21世纪高等学校规划教材 | 计算机应用

从实例中学C/C++程序设计

何克右 主编

清华大学出版社

21世纪高等学校规划教材

从实例中学C/C++程序设计

何克右 主编

清华大学出版社
北京

内 容 简 介

本书通过192个精心挑选的实例分析和解答，阐述了C/C++程序设计的方法和技巧。本书内容既涉及C/C++语言的使用方法，包括程序控制结构、数组和结构体、函数、指针等；典型数据结构的定义和应用，包括顺序表、链表、二叉树等；也涉及程序设计中常用算法的基本思想和应用方法，包括递推和迭代、穷举、递归、贪心法、分治法、回溯法、动态规划等。这些内容的学习和掌握，对于提高读者的程序设计能力大有裨益。

本书语言简洁、通俗易懂，注重理论与实践相结合。全书实例丰富，每个实例的思路分析清晰，逻辑性强。书中所有程序均在Visual C++ 6.0上运行通过。

本书可作为高等院校计算机专业和相关专业程序设计课程的教学参考书，也可作为数据结构和算法的课外辅导用书，还可供有兴趣参加各类程序设计竞赛的读者作为基础训练用书，同时可供各类程序设计培训班学员和C/C++语言自学者参考。

图书在版编目(CIP)数据

从实例中学C/C++程序设计/何克右主编. --北京：清华大学出版社，2014
21世纪高等学校规划教材·计算机应用
ISBN 978-7-302-35058-3

Ⅰ. ①从… Ⅱ. ①何… Ⅲ. ①C语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2014)第006770号

责任编辑：闫红梅　赵晓宁
封面设计：傅瑞学
责任校对：焦丽丽
责任印制：沈　露

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 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：28.75　　**字　　数**：697千字
版　　次：2014年3月第1版　　**印　　次**：2014年3月第1次印刷
印　　数：1～2000
定　　价：54.50元

产品编号：057350-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程（简称‘质量工程’）”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上；精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展，顺应并符合 21 世纪教学发展的规律，代表并引领课程发展的趋势和方向）、示范性（教材所体现的课程体系具有较广泛的辐射性和示范性）和一定的前瞻性。教材由个人申报或各校推荐（通过所在高校的“编委会”成员推荐），经“编委会”认真评审，最后由清华大学出版

社审定出版。

目前，针对计算机类和电子信息类相关专业成立了两个“编委会”，即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括：

(1) 21世纪高等学校规划教材·计算机应用——高等学校各类专业，特别是非计算机专业的计算机应用类教材。

(2) 21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21世纪高等学校规划教材·信息管理与信息系统。

(6) 21世纪高等学校规划教材·财经管理与应用。

(7) 21世纪高等学校规划教材·电子商务。

(8) 21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人：魏江江

E-mail：weijj@tup.tsinghua.edu.cn

前言

目前，国内高校普遍开设了“高级语言程序设计”之类的课程，绝大多数高校将C语言或C++语言作为程序设计语言课程的首选语言。“高级语言程序设计”是计算机类专业必修的重要专业基础课程，也是计算机专业入门课程，在课程体系中地位十分重要。这一课程的学习效果将直接关系到学生对“数据结构”、“操作系统”、“编译原理”等课程的学习，锻炼出的程序设计能力也将直接关系到学生的软件开发能力。

在多年的教学实践中，我们发现学生在学习程序设计过程中通常会面临一个“概念知识点都能理解，但就是编写不出程序”的问题。如何解决好这样的问题，我们也进行了探索和实践。在教学过程中，精心设计实例，给学生一个比较实际的切入点，通过老师的分析和讲解，使学生感觉能够入手，然后再通过将此实例不断修改、扩充，引导学生参与到程序的设计过程中，并通过实例问题的不断扩展和同一个问题的多种解决方法，有效开阔学生的思路，使得学生能获得用高级语言进行程序设计的技能，培养学生的独立思考能力和一定的自主创新能力。

本书就是我们多年教学实践过程的一个成果总结。全书通过192个精心挑选的实例的分析和解答，阐述了C/C++程序设计的方法和技巧。从内容上组织为7章。第1章是程序设计的概述，介绍程序设计语言和算法、程序设计的步骤和结构化程序设计方法等；第2章是程序控制结构，包括选择结构和循环结构程序设计的方法、递推和迭代、穷举法的基本思想和应用方法等；第3章是数组和结构体，包括数组的变换、方阵的构造、顺序表等内容；第4章是函数，以函数的使用和递归法的应用作为主要内容；第5章是指针，在介绍指针的概念的基础上，重点阐述了和指针应用密切相关的链表和二叉树的操作处理方法；第6章是算法，介绍了贪心法、分治法、回溯法和动态规划等算法的基本思路和应用方法；第7章从实践的角度阐述了如何通过实践提高自己的程序设计能力。

本书第1、第3和第4章由何克右编写，第5和第7章由刘传文编写，第2章由闵联营编写，第6章由谭新明编写，全书由何克右统稿。

由于作者水平有限，书中难免有不足之处，恳请读者批评指正。

编　者

2014年2月

目 录

第1章 程序设计概述

程序设计是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具，编写出这种语言下的程序。程序设计过程包括分析、设计、编码、测试、排错等不同阶段。

1.1 程序设计语言和算法

1.1.1 程序设计语言

要使计算机完成各种预定的操作，不仅应该告诉计算机做什么，而且还要告诉计算机如何做，这都是通过计算机执行一条条指令来完成的。

指令是指挥计算机完成某种操作的命令，它在计算机中是以一组二进制代码来表示的，一条指令对应计算机的一定动作。一台计算机所有指令的集合称为这台计算机的指令系统。指令系统的完善和齐全程度在一定程度上反映了这台计算机的功能与作用的强弱，是由计算机在硬件设计时所决定的。不同的 CPU 具有不同的指令系统，通过执行各种指令可以使计算机完成预定的操作。

用计算机进行数据处理时，要把处理过程的内容、步骤和运算规则用一系列指令表达出来，这一系列指令的有序集合就称为程序。程序通过输入设备送入计算机的存储器中存储起来，然后根据程序的要求一条条执行其中的指令，这样计算机的各部件就会在程序控制下自动完成指令规定的各种操作，操作完毕后，通过输出设备送出结果。这就是存储程序的基本思想，它是由美国计算机科学家冯·诺依曼提出来的。

程序是用计算机程序设计语言编写的。程序设计语言是人们为了描述计算机解决问题时的计算过程而设计的一种具有语法语义描述的记号。程序设计语言在发展的过程中经历了由低级到高级的发展过程，可以分为机器语言、汇编语言和高级语言。

C++语言是一种面向过程和面向对象都适用的混合型语言，是在 C 语言的基础上逐步发展和完善起来的，而 C 语言则是在吸收了其他语言的一些优点后逐步发展为实用性很强的语言。

由于 C++语言应用广泛，本书采用 C++语言编写程序。

1.1.2 算法的概念

日常生活中，无论做什么事情都要有一定的步骤，算法是为解决一个问题而采取的方法

和步骤。程序设计的关键是将解决问题的方法和步骤(即算法)描述出来,因而算法是程序设计的核心,只要设计好了算法,就可以采用任何程序设计语言来实现。可见算法在程序设计中起着举足轻重的作用,著名计算机科学家尼·沃思曾提出一个公式:

程序=数据结构+算法

数据结构是对数据的描述,在程序中指数据的类型和数据的组织形式。算法是程序的基石,数据结构是加工的对象。算法是解决计算机"做什么"和"怎么做"的问题,而程序中的操作语句,实际上就是算法的体现。

可以说,不了解算法,就无法进行程序设计。因此程序设计初学者一定要重视算法的设计,多了解、掌握和积累一些计算机常用的算法,不要急于编写程序,应该养成编写程序之前先把算法设计好的习惯。实际上,编写程序的大部分时间还是用在算法的设计上,把一个设计好的算法用具体的程序设计语言表达出来,是一件比较容易的事情。

1. 算法的特点

一个算法具有以下特点。

(1) 有穷性:一个算法在执行有限步之后必须终止。即每条指令的执行次数必须是有限的。

(2) 确定性:一个算法所给出的每一步计算步骤必须是精确定义的。即每一条指令的含义必须是明确的,无二义性。

(3) 可行性:算法中要执行的每一步计算步骤都可在有限时间内完成。即每一条指令的执行时间都是有限的。可行性与有穷性和确定性是相容的。

(4) 输入:一个算法一般具有零个或多个输入信息,这些输入量是算法所需的初始数据,它取自某一个特定的集合。

(5) 输出:一个算法一般有一个或多个输出信息,它是算法对输入信息的运算结果。

2. 基本结构

1966 年,Bohra 和 Jacopinij 提出了顺序结构、选择结构和循环结构 3 种基本结构。经过理论证明,无论多么复杂的算法,都可以表示为这 3 种基本结构的组合。

1) 顺序结构

顺序结构是 3 种基本结构中最简单的一种,算法在执行过程中会按照语句的先后顺序依次执行。例如,A、B 两个操作,在执行完 A 的操作之后,才能执行 B 的操作。

2) 选择结构

选择结构也称为分支结构,是指在算法执行过程中根据判定条件的真假来选择执行下一步的操作。例如,A、B 两个操作,当满足判断条件 P 时,执行操作 A,当不满足判断条件 P 时,执行操作 B。在算法的一次执行中,操作 A 和操作 B 只可能执行其中的一个。

3) 循环结构

循环结构用于重复执行相似或相同的操作。循环结构的特点是在给定条件 P 成立时,反复执行某个操作段 A。通常我们称给定条件 P 为循环条件,称反复执行的操作段 A 为循环体。循环结构一般分成两种情形:一种是当型循环,一种是直到型循环。在程序执行时,当型循环是先判断条件 P 是否成立,当条件 P 成立时,执行操作段 A 中的语句,然后再判断

条件 P，条件 P 成立时，再执行操作段 A 的语句，这样循环往复，直到当条件 P 不成立时，才退出循环体，执行后继的操作。

直到型循环是先执行一次操作段 A 中的语句，然后判断条件 P 是否成立，如果条件 P 不成立，则继续执行操作段 A 中的语句，然后再判断，如此往复。直到所给的条件成立时，才退出循环程序。

以上 3 种结构具有以下共同特性：

- 只有一个入口和一个出口；
- 结构中的每一个部分都有可能被执行到；
- 结构内不存在“死循环”。

1.1.3 算法的表示方法

一个算法可以用自然语言、传统流程图、N-S 流程图或伪代码等方式来描述。

传统流程图用一些图框、流程线以及一些文字说明来描述操作过程。用流程图表示算法，更加直观、易于理解。常用流程图符号如图 1-1 所示。

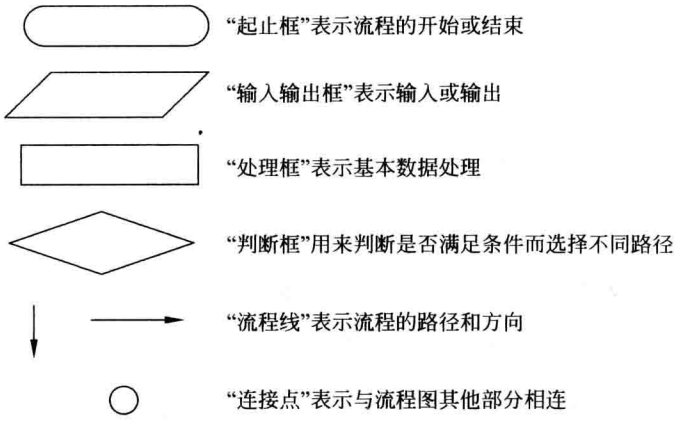


图 1-1 常用流程图符号

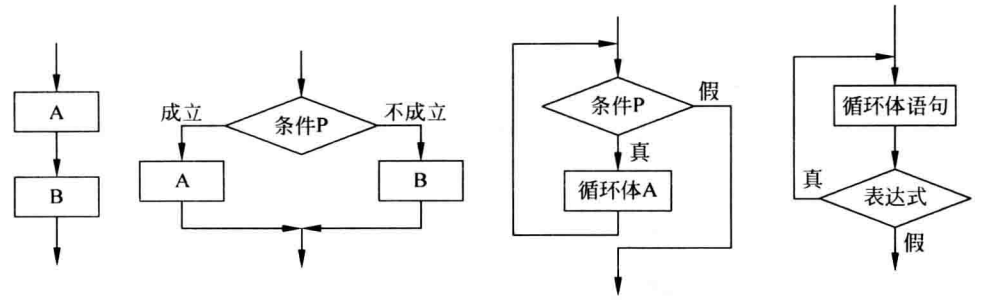
上面提出的 3 种基本结构用传统流程图表示如图 1-2 所示。其中，顺序结构如图 1-2(a)所示，分支结构如图 1-2(b)所示，当型循环结构如图 1-2(c)所示，直到型循环结构如图 1-2(d)所示。

传统流程图虽然形象直观，但对流程线的使用没有限制，使用者可以不受限制地使流程随意跳转，流程图可能变得毫无规律，不便于阅读。为了提高算法表示的质量，使算法更便于阅读，人们对流程图的表示方法进行了改进。1973 年，美国学者 I. Nassi 和 B. Shneiderman 提出了 N-S 流程图。这种流程图去掉了带有箭头的流程线，全部的算法写在一个矩形框之内，在矩形框内可以包含其他从属于它的框图，从而更适合于结构化程序设计，因此更多地被应用于算法设计中。

N-S 流程图用以下流程图符号：

(1) 顺序结构：如图 1-3(a)所示。A 和 B 两个框依次放置组成一个顺序结构。

(2) 分支结构：如图 1-3(b)所示。当条件 P 成立时，执行 A 操作，P 不成立时，执行 B

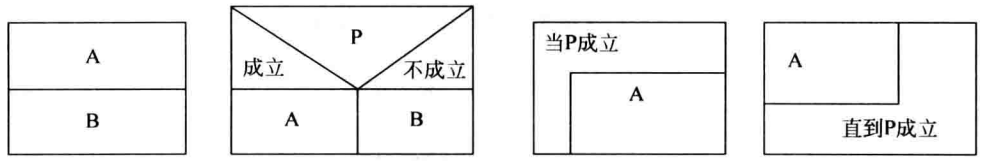


(a) 顺序结构　(b) 分支结构　(c) 当型循环结构　(d) 直到型循环结构

图 1-2　用传统流程图表示的 3 种基本结构

操作。

(3) 循环结构：当型循环用图 1-3(c)表示。当条件 P 成立时，反复执行 A 操作，直到条件 P 不成立为止。直到型循环用图 1-3(d)表示。条件 P 不成立时反复执行 A 操作，直到条件 P 成立为止。



(a) 顺序结构　(b) 分支结构　(c) 当型循环　(d) 直到型循环

图 1-3　N-S 流程图

用以上的三种 N-S 流程图的基本框，可以组成复杂的 N-S 流程图，用来表示算法。N-S 流程图就像一个多层的盒子，所以也称为盒图。

【实例 1-1】 选手的得分。

在歌唱比赛中有 10 位评委为选手评分，现要求输入 10 位评委对某位选手的评分(设给定评分范围为 0～10 分)后，输出该选手的最高得分、最低得分和平均分(计算平均分时去掉一个最高分和一个最低分)。

(1) 用自然语言表示算法。

① 初始化程序中用到的各个数据量的值。选手总分 sum＝0，当前最高分 max＝0，当前最低分 min＝10。

② 打分的评委号 i＝1。

③ 所有评委评分完毕了吗(即 i＞10 吗)？全部评分完毕转第⑧步，否则往下继续执行。

④ 输入当前评委 i 的评分 num。

⑤ 如果 num 大于最高分 max，则修改最高分 max 为 num，即 max＝num。

⑥ 如果 num 小于最低分 min，则修改最低分 min 为 num，即 min＝num。

⑦ 将 num 累加到总分上，即 sum＝sum＋num。

⑧ 打分评委号加 1，即 i＝i＋1，准备下一个评委打分，转第③步。

⑨ 输出最高得分 max、最低得分 min 和平均分(sum－max－min)/8。

(2) 用传统流程图表示。

用传统流程图表示如图 1-4 所示。

(3) 用 N-S 盒图表示。

用 N-S 盒图表示如图 1-5 所示。

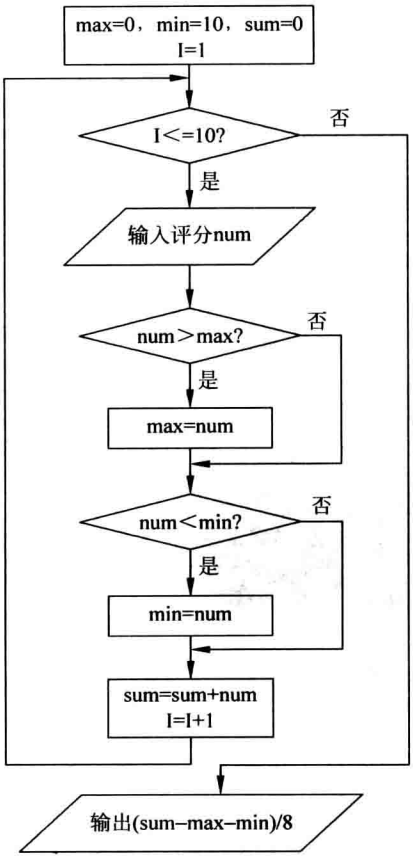


图 1-4　用传统流程图表示的算法

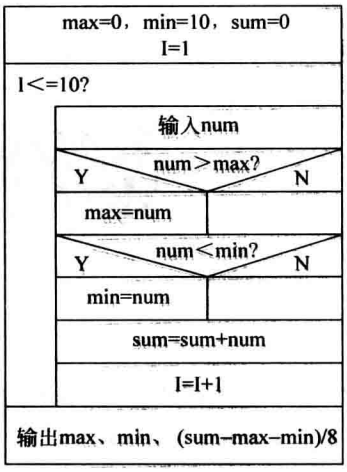


图 1-5　用 N-S 盒图表示的算法

(4) 用伪代码表示。

```
初始化最高分 max = 0.0,最低分 min = 10.0,总分 sum = 0.0;
当前打分评委号 i = 1
While   (i <= 10)
{
    输入评委 i 的评分 score;
    if  (score > max)   则 max = score;
    if  (score < min)   则 min = score;
    总分 sum = sum + score;
    评委号 i = i + 1
}
平均分 avg = (sum - max - min)/(10 - 2);
输出选手最高得分 max、最低得分 min 和平均分 avg;
程序结束
```

(5) 源程序及运行结果。

```
#include <iostream>
using namespace std;
int main()
{
    float  score,max,min,sum,avg;
    max = 0.0;     min = 10.0;       sum = 0.0;
    for (int i = 1;i <= 10;i++)
    {
        cin >> score;
        if (score > max)  max = score;
        if (score < min)  min = score;
        sum = sum + score;
    }
    avg = (sum - max - min)/(10 - 2);
    cout <<"Max = "<< max <<"   Min = "<< min;
    cout <<"    Average = "<< avg << endl;
    return 0;
}
```

在 Visual C++6.0 中,编译并执行以上程序,运行结果如图 1-6 所示。

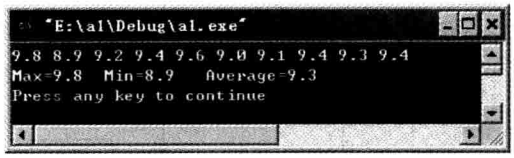


图 1-6 实例 1-1 运行结果

说明:

(1) 本书为了版面整洁,在表示运行结果时,不用图 1-6 所示的图片,而是直接将运行结果复制下来,采用如下的描述方法。

编译并执行以上程序,可得到如下所示的结果。

```
9.8 8.9 9.2 9.4 9.6 9.0 9.1 9.4 9.3 9.4
Max = 9.8   Min = 8.9    Average = 9.3
Press any key to continue
```

(2) 本书的程序采用 C++语言描述,又由于全书实例主要用于讲解结构化程序设计的方法,没有涉及面向对象的相关内容。因此,对于学习 C 语言的读者而言,本书亦是一本很好的参考书。书中绝大多数源程序只需将有关的输入输出流改写成 C 语言中的输入输出库函数,即可正确编译运行。

例如,上面的源程序用 C 语言描述为:

```
#include <stdio.h>                          /* 改写为包含 C 的输入输出头文件 */
int main()
{
    float  score,max,min,sum,avg;
    max = 0.0;     min = 10.0;       sum = 0.0;
    for (int i = 1;i <= 10;i++)
```

```
    {
        scanf("%f",&score);                    /* 改写为C的格式化输入函数 scanf */
        if (score>max)   max=score;
        if (score<min)   min=score;
        sum=sum+score;
    }
    avg=(sum-max-min)/(10-2);
    printf("Max=%.2f  Min=%.2f  Average=%.2f\n",max,min,avg);
                                           /* 改写为C的格式化输出函数 printf */
    return 0;
}
```

1.2 程序设计的步骤和方法

1.2.1 编写程序解决问题的一个例子

下面举例说明如何通过编写程序来解决实际问题。

【实例 1-2】 最简真分数的个数。

以 2010 为分母的最简真分数有多少个？所谓最简真分数是一个分数的分子小于分母，且分子分母无公因数。

(1) 问题分析。

这是一道小学奥数试题。它考察的是学生对集合包含和容斥知识的掌握情况。

由于 2010＝2×3×5×67，因此以 2010 为分母的最简真分数的分子必须小于 2010 且不能被 2、3、5 或 67 整除。

解决这个问题的计算过程如下：

在 1～2010 共 2010 个数中，

- 能被 2 整除的数有 2010÷2＝1005(个)；
- 能被 3 整除的数有 2010÷3＝670(个)；
- 能被 5 整除的数有 2010÷5＝402(个)；
- 能被 67 整除的数有 2010÷67＝30(个)；
- 能同时被 2 和 3 整除的数有 2010÷(2×3)＝335(个)；
- 能同时被 2 和 5 整除的数有 2010÷(2×5)＝201(个)；
- 能同时被 2 和 67 整除的数有 2010÷(2×67)＝15(个)；
- 能同时被 3 和 5 整除的数有 2010÷(3×5)＝134(个)；
- 能同时被 3 和 67 整除的数有 2010÷(3×67)＝10(个)；
- 能同时被 5 和 67 整除的数有 2010÷(5×67)＝6(个)；
- 能同时被 2、3 和 5 整除的数有 2010÷(2×3×5)＝67(个)；
- 能同时被 2、3 和 67 整除的数有 2010÷(2×3×67)＝5(个)；
- 能同时被 2、5 和 67 整除的数有 2010÷(2×5×67)＝3(个)；
- 能同时被 3、5 和 67 整除的数有 2010÷(3×5×67)＝2(个)；
- 能同时被 2、3、5 和 67 整除的数有 2010÷(2×3×5×67)＝1(个)。

这样，1～2010 中能被 2 或 3 或 5 或 67 整除的数有

(1005＋670＋402＋30)－(335＋201＋15＋134＋10＋6)＋(67＋5＋3＋2)－1

＝2107－701＋77－1

＝1482(个)

因此，1～2010 中既不能被 2 整除，也不能被 3 整除，也不能被 5 整除，也不能被 67 整除的数有 2010－1482＝528 个。

即以 2010 为分母的最简真分数有 528 个。

可以看出，上面的计算过程还是比较繁琐的，需要认真仔细。

下面讲一个真实的故事。2010 年时，一个小朋友问到我这个问题，我跟他讲了上面方法后，让他自己计算。他计算完后，问我正确的答案，当时我正好在上网，因此编写了一个简单的程序解决这个问题。

(2) 编程思路。

用一个变量 cnt 来保存最简真分数的个数，初始值为 0。

对 1～2010 中的每一个数 num，进行判断，这是一个循环，写成

```
for(num = 1; num <= 2010;num++)
```

循环体中的判断方法为：如果 num 既不能被 2 整除，也不能被 3 整除，也不能被 5 整除，也不能被 67 整除，则计数。写成

```
if(num % 2!= 0 && num % 3!= 0  && num % 5!= 0  && num  % 67!= 0)
      cnt++;
```

最后，输出结果 cnt。编写一个简单的程序，就得到问题的答案。

(3) 源程序及运行结果。

```
#include <iostream>
using namespace std;
int main()
{
    int cnt,num;
    cnt = 0;
    for(num = 1; num <= 2010;num++)
        if(num % 2!= 0 && num % 3!= 0  && num % 5!= 0  && num  % 67!= 0)
          cnt++;
    cout << cnt << endl;
    return 0;
}
```

编译并执行以上程序，可得到如下所示的结果。

```
528
Press any key to continue
```

通过这个实例，可以体会，编写程序让计算机解决问题有时是一个很好很有趣的事情。上面的实例再引申为，如统计分母在指定区间[10,100]的最简真分数共有多少个？显然，如果人工计算，因为需要对分母为 10～100 之间的 91 个数每个进行穷举计算，非常繁琐耗时，