国内首本系统论述ACM/ICPC竞赛的C语言程序设计教程！

ACM/ICPC资深教练授课经验总结，融合大量优秀竞赛作品实战技能！

清华 开发者书库

C Language Programming
Practi Contest

C语言程序设计

零基础ACM/ICPC竞赛实战指南

王建芳◎编著
Wang Jianfang

清華大學出版社

清华
开发者书库

C Language Programming
Practical Guide for ACM/ICPC Contest

C语言程序设计

零基础ACM/ICPC竞赛实战指南

王建芳◎编著
Wang Jianfang

清华大学出版社
北京

内 容 简 介

本书是专为C语言爱好者及ACM/ICPC参赛者编写的入门级教程，针对C语言学习过程中普遍存在的重理论轻实践、重语法轻编程的现象，通过贯穿全书的大量实例来介绍C语言编程的方法和技巧。全书分为三个部分：第一部分介绍C语言的基础性语法，包括标准程序框架、数据类型和控制结构；第二部分介绍了常见的OJ(Online Judge)平台、使用方法及OJ系统的基本输入与输出的常见类型；第三部分通过实例介绍了数组、函数和结构体编程过程中常用的知识点。

本书可以作为“C语言程序设计”课程的基础教材，也可作为参加ACM/ICPC竞赛的指导用书，并可作为各高校和相关培训机构的教学参考书。

图书在版编目(CIP)数据

C语言程序设计：零基础ACM/ICPC竞赛实战指南/王建芳编著. --北京：清华大学出版社，2015
(清华开发者书库)
ISBN 978-7-302-40116-2

Ⅰ. ①C…　Ⅱ. ①王…　Ⅲ. ①C语言－程序设计　Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2015)第089501号

责任编辑：盛东亮
封面设计：李召霞
责任校对：白　蕾
责任印制：宋　林

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
课件下载：http://www.tup.com.cn，010-62795954
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：186mm×240mm　　**印　　张**：13.75　　**字　　数**：308千字
版　　次：2015年5月第1版　　**印　　次**：2015年5月第1次印刷
印　　数：1～2500
定　　价：35.00元

产品编号：063522-01

前言
PREFACE

C语言功能丰富、表达能力强、使用灵活方便，20世纪90年代以来，C语言迅速在全世界普及推广。C语言具有高级语言的优点又有低级语言的特性，既适合编写操作系统软件，又能方便地开发领域应用软件。目前，C语言程序设计已经成为最为广泛的一门程序设计课程。依据基于世界范围内的资深软件工程师和第三方供应商提供作为指数的TIOBE开发语言排行榜（每月发布一次），C语言排名一直名列前茅。C语言是实践性很强的一门课程，必须不断地练习编程。在信息技术飞速发展的今天，如何将理论与实践有机结合，有效地推进素质教育和高水平人才培养，是新时期IT人才面临的新课题。

ACM国际大学生程序设计竞赛（ACM International Collegiate Programming Contest，ACM/ICPC）是由美国计算机协会（Association for Computing Machinery，ACM）主办，其目的是使大学生充分展示分析问题和运用计算机解决问题的能力。ACM/ICPC作为一项世界性的竞赛活动，自1970年开始至今，是世界范围内历史最悠久、规模最大的程序设计竞赛。正好迎合了当今社会对创新性IT人才的需求，竞赛较全面地考验学生对知识的综合运用能力、创造性地分析解决问题能力，所以在IT界具有超凡的影响力。该项赛事极大地提高了参赛同学的学习热情、实践动手能力、团队合作能力和创造创新能力。ACM/ICPC在线评判（Online Judge，OJ）系统是该项比赛的评判事务处理平台，提供了一个基于B/S结构的多用户在线系统，允许用户在线提交自己的解题代码，系统自动编译运行给出评判结果，并根据用户解题数和用时综合排出名次。

针对C语言学习过程中普遍存在的重理论轻实践、重语法讲解轻编程思想的现象，本书将“A+B”的示例程序贯穿全书，将ACM竞赛平台OJ系统用于C语言的教学讲解和自学过程中。全书分为三个部分：第一部分为C语言的基础性语法，包括标准程序框架，数据类型和控制结构；第二部分针对常见的OJ平台、使用方法及OJ系统的基本输入与输出等常见类型进行讲解；第三部分为数组、函数和结构体。

本书主要特点：①所讲解的程序框架是ACM/ICPC通用的标准框架；②采用实例讲解方法引出理论；③每个例程均已通过测试，确保能够正确编译并运行；④详细讲解如何使用OJ系统进行编程实践。

本书适用人群：①大学本科一年级没有学过C语言的学生；②已学过或正在学C语言，但对已学内容不得要领的学生；③有强烈参加ACM/ICPC竞赛愿望的学生；④大学本科四年级考取研究生，复试阶段需要上机复试C语言的学生；⑤有强烈提高自己编程能力

欲望，但苦于找不到合适训练方法和习题的读者。

本书程序运行的操作系统为 Windows 7，计算机硬件配置为 Intel(R) Core(TM) i5 CPU M480 @2.67GHz，系统类型为 64 位操作系统。

本书作者是长期从事 ACM/ICPC 竞赛指导和 C 语言教学的一线教师，同时也一直致力于 C 语言教学改革，近三年来所带学生在计算机软件类学科竞赛中获得省级以上奖励三百余人次。

在本书的编写过程中，部分题目参考或改编自杭州电子科技大学和北京大学的在线评判(OJ)系统，在此表示感谢！

由于作者能力和水平所限，加之时间仓促，书中不足之处恳请广大专家、读者批评指正！在 C 语言程序设计竞赛相关书籍中，希望本书抛出的“砖”能够引出更多的“玉”！作者邮箱 hpuacm@126.com。

编著者

2015 年 2 月

目录

CONTENTS

第1章 死记硬背

作为全书的开篇，本章讲述以下内容，为即将开始的C语言程序设计做必要的准备工作。

(1) 为什么要学习C语言。

(2) 程序设计与盖房子的类比。

(3) 完成并记住第一个完整的程序框架。

1.1 引子

为什么每个程序开发者都应该学习C语言？

每个程序开发者在编程生涯中都应该学习C语言，因为它有太多难以忽视的好处。除了会给你提供更多的工作机会，C语言还会教给你更多关于计算机的相关知识。C语言提供的裨益，简单列举如下：

(1) 相比较其他的编程语言(像C++，Java)，C语言是低级语言。总体来说，低级的编程语言可以让你更好地了解计算机内部构造和原理。

(2) C语言能够让你深入系统底层。Windows、UNIX、Linux、Mac、Android、OS/2等操作系统，没有一个例外，都是用C语言编写的。如果不懂C语言，怎么可能深入到这些操作系统中去呢？更不要说去写它们的内核程序了。

(3) C语言程序比其他语言写的程序，实现相同的功能，用的代码行数更少，而它带来的运行效率却更高。

(4) 如果你学习过C语言，就能学习现在任何高级编程语言。因为所有高级语言都是以C语言为基础(像Java、C++、C#等)。所以，如果你想在程序设计方面有所建树，就必须去学它。

(5) 很多新型语言都是衍生自C语言，例如C++，Java，C#及开发基于Android的应用程序。掌握了C语言，可以说就掌握了很多门语言，经过简单的学习，就可以用新型的语言进行开发，这再一次验证了C语言是程序设计的重要基础。

(6) 任何里面有微处理器的设备都支持C语言。从微波炉到智能手机，都是由C语言技术来推动的。

(7) 即使当今比较知名的公司招聘程序员，在部分考试中都要考查C语言的基本知识和技能。想加入IT行业，就一定要掌握好C语言，因为任何算法都可以用C语言实现。

(8) 目前，计算机科学与技术专业的大四本科生所参加的研究生入学考试，在复试的机试环节中大部分高校也是通过OJ系统来提交代码，而这些代码都能够用C语言来实现。

C语言在计算机专业中所处地位，如果以一棵树来比喻的话，C语言可以说处于树根的位置，如图1-1所示。

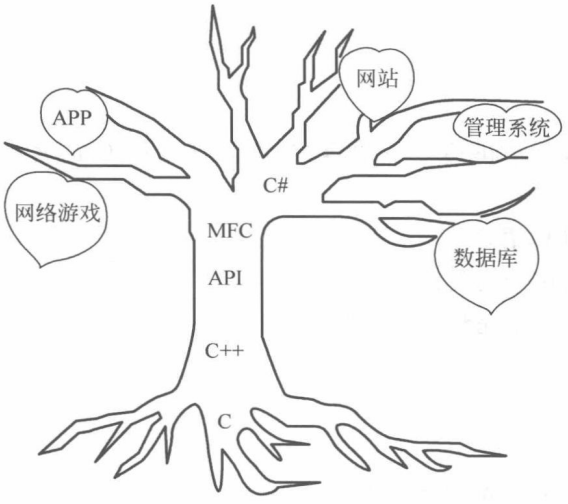

图1-1 语言之树

在正式学习C语言程序设计之前，将程序设计与盖房子做一类比，看看二者有哪些相似之处。

1. 类比：盖房子与程序设计

如果要盖一栋房子，需要做哪些事情呢？

盖房子的基本过程可分解为以下步骤：

(1) 在哪里盖？

(2) 怎么盖？

(3) 盖什么？

(4) 遇到问题怎么办？

(5) 盖房子过程中需要的辅助材料有哪些？

接着一一回答上面的问题：

(1) 在哪里盖？

就是要圈地的问题。

(2) 怎么盖？

盖房子需要一些基本的规则和步骤，比如先打地基，地基打成什么样子，然后砌墙，墙的厚度、高度为多少等。

(3) 盖什么?

盖什么功能的房子,是人居住的还是厂房?

(4) 遇到问题怎么办?

如果盖房子过程中出现了问题,是推倒重来,还是进行加固性的纠正?

(5) 盖房子过程中需要的辅助材料有哪些?

比如说需要打支子,但房子盖好后这些架子将会拆掉的。

2. 将C语言编程与盖房子类比

(1) 在哪里盖房子 VS 在哪里编写代码:开发环境(Dev C++、Visual C++ 6.0 等)。

(2) 怎么盖 VS 怎么编写程序:基本语法规则。

(3) 盖什么 VS 编写什么:编写什么功能的程序。

(4) 遇到问题怎么办:调试、纠错。

(5) 辅助:变量的定义、注释及语句的提示,等等。

3. 用任何语言编程都要考虑的问题

(1) 在哪里编写代码?

(2) 怎么编写?

(3) 编写什么?

(4) 遇到问题怎么办?

(5) 哪些地方需要辅助性的语句?

注:本书C语言开发环境统一使用 Dev-C++ 4.9.9.2。

小知识

Dev-C++是一个C&C++开发工具,使用 Delphi/Kylix 开发,是一款自由软件,遵守GPL协议。它使用 MinGW/GCC/Cygwin 编译器,遵循C/C++标准。

DEV C++已被全国青少年信息学奥林匹克联赛设为C++语言指定编译器,同时也是ACM-ICPC竞赛官方指定C语言的编译器之一。

Dev-C++官网:http://sourceforge.net/projects/dev-c++/files/Binaries/

特别提示　Dev- C++安装说明见附录A;Dev- C++使用说明见附录B。

1.2　死记硬背

1.2.1　编程基本步骤

C语言编程的基本步骤分为如下七个步骤:

(1) 打开 Dev-C++开发环境;

(2) 新建文件;

(3) 保存并命名文件;

(4) 编写代码;

(5) 编译;

(6) 调试;

(7) 运行并测试程序。

小技巧

快速建立自己的C语言程序文件,即以c为后缀的文件(**.c)。

注意:快速建立自己的.c文件,此方法适应于计算机只安装一个能够打开C语言的开发环境,即只安装一个Dev-C++软件的计算机。

---

(1) 在D盘先建立一个文件夹(例如:命名为"我的C语言代码");

(2) 打开文件夹,在空白处右击;

(3) "新建"→"文本文件";

(4) 重命名"新建文本文档.txt"更改为"自己命名的文件名.c"(一定要将原来的扩展名".txt"更改为".c");

(5) 直接双击刚才建立的.c文件,即可编写代码。

如果上述第(4)步没有显示"新建文本文档.txt"的后缀.txt,可以按以下方法操作:

以Windows 7为例:打开"我的电脑",选择"工具"菜单(如果没有"工具"菜单,按Alt键),如图1-2所示。

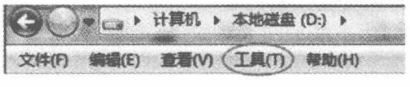

图1-2 "工具"菜单

然后在"文件夹选项"(见图1-3)中的"查看"选项卡中去掉"隐藏书籍文件类型的扩展名"的勾选,如图1-4所示。

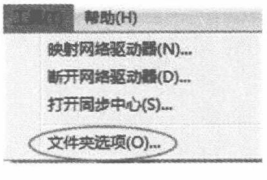

图1-3 文件夹选项

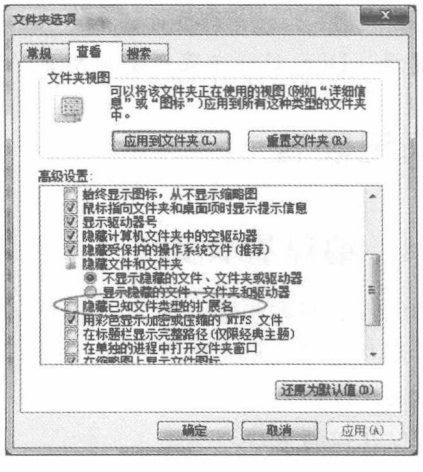

图1-4 "查看"选项卡

特别提示　新建文件，命名文件名时注意事项：不可默认，不可用数字开头，最好不要用汉字命名。

1.2.2　记死

【实例分析 1-1】　实现 A+B，详细代码如程序 1-1 所示。将程序 1-1 在编程环境 Dev C++中编译并运行，看看结果是什么？

程序 1-1　A+B 问题

```
#include<stdio.h>
int main()
{
printf("%d\n",1+1);          //输出语句
return 0;
}
```

看看程序 1-1 的运行结果如何？

下面对此程序作以简单解释：

```
#include<stdio.h>          //基本输入与输出头文件，如果程序中出现 printf()函数，必须有此语句
int main                   //主函数，任何程序必须包含主函数
{                          //括号表示函数体的范围，里面有两行命令
    printf("%d\n",1+1);//输出"2"，%d 表示输出类型，\n 表示换行
return();                  //表示返回值
}
```

注释说明

单行注释：//注释内容

多行注释：/*注释内容*/

如果编译没有问题，运行后屏幕一闪而过，则需要在程序 1-2 的代码中加上阴影部分的两行代码，具体详细代码如程序 1-2 所示。

程序 1-2　A+B 问题的改进

```
#include<stdio.h>
#include<stdlib.h>      //增加 system 的头文件
int main()
{
printf("%d\n",1+1);      //输出语句
system("pause");         //增加屏幕暂停语句
return 0;
}
```

注意：在某些C语言的编程环境中，不需要增加system("pause")；这条语句，比如Visual C++ 6.0。

常见错误举例

初次编写程序1-2常见的错误有：

(1) 括号没配对。头文件一定是尖括号< >，其他的括号均为圆括号()。

(2) 大小写没注意。C语言对大小写是敏感的。

(3) 所有的字符均是英文字符，尤其是逗号(英文逗号","与中文逗号"，"的不同)、分号(英文分号";"与中文分号"；"的不同)以及双引号(英文双引号""与中文引号"“”"的不同)。

(4) 没有空格。空格一定要有，例如int与main()之间，return与0之间。

上述程序有一个遗憾：计算的数据是事先确定的。为了计算1+2和2+3，即在运行过程中根据需要输入两个数据，我们不得不编写两个，甚至多个程序。可不可以让程序读取键盘输入，并根据输入内容计算结果呢？答案是可以。

【实例分析1-2】 A+B带有输入与输出的完整的程序代码，详细代码如程序1-3所示。

程序1-3 带有输入与输出的A+B问题

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    printf("%d\n",a+b);
    system("pause");
return 0;
}
```

常见错误举例

scanf()中的占位符和变量的数据类型没有一一对应。且每个变量前要在相应的位置加上&符号。

该程序复杂了许多。简单地说，第一条语句"int a,b"声明了两个整型(即整数类型)变量a和b，然后读取键盘输入，并放到a和b中。注意a和b前面的&符号，读者可将&去掉，看一下运行结果如何。

现在，程序已经读入了两个整数，可以在表达式中自由使用它们，就好比使用1 2，55 45这样的常数。

程序1-3为今后学习程序设计，尤其是程序1-3中阴影部分，为程序设计的基本框架。

此程序框架不一定要死记，但一定要学会灵活运用。

以后的C语言基本上以此框架为基础进行编写代码。

小技巧

从别的非编程环境中(网页或者其他文档中)粘贴代码到Dev-C++方法:

先粘贴到文本记事本上;

再粘贴到运行环境里面;

可以过滤掉一些特殊的字符;

可以说文本记事本是万能的文本过滤器。

记住:在C语言编程过程中,所有的事情,哪怕是一个变量,一定要自己创造(定义),然后才能使用。

问题是:怎么知道哪些有,哪些没有?

这就是编程语言的语法规则,也是接下来学习的重点。

1.3 初学者方法

对于初学者,尤其是自学者来说,学习计算机语言最好的方法是"模仿程序,改造例程,举一反三"。

当然,对于没有学过任何计算机语言的初学者,最好还是先阅读教程,对于书本中提到的例程,一时不理解,可以先背会,在学习后面章节的过程中慢慢理解,以培养自学能力。

学习语言,必须注意每一个细节,书上的例子代码一定要亲自敲一遍,编译、执行并跟书上的一致才能算是学完了一个例子,如果不一致,要仔细找原因。

编写运行正确代码一段时间后,要会改造书本上的例子,学会举一反三,将改造加工的例程仔细地归类保存,并且在源代码中写上简短的注释,阐述例子的意图。所谓"好记性不如烂笔头",就是这个道理。

例程之后就是练习了,建议初学者以OJ试题库(本书第六章讲解OJ系统编程的基本方法)作为练习的重点,找适合自己水平的习题,由浅入深,这是提高编程能力最重要的方法。

仔细读书、认真编写源代码、独立完成习题,外加更进一步的实验,最后将所有的代码保存,成为自己的经验和财富,绝对的辛苦能够换来事半功倍的效果。

最后,还有非常重要的一点——代码风格,从开始学习就必须强迫自己模仿最优秀的代码风格。

总之,要掌握一门语言,需要以下几点:

(1) 重视实验:哪怕不理解背后的道理,至少要清楚现象。

(2) 学会模仿:把学习的焦点集中在最有趣的地方。

(3) 如果直观的解决方案行得通,就不必追究其背后的机理。

(4) 如果对一个东西不理解,就不要对它修改;如果非改不可,则应根据自己的直觉和猜测尝试各种改法,而不必过多考虑“为什么要这样”。

(5) 当有一定的分析、解决问题的能力后,再寻找更多的资料进一步学习一些重要的概念和原理。

要想把事情做好,必须学得透彻,通过练习来提高,但没有必要操之过急。