

北京希望电脑公司计算机系列丛书

C

易学 C 语言基础

姚明廷 编著



海洋出版社

北京希望电脑公司计算机技术丛书

易学 C 语言基础

姚明廷 原著
王真华 改编
希 望 审校



0024749

海洋出版社
1993·北京

内 容 提 要

本书浅显易懂地讲述了 C 语言的基本结构, C 语言的基本语法及 C 语言的规则。读者可以循序渐进地学习本书, 本书在每章均附有例题程序, 读者可以试运行, 并看看运行结果是否与本书相符。希望本书的出版能对读者有所帮助。

需要本书的用户请直接与北京 8721 信箱联系, 邮政编码 100080, 电话 256329。

版 权 声 明

本书繁体字中文版原书名为《易学 C 语言基础》由松岗电脑图书资料股份有限公司出版, 版权归松岗公司所有。本书简体字中文版版权由松岗公司授予北京希望电脑公司, 由北京希望电脑公司和海洋出版社独家出版、发行。未经出版者书面许可, 本书的任何部分不得以任何形式或任何手段复制或传播。

(京) 新登字 087 号

北京希望电脑公司计算机技术丛书
易学 C 语言基础

姚明廷 原著
王真华 改编
希 望 审校

海洋出版社 (北京市复兴门外大街 1 号)

海洋出版社发行 双青印刷厂印刷

开本: 787×1092 1/16 印张: 19.75 字数 449 千字

1993 年 10 月第一版 1993 年 10 月第一次印刷

印数: 1-5000 册

ISBN7-5027-3837-5/TP·237

定价: 25.00 元

JS363/09

前 言

C 语言是目前计算机界流行的一种编程方式，学习 C 语言已成为一种必然趋势，目前市面上虽然已经有许多有关 C 语言编程方面的书，但一般都比较高深，对初学者不太适合。作者正是为了满足初学者的需求，编写本书，本书也可作为学生和老师的教材，希望本书的出版能给读者有所帮助。

本书共分为十二章及五个附录，第一章 C 语言简介；第二章数据类型；第三章运算符；第四章流程控制；第五章函数；第六章数组；第七章指针；第八章字符串；第九章结构与联合；第十章文件；第十一章前置处理程序；第十二章变量的等级；附录一扩充 ASCII 码表；附录二 Turbo C 函数一览表；附录三常见错误信息；附录五 DOS ROM BIOS 程序。

由于松岗电脑图书股份公司和北京希望电脑公司的通力合作，才得以使本书顺利出版，在此表示衷心感谢。

笔者编写此书虽力求完善，错误之处仍在所难免，希望各位先生、前辈及同学们不吝指正，在此表示最诚致的谢意。

编 者

1993 年 9 月

目 录

第一章 C语言简介	(1)
第一节 C的历史	(1)
第二节 C语言的特性	(2)
第三节 Turbo C简介	(5)
第四节 IDE集成工作环境	(6)
第五节 C的基本语法	(10)
第六节 C语言的规则	(13)
第七节 编译链接C程序	(15)
第八节 Turbo C编译程序的目录设定	(16)
第九节 保留字说明	(22)
第二章 数据类型	(25)
第一节 数字系统	(25)
第二节 2的补数(2's complement)	(28)
第三节 整数的声明与赋值	(29)
第四节 基本数据类型	(31)
第五节 数据类型的修饰字	(42)
第六节 表达式的类型转换	(46)
第七节 自定义类型数据	(47)
第三章 运算符	(51)
第一节 算术运算符	(51)
第二节 递增(Increment)和递减(Decrement)运算符	(52)
第三节 关系运算符	(53)
第四节 逻辑(Logical)运算符	(54)
第五节 比特运算符	(55)
第六节 指定运算符(Assignment Operators)	(64)
第七节 条件运算符	(66)
第八节 类型转换运算符(Cast Operators)	(67)
第九节 其他运算符	(69)
第十节 运算符执行的优先顺序	(70)
第四章 流程控制	(72)
第一节 if_else语句	(72)
第二节 switch语句	(78)
第三节 for语句	(81)
第四节 while循环	(87)

第五节	do_while 语句	(91)
第六节	goto 语句	(92)
第五章	函 数	(94)
第一节	何谓函数	(94)
第二节	函数的组成	(94)
第三节	函数的类型	(96)
第四节	输入输出函数	(100)
第五节	传值调用于传址调用	(119)
第六节	返回的概念	(122)
第六章	数 组	(127)
第一节	一维数组	(127)
第二节	数组初值	(129)
第三节	数组的应用	(131)
第四节	二维数组	(139)
第五节	二维数组的应用	(141)
第六节	魔术方阵	(142)
第七节	多维数组	(145)
第七章	指 标	(147)
第一节	8086 / 8088 记忆性定址	(147)
第二节	何谓指针(pointer)	(148)
第三节	指针的种类	(150)
第四节	指针的初值	(151)
第五节	指针的运算	(152)
第六节	指针于函数	(154)
第七节	指针于数组	(159)
第八节	指针数组	(162)
第九节	指针的指针	(166)
第十节	main 函数的参数	(167)
第八章	字 串	(170)
第一节	字串的基本性质	(170)
第二节	库存字串函数	(171)
第三节	自行拷贝库存的字串函数	(180)
第四节	自定义字串函数	(184)
第五节	制作函数文件	(192)
第六节	如何使用函数库	(197)
第九章	结构于联合	(200)
第一节	结构的类型	(200)
第二节	结构成员于初值	(201)
第三节	typedef	(204)

第四节	结构数组	(205)
第五节	结构的地址运算	(207)
第六节	结构指针	(210)
第七节	嵌套结构	(212)
第八节	比特结构	(214)
第九节	联合	(218)
第十节	寄存器于联合	(220)
第十章	文件	(225)
第一节	C 语言文件处理基本概念	(225)
第二节	高级打开、关闭文件函数	(227)
第三节	高级的文件 I/O 函数	(229)
第四节	随机存取	(240)
第五节	低阶文件函数	(245)
第六节	标准输入、输出设备	(248)
第七节	文件处理相关函数	(250)
第十一章	前置处理程序	(255)
第一节	宏定义指令	(255)
第二节	常数宏	(256)
第三节	字符串取代宏	(257)
第四节	宏函数	(259)
第五节	含入文件	(263)
第六节	条件编译	(267)
第十二章	变量的等级	(271)
第一节	变量的使用范围	(271)
第二节	存储类别	(274)
第三节	auto	(276)
第四节	register	(276)
第五节	extern	(277)
第六节	static	(278)
附录一	扩充 ASCII 码表	(280)
附录二	Turbo C 函数一览表	(286)
附录三	TCC 的参数说明	(293)
附录四	常见错误信息	(295)
附录五	DOS ROM BIOS 常式	(303)

例题目录

第一章 C语言简介	(1)
Ch1_1 一个由上而下设计的C程序结构	(4)
Ch1_2 说明C语言的程序模式	(11)
Ch1_3 C语言程序模式说明	(12)
第二章 数据类型	(25)
Ch2_1 显示各数据类型的长度	(31)
Ch2_2 char类型数据含字符和数值二种特性	(32)
Ch2_3 字符类型数据的运算	(32)
Ch2_4 字符类型数据的运算	(33)
Ch2_5 char类型数据使用范围	(34)
Ch2_6 int类型数据的各种表示法	(36)
Ch2_7 int类型数据的溢位	(36)
Ch2_8 浮点数表示法	(37)
Ch2_9 用int检查float类型数据存储方式	(38)
Ch2_10 浮点数的精确度	(39)
Ch2_11 浮点数正确的书写方法	(39)
Ch2_12 浮点数的精确度	(40)
Ch2_13 浮点数的精确度	(41)
Ch2_14 浮点数的数据格式	(42)
Ch2_15 short修饰字	(43)
Ch2_16 long修饰字	(43)
Ch2_17 long类型数据的表示法	(44)
Ch2_18 unsigned与signed比较	(45)
Ch2_19 运算结果的类型	(46)
Ch2_20 union和struct的比较	(47)
Ch2_21 列举类型——洗牌	(49)
第三章 运算符	(51)
Ch3_1 余数运算%的使用	(51)
Ch3_2 递增、递减运算的效率	(52)
Ch3_3 递增、递减运算的前置与位置	(53)
Ch3_4 关系表达式的传回值	(53)
Ch3_5 AND和OR逻辑的真值表	(54)
Ch3_6 NOT逻辑表达式的传回值	(55)
Ch3_7 用&比特运算符转变大小写字母	(56)

Ch3_8	用 位逻辑运算符改变大小写	(57)
Ch3_9	用 · 比特运算符观察变量是否相等	(58)
Ch3_10	用 ~ 模拟 ~ 的效果	(58)
Ch3_11	用 · 作数据加密	(59)
Ch3_12	用 ~ 比特运算符求负数	(60)
Ch3_13	用 ~ 比特运算符改变大小写	(61)
Ch3_14	用比特运算符观察变量内容	(62)
Ch3_15	用位移比特运算符模拟 ROL 旋转指令	(63)
Ch3_16	运算的效率的比较	(65)
Ch3_17	连续给值与指定运算	(65)
Ch3_18	if-else 求绝对值	(66)
Ch3_19	? 和 #define	(67)
Ch3_20	#define 的误用	(67)
Ch3_21	类型转换的用法	(68)
Ch3_22	类型转换的必要性	(68)
Ch3_23	取址与指针	(69)
第四章	流程控制	(72)
Ch4_1	计程车车资计算 if 语句	(73)
Ch4_2	计程车车资计算 if_else 语句	(73)
Ch4_3	计程车车资计算?: 运算	(74)
Ch4_4	嵌套 if	(74)
Ch4_5	闰年判断嵌套 if 语句	(76)
Ch4_6	闰年判断逻辑关系运算符	(76)
Ch4_7	闰年判断给值判断式	(77)
Ch4_8	switch 多重选择	(79)
Ch4_9	数字转中文字 switch 的应用	(80)
Ch4_10	for 循环的应用标准写法	(81)
Ch4_11	for 循环的应用省略写法	(82)
Ch4_12	for 循环的应用合作写法	(83)
Ch4_13	九九乘法表嵌套 for 循环	(83)
Ch4_14	时间显示无穷 for 循环	(84)
Ch4_15	数字输入程序 continue	(85)
Ch4_16	while 循环的应用标准写法	(87)
Ch4_17	字符串复制 while 的应用	(88)
Ch4_18	九九乘法表嵌套 while 循环	(88)
Ch4_19	密码输入程序	(89)
Ch4_20	时间显示 do while 循环	(91)
Ch4_21	goto 语句	(92)

第五章 函数	94)
Ch5_1 K&R 和 ANSI 函数格式的比较	(95)
Ch5_2 参数类型的一致性	(96)
Ch5_3 函数(传回值)的类型	(96)
Ch5_4 K&R 函数不指定类型即表 int	(97)
Ch5_5 void 类型函数	(98)
Ch5_6 void 类型函数的声明	(98)
Ch5_7 声明参数个数与类型	(99)
Ch5_8 利用 getch()暂停程序用户按下任意键可继续	(101)
Ch5_9 用 getch()取得用户按键	(101)
Ch5_10 用 getch()读取文件数据请用重定向(redirection)执行本程序	(101)
Ch5_11 用 getch()读取文件数据	(102)
Ch5_12 用 putch()显示数据	(103)
Ch5_13 getchar()和 putchar()	(104)
Ch5_14 putchar()转向能力	(105)
Ch5_15 gets()和 puts()	(106)
Ch5_16 用 gets()读取文件数据请用重定向(redirection)执行本程序	(106)
Ch5_17 printf()的格式字符串-控制字符串靠左	(108)
Ch5_18 printf()的格式字符串+在正数前加上+号	(109)
Ch5_19 printf()的格式字符串' '在正数前加上' '	(109)
Ch5_20 printf()的格式字符串#的用法	(110)
Ch5_21 printf()的格式字符串 0 在数字前补 0	(111)
Ch5_22 printf()的格式字符串数据长度的设定	(111)
Ch5_23 printf()的格式字符串可变量据长度的设定	(112)
Ch5_24 printf()的格式字符串数据长度的设定	(114)
Ch5_25 scanf()的用法变量的传址调用	(114)
Ch5_26 scanf()的格式字符串分隔字符	(115)
Ch5_27 scanf()的格式字符串输入数据长度的限制	(116)
Ch5_28 scanf()的格式字符串%s 无法输入空白数据	(118)
Ch5_29 Call by Value	(119)
Ch5_30 Call by Address	(121)
Ch5_31 一般函数求阶乘 $N! = N * (N-1) * (N-2) \dots * 2 * 1$	(123)
Ch5_32 递归函数求阶乘 $N! = N * (N-1)!$	(124)
Ch5_33 求 X^Y 用 math.h 的 pow(2)用一般函数 pow(13)用递归 函数 pow2	(125)
第六章 数组	(127)
Ch6_1 数组的大小测试	(128)
Ch6_2 数组的地址	(128)
Ch6_3 数组的初值	(129)

Ch6_4	求数组的平均值	(130)
Ch6_5	泡沫排序法	(132)
Ch6_6	选择排序法	(133)
Ch6_7	插入排序法	(135)
Ch6_8	库存快速排序法	(136)
Ch6_9	快速排序法	(137)
Ch6_10	二维数组的大小	(139)
Ch6_11	二维数组的地址	(140)
Ch6_12	二维数组的初值及二维数组的操作	(140)
Ch6_13	字符串排序	(141)
Ch6_14	4X4 奇数魔术阵	(143)
Ch6_15	5X5 奇数魔术方阵	(144)
Ch6_16	三维数组的操作	(146)
第七章 指 标	(147)
Ch7_1	指针的长度	(149)
Ch7_2	指针的种类不正确的操作	(150)
Ch7_3	指针的种类	(151)
Ch7_4	指针的初值	(152)
Ch7_5	指针的运算	(152)
Ch7_6	指针的运算	(153)
Ch7_7	指针的传址调用	(154)
Ch7_8	指针的运算	(155)
Ch7_9	指向函数的指针	(156)
Ch7_10	指向函数的指针	(157)
Ch7_11	指向函数的指针	(157)
Ch7_12	指针与数组	(159)
Ch7_13	指针与下标	(160)
Ch7_14	数组的地址是常数这是无法编译的程序, 会产生 Lvalue required in function main 的错误	(161)
Ch7_15	指针与二维数组	(161)
Ch7_16	指针数组	(162)
Ch7_17	数组长度不足的错误	(164)
Ch7_18	字符串排序	(164)
Ch7_19	指针的指针	(166)
Ch7_20	main()函数的参数未传参数的情况	(167)
Ch7_21	main()函数的参数传入参数的情况	(167)
第八章 字 串	(170)
Ch8_1	strcpy()	(172)
Ch8_2	strncat()	(172)

Ch8_3	strchr()	(173)
Ch8_4	strrchr()	(174)
Ch8_5	strstr()	(174)
Ch8_6	strcmp()&strncmp()	(175)
Ch8_7	stricmp()&strncmpi()	(176)
Ch8_8	strcpy()&strncpy()	(177)
Ch8_9	strdup()	(178)
Ch8_10	strspn()&strcspn()	(178)
Ch8_11	strtok()	(179)
Ch8_12	stradd()	(180)
Ch8_13	stcopy()	(181)
Ch8_14	stcomp()	(181)
Ch8_15	strlen()	(182)
Ch8_16	replicate()	(183)
Ch8_17	reverse()	(183)
Ch8_18	left()	(184)
Ch8_19	right()	(185)
Ch8_20	substr()	(186)
Ch8_21	stuff()	(187)
Ch8_22	space()	(188)
Ch8_23	pad()	(189)
Ch8_24	lpad()	(190)
Ch8_25	trim()	(190)
Ch8_26	ltrim()	(191)
第九章	结构与联合	(200)
Ch9_1	结构的声明与初值	(201)
Ch9_2	结构的声明与初值	(202)
Ch9_3	结构的元素使用	(203)
Ch9_4	结构的指定运算	(203)
Ch9_5	typedef 的应用	(204)
Ch9_6	typedef 与结构	(205)
Ch9_7	结构数组	(206)
Ch9_8	结构数组元素与结构指定运算	(206)
Ch9_9	结构成员的地址运算	(207)
Ch9_10	结构的地址运算	(209)
Ch9_11	结构的地址	(209)
Ch9_12	结构的指针	(211)
Ch9_13	嵌套结构	(212)
Ch9_14	>> 运算与比特字段	(214)

Ch9_15	运算与比特字段	(215)
Ch9_16	读取键盘状态	(216)
Ch9_17	联合与结构的比较	(218)
Ch9_18	存储器共用的情况	(218)
Ch9_19	用联合读取键盘状态	(219)
Ch9_20	读取键盘状态	(223)
第十章	文 件	(225)
Ch10_1	文件缓冲区的大小	(228)
Ch10_2	文件字符 I/O fgetc()函数	(230)
Ch10_3	文件字符 I/O fgetc()函数	(231)
Ch10_4	读取二元文件数据	(231)
Ch10_5	文件字符 I/O fputc()函数	(232)
Ch10_6	文件字符 I/O ungetc()函数	(233)
Ch10_7	文件字符串 I/O fgets()函数	(234)
Ch10_8	用 fgetc()函数编写 fgets()函数	(235)
Ch10_9	文件字符串 I/O fputs()函数	(236)
Ch10_10	文件格式化 I/O sprintf() fscanf()函数	(236)
Ch10_11	文件记录 I/O fread() fwrite()函数	(239)
Ch10_12	文件长度	(242)
Ch10_13	显示 DBF 数据	(243)
Ch10_14	文件缓冲区的大小	(246)
Ch10_15	低级文件的复制	(247)
Ch10_16	用 stdio 显示文件	(249)
Ch10_17	用 stdprn 打印文件	(249)
Ch10_18	用 PRN 打印文件	(250)
Ch10_19	列出文件	(253)
第十一章	前置处理程序	(255)
Ch11_1	const 和 #define 的比较	(256)
Ch11_2	const 和 #define 的错误用法	(257)
Ch11_3	判别三角形的类别	(257)
Ch11_4	宏函数的用法	(259)
Ch11_5	宏函数和一般函数的比较	(260)
Ch11_6	宏函数的陷阱	(261)
Ch11_7	宏函数的正确写法	(261)
Ch11_8	显示 1-10 的平方值	(262)
Ch11_9	未使用含入文件的库存函数	(264)
Ch11_10	使用含入文件的库存函数	(264)
Ch11_11	自定义类型的库存函数	(264)
Ch11_12	自定义含入文件	(265)

Ch11_13 中英文书框.....	(268)
第十二章 变量的等级	(271)
Ch12_1 变量的使用的范围.....	(271)
Ch12_2 区块与区域变量.....	(273)
Ch12_3 区域变量的地址.....	(274)
Ch12_4 整型变量的地址.....	(275)
Ch12_5 auto 变量	(276)
Ch12_6 register 变量	(276)
Ch12_7 extern 变量.....	(277)
Ch12_7_1 extern 变量	(278)
Ch12_8 static 区域变量	(279)

第一章 C 语言简介

本章介绍关于 C 语言的背景资料，并介绍 Turbo C 的基本用法。

第一节 C 的历史

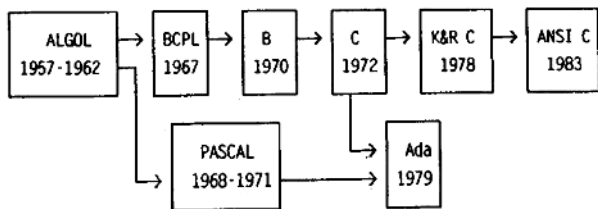
C 语言本身就如同它的名字一样地简洁，究竟 C 这个名字是由何而来的呢？

1972 年贝尔实验室(Bell Lab)的 Dennis Ritchie 和 Ken Thompson 在设计 UNIX 操作系统的时候创造了 C 语言。Ken Thompson 曾在加州柏克莱大学开发 Pascal 语言，后来加入贝尔实验室，参加了 MULTICS 方案，这个方案是贝尔实验室、麻省理工学院(MIT)和通用电器公司(GE)合作，共同开发一套交互式、多用户、多任务的操作系统。不过贝尔实验室不久后就退出这个方案，但 Thompson 仍锲而不舍的在 DEC PDP-7 的机器上用汇编语言编写这套操作系统。他在贝尔实验室的同事 Brian Kernighan 将这套单用户的操作系统称为 UNIX。

接着 Thompson 把 UNIX 由 PDP-7 移植到 DEC PDP-11 的机器，并参照 Martin Richards 的 BCPL 语言 (Basic Combined Programming Language) 开发一种名为 B 语言。Ritchie 加入贝尔实验室的 UNIX 开发小组后，对于 B 语言很有兴趣，在 1972 年时，将 B 语言改写成 NB 也就是 C 语言。C 的名称其实就源自于 B。Ritchie 并且用 C 语言改写 UNIX 操作系统。在这里我们也可以看到 C 语言和 UNIX 操作系统之间深厚的渊源。C 是在 UNIX 下开发出来的，但后来 UNIX 系统又是用 C 写成的。

1978 年 Brian W. Kernighan 和 Dennis M. Ritchie 合著的 The C Programming Language，被誉为是 C 语言的经典之作，书中详述了 C 语言的基本规则。以后 C 语言都以此书作为重要依据，这些 C 通称为 K&R 版本。1983 年美国国家标准局 (American National Standard Institute，简称 ANSI) 定义了一个新版本，称为 ANSI 版本，这是一个标准版本的 C。

C 程序语言的演变



第二节 C 语言的特性

一、C 是一种高级语言

传统上, 计算机语言可区分为两大类:

- (1) 高级语言 (High-Level Language), 如 COBOL, BASIC, PASCAL, ADA 等等。这类语言接近人类所用的语言, 所设计出来的程序具有结构化, 也能在不同的计算机上执行。
- (2) 低级语言 (Low-Level Language), 通常指“汇编语言”(Assembly Language), 这类语言接近机器语言, 设计出来的程序, 占主存储器空间较少, 执行效率又比高级语言来得高。

C 出身于小型计算机 (Mini Computer), 由语法十分简便的语言 B C P L 与 B 所延伸而成, 因此其设计哲学崇尚精简, 但其关键字又含有高级语言的特性 (参照下面 C 关键字列表), 又由于 C 最初是为设计操作系统而制作的, 所以非常注重执行效率, 因此 C 语言具有高低级语言二者的优点, 常被称为是一种高级语言, 它拥有高级语言的多种数据类型及高度的可移植性 (Portability), 以及低级语言直接操作位 (Bits)、字节 (Bytes)、地址 (Address) 及 CPU 的能力。

ANSI C 保留字

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

二、C 具有可移植性 (Portability)

所谓可移植性 (Portability) 的高低是衡量一个程序在不同类型计算机上执行的难易程度, 例如: 按 ANSI 版本编写的 C 程序, 能在 IBM PC 上编译、链接及运行; 也能在 VAX 小型及 IBM 大型甚至 CRAY 超级计算机上运行, 也就是说, ANSI 版本的 C 能在不同类型的计算机上运行, 若想把 ANSI C 程序移植到另一种类型的计算机上时, 可完全不修改源程序, 只需重新编译, 链接就能在不同类型的计算机上运行, 而且得到相同的结果。

C 如何达到其高度的可移植性呢? 主要它是利用两种重要类型的文件:

- (1) 头文件 (Header Files), 又称为包含文件 (Include Files),
- (2) 函数档 (Function Libraries), 又称为函数库 (Libraries),

凡是跟硬件设备或特殊操作系统有密切关系指令, 都放在头文件或函数库之内, 不直

接编写在 C 源程序中，如此可以分离 C 源程序与计算机硬件设备的直接关系。头文件及函数库的内容因计算机种类而不同，C 源程序在不同计算机上重新编译时，会引用存放在该计算机上的头文件；在链接时会与该计算机上的函数库链接起来，于是就产生可在该计算机上执行的程序了。

三、C 是一种结构化的语言

在计算机术语中所谓结构化指的是程序由以下三种基本模式所组成的：

- (一) 顺序(Sequence)：即依照指令编写的顺序，一个指令接一个指令的依序执行。
- (二) 选择(Selection)：程序在执行的过程中有些时候要去判断一些情况，然后再根据这些判断的结果，选择适当的执行程序。
- (三) 重复(Repetition)：有些时候程序中有些反复的动作要不断的执行，我们会利用一个循环去执行这些程序。

一个结构化的程序，比较容易了解，修改也较方便，可以降低软件维护的成本。

在 C 中，我们可以在这三种模式下完成所有的工作。此外在 C 中除了指令具有结构化的特性，我们还可以把一组指令放在一个由一对 { } 所组成的区块 (block) 中，这些区块也具有结构化的特性。

四、C 适合模块化 (Modular)

C 的区块和结构化的特点，使 C 特别适合模块化，所谓模块化就是预先设计好一组的工具程序，真正编写程序时，再通过前面三种结构把这些工具程序组合起来。模块化可提高程序开发的效率，降低开发成本，此类程序也比较容易维护。在 C 中模块化最具体的表现就是函数，事实上，C 程序本身就完全是由函数所组成的。如在前面所提到的，一般的 C 在不同的机器上，会提供不同的函数库，供连接用。这些函数库中的函数，因机器不同可能有不同编写的方法，但每个相对应的函数却都有相同的函数名称，所以对程序员来说，只要知道标准函数库每个函数名称所提供的功能即可，而利用这个标准函数库中函数所设计的程序，也可以很快的移植到不同的机器上。

标准函数库中的函数，如果不够用，我们还可以向外面购买各式各样特殊功能的程序库，例如绘图、数据库、中文系统等等。当然，随时把自己所开发的程序模块化，作成新的函数是最重要的一件事了。

五、C 适合由上而下的设计 (Top to Down)

所谓 Top to Down 由上而下的设计概念是在系统分析上常用的一种方法，其概念主要源自模块化的概念。我们在设计程序时，可以先把所要处理的工作分作一些细项，然后针对每一细项再一步一步往下细分。这种系统分析的方式可以使我们在第一步骤，专注于处理属于该层次的问题，并可以清晰的掌握系统的模式。例如我们想要设计一个客户管理系统，我们要考虑这样的系统要包含哪些功能，譬如说：我们想要包括新增、删除、修改、查询、打印等五项功能。接着我们再去考虑在新增时，要如何处理数据输入，每个数据项处理的方法等下一层的问题。如果把上述事项处理流程用伪代码的方式表示出来。