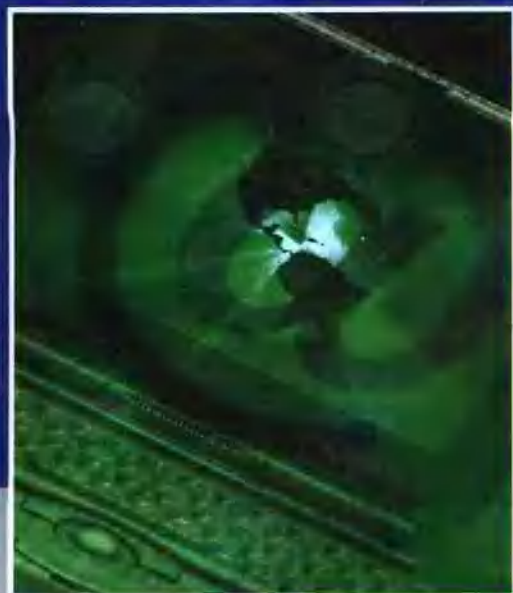


高等学校教材



◎ 主编 郭清华 赵英红

C

语言程序设计

The C Programming
Language



中国石油大学出版社

高等学校教材

C 语言程序设计

主 编 郭清华 赵英红

副主编 李中军 陈 晶

中国石油大学出版社

图书在版编目 (CIP) 数据

C 语言程序设计/郭清华, 赵英红主编. —东营: 中国石油大学出版社, 2007. 8

ISBN 978-7-5636-2435-5

I. C… II. ①郭…②赵… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 128129 号

书 名: C 语言程序设计
主 编: 郭清华 赵英红
副主编: 李中军 陈 晶

.....
责任编辑: 刘玉兰 (0546-8391810)

.....
出 版 者: 中国石油大学出版社 (山东 东营, 邮编 257061)
印 刷 者: 东营市新华印刷厂
电子邮箱: eyi0213@163.com
发 行 者: 中国石油大学出版社 (电话 0546-8392062)
开 本: 185×260 印张: 15.25 字数: 390 千字
版 次: 2007 年 8 月第 1 版第 1 次印刷
定 价: 21.80 元

版权所有, 翻印必究。举报电话: 0546-8391810

本书封面覆有中国石油大学出版社标志的激光防伪膜。

本书封面贴有中国石油大学出版社标志的电码防伪标签, 无标签者不得销售。

前 言

《C 语言程序设计》是计算机及相关专业的专业基础课程之一，也是学习软件开发的入门课程，它担负着培养学生科学的程序设计理念的重任。通过学习 C 语言程序设计的基本方法，让学生在初步掌握一门程序设计语言的基础上，形成良好的结构化程序设计的理念。

本书针对职业院校学生的特点，结合技能型人才培养的目标，注重实用性和实验环节，通过实例逐步展开内容讲述，学习者可以通过学习实例、模仿实例到独立编程，逐步达到掌握编程技术的目的。

全书共分 13 章，以程序设计为主线，全面介绍了结构化程序设计的基本方法和 C 语言的特点。第 1 章介绍了 C 语言的发展、特点，C 程序的基本结构和上机基本方法；第 2 章介绍了算法的概念和 N-S 流程图表示的算法；第 3 章讨论了 C 语言的基本数据类型；第 4 章介绍了 C 语言的运算符与表达式；第 5~7 章讨论了 C 语言的结构程序设计方法（包括顺序、分支和循环）；第 8 章讨论了数组；第 9 章讨论了函数；第 10 章介绍了 C 语言的预处理命令；第 11 章介绍了 C 语言最重要的特色“指针”；第 12 章集中介绍了 C 语言结构体、联合体、枚举和位运算；第 13 章介绍了文件的概念和各种操作。

C 语言是一门实践性很强的课程，实践是学好本课程十分重要的环节，为此，本书在最后提供了 16 个上机实验，基本上按照 C 语言教学规律，安排一周一次上机实验。16 个实验涵盖了 C 语言的基本知识和程序设计常用算法，旨在进一步巩固对基本知识的理解和掌握，提高学生的逻辑分析、抽象思维和程序设计能力，培养优良的程序设计风格。

本书由郭清华、赵英红主编。本书各章习题答案和教材相关电子课件可访问：<http://www.edubzvc.com.cn/jsjx.asp> 下载，或通过电子信箱 guoqinghua@sohu.com 联系作者索取。

由于作者水平有限，书中错误之处在所难免，敬请读者批评指正。

编者
2007 年 6 月

目 录

第1章 C语言概述	1
1.1 C语言出现的历史背景	1
1.2 C语言的特点	2
1.3 C程序结构	2
1.3.1 简单的C程序介绍	2
1.3.2 C程序结构	4
1.4 C语言基本语法成分	5
1.5 C语言程序的编辑、编译、运行（C程序的上机步骤）	6
习 题 1	7
第2章 算 法	8
2.1 算法的概念	8
2.2 简单算法举例	8
2.3 算法的特性	11
2.4 用N-S图表示算法	12
习 题 2	14
第3章 基本数据类型	16
3.1 C的数据类型	16
3.2 常量与变量	17
3.2.1 常量	17
3.2.2 变量	17
3.3 整型数据	18
3.3.1 整型常数的表示方法	18
3.3.2 整型变量	18
3.4 实型数据	20
3.4.1 实型常量的表示方法	20
3.4.2 实型变量	21
3.4.3 实型常量的类型	22
3.5 字符型数据	22
3.5.1 字符常量	22
3.5.2 字符变量	23
3.5.3 字符数据在内存中的存储形式及其使用	23
3.5.4 字符串常量	24
3.6 变量赋初值	25
3.7 各类数值型数据（整型、实型、字符型）的混合运算	25

习 题 3	27
第 4 章 运算符与表达式	28
4.1 C 运算符简介	28
4.2 算术运算符和算术表达式	28
4.3 赋值运算符和赋值表达式	30
4.4 自增、自减运算符	32
4.5 逗号运算符和逗号表达式	33
习 题 4	33
第 5 章 顺序程序设计	35
5.1 C 语句概述	35
5.2 输入/输出及其 C 语言的实现	37
5.3 格式输入/输出	37
5.3.1 printf 函数 (格式输出函数)	37
5.3.2 scanf 函数 (格式输入函数)	42
5.4 其他输入/输出函数	44
5.4.1 putchar 函数 (字符输出函数)	44
5.4.2 getchar 函数 (字符输入函数)	45
5.5 顺序结构程序设计举例	45
习 题 5	47
第 6 章 选择 (分支) 结构程序设计	49
6.1 关系运算符和关系表达式	49
6.1.1 关系运算符及其优先次序	49
6.1.2 关系表达式	50
6.2 逻辑运算符和逻辑表达式	50
6.2.1 逻辑运算符及其优先顺序	50
6.2.2 逻辑表达式	51
6.3 if 语句	52
6.3.1 if 语句三种形式	52
6.3.2 if 语句的嵌套	54
6.3.3 条件运算符 (?:)	57
6.4 switch 语句	58
6.5 程序举例	60
习 题 6	62
第 7 章 循环结构程序设计	64
7.1 while 语句	64
7.2 do-while 语句	66
7.3 for 语句	67
7.4 语句标号和 goto 语句	71
7.4.1 语句标号	71
7.4.2 goto 语句 (无条件转移语句)	71

7.5 循环的嵌套	72
7.6 几种循环的比较	73
7.6.1 循环结构的基本组成部分	73
7.6.2 几种循环的比较	73
7.7 break 和 continue 语句	75
7.7.1 break 语句	75
7.7.2 continue 语句	76
7.8 应用举例	77
习 题 7	79
第8章 数 组	81
8.1 一维数组	81
8.1.1 一维数组的定义	81
8.1.2 一维数组的初始化	82
8.1.3 数组元素的引用	82
8.1.4 一维数组的应用	83
8.2 二维数组	86
8.2.1 二维数组的定义	86
8.2.2 二维数组的初始化	87
8.2.3 二维数组元素的引用	87
8.2.4 二维数组应用举例	87
8.2.5 多维数组	89
8.3 字符数组	89
8.3.1 字符数组的定义、初始化及引用	89
8.3.2 字符串与字符数组	90
8.3.3 字符串（字符数组）处理函数	92
8.3.4 字符数组应用举例	94
习 题 8	96
第9章 函 数	98
9.1 函数概述	98
9.2 函数定义的一般形式	99
9.3 函数的参数和返回值	101
9.3.1 形式参数与实际参数	101
9.3.2 函数的返回值	102
9.4 函数的调用	102
9.4.1 函数调用的一般方法	102
9.4.2 函数的声明	103
9.5 函数的嵌套调用和递归调用	104
9.5.1 函数的嵌套调用	104
9.5.2 函数的递归调用	106
9.6 数组作为函数参数	107

9.6.1 数组元素作为函数参数	107
9.6.2 数组名作为函数的形参和实参	108
9.7 变量的作用域	110
9.7.1 局部变量	110
9.7.2 全局变量	111
9.8 变量的存储类别 (生存期)	113
9.8.1 动态存储方式	113
9.8.2 静态存储方式	114
9.9 内部函数和外部函数	116
9.9.1 内部函数	116
9.9.2 外部函数	116
9.10 多个源程序文件的编译和连接	117
习 题 9	118
第 10 章 编译预处理	119
10.1 宏 定 义	119
10.1.1 不带参数宏定义 (简单替换)	119
10.1.2 带参数宏定义	120
10.2 文件包含	122
10.3 条件编译	123
10.3.1 #ifdef~#endif 和 #ifndef~#endif 命令	123
10.3.2 #if~#endif	123
习 题 10	124
第 11 章 指 针	126
1.1 变量的地址和指针变量	126
11.1.1 指针、指针变量概念及变量的存取方式	126
11.1.2 指针变量的定义	128
11.1.3 指针变量的赋值	128
11.1.4 指针变量的引用	128
11.1.5 指针变量作为函数的参数	130
11.2 数组的指针和指向数组的指针变量	132
11.2.1 指向数组的指针变量 (指向数组元素的指针变量)	132
11.2.2 数组名作为函数参数	134
11.2.3 指向多维数组的指针和指针变量	136
11.3 字符串的指针和指向字符串的指针变量	140
11.3.1 字符串的表示形式	140
11.3.2 字符串指针作为函数参数	141
11.3.3 字符数组和字符指针的区别	142
11.4 指向函数的指针变量	143
11.4.1 函数的指针 使用函数指针调用函数	143
11.4.2 用指向函数的指针作为函数的参数	145

11.5 返回指针值的函数	146
11.6 指针数组与指向指针的指针	148
11.6.1 指针数组	148
11.6.2 指针的指针	149
11.6.3 指针数组作为 main 函数的参数	150
11.7 指针运算举例	151
习 题 11	154
第 12 章 结构体、联合体、枚举、位运算	155
12.1 结构体类型	155
12.1.1 结构体类型和结构体变量	155
12.1.2 结构体变量的引用	157
12.1.3 结构体变量的初始化	158
12.2 结构体数组	159
12.2.1 结构体数组的定义	159
12.2.2 结构体数组的初始化	159
12.3 结构体指针变量	160
12.3.1 结构体指针变量	160
12.3.2 结构体变量、结构体指针变量作函数参数	161
12.4 联合体（共同体）	163
12.4.1 联合体类型、联合体类型变量的定义	163
12.4.2 联合体变量的引用	163
12.5 枚举类型	166
12.6 用 typedef 定义类型	167
12.7 位 运 算	168
习 题 12	170
第 13 章 文 件	172
13.1 文件概述	172
13.2 文件类型指针	173
13.3 文件的打开与关闭	173
13.3.1 文件的打开（fopen 函数）	173
13.3.2 文件的关闭（fclose 函数）	175
13.4 文件的读写	175
13.4.1 字符读写函数	175
13.4.2 字符串读写函数	178
13.4.3 格式化读写函数	179
13.4.4 数据块读写函数（一般用于二进制文件读写）	179
13.5 文件的定位	180
13.5.1 rewind（重返文件头函数）	181
13.5.2 fseek（位置指针移动函数）	181
13.5.3 ftell（获取当前位置指针函数）	182

13.6 出错的检测	182
习 题 13	183
实验部分.....	184
实验 1 Turbo C 集成开发环境和 C 程序实现	184
实验 2 基本数据类型的使用	187
实验 3 运算符与表达式	190
实验 4 顺序结构程序设计	193
实验 5 选择结构程序设计	195
实验 6 循环结构程序设计	198
实验 7 多重循环结构程序设计	201
实验 8 一维数组与二维数组	203
实验 9 字符数组与字符串	206
实验 10 函数的定义、声明和调用	209
实验 11 多文件程序的实现	213
实验 12 编译预处理	216
实验 13 指 针	218
实验 14 结构体、位运算	220
实验 15 文本文件操作	223
实验 16 二进制文件操作	226
附录一 ASCII 码表	230
附录二 C 运算符的优先次序	231
附录三 程序常见错误分析	232

第1章 C语言概述

1.1 C语言出现的历史背景

C语言是目前国际上流行的、很有发展前途的计算机高级语言。它适合作为“系统描述语言”，既可以用来编写系统软件，也可以用来编写应用程序。

早期的操作系统等系统软件主要采用汇编语言编写。由于汇编语言依赖于计算机硬件，程序的可读性、可移植性都比较差，为了提高可读性和可移植性，人们希望采用高级语言编写这些软件，但是一般的高级语言难以实现汇编语言的某些操作，特别是针对硬件的一些操作（如内存地址的读写、直接操作硬件、二进制位的操作等）。人们设法寻找一种既具有一般高级语言特性，又具有低级语言特性的语言，C语言就在这种情况下应运而生。

C语言的发展经历了如下的过程：

ALGOL60 → CPL → BCPL → B → C

其中：

（1）ALGOL60：一种面向问题的高级语言。ALGOL60离硬件较远，不适合编写系统程序。

（2）CPL（Combined Programming Language，组合编程语言）：CPL是在ALGOL60基础上更接近硬件的一种语言。CPL规模较大，实现困难。

（3）BCPL（Basic Combined Programming Language，基本的组合编程语言）：BCPL是对CPL进行简化后的一种语言。

（4）B语言：是对BCPL进一步简化所得到的一种很简单接近硬件的语言。B语言取BCPL的第一个字母。B语言精练、接近硬件，但过于简单，数据无类型。B语言诞生后，Unix开始用B语言改写。

（5）C语言：是在B语言基础上增加数据类型而设计出的一种语言。C语言取BCPL的第二个字母。C语言诞生后，Unix很快用C语言改写，并被移植到其他计算机系统。

最初Unix操作系统是采用汇编语言编写的，B语言版本的Unix是第一个用高级语言编写的Unix。在C语言诞生后，Unix很快用C语言改写，C语言良好的可移植性很快使Unix从PDP计算机移植到其他计算机平台。随着Unix的广泛应用，C语言也得到推广。从此，C语言和Unix像一对孪生兄弟，在发展中相辅相成，Unix和C很快风靡全球。

从C语言的发展历史可以看出，C语言是一种既具有一般高级语言特性，又具有低级语言特性的程序设计语言。C语言从一开始就是用于编写大型、复杂系统软件的，当然C语言也可以用来编写一般的应用程序。

常见的C语言开发环境有Borland公司的Turbo C、Turbo C++、Borland C++、C++

Builder(Windows 版本)和 Microsoft 公司的 Microsoft C、Visual C++(Windows 版本)。

1.2 C 语言的特点

C 语言是从“组合编程语言”(CPL)发展而来,它既具有一般高级语言的特性,又具有低级语言的特性。C 语言具有以下特点:

1. C 语言的语言成分简洁、紧凑,书写形式自由

C 语言程序书写形式自由,主要用小写字母表示,压缩了一切不必要的成分。C 语言程序比其他许多高级语言简练,源程序短,因此输入程序时工作量小。

2. C 语言拥有丰富的数据类型

C 语言具有整型、实型、字符型、数组类型、指针类型、结构体类型、共同体类型等数据类型,能方便地构造更加复杂的数据结构(如使用指针构造链表、树、栈)。

3. C 语言的运算符丰富,功能更强大

C 语言的运算符包含广泛,运算类型丰富,共包括 34 种运算符,灵活使用这些运算符可以实现其他高级语言难以实现的运算。

4. C 语言是结构化的程序设计语言

① C 语言具有结构化的控制语句。

如: if/else、switch/case、for、while、do/while 语句。

② 函数是 C 语言程序的模块单位。

用函数作为程序的模块单位,便于实现程序的模块化。

5. C 语言对语法限制不严格,程序设计灵活

C 语言不检查数组下标越界,不限制对各种数据的转化(编译系统可能对不合适的转化进行警告,但不限制),不限制指针的使用,程序正确性由程序员保证。

6. C 语言编写的程序具有良好的可移植性

C 语言编制的程序基本上不需要修改或只需要少量修改就可以移植到其他计算机系统或其他操作系统。

7. C 语言可以实现汇编语言的大部分功能

① C 语言可以直接操作计算机硬件,如寄存器、各种外设 I/O 端口等。

② C 语言的指针可以直接访问内存物理地址。

③ C 语言类似汇编语言的位操作可以方便地检查系统硬件的状态。

由于以上特点,C 语言适合编写系统软件。

8. C 语言编译后生成的目标代码小,质量高,程序的执行效率高

C 语言的目标代码只比汇编代码效率低 10%~20%。

1.3 C 程序结构

1.3.1 简单的 C 程序介绍

例 1-1

```
main()
```

```
{  
    printf("This is a C program.\n");  
}
```

说明：本程序的功能是输出一行信息“This is a C program.”。

其中：

(1) `main` 表示“主函数”。每个 C 语言程序都必须有一个 `main` 函数，它是每一个 C 语言程序的执行起始点，即程序都从主函数开始执行，而不论主函数的位置如何。

(2) 用“{}”括起来的是主函数 `main` 的函数体，`main` 函数中的所有操作（或语句）都在这一对“{}”之间。也就是说，`main` 函数的所有操作都在 `main` 函数体中。

(3) 本程序主函数 `main` 中只有一条语句，它是 C 语言的库函数，功能是用于程序的输出（显示在屏幕上），本例用于将一个字符串“This is a C program.\n”的内容输出。其中“\n”表示回车换行。即在屏幕上显示：

This is a C program.

(4) 注意：每条语句用“;”结束，“;”是 C 语言的语句结束符。

(5) C 语言中关键字都用小写字母表示，是一种“小写式”的语言。

例 1-2

```
main()                /* 计算两数之和 */  
{  
    int a,b,sum;        /* 这是定义变量 */  
    a=123;b=456;        /* 以下 3 行为 C 语句 */  
    sum=a+b;  
    printf("sum=%d\n",sum);  
}
```

说明：本程序计算两数之和，并输出结果。

(1) 同样，此程序也必须包含一个 `main` 函数作为程序执行的起点。“{}”之间为 `main` 函数的函数体，`main` 函数的所有操作均在 `main` 函数体中。

(2) “/* */”括起来的部分是一段注释，注释只是为了改善程序的可读性，在编译、运行时不起作用（事实上编译时会跳过注释，目标代码中不会包含注释）。注释可以放在程序任何位置，并允许占用多行，只是需要注意“/*”、“*/”匹配，一般不要嵌套注释。

(3) “`int a,b,sum;`”是变量声明，声明了三个具有整数类型的变量 `a`、`b`、`sum`。C 语言的变量必须先声明再使用。

(4) “`a=123;b=456;`”是两条赋值语句，将整数 123 赋给整型变量 `a`，将整数 456 赋给整型变量 `b`。

也可以将两条语句写成两行，即：

```
a=123;  
b=456;
```

由此可见，C 语言程序的书写可以相当随意，但是为了保证容易阅读要遵循一定的规范。

(5) “`sum=a+b;`”是将 `a`、`b` 两变量内容相加，然后将结果赋值给整型变量 `sum`。此时 `sum` 的内容为 579。

(6) “printf(“sum=%d\n”,sum);”是调用库函数输出 sum 的结果。“%d”为格式控制,表示 sum 的值以十进制整数形式输出。程序运行后,输出:

sum=579

例 1-3

```
main()                /* 主函数 */
{
    /* main 函数体开始 */
    int a,b,c;         /* 声明部分定义变量 */
    scanf("%d,%d",&a,&b);
    c=max(a,b);        /* 调用 max, 将调用结果赋给 c */
    printf("max=%d",c);
}
/* main 函数体结束 */
int max(int x,int y)  /* 计算两数中较大的数 */
{
    /* max 函数体开始 */
    int z;             /* 声明部分, 定义变量 */
    if(x>y)z=x;
    else z=y;
    return z;          /* 将 z 值返回, 通过 max 带回调用处 */
}
/* max 函数体结束 */
```

说明: 本程序输入两个整数, 计算两者中较大的数, 并输出。

本程序包括两个函数, 其中主函数 main 仍然是整个程序执行的起点, 函数 max 计算两数中较大的数。

(1) 主函数 main 调用 scanf 函数获得两个整数, 存入 a、b 两个变量, 然后调用函数 max 获得两个数中较大的值, 并赋给变量 c, 最后输出变量 c 的值。

(2) “int max(int x,int y)”是函数 max 的函数首部, 定义函数的函数名、函数的参数及返回值的类型。

(3) 函数 max 同样也用“{ }”将函数体括起来。max 的函数体是函数 max 的具体实现, 从参数表获得数据, 处理后得到结果 z, 然后将 z 返回调用函数 main。

(4) 本例还表明函数除了调用库函数外, 还可以调用用户自己定义编制的函数。

1.3.2 C 程序结构

综合上述三个例子, 我们对 C 语言程序的基本组成和形式(程序结构)有了一个初步了解:

1. C 程序由函数构成(函数是 C 程序的基本单位)

① 一个 C 源程序至少包含一个 main 函数, 也可以包含一个 main 函数和若干个其他函数。函数是 C 程序的基本单位。

② 被调用的函数可以是系统提供的库函数, 也可以是用户根据需要自己编写设计的函数。C 是函数式的语言, 程序的全部工作都由各个函数完成, 编写 C 程序就是编写一个个函数。

③ C 库函数非常丰富, ANSI C 提供 100 多个库函数, Turbo C 提供 300 多个库函数。

2. main 函数(主函数)是每个程序执行的起始点

一个C程序总是从main函数开始执行,而不论main函数在程序中的位置。可以将main函数放在整个程序的最前面,也可以放在整个程序的最后,或者放在其他函数之间。

3. 一个函数由函数首部和函数体两部分组成

① 函数首部:一个函数的第一行。一般格式为:

返回值类型 函数名([函数参数类型 1,函数参数名 1][, ...,函数参数类型 2,函数参数名 2])

注意:函数可以没有参数,但是后面的一对“()”不能省略,这是格式的规定。

② 函数体:函数首部下用一对“{}”括起来的部分。如果函数体内有多个“{}”,最外层是函数体的范围。函数体一般包括声明部分和执行部分两部分,即:

```
{  
    [声明部分]:在这部分定义本函数所使用的变量  
    [执行部分]:由若干条语句组成命令序列(可以在其中调用其他函数)  
}
```

4. C程序书写格式自由

① 一行可以写几条语句,一条语句也可以写在多行上。

② C程序没有行号,也没有FORTRAN、COBOL那样严格规定书写格式(语句必须从某一列开始)。

③ 每条语句的最后必须有一个“;”表示语句的结束。

5. 可以使用“/* */”对C程序中的任何部分作注释

注释可以提高程序可读性,使用注释是编程人员的良好习惯。

① 编写好的程序往往需要修改、完善,事实上没有一个应用系统是不需要修改、完善的。很多人会发现自己编写的程序在经历了一段时间以后,由于缺乏必要的文档、注释,连自己都很难再读懂,需要花费大量时间重新思考、理解原来的程序。如果一开始编程就对程序进行注释,刚开始麻烦一些,但以后可以节省大量的时间。

② 一个实际的系统往往是多人合作开发的,程序文档、注释是其中重要的交流工具。

③ 在软件开发过程中,还可以将注释用于程序的调试,暂时屏蔽一些语句。

例如,在调试程序时暂时不需要运行某段语句,但又不希望立即从程序中删除它们,可以使用注释符将这段语句框起来,暂时将其屏蔽,以后可以方便地恢复。

6. C语言本身不提供输入/输出语句,输入/输出的操作通过调用库函数(scanf、printf)完成

输入/输出操作涉及具体的计算机硬件,把输入/输出操作放在函数中处理,可以简化C语言和C的编译系统,便于C语言在各种计算机上实现。不同的计算机系统需要对函数库中的函数做不同的处理,以便实现同样或类似的功能。

不同的计算机系统除了提供函数库中的标准函数外,还按照硬件的情况提供一些专门的函数,因此,不同计算机系统提供的函数数量、功能会有一定差异。

1.4 C语言基本语法成分

(1) 标识符(名字):用来标识变量名、符号常量名、函数名、数组名、类型名等实体(程序对象)的有效字符序列。标识符由用户自定义(取名字)。

C语言标识符定义规则:

① 标识符只能由字母、数字和下划线三种字符组成，且第一个字符必须为字母或下划线。

例如：

合法的标识符：sum, average, _total, Class, day, stu_name, p4050。

不合法的标识符：M.D.John, \$123, #33, 3D64, a>b。

② 大小写敏感。C 程序员习惯于变量名小写，常量名大写，但不绝对。

例如：sum 不同于 Sum；BOOK 不同于 book。

③ ANSI C 没有限制标识符长度，但各个编译系统都有自己的规定和限制（TC 规定 32 个字符，MSC 规定 8 个字符）。

例如：student_name、student_number，如果取 8 个字符，这两个标识符是相同的。

④ 标识符不能与“关键词”同名，也不能与系统预先定义的“标准标识符”同名。

⑤ 建议：标识符命名应当有一定的意义，做到见名知义。

(2) 关键词（保留字）：C 语言规定的具有特定意义的字符串。

(3) 运算符：运算符将常量、变量、函数连接起来组成表达式，表示各种运算。运算符可以由一个或多个字符组成。

运算符根据参与运算的操作数的个数分为单目、双目、三目运算符。

(4) 分隔符：逗号、空格等。起分隔、间隔作用。

(5) 注释符：“/*”和“*/”构成一组注释符。编译系统将“/*……*/”之间的所有内容看作注释，编译时编译系统忽略注释。

1.5 C 语言程序的编辑、编译、运行（C 程序的上机步骤）

1. 源程序、目标程序、可执行程序的概念

① 程序：为了使计算机能按照人们的意志工作，就要根据问题的要求，编写相应的程序。程序是一组计算机可以识别和执行的指令，每一条指令使计算机执行特定的操作。

② 源程序：程序可以用高级语言或汇编语言编写，用高级语言或汇编语言编写的程序称为源程序。C 语言源程序的扩展名为“.c”。

源程序不能直接在计算机上执行，需要用“编译程序”将源程序翻译为二进制形式的代码。

③ 目标程序：源程序经过“编译程序”翻译所得到的二进制代码称为目标程序。目标程序的扩展名为“.obj”。

目标代码尽管已经是机器指令，但是还不能运行，因为目标程序还没有解决函数调用问题，需要将各个目标程序与库函数连接，才能形成完整的可执行的程序。

④ 可执行程序：目标程序与库函数连接形成的完整的可在操作系统下独立执行的程序称为可执行程序。可执行程序的扩展名为“.exe”（在 DOS/Windows 环境下）。

2. C 语言程序的上机步骤

① 输入与编辑源程序；

② 编译源程序，产生目标代码；

③ 连接各个目标代码、库函数，产生可执行程序；

④ 运行程序；

⑤ 修改错误，调试程序；

⑥ 查看运行结果。

具体操作参见上机指导“实验1”。

习 题 1

1.1 请简述 C 语言的主要特点。

1.2 C 语言的主要用途是什么？它和其他高级语言相比有什么异同？

1.3 写出一个 C 程序的构成。

1.4 C 语言以函数为程序的基本单位有什么好处？

1.5 请编写一个 C 程序，输出以下信息：

Hello World!

1.6 编写一个 C 程序，输入 a、b 两个值，输出它们的和与差。

1.7 上机运行本章例题，熟悉系统的上机方法与步骤。