



新浪微博：C语言图书
腾讯QQ：4006751066

C语言学习路线图



C语言必须知道的 300个问题



明日科技 编著

本书提供了内容丰富的配套资源，可以登录www.tup.com.cn，找到本书后，在该页面的“网络资源”超链接处下载。也可以访问本书的新浪微博，根据提示链接下载。

清华大学出版社

C 语言学习路线图

C 语言必须知道的 300 个问题

明日科技 编著

清华大学出版社

北 京

内 容 简 介

本书以基础知识为框架，介绍了 C 语言各部分知识所对应的常见开发疑难问题，并作了透彻地解析。本书内容包括初识 C 语言，一个简单的 C 程序，算法入门，常用数据类型，运算符与表达式，输入/输出函数，选择、分支结构程序设计，循环结构，数组，函数编程基础，指针解析，常用数据结构，位运算操作符，存储管理，预处理和函数类型，文件的读写操作和图形图像处理。

书中列出的 300 个疑难问题，都是一线开发人员在实际项目中经常会遇到的。针对这些问题，本书给出了详尽的解决方案，并作了技术上的解析。通过对本书的学习，读者能够迅速提高解决实际问题的能力。

本书提供了源程序、素材，提供了相关的模块库、案例库、素材库、题库等多种形式的辅助学习资料，还提供迅速及时的微博、QQ、论坛等技术支持。

本书涉及内容丰富经典，讲解细致，适合各种水平的 C 语言学习者进行学习查阅，还可以作为软件开发人员的参考资料。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目 (CIP) 数据

C 语言必须知道的 300 个问题/明日科技编著. —北京：清华大学出版社，2012.2
(C 语言学习路线图)

ISBN 978-7-302-27667-8

I. ①C… II. ①明… III. ①C 语言-程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 267885 号

责任编辑：赵洛育 刘利民

版式设计：文森时代

责任校对：张兴旺

责任印制：

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 刷 者：

装 订 者：

经 销：全国新华书店

开 本：185mm×260mm 印 张：25.75 字 数：595 千字

版 次：2012 年 2 月第 1 版 印 次：2012 年 2 月第 1 次印刷

印 数：1~5000

定 价：49.80 元

产品编号：045364-01

前言

Preface



学会站在巨人的肩膀上!

程序员在求职时，经常会被问到有关开发经验的问题。例如，“从事了多少年的开发？”、“参与了哪些大型项目？”……为何面试官对项目经验丰富的求职者青睐有加？究其根源，是由于程序开发是一门实践性非常强的学科。正所谓“纸上得来终觉浅，绝知此事要躬行”。编程经验是程序开发者在长期的实践中逐步积累的宝贵的技术资源库，其中包含各种习惯用法、知识理论、代码片段和应用技巧等。要想成为经验丰富的编程高手，除了勤学苦练，也要学会站在前人的肩膀上，通过分析别人的代码而有所领悟，或者借鉴他人的经验技巧形成自己的技能，在认识错误与解决问题中不断进步。这也是每个编程者的必由之路。

本书汇集了 300 个一线开发人员常见的疑难问题，每个问题都给出了详细的解答与点评，图文并茂、难易并举，通过对本书的学习，读者可以尽享前人的开发经验，迅速提升个人的开发能力。

本书内容

本书以基础知识为框架，介绍了 C 语言各部分知识所对应的常见开发疑难问题，并作了透彻地解析。全书共分为 17 章，主要内容包括初识 C 语言，一个简单的 C 程序，算法入门，常用数据类型，运算符与表达式，输入/输出函数，选择、分支结构程序设计，循环结构，数组，函数编程基础，指针解析，常用数据结构，位运算操作符，存储管理，预处理和函数类型，文件的读写操作和图形图像处理。

为了更清晰地阐述问题并给出详尽的解决方案，本书设置了以下栏目。

问题阐述

对问题进行详细阐述，对复杂问题给出问题产生的条件，并对问题中需要解决的部分作出说明。

专家解答

根据问题进行具体分析，给出相应的解题思路及正确答案，并提供该问题涉及的技术知识。

专家点评

对问题及解答进行总结，为以后的问题解决提供思路；此外，还会列出一些有关此问题应该注意的事项，以及对该问题的一些拓展。



本书特色

贴近应用

本书精选的疑难问题都是在实际项目开发中经常会遇到的，主要目的就是为读者答疑解惑。

横向链接

本书知识框架与《C 语言开发入门及项目实战》一书相对应，可以在使用《C 语言开发入门及项目实战》一书进行基础学习之后，使用本书丰富并提高技能。

解析透彻

本书对每个问题的相关知识都作了细致地讲解，并进行知识拓展，使读者不仅知其然而且知其所以然。

授人以渔

本书在讲解技术的同时，还注重对读者能力的培养，使读者真正掌握分析问题与解决问题的能力。

本书配套资源

本书提供了内容丰富的配套资源，包括源程序、素材，以及模块库、案例库、题库、素材库等多项辅助内容，读者可以通过如下方式获取。

第 1 种方式：

(1) 登录 www.tup.com.cn，在网页右上角的搜索文本框中输入本书书名（注意区分大小写和留出空格），或者输入本书关键字，或者输入本书 ISBN 号（注意去掉 ISBN 号间隔线“-”），单击“搜索”按钮。

(2) 找到本书后单击超链接，在该书的网页下侧单击“网络资源”超链接，即可下载。

第 2 种方式：

访问本书的新浪微博：C 语言图书，找到配套资源的链接地址进行下载。

读者人群

本书非常适合以下人员阅读：

- 从事 C 语言编程的开发人员
- 有一定语言基础，想进一步提高技能的人员
- 大中专院校的老师 and 学生
- 即将走上相关工作岗位的大学毕业生
- 相关培训机构的老师和学员
- C 语言编程爱好者



Note



读者服务&本书勘误

读者在使用本书过程中遇到的所有问题，均可通过以下方式联系我们。

1. 新浪微博：C 语言图书。

及时发布读者答疑、本书勘误、配套资料更新等内容。

2. 腾讯 QQ 群：4006751066

3. 登录网站：www.mingribook.com，在论坛、勘误发布、读者纠错、技术支持、读者之家等栏目中的相关模块中提问、留言或查看。

本书作者

本书由明日科技组织编写，参与编写的有孙秀梅、曹飞飞、王雪、朱晓、赵永发、李鑫、潘凯华、刘欣、李慧、高春艳、王小科、赵会东、李继业、赛奎春、王国辉、陈丹丹、李伟、杨丽、李丽、刘龄龄、王明招、孙茜、陈英、肖鑫等。

由于作者水平有限，疏漏和不足之处在所难免，敬请广大读者朋友批评指正。

编者



Note

目 录

Contents



第 1 章 初识 C 语言..... 1

- 问题 1 C 语言是如何发展起来的? 2
- 问题 2 什么是 ANSI 标准? 2
- 问题 3 C 语言编写程序的优点有哪些? ... 3
- 问题 4 如何规避 C 语言的不足之处? 4
- 问题 5 C 语言的应用领域有哪些? 5
- 问题 6 什么是 C99 标准? 与 C89 标准
相比, C99 标准有哪些
新特性? 6
- 问题 7 C 语言是 C++ 的子集吗? 10
- 问题 8 C 语言程序的开发过程是
怎样的? 11
- 问题 9 什么是编译程序和解释程序? 12
- 问题 10 ANSI C 的编译限制有哪些? 13

第 2 章 一个简单的 C 程序..... 14

- 问题 11 C 语言的入口函数是什么? 15
- 问题 12 C 语言程序由哪些部分组成? 16
- 问题 13 如何在 Turbo C 2.0 中输入一个
程序? 16
- 问题 14 如何在 Visual C++ 6.0 中运行
一个 C 程序? 18
- 问题 15 如何在 Visual C++ 2008 中运行
一个 C 程序? 22
- 问题 16 如何提高程序的可读性? 26
- 问题 17 什么是关键字? C 语言的关键
字有哪些? 27
- 问题 18 什么是标识符? 使用标识符的
注意事项是什么? 29
- 问题 19 void 关键字都有哪些用途? 30
- 问题 20 什么是匈牙利命名约定? 它是否

是好的约定? 31

第 3 章 算法入门..... 33

- 问题 21 为什么说算法是程序设计的
灵魂? 34
- 问题 22 算法的特性有哪些? 34
- 问题 23 如何评价一个算法的好坏? 35
- 问题 24 算法的表示方法都有哪些? 36
- 问题 25 算法的基本结构是什么? 40
- 问题 26 算法有哪几类? 42
- 问题 27 算法的效率度量方法有哪些? .. 42
- 问题 28 什么是算法的时间复杂度? 43
- 问题 29 什么是算法的空间复杂度? 44
- 问题 30 什么是分治法算法思想? 45

第 4 章 常用数据类型..... 46

- 问题 31 声明变量和定义变量的区别
是什么? 47
- 问题 32 在开发时如何决定使用哪种
数据类型? 48
- 问题 33 什么是常量? 如何区分常量
和变量? 48
- 问题 34 各种数据类型所占的内存是
多少? 49
- 问题 35 字符与字符串的差别有
哪些? 50
- 问题 36 变量是否必须初始化? 51
- 问题 37 为什么会发生数据溢出? 如何
避免数据溢出? 52
- 问题 38 局部变量和全局变量能否
重名? 53



- 问题 39 全局变量可不可以定义在可被多个.C 文件包含的头文件中? 为什么? 53
- 问题 40 如何引用一个已经定义过的外部变量? 54
- 问题 41 全局变量和局部变量的存储方式有什么区别? 55
- 问题 42 整型常量的存储形式是怎样的? 55
- 问题 43 整型常量的表示形式有几种? 56
- 问题 44 使用了没定义的变量会有什么现象? 56
- 问题 45 static 关键字有什么作用? 57
- 问题 46 const 关键字有什么作用? 59
- 问题 47 const 与#define 相比有何优点? 60
- 问题 48 sizeof 不是函数吗? 61
- 问题 49 float 类型数如何与 0 值比较? 63
- 问题 50 静态变量与自动变量的区别有哪些? 64

第 5 章 运算符与表达式 66

- 问题 51 运算符的优先级和结合性是怎样的? 67
- 问题 52 如何区分“,”是运算符还是分隔符? 68
- 问题 53 C 语言如何解释 $x=a+=b+c$? 69
- 问题 54 $x=x+1$, $x+=1$, $x++$, 哪个效率最高? 70
- 问题 55 什么是运算符的目? 怎样进行区分? 70
- 问题 56 使用“++”和“-”运算符需要注意些什么? 71
- 问题 57 如何理解 $i++j$? 71
- 问题 58 赋值表达式中什么是左值和右值? 数组名作为左右值时又具有怎样的意义? 72
- 问题 59 如何确定条件表达式的结果的

数据类型? 73

- 问题 60 “%”运算符是否可以对小数的进行运算? 74
- 问题 61 “/”运算符得到的结果一定为整数吗? 75
- 问题 62 在进行多种数据类型混合运算的时候, 数据类型自动转换有哪些规则? 76
- 问题 63 C 语言中有哪些简化的运算表达式? 77
- 问题 64 使用逻辑表达式需要注意哪几点问题? 77
- 问题 65 $i++*i++$ 这样的表达式为什么不能得到预期的结果? 78
- 问题 66 $a[i]=i++$; 这样的代码正确吗? ... 79
- 问题 67 编写表达式时需要注意什么? 79
- 问题 68 如何理解 $c=a,b$? 80
- 问题 69 为无符号类型变量赋值时, 数据类型应怎样转换? 81
- 问题 70 C 语言表达式的求值顺序总是按照运算符的结合性保证“自左至右”或者“自右至左”吗? 81

第 6 章 输入/输出函数 83

- 问题 71 函数 printf() 的基本格式是什么? 84
- 问题 72 如何认识 printf() 函数的格式字符? 85
- 问题 73 函数 printf() 的标志有几种? 如何使用? 91
- 问题 74 如何控制输出最小宽度? 91
- 问题 75 如何控制输出精度? 92
- 问题 76 如何控制输出长度? 93
- 问题 77 如何动态设置输出宽度和精度? 94
- 问题 78 printf() 函数的返回值是什么? 95
- 问题 79 如何理解输出列表? 96



Note



问题 80	函数 scanf()的基本格式是什么?	96	问题 104	if 语句与 switch 语句的优缺点是什么?	127
问题 81	scanf()函数的格式字符是什么?	97	问题 105	switch 语句中的 default 关键字是否必须?	128
问题 82	使用 scanf()函数应注意的问题是什么?	100	问题 106	break 关键字在 switch 语句中应注意什么?	128
问题 83	scanf()函数的返回值是什么?	101	问题 107	如何正确判断 if 和 else 的匹配?	130
问题 84	如何使用 getchar()函数?	102	问题 108	switch 和 case 后的表达式值的类型是否可以浮点型?	132
问题 85	getch()函数如何使用?	104	问题 109	区段划分型条件有什么技巧?	133
问题 86	如何应用 gets()函数?	104	问题 110	default 必须写在所有 case 之后吗?	134
问题 87	如何应用 putchar()函数?	105	第 8 章 循环结构		136
问题 88	puts()函数该如何应用?	106	问题 111	循环结构的基本概念是什么?	137
问题 89	如何控制多数值的输入?	107	问题 112	while 语句的基本格式是什么?	138
问题 90	如何输入字符数组?	108	问题 113	while 循环应注意什么问题?	139
第 7 章 选择、分支结构程序设计		110	问题 114	for 循环语句的基本格式是什么?	140
问题 91	5>4>3 为什么不成立——谈谈关系表达式的值	111	问题 115	for 语句的三个表达式都是必须的吗?	142
问题 92	=和==如何区分?	112	问题 116	do...while 语句的基本格式是什么?	144
问题 93	什么叫逻辑短路?	113	问题 117	分号在循环体中的作用?	146
问题 94	if 语句的基本形式有哪些? 如何应用?	114	问题 118	while 与 do...while 的区别?	148
问题 95	浮点数的相等比较是否可以用==?	116	问题 119	什么是循环嵌套?	149
问题 96	关系运算符和数学不等号有什么区别?	117	问题 120	循环嵌套的结构是怎样的?	151
问题 97	if 语句后面一定不能写分号吗?	118	问题 121	如何正确使用循环嵌套?	153
问题 98	这个程序为什么多执行了好多语句?	119	问题 122	死循环是怎样产生的?	154
问题 99	不用关系表达式和逻辑表达式做条件	120	问题 123	怎样提高循环语句的效率?	156
问题 100	怎样理解复合语句中的变量?	122	问题 124	continue 语句的基本作用是什么?	158
问题 101	如何进行 if 语句的嵌套?	123			
问题 102	条件运算符“?:”怎样应用?	124			
问题 103	switch 语句的基本格式是什么?	125			





- 问题 125 break 语句的基本作用是什么? 159
- 问题 126 goto 语句的基本格式是什么? 如何使用? 161
- 问题 127 goto 语句的缺陷是什么? 163
- 问题 128 如何选择循环语句? 165
- 问题 129 如何判定循环结束和提前结束? 165
- 问题 130 如何避免循环中的初值错误问题? 167

第 9 章 数组 170

- 问题 131 什么是数组? 其存储有何特点? 171
- 问题 132 数组的维数该如何理解? 171
- 问题 133 一维数组是怎样定义的? 172
- 问题 134 如何引用一维数组元素? 173
- 问题 135 如何初始化一维数组? 174
- 问题 136 如何设计数组的排序算法? 175
- 问题 137 如何定义二维数组? 176
- 问题 138 如何引用二维数组元素? 178
- 问题 139 如何初始化二维数组? 179
- 问题 140 如何定义字符数组? 180
- 问题 141 如何初始化字符数组? 180
- 问题 142 如何引用字符数组? 181
- 问题 143 如何进行字符数组的复制? 182
- 问题 144 如何进行字符数组的连接? 183
- 问题 145 如何进行字符串的比较? 185
- 问题 146 如何测定字符串的长度? 187
- 问题 147 如何进行字符串大小写的相互转换? 188
- 问题 148 如何计算字符串中有多少个单词? 190
- 问题 149 gets()函数和 scanf()函数在输入字符串时有何区别? 191
- 问题 150 puts()函数和 printf()函数在输出字符串时有何区别? 192

- 问题 151 数组与指针的区别是什么? .. 193
- 问题 152 为什么作为函数形参的数组和指针可以互换? 195
- 问题 153 为什么数组名作参数传递给子函数时, 子函数可以改变主函数中数组的值? 196
- 问题 154 C 语言中有动态数组吗? 197
- 问题 155 如何实现动态二维数组? 199
- 问题 156 strcpy()函数可以复制字符串的一部分吗? 201
- 问题 157 字符串和字符数组有什么区别? 202
- 问题 158 '\0'和'\0'有什么区别? 203
- 问题 159 字符数组占用内存怎样算? 204
- 问题 160 用字符数组和指针两种方式定义的字符串有什么不同? 204

第 10 章 函数编程基础 206

- 问题 161 什么是函数? 如何分类? 207
- 问题 162 如何定义无参函数? 209
- 问题 163 如何定义有参函数? 210
- 问题 164 什么是空函数? 作用是什么? 210
- 问题 165 什么是形参和实参? 如何使用? 211
- 问题 166 如何从函数返回? 213
- 问题 167 函数返回值你了解多少? 214
- 问题 168 如何进行函数的一般调用? 215
- 问题 169 函数调用的基本方式有几种? 各是什么? 216
- 问题 170 函数调用应具备哪些条件? 216
- 问题 171 如何进行函数的嵌套调用? 218
- 问题 172 什么是递归调用? 如何实现? 219
- 问题 173 函数如何将数组元素作为



Note



实参?	222	问题 199	如何用指针表示多维数组?	260
问题 174	如何将数组名作为函数参数?	问题 200	如何使用指针操作多维数组?	261
问题 175	如何将多维数组名作为函数参数?	问题 201	如何用指针为函数传递数组?	263
问题 176	什么是局部变量?	问题 202	如何用指针表示字符串?	264
问题 177	什么是全局变量? 如何应用?	问题 203	如何使用字符串指针作为函数参数?	265
问题 178	存储方式有哪几种? 分别是什么?	问题 204	字符数组和字符指针的区别是什么?	266
问题 179	如何使用 auto 关键字?	问题 205	什么是指针数组?	267
问题 180	什么是静态变量? 如何实现?	问题 206	如何使用指针数组处理字符串?	268
问题 181	什么是寄存器变量? 如何实现?	问题 207	如何将指针数组作为函数的参数?	269
问题 182	如何声明外部变量?	问题 208	什么是指向指针的指针?	270
问题 183	如何调用编译后的函数?	问题 209	二级指针如何应用于二维数组?	271
问题 184	如何限定外部变量的使用范围?	问题 210	如何实现二级指针对二维数组的操作?	272
问题 185	如何使用函数调用实现对字符串的统计?	问题 211	二级指针如何操作字符串数组(指针数组)?	273
问题 186	main()函数有什么作用?	问题 212	如何理解返回指针的函数?	275
问题 187	什么是内部函数?	问题 213	什么是指向函数的指针?	277
问题 188	什么是外部函数? 怎么用?	问题 214	如何用 const 控制指针?	278
问题 189	static()函数与普通函数有什么区别?	问题 215	什么是“野指针”?	279
问题 190	形参和实参有什么区别?	问题 216	main()函数的指针数组形参是怎么回事?	279
第 11 章 指针解析	248	问题 217	void 指针就是空指针吗? 它有什么作用?	281
问题 191	什么是指针? 什么是指针变量?	问题 218	指针是一种特殊的变量, 只能用来保存地址。这句话对吗?	282
问题 192	如何创建指针?	问题 219	字符指针、浮点数指针以及函数指针这三种类型的变量哪个占用的内存最大? 为什么?	282
问题 193	如何初始化指针?			
问题 194	如何使用指针?			
问题 195	函数中如何传递指针?			
问题 196	指针、数组和地址之间的关系是什么?			
问题 197	如何进行指针运算?			
问题 198	如何使用指针操作数组?			





问题 220 一个 32 位的机器, 该机器的
指针是多少位? 283

第 12 章 常用数据结构 284

问题 221 空结构体所占的内存是
多少? 285

问题 222 在 C 语言中, 一个结构体可以
包含指向自己的指针吗? 286

问题 223 struct person {...}; person a;
为什么编译出错? 287

问题 224 怎样从/向数据文件读/写
结构? 289

问题 225 枚举与 #define 宏的区别有
哪些? 290

问题 226 如何看待枚举类型, 枚举类型
的优点是什么? 291

问题 227 关键字 typedef 的功能是
什么? 292

问题 228 类型定义是否允许嵌套? 294

问题 229 typedef 与 #define 宏的相似之处
与不同之处是什么? 295

问题 230 什么是散列法? 296

问题 231 大小端模式对 union 类型数据
有什么影响? 297

问题 232 如何为联合体变量赋
初值? 298

问题 233 如何证明联合体变量的所有成员
是共享一个内存单元的? 300

问题 234 堆和栈的区别是什么? 301

问题 235 举例说明, 什么是静态链表?
什么是动态链表? 302

问题 236 单向链表、双向链表和循环链
表有什么区别? 304

问题 237 如何在链表中的指定位置插入
结点? 305

问题 238 如何删除链表中指定位置的
结点? 306

问题 239 如何创建一个动态链表? 308

问题 240 指向结构体数组的指针如何
应用? 310

第 13 章 位运算操作符 312

问题 241 什么是位运算? 位运算符包括
哪些? 313

问题 242 移位运算中如何补位? 314

问题 243 移位运算符与加减运算符的
优先级哪个较高? 315

问题 244 什么是循环移位? 316

问题 245 什么是位段? 其优点是
什么? 317

问题 246 如何正确使用位段? 318

问题 247 数据在计算机中的存储单位有
哪些? 有几种存储形式? 320

第 14 章 存储管理 322

问题 248 与内存息息相关的重要概念
有哪些? 323

问题 249 指针指向不合法引起了哪些
内存问题? 324

问题 250 内存分配与释放引起的常见
问题有哪些? 325

问题 251 什么是内存越界? 什么是内存
泄露? 二者是如何产生的? ... 326

问题 252 C 语言提供了哪些动态内存
分配函数? 327

问题 253 malloc() 函数与 calloc() 函数有
什么区别? 328

问题 254 内存耗尽怎么办? 328

问题 255 动态内存会被自动释
放吗? 330

问题 256 高位优先与低位优先的不同
之处是什么? 330

问题 257 free() 和 delete() 怎样处理
指针? 331

问题 258 怎样利用好敏感的内存
资源? 333

第 15 章 预处理和函数类型 335

问题 259 在头文件中 #if、_STDC_ 等字符
起什么作用? 336

问题 260 如何书写多条语句宏? 337



Note



问题 261	预处理中#和##运算符是什么意思?	338	问题 282	scanf()函数的格式控制包括哪些?	364
问题 262	一个头文件可以包含另一个头文件吗?	339	问题 283	printf()函数和 scanf()函数格式符的修饰符“*”有什么作用?	366
问题 263	#include<>和#include“”有什么区别?	340	问题 284	fscanf()函数、fprintf()函数与 scanf()函数和 printf()函数有什么不同?	367
问题 264	什么是无参宏定义?	341	问题 285	如何判断文件的结束?	368
问题 265	什么是带参宏定义?	342	第 17 章 图形图像处理	371	
问题 266	怎样写参数个数可变的宏?	343	问题 286	为什么在使用图形函数时要首先初始化图形模式?	372
问题 267	#pragma 预处理的作用是什么?	345	问题 287	怎样初始化图形模式?	372
问题 268	条件编译的表达形式有哪些?	346	问题 288	初始化时提示“BGI Error: Graphics not initialized (use 'initgraph')”怎么办?	374
问题 269	如何应用内部函数?	347	问题 289	怎样利用 C 语言建立独立的图形运行程序?	375
问题 270	如何应用外部函数?	348	问题 290	TC 中有几个画线函数? 怎么使用?	376
第 16 章 文件的读写操作	351		问题 291	TC 中有几个画矩形函数? 怎么使用?	377
问题 271	各个读写操作的区别是什么?	352	问题 292	TC 中有几个画圆函数? 怎么使用?	379
问题 272	C 语言文件有哪几类?	354	问题 293	如何使用 C 语言填充封闭图形?	380
问题 273	怎样写数据文件,使之可以在不同字大小、字节顺序或浮点格式的机器上读入?	355	问题 294	TC 中有几个和光标有关的函数? 怎样使用?	382
问题 274	能否使用 fflush()函数清除多余的输入?	356	问题 295	如何在图形模式下输出文本?	383
问题 275	fopen()函数打开文件失败的原因是什么?	357	问题 296	背景色、线条颜色和填充颜色有什么区别? 何时使用?	386
问题 276	为什么打开文件后要及时关闭?	358	问题 297	怎样记住那么多的颜色?	387
问题 277	文件的打开方式有哪些?	358	问题 298	线条样式和填充样式都有哪些? 怎样设置?	388
问题 278	如何正确使用 putchar()函数和 getchar()函数?	360	问题 299	怎样复制图形?	392
问题 279	getchar()函数、getch()函数和 getche()函数的区别是什么?	361	问题 300	怎样在 C 语言中制作动画?	394
问题 280	使用 printf()函数和 scanf()函数需要注意什么?	362			
问题 281	printf()函数有哪些参数?	363			



第 1 章

初识 C 语言

- »» C 语言是如何发展起来的?
- »» 什么是 ANSI 标准?
- »» C 语言编写程序的优点有哪些?
- »» 如何规避 C 语言的不足之处?
- »» C 语言的应用领域有哪些?
- »» 什么是 C99 标准? 与 C89 标准相比, C99 标准有哪些新特性?
- »» C 语言是 C++ 的子集吗?
- »» C 语言的开发过程是怎样的?
- »» 什么是编译程序和解释程序?
- »» ANSI C 的编译限制有哪些?



问题 1 C 语言是如何发展起来的?

问题阐述

C 语言产生、发展至今都经历了哪些阶段? 是如何发展成熟的?

专家解答

在学习 C 语言之前, 要先了解一下 C 语言的发展。

最初的操作系统等软件主要是使用汇编语言进行编写, 由于汇编语言有依赖于硬件、程序可读性和移植性都比较差等特点, 而高级语言又难以实现直接对硬件等进行操作, 故需发展一种新的语言, 使其既具有高级语言的特性, 又具有低级语言的特性。于是, C 语言应运而生。

1970 年, UNIX 的开山鼻祖——美国贝尔实验室的 Ken Thompson 设计出了既简单又很接近硬件的 B 语言(取 BCPL 的第一个字母), 并用 B 语言编写了第一个 UNIX 操作系统。

1972 年, Dennis Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母)。C 语言既保持了 BCPL 和 B 语言的优点(精练、接近硬件), 又克服了它们的缺点(过于简单、数据无类型等)。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。

1975 年, UNIX 第 6 版公布后, C 语言的突出优点引起了人们的普通关注。1977 年出现了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》, 简化了 C 移植到其他机器时所做的工作。

1978 年, Brian W.Kernighan 和 Dennis M.Ritchie 共同出版了影响深远的《The C Programming Language》一书。

最初, C 语言运行于 AT&T 的多用户、多任务的 UNIX 操作系统上。后来, Ritchie 用 C 语言改写了 UNIX C 的编译程序, UNIX 操作系统的开发者 Ken Thompson 又用 C 语言成功地改写了 UNIX, 从此开创了编程史上的新篇章。由此, UNIX 成为第一个不是用汇编语言编写的主流操作系统。

专家点评

C 语言在当前仍是比较流行的计算机高级语言, 被广泛应用于各领域。C 语言也是计算机编程的基础语言学科, 所以大家应该学好 C 语言, 为编程打下坚实的基础。

问题 2 什么是 ANSI 标准?

问题阐述

在使用 C 语言进行编程时, 最常被提及的就是 ANSI C 标准, 那么什么是 ANSI C 标





准？它是怎样被建立起来的？

专家解答

1983 年，美国国家标准协会（ANSI）委任一个委员会 X3J11 对 C 语言进行标准化并于当年颁布了第一个 C 语言草案（83ANSI C），后又于 1987 年颁布了另一个 C 语言标准草案（87ANSI C）。经过漫长而艰苦的工作，该委员会的研究成果于 1989 年 12 月 14 日被正式批准为 ANSI X 3.159—1989 并于 1990 年春天颁布。ANSI C 主要标准化了现存的实践，同时增加了一些来自 C++ 的内容（主要是函数原型）并支持多国字符集（包括备受争议的三字符序列）。此外，ANSI C 标准还规定了 C 运行期库例程的标准。

一年后，该标准被接受为国际标准——ISO/IEC 9899:1990 并被广泛采用。在美国国内，该标准（在这里它被称做 ANSI/ISO 9899—1990 [1992]）也取代了早先的 X3.159。作为一个 ISO 标准，它会以发行技术勘误和标准附录的形式不断更新。

1994 年，技术勘误 1（TC1）对标准中 40 个地方作了修正，多数都是小的修改或明确；而标准附录 1（NA1）增加了大约 50 页的新材料，多数是规定国际化支持的新库函数。1995 年，TC2 增加了更多的小修改。

1999 年，ANSI 标准的一个重大修订——C99 颁布，并于 2000 年 3 月被 ANSI 采用。

ANSI 标准的数个版本，包括 C99 和原始的 ANSI 标准，都涉及一个“基本原理”（Rational），解释它的许多决定并讨论了很多细节问题，包括本文中提及的某些内容。不过由于未得到主流编译器厂家的支持，C99 并未被广泛采用。

专家点评

目前流行的 C 编译器一般都是以 ANSI 标准为基础的，各种版本的 C 语言编译器系统虽然基本部分是相同的，但是也有一些不同，读者在使用的时候要注意识别。

问题 3 C 语言编写程序的优点有哪些？

问题阐述

C 语言能够存在并发展至今，其生命力之强可见一斑。这其中一定是有着某些不可替代的优点，那么 C 语言编写程序的优点都有哪些呢？

专家解答

为了方便读者理解，下面对 C 语言的每条特点进行详细的解说。

- （1）程序结构简洁、紧凑、规整，表达式简练、使用灵活。
- （2）编写的程序可读性强，编译效率高。
- （3）具有丰富的运算符，多达 34 种。丰富的数据类型与丰富的运算符相结合，使 C 语言具有表达灵活和效率高等特点。



Note



(4) 数据类型种类繁多。C 语言具有 5 种基本的数据类型和多种构造数据类型以及复合的导出类型，同时还提供了与地址密切相关的指针机器运算符。指针可以指向各种类型的简单变量、数组、结构和联合，乃至函数等。此外，C 语言还允许用户自己定义数据类型。

(5) 是一种结构化程序设计语言，特别适合大型程序的模块化设计。C 语言具有编写结构化程序所必需的基本流程控制语句，C 语言程序是由函数集合构成的，函数各自独立，并且作为模块化设计的基本单位。

说明：

C 语言的源文件，可以分割成多个源程序，分别进行编译，然后连接起来构成可知性的目标文件，为开发大型软件提供了极大的方便。C 语言还提供了多种存储属性，使数据可以按其需要在相应的作用域起作用，从而提高了程序的可靠性。

(6) 语法限制不太严格，程序设计自由度大。例如，对数组下标越界不作检查，由程序编写者自己保证程序的正确。一般的高级语言语法检查比较严，能检测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，因此放宽了语法的检查。程序员应当仔细检查程序，保证其正确，而不要过分依赖 C 语言编译程序去查错。

(7) 允许直接访问物理地址，能进行位 (bit) 操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此，C 语言既具有高级语言的功能，又兼容低级语言的许多功能，可用来编写系统软件。

(8) 生成的目标代码质量高，程序执行效率高。它一般只比汇编程序生成的目标代码率低 10%~20%。

(9) 具有较高的可移植性。它的语句基本上无须修改就能用于各种型号的计算机和各种操作系统。

C 语言是处于汇编语言和高级语言之间的一种中间型程序设计语言，常被称为中级语言。它既有高级语言的基本特点，又具有汇编语言面向硬件和系统，可以直接访问硬件的功能。

专家点评

C 语言的这些优点，读者仅通过这里的介绍还不能深刻理解和体会，待对 C 语言有了一定的了解之后再回顾一下，就会体会到这些优点了。但由于 C 语言的限制少、灵活性大、功能强，所以对程序员有较高的要求。在使用 C 语言进行编程时，需要有足够的细心和耐心。

问题 4 如何规避 C 语言的不足之处？

问题阐述

C 语言虽然具有功能强、灵活性大、可移植性强、应用广泛等优点，但是同时也存在

