

TURING 图灵程序设计丛书

日本C语言入门第一书

原版畅销20万册



明解 C语言

〔日〕柴田望洋 著
管杰 罗勇 译

 人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

013034532

TP312C

2170

明解 语言



[日] 柴田望洋 著
管杰 罗勇 译



北航

C1642223

TP312C
2170

人民邮电出版社

北京

图书在版编目 (CIP) 数据

明解 C 语言 / (日) 柴田望洋著 ; 管杰, 罗勇译.
—北京 : 人民邮电出版社, 2013.5
(图灵程序设计丛书)
ISBN 978-7-115-29979-6

I . ①明… II . ①柴… ②管… ③罗… III . ①
C 语言—程序设计—教材 IV . ① TP312

中国版本图书馆 CIP 数据核字 (2012) 第 267752 号

内 容 提 要

本书是日本的 C 语言经典教材, 自出版以来不断重印、修订, 被誉为“C 语言圣经”。作者在日本 IT 界家喻户晓, 出版过一系列极富影响力的计算机教材和参考书。其简洁、通俗的文风深受读者的喜爱。

本书图文并茂, 示例丰富, 设有 190 段代码和 164 幅图表, 对 C 语言的基础知识进行了彻底剖析, 内容涉及数组、函数、指针、文件操作等。对于 C 语言语法以及一些难以理解的概念, 均以精心绘制的示意图, 清晰、通俗地进行讲解。

本书适合 C 语言初学者阅读。

图灵程序设计丛书

明解 C 语言

-
- ◆ 著 [日] 柴田望洋
 - 译 管 杰 罗 勇
 - 责任编辑 傅志红
 - 执行编辑 乐 馨 张 靖

 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京天宇星印刷厂印刷

 - ◆ 开本: 800×1000 1/16
印张: 21.75
字数: 514 千字 2013 年 5 月第 1 版
印数: 1-5000 册 2013 年 5 月北京第 1 次印刷
著作权合同登记号 图字: 01-2012-3527 号
ISBN 978-7-115-29979-6
-

定价: 59.00 元

读者服务热线: (010)51095186 转 604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

目录

Content

第1章 初识C语言

1-1 显示计算结果

计算整数的和并显示结果	2
程序和编译	2
注释	3
固定代码	4
格式化输出函数printf	4
语句	5
计算并显示整数的差	5
格式化字符串和转换说明	6
符号的称呼	7
无格式化输出	8
字符串常量	9

转义字符	9
------------	---

1-2 变量

常量和变量	10
声明多个变量	11
赋值	11

1-3 输入和显示

通过键盘进行输入	12
格式化输入函数scanf	12
乘法运算	13
输出函数puts	14

2-1 运算

四则运算	18
运算符和操作数	18
商和余数	19
乘除运算符和加减运算符	19
使用printf函数打印%	19
获取整数的最后一位数字	20
多个转换说明	21
单目运算符	22
赋值运算符	23
表达式和赋值表达式	23

第2章 运算和数据类型

表达式语句	23
-------------	----

2-2 数据类型

求平均值	24
数据类型	24
int类型和double类型	25
数据类型和对象	26
整数常量和浮点数常量	27
double类型的运算	27
数据类型和运算	28
类型转换	30
转换说明	32

第3章 分支结构程序

3-1 if 语句

程序的流程	36
if 语句 (1)	36
奇数的判定	37
if 语句 (2)	38
奇数和偶数	39
判断	39
非0的判断	40
语法结构	40
相等运算符	42
比较余数	43
关系运算符	44

嵌套的if语句	45
计算较大值	46
计算三个数的最大值	47
条件运算符	48
差值计算	49
复合语句 (程序块)	50
判断季节	52
逻辑运算符	52

3-2 switch语句

程序的流程	54
switch语句和if语句	57
选择语句	57

4-1 do语句

do语句	60
复合语句 (程序块) 中的声明	61
逻辑非运算符	61
逆向显示整数值	62
计算整数的位数	63
初始化	64
复合赋值运算符	66
后置递增运算符和 后置递减运算符	67

4-2 while语句

while语句	68
字符常量	69
putchar	69

第4章 程序的循环控制

用递减运算符简化程序代码	70
数据递增	71
限定次数的循环操作	72
前置递增运算符和 前置递减运算符	73

4-3 for语句

for语句	74
循环语句	75
固定次数的循环	76

4-4 多重循环

九九乘法表	78
多重循环	79
长方形	80

直角三角形	80
4-5 程序的组成元素和格式	
关键字	82
标识符	82
分隔符	83

运算符	83
常量和字符串常量	83
自由的书写格式	84
连接相邻的字符串常量	85
缩进	85

第5章 数组

5-1 数组

数组	88
数组和for语句	90
数组初始化	92
数组的复制	93
输入数组元素的值	94
对数组进行倒序排列	94
使用数组进行成绩处理	96
对象式宏	96
赋值表达式	98
及格学生一览表	100
数组的元素个数	100
成绩分布图	101

5-2 多维数组

矩阵	102
----------	-----

5-3 质数计算

质数	104
质数计算程序（第1版）	104
大整数	104
break语句	104
质数计算程序（第2版）	106
质数计算程序（第3版）	107
质数计算程序（第4版）	108
质数计算程序（第5版）	110
逗号运算符	110

第6章 函数

6-1 什么是函数

main函数	114
库函数	114
函数定义和函数调用	114

三个数中的最大值	118
平方差	119
幂	120
值传递	120

调用其他函数	121	函数的通用性	129
6-2 函数设计		数组的传递	130
没有返回值的函数	122	对接收到的数组进行写入处理	132
通用性	122	const类型的修饰符	133
不含形参的函数	124	顺序查找	134
函数返回值的初始化	125	哨兵查找法	135
作用域	125	表达式语句和空语句	137
计算最高分的程序	126	多维数组的传递	138
文件作用域	126	6-3 作用域和存储期	
声明和定义	127	作用域和标识符的可见性	140
函数原型声明	127	存储期	142
头文件和文件包含指令	128		

7-1 基本数据类型和数

基本数据类型	148
基数	149
基数转换	150

7-2 整型和字符型

字符型和整型	152
<limits.h>头文件	154
字符型	155
sizeof运算符	156
size_t型和typedef声明	157
整型的灵活运用	157
整型常量	158
整型常量的数据类型	158
内部表示和位	160
无符号整数的内部表示	160
有符号整数的内部表示	161
反码表示法和补码表示法	162
窥探整数内部	163
按位操作的逻辑运算	164

第7章 基本数据类型

位移运算符	166
位数的计算	168
求出unsigned型的位数	168
显示位的内容	169
整数的显示	170
数据溢出和异常	171

7-3 浮点型

浮点型	172
浮点型常量	173
循环的控制	174
<math.h>头文件	175

7-4 运算

运算符一览	176
优先级	176
结合性	176
数据类型转换	178
sizeof运算符	180
sizeof运算符和数组	181

第8章 动手编写各种程序吧

8-1 函数式宏

函数和数据类型	184
函数式宏	185
函数和函数式宏	186
不带参数的函数式宏	187
函数式宏和逗号运算符	188

8-2 枚举类型

枚举类型	190
枚举常量	192
命名空间	193

8-3 递归

阶乘	194
----------	-----

最大公约数	196
-------------	-----

问题和递归	197
-------------	-----

8-4 输入输出和字符

数字字符计数	198
getchar函数	198
EOF	198
字符和数值	199
字符	200
转义字符	203
复制	204

9-1 什么是字符串

字符串字面量	208
字符串字面量的长度	208
字符串	210
字符数组的初始化赋值	211
空字符串	212
字符串的读取	212
格式化显示字符串	213

9-2 字符串数组

字符串数组	214
-------------	-----

第9章 字符串的基本知识

读取字符串数组中的字符串	215
--------------------	-----

9-3 字符串处理

字符串长度	216
遍历字符串	218
数字字符的出现次数	219
字符串数组的参数传递	220
大小写字符转换	222

第 10 章 指针

10-1 指针

函数的参数	226
变量和对象	227
地址	227
取址运算符	228
指针	229
指针运算符	231

10-2 指针和函数

作为函数参数的指针	232
二值互换	234

引用传递 (C++语言)	235
计算和与差	236
scanf函数和指针	236
将两个值升序排列	237
指针的类型	238
标量型	239

10-3 指针和数组

指针和数组	240
数组的传递	244

11-1 字符串和指针

字符串和指针	248
数组和指针的相同点	249
数组和指针的不同点	250
字符串数组	252

11-2 通过指针操作字符串

字符串和指针	254
判断字符串长度	254

第 11 章 字符串和指针

const	254
使用指针进行遍历	255
字符串的复制	256
不正确的字符串复制	258
返回指针的函数	259

11-3 字符串处理库函数

字符串处理函数	260
字符串转换函数	264

第 12 章 结构体

12-1 结构体

排序	268
冒泡排序法	269
数据关联性	270
结构体	272
结构体成员 (.运算符)	274
成员的初始化	275
结构体成员 (->运算符)	276
结构体和typedef	278
结构体和程序	279

聚合类型	280
命名空间	280
返回结构体的函数	281
结构体数组	282
派生类型	282
表示日期和时间的结构体	284

12-2 作为成员的结构体

表示坐标的结构体	286
表示具有定位功能的汽车的结构体	286

13-1 文件与流

文件	290
流	290
标准流	291
FILE型	291
打开文件	292
关闭文件	294
打开与关闭文件示例	295
文件数据汇总	296
写入日期和时间	298

第 13 章 文件处理

获取上一次运行时的信息	300
标准输入输出	302
显示文件内容	302
文件的复制	304

13-2 文本和二进制

在文本文件中保存实数	306
文本文件和二进制文件	307
在二进制文件中保存实数	308
显示文件自身	310

附录 1 C语言简介

C语言的历史.....	314
K&R——C语言的圣经.....	314

C语言标准规范.....	314
--------------	-----

附录 2 printf函数与scanf函数

printf函数.....	318
---------------	-----

scanf函数.....	322
--------------	-----

致谢.....	326
---------	-----

参考文献.....	326
-----------	-----

索引.....	327
---------	-----

版权声明.....	338
-----------	-----

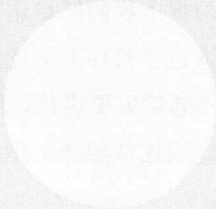


第 1 章

初识C语言

如果说熟悉了一件事就能大幅进步的话，那么长期从事这件事并已完全熟练的人就应该是高手了。但现实并非如此，就拿体育训练来说，假如训练的方式是错误的，只会越练习越差。编程也是如此，仅仅熟练是不够的。

总之，任何事情开始的时候，都需要先试试水。本章就带领大家尝试一下简单的C语言编程。



1-1 显示计算结果

计算整数的和并显示结果

电脑也称为电子计算机，对它来说，任何任务都是通过计算来完成的。那么就让我们使用C语言来进行下面的计算吧。

计算整数 15 和 37 的和，并显示结果。

在编辑器中键入代码清单 1-1 所示的程序代码。C 语言程序是区分大小写和全半角字符的，请大家在书写的时候特别注意。

代码清单 1-1

```
/*  
    显示整数 15 和 37 的和  
*/  
  
#include <stdio.h>  
  
int main(void)  
{  
    printf("%d", 15 + 37);    /* 用十进制数显示整数 15 和 37 的和 */  
  
    return (0);  
}
```

运行结果

52

► 为了方便读者阅读，本书中的代码使用了蓝色、斜体、`consolas`体和 Courier New 体来进行区分。大家在书写程序代码的时候可以忽略这些字体设置。

程序和编译

如代码清单 1-1 所示，人们通过字符序列创建出的程序被称为源程序（source program），用来保存源程序的文件被称为源文件（source file）。

习惯上我们把 C 语言源文件的扩展名约定为“.c”，例如我们可以把源文件保存为 list0101.c。

通过字符序列创建出的程序，需要转换为计算机能够理解的位序列，也就是 0 和 1 的序列。源程序通常需要进行如图 1-1 所示的翻译操作之后才能执行（关于位的介绍请参考第 7 章）。

完成这些翻译工作之后运行程序，屏幕上就能显示出结果 52 了。

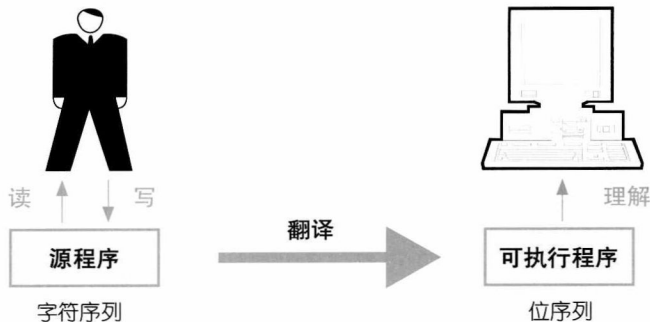


图 1-1 源程序和可执行程序

► 编译器和运行环境不同时，翻译的步骤和程序的执行方法也不同，请大家参考各自编译器的说明书。在后面的专题 1-1 中会对翻译和编译器等术语进行说明。

源程序中如果有拼写错误，翻译的时候就会发生错误，并显示出相应的**诊断消息**。出现这种情况时请仔细检查键入的程序代码，纠正错误之后再行编译。

程序中有着大量 # 和 { 等符号，大家可能不理解它们的意思。不过没关系，我们慢慢来，一点一点学习。

► 稍后我们还会对符号的称呼进行总结。

注释

源程序中 /* 和 */ 之间的部分，称为**注释** (comment)^①。有没有注释以及注释的内容如何，其实对程序的运行并没有什么影响。编程者用简洁明了的语言将程序想要表达的意思标注在程序旁，这样能提高程序的可读性。

► 代码清单 1-1 这样简短的程序还好，如果碰到他人编写的长篇程序，理解起来就非常困难了。另外，即使是自己编写的程序也不可能永远都记得住。因此，适当加一些注释不仅能方便他人，也能方便编程者自己。

■ 注意 ■

请大家在源程序中，用简洁的语言把想要表达的意思以**注释**的形式记录下来。

从程序中可以看出，注释也可以是多行的。但是请大家注意不要把结束注释用的符号误写成 “/*”，否则后面的程序都会被解释为注释。

① C99支持单行注释，即“//……”这种形式，“//”之后直到行尾的内容为注释。（本书脚注均为译者注）

固定代码

如图1-2所示，删除程序中的注释。白底以外的部分是一段固定代码，它的含义之后会详细介绍，请大家牢记这段代码。

现阶段我们暂时先照搬这段代码，其余的部分由自己编写。

```
#include <stdio.h>

int main(void)
{
    printf("%d", 15 + 37);

    return (0);
}
```

图 1-2 程序和固定代码

格式化输出函数 printf

printf 函数可以在显示器上进行输出操作(**printf** 末尾的 f 源自 format(格式化)这个单词)。

如果想要使用某个函数的功能，就必须通过**函数调用** (function call) 来实现。调用**printf** 函数显示15和37的和的过程如图1-3所示。



图 1-3 调用 printf 函数

调用此函数即发出了“显示这些内容”的请求，然后通过括号中的**实参** (argument) 来传递想要显示的内容。另外，如本例所示，当实参超过两个时，需要用逗号隔开。

printf 函数的第一个实参 **"%d"** 指定了输出格式，它告诉程序：以十进制数的形式显示后面的实参。因此，通过调用 **printf** 函数显示出了第二个实参 **15+37** 的值，即 **15** 与 **37** 的**和** **52**。

► **"%d"** 中的 d 源自 decimal (十进制数)。十进制和二进制的介绍将在 7-1 节进行介绍，八进制数和十六进制数将在 7-2 节进行介绍。**printf** 函数的详情，请参考附录 2 中的总结。

■ 注意 ■

函数调用是申请进行处理请求，而调用函数时的一些辅助指示则通过**实参**来发出。

语句

请大家仔细观察之前的程序代码，调用 `printf` 函数的时候使用了分号，那段固定代码中也使用了分号（`return (0);`）。这里的分号就相当于中文里的句号。

正如在句子末尾加上句号才能构成完整的一句话，C 语言也需要在末尾加上分号来构成正确的语句（statement）。

■ 注意 ■

原则上语句必须以分号结尾。

计算并显示整数的差

代码清单 1-2 所示程序的功能是计算并显示 15 减去 37 的差。

代码清单 1-2

```
/*  
 计算并显示 15 减去 37 的差  
*/  
  
#include <stdio.h>  
  
int main(void)  
{  
    printf("%d", 15 - 37); /* 用十进制数显示 15 减去 37 的值 */  
  
    return (0);  
}
```

运行结果

-22

运行程序就会显示结果 -22。可以看到当计算结果为负数时，会自动加上负号。

专题 1-1 翻译阶段和编译

运行 C 语言之前，理论上要经过 8 个翻译阶段（translation phase）。另外，运行源代码还需要安装必要的软件环境，也就是编译器^①。

大多数 C 语言编译器都是通过编译方式（如本文中描述的方式）把源代码翻译成计算机能够直接理解执行的形式。但是也存在逐行解释然后执行的解释方式（执行速度比较缓慢）。

^① 即符合 C 语言规范的实现（implementation）。

格式化字符串和转换说明

程序运行的时候如果只显示和或者差的值，理解上会比较困难，接下来我们让结果显示得更加人性化一些，请看代码清单 1-3 所示程序。

这次我们把 `printf` 函数的第一个实参设置得更长更复杂一些。

代码清单 1-3

```
/*
   人性化地显示 15 与 37 的和
*/

#include <stdio.h>

int main(void)
{
    printf("15 与 37 的和是 %d。 \n", 15 + 37);    /* 显示结果后换行 */

    return (0);
}
```

运行结果

15 与 37 的和是 52。

清单中的蓝色底纹部分是 `printf` 函数的第一个实参，被称为**格式化字符串**（format string）。

格式化字符串中的 `%d` 指定了实参要以十进制数的形式显示，这就是**转换说明**（conversion specification）。格式化字符串中没有指定转换说明的字符基本上都会原样输出。

格式化字符串结尾的 `\n` 是代表**换行**（new line）的符号，`\` 和 `n` 组成了一个特殊的“换行符”。

► 在字符串结尾加上换行符的必要性请参考专题 1-2。

本程序的显示结果如图 1-4 所示。

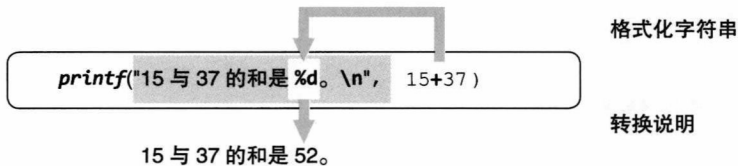


图 1-4 格式化字符串和转换说明

● 练习 1-1

编写一段程序，计算出 15 减去 37 的结果，并以“15 减去 37 的结果是 -22。”的格式进行显示。