘积

实 现

```
#include <stdio.h>
main ()
{
    int x, y, m; /* 定义整型变量 x, y, m */
    printf (" Please input x and y \n"); /* 输出提示信息 */
    scanf ("%d %d", &x, &y); /* 读入两个乘数，赋给 x, y 变量 */
    m=x*y; /* 计算两个乘数的积，赋给变量 m */
    printf ("%d * %d = %d \n", x, y, m); /* 输出结果 */
} →
```

程序输出结果为：

```
Please input x and y
5 15
5 * 15 = 75
```

相关知识

一、C 语言的数据类型

C 语言的数据类型就是在 C 程序中所允许出现的数据种类。不同类型的数据存储的数据范围、格式、表达方式及特点亦各不相同。C 语言程序中的每一个数据都属于唯一的数据类型。

C 语言的数据类型一般分为基本类型、构造类型、指针类型和空类型四大类，如图 2-1 所示。

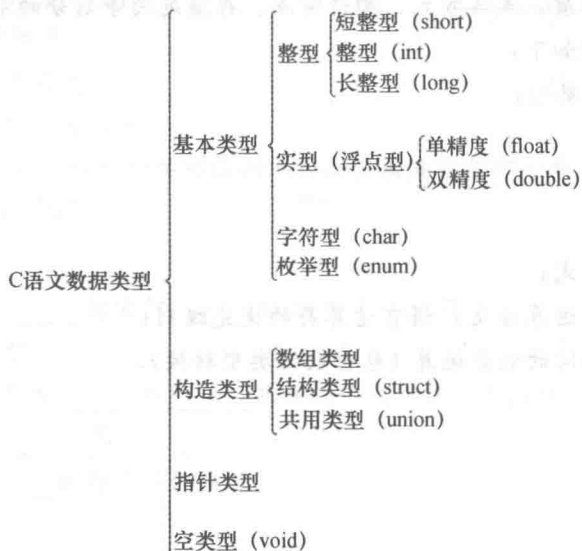


图 2-1 C 语言的数据类型

二、标识符

程序处理的对象是数据，而数据是以某种特定的形式存在的。标识符就是给程序中的实体（变量、常量、函数、类型、数组、结构体等）所起的标志性的名称。其主要作用就是“标识”，区分 C 程序中的不同类型的对象。

C 程序中的标识符分为两大类：系统定义标识符和用户定义标识符。

（一）系统定义标识符

系统定义标识符是指由系统定义的，具有固定名字和特定含义的，不能用于其他用途的标识符。且系统定义标识符都只能用小写字母书写。下面是 C 语言中系统定义标识符的几种分类：

1. 标识数据类型

int, char, long, float, double, short, unsigned, struct, union, enum, auto, extern, static, register, typedef, void。

2. 标识控制流

if, else, switch, case, default, do, while, for, break, continue, return, goto。

3. 标识预处理功能

define, include, undef, ifdef, endif, line。

(二) 用户定义标识符

用户在对自己使用的程序中的操作对象进行命名, 来对各个对象进行区分时所确定的名称就是自定义标识符。自定义标识符不能随意构造, 用户在定义标识符时要符合 C 语言中的语法规则。

C 语言中标识符的构造规则如下:

(1) 标识符只能以字母或者下划线开头。

(2) 标识符只能由字母、数字、下划线三种字符组成。

(3) 系统已经定义的一些标识符和关键字是系统专用的, 不允许用户作为自定义对象的标识符来使用。

(4) C 语言将大小写字母作为不同的字符处理。

(5) 自定义标识符一般应做到见名知义, 以提高程序的可读性。如用 sum 表示和, name 表示姓名, score 表示成绩, max 表示最大等。

因此标识符中只能出现:

●阿拉伯数字: 0~9, 共 10 个。

●英文字母: 大小写字母 A~Z, a~z, 共 52 个。

●下划线: _, 共 1 个。

标识符对于所使用的字符个数本身并没有限制, 但是在不同的 C 语言版本具体使用中, 对标识符的长度有不同要求。大多数系统取前 8 个字符有效。因此, 在定义标识符时, 为了程序的可移植性和可读性, 我们要求一般不要超过 8 个字符。

正确的标识符举例: M, sum 1, _total, ab_1_2。

错误的标识符举例: 11abc, for, _abc, \$12。

三、常量

常量是指在程序运行中值不能发生改变的量。

(一) 整型常量

整型常量又称为整数。C 语言程序中有三种不同的表示形式: 八进制、十进制、十六进制。

(1) 八进制整数: 采用 8 个不同的数码符来表示的数。由数字 0~7 组成, 逢 8 进位。为了以示区别, 在八进制整数前加上一个 0 标识。

(2) 十进制整数：采用 10 个不同的数码符来表示的数。由数字 0~9 组成的整数，逢 10 进位。

(3) 十六进制整数：采用 16 个不同的数码符来表示的数。由数字 0~9，a~f 组成的整数，逢 16 进位。为了以示区别，在十六进制整数前加一个“0x”标识。

整型数还可以分为长整型数 (long int)、短整型数 (short int)、无符号整型数 (unsigned int) 三种。

长整型数在数值后面加上一个标示符 L；短整形数不加标识符；无符号整型数在数值后面加一个标识符 U。

无符号数与有符号数的主要区别是数的最高位不作符号处理，表示数的绝对范围是有符号数的两倍。相应的无符号整型有无符号基本整型 (unsigned int)、无符号短整型 (unsigned short int) 和无符号长整型 (unsigned long int)。

这六种整型数据的长度和取值范围如表 2-1 所示。

表 2-1 整型数据类型表

	类型标识符	数据长度	取值范围
有符号整数 (signed)	短整型 signed short	16 位	$-2^{15} \sim (2^{15}-1)$
	基本整型 signed int	16 位	$-2^{15} \sim (2^{15}-1)$
	长整型 signed long	32 位	$-2^{31} \sim (2^{31}-1)$
无符号整数 (unsigned)	短整型 unsigned short	16 位	$0 \sim (2^{16}-1)$
	基本整型 unsigned int	16 位	$0 \sim (2^{16}-1)$
	长整型 unsigned long	32 位	$0 \sim (2^{32}-1)$

(二) 实型常量

实型常量又称为实数或者浮点数。C 语言中的实数只能用十进制来表示。实型常量用小数形式和指数形式表示。

(1) 小数形式：小数形式由数字 0~9 和小数点部分组成（必须要有小数点）。例如，3.47，13.25，123 等都是合法的实型数的表示方法。

(2) 指数形式：指数型的实数采用的是科学计数法。例如，12e2 或者 12E2 都代表 12×10^2 。但是要注意字母 e 或者 E 的前后必须要有数字，且 e 或者 E 后面的指数必须为整数。例如，123e4，123E4，123.4e4，12.3e-4 均为合法指数形式，而 E5，12e0.2，.e3，5E 等均为不合法的指数形式。

实型常量还可以分为单精度实型 (float) 和双精度实型 (double) 两种。一般系统中，单精度实型在存储时占用 4 个字节，有 32 比特，有 7 位有效数字。双精度实型在存储时占用 8 个字节，有 64 比特，有 15~16 位有效数字。还有一种长双精度型 (long double) 用得较少。

这三种实型数据的长度和取值范围如表 2-2 所示。

表 2-2 实型数据类型表

	类型标识符	数据长度	取值范围
单精度	float	32 位	$10^{-38} \sim 10^{38}$
双精度	double	64 位	$10^{-308} \sim 10^{308}$
长双精度	long double	80 位	$10^{-4932} \sim 10^{4932}$

(三) 字符常量

字符常量是用单引号''括起来的一个字符，如'a','B'等都是字符常量。其中，单引号是作为字符常量的定界符使用，并不表示字符常量本身。C 语言中常用的字符常量有一般字符常量和转义字符常量两种。

1. 一般字符常量

字符常量与计算机系统多使用的字符集有关，在中小型计算机上大多采用的是 ASCII 字符集。在 C 语言中，一个字符常量还具有数值。字符常量的值就是由该字符的 ASCII 码值确定，可以用整型数据来描述。例如'A'的 ASCII 码为 65。并且小写和大写要相区别，如'a'和'A'是不同的字符常量。因此，在 C 程序中，有时需要将某个字符常量赋给一个变量，实际上是将该字符常量的 ASCII 码值赋给变量。同理，若 C 语言中需要进行字符间的比较，实际上也是将各个字符的 ASCII 码值进行大小比较。

2. 转义字符

在 C 语言中还有一种特殊的字符型常量，是以一个反斜杠“\”开头的字符，后面加上一个数字或字符的特殊表示形式，用于表示 ASCII 码字符集内的控制代码和某些功能定义。我们称为“反斜杠字符常量”或者“转义字符”。“转义字符”代表一个字符，在内存中只占用一个字节。

C 语言中常见的转义字符功能表如表 2-3 所示。

表 2-3 转义字符功能表

字符形式	含义
\n	换行，当前位置移到下一行开头
\r	回车，当前位置移到本行开头
\f	换页，当前位置移到下页开头
\t	水平制表，横向跳格
\b	退格，当前位置移到前一列
\v	垂直制表，竖向跳格
\'	单引号字符
\"	双引号字符
\\	反斜杠字符
\ddd	1~3 位 8 进制数表示的字符
\xhh	1~2 位 16 进制数表示的字符

(四) 字符串常量

多个单独的字符构成字符串。C 语言允许的字符串常量是由一对双引号" "括起来的字符序列。例如" A"," CHINA"," 1234"," how are you" 等均为合法的字符串常量。

要特别注意的是字符常量和字符串常量的区别。

(1) 字符常量和字符串常量的书写格式不同。字符常量是用单引号' '括起来的, 而字符串常量是用双引号" "括起来的。例如' a' 是字符常量, 而" a" 是字符串常量, 二者形式类似, 但是含义不同。

(2) 字符常量和字符串常量的存储方式不同。C 语言规定, 在计算机中存储一个字符串时, 编译系统自动在每个字符串结尾处加上一个特定的控制符' \0' 来表示该字符串结束。也就是说如果系统要存储字符串" a" 时, 需要 2 个字节的内存, 分别来存储字符' a' 和' \0'。而若系统要存储字符' a' 时, 只需要一个字节的长度。也就是说字符串存储的长度比实际长度多一位。

例如, 若存储一个字符串" goodbye" 需要 8 个字节的长度, 如图 2-2 所示。

g	o	o	d	b	y	e	\0
---	---	---	---	---	---	---	----

图 2-2 字符串存储示意图

(五) 符号常量

在 C 语言中, 若使用符号来替代常量, 该符号称为符号常量。C 语言程序中的符号常量通常采用大写英文字母, 以示区别于一般的变量。当某一个常量在程序中多次出现时, 就可以使用符号常量来替代, 不仅可以方便地修改源程序, 同时也可以提高程序的可读性和可移植性。符号常量在使用前必须先定义, 其定义的格式如下:

```
# define 符号常量名 常量
```

例如:

```
# define NULL 0
```

用符号常量 NULL 来替代常量 0;

要注意的是, 每个符号常量定义式一次只能定义一个符号常量, 且每个符号常量定义式占据一行。

例如:

```
# define NULL 0
```

```
# define N 1
```

```
# define M -1
```

其中, NULL, N, M 是符号常量, 它们分别替代常量为 0, 1, -1。

四、变量

变量是指在程序运行过程中值可以发生改变的量。变量在使用前必须先进行定义, 指定变量的名称和数据类型。当用户给定变量的名称和初始值后, 系统就会在编译和运行程序时, 按照变量的数据类型在内存中分配相应大小的存储单元, 来存放变量的值。同时用

户还可以由赋值语句对程序中的变量赋予初始值。

(一) 变量名称

用户可以根据程序的要求任意设定所需要的变量名。但是在定义变量名称时要注意,变量名是标识符,必须要符合标识符构造的规则(详见 2.2 节标识符内容)。

C 语言规定,变量定义的格式为

数据类型符 变量名表

例如:

```
int a;      /* 定义整型变量 a */  
float b;    /* 定义单精度实型变量 b */
```

变量名实际上是一个符号地址,系统在编译时给每一个变量名分配一个内存地址。程序若想从变量中取值,实际上是通过变量名来找到相应存储地址后,从存储单元中读取数据。

(二) 变量的类型

C 语言中的变量,根据数值的性质可以分为两大类:基本数据类型变量和构造数据类型变量。其中,每一个大类型又可以分为若干个不同类型的变量。我们主要来看基本数据类型变量,如整型变量、实型变量和字符型变量等。另外还有指针型变量、结构体型变量、共用体型变量等,这些将在后续章节中详细介绍到。

1. 整型变量

整型变量用来存放整型数值。整型变量可分为:基本型(int),短整型(short int 或 short),长整型(long int 或 long)和无符号型(unsigned int, unsigned short, unsigned long)等。

前三种整型变量存储单元的最高位为符号位。0 表示为正,1 表示为负。无符号型变量存储单元的所有位均表示数值。具体可参看表 2-1。

在使用整型变量时一定要注意数值的范围,超过该变量允许的使用范围将导致错误的结果。

定义整型变量同样遵守变量的定义。

例如:

```
int a;      /* 定义整型变量 a */  
unsigned short c; /* 定义无符号短整型变量 c */  
long e;     /* 定义长整型变量 e */
```

2. 实型变量

实型变量分为单精度型(float)和双精度型(double)两类(双精度类型数据用得较少,我们不做讨论)。其存放数据的差别是:单精度变量占 4 个字节内存单元,有 7 位有效数字,数值范围在 $3.4\text{e}-38 \sim 3.4\text{e}+38$ 之间。而双精度变量占有 8 个字节内存单元,有 15~16 位有效数字,数值范围为 $1.7\text{e}-308 \sim 1.7\text{e}+308$ 。具体可参考表 2-2。

定义实型变量:

例如: