

深入  
浅出 系列规划教材

# 深入浅出

## C语言程序设计 (第2版)习题集和编程指导

李俊菽 张晴晖 徐伟恒 主 编  
秦明明 强振平 邢丽伟 副主编

清华大学出版社



深入  
浅出 系列规划教材

深入  
浅出

# C语言程序设计

## (第2版)习题集和编程指导

李俊菽 张晴晖 徐伟恒 主 编  
秦明明 强振平 邢丽伟 副主编

清华大学出版社  
北京

## 内 容 简 介

本书是与清华大学出版社出版的《深入浅出 C 语言程序设计(第 2 版)》(书号: 9787302391791)一书相配套的辅助教材。书中对主教材各章的重点和难点进行总结,并对各章课后习题进行精讲,最后还给出大量编程题的题库。

本书概念清楚,习题讲解清晰易懂,可以作为高等学校计算机相关专业的辅助教材;各章同步练习和编程题题库均选自历年全国计算机等级考试二级 C 语言的试题,也适合作为全国计算机等级考试的备考用书,还可作为各类认证考试的参考书,以及计算机工程技术人员的参考书。

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

深入浅出 C 语言程序设计: 习题集和编程指导/李俊菽, 张晴晖, 徐伟恒主编. --2 版. --北京: 清华大学出版社, 2016

深入浅出系列规划教材

ISBN 978-7-302-42843-5

I. ①深… II. ①李… ②张… ③徐… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 028843 号

责任编辑: 白立军

封面设计: 傅瑞学

责任校对: 徐俊伟

责任印制: 何 芊

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto: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: 185mm×260mm 印 张: 22.5 字 数: 537 千字

版 次: 2010 年 8 月第 1 版 2016 年 6 月第 2 版 印 次: 2016 年 6 月第 1 次印刷

印 数: 1~2000

定 价: 44.50 元

产品编号: 066674-01



为什么开发深入浅出系列丛书?

目的是从读者角度写书,开发出高质量的、适合阅读的图书。

“不积跬步,无以至千里;不积小流,无以成江海。”知识的学习是一个逐渐积累的过程,只有坚持系统地学习知识,深入浅出,坚持不懈,持之以恒,才能把一类技术学习好。坚持的动力源于所学内容的趣味性和讲法的新颖性。

计算机课程的学习也有一条隐含的主线,那就是“提出问题→分析问题→建立数学模型→建立计算模型→通过各种平台和工具得到最终正确的结果”,培养计算机专业学生的核心能力是“面向问题求解的能力”。由于目前大学计算机本科生培养计划的特点,以及受教学计划和课程设置的原因,计算机科学与技术专业的本科生很难精通掌握一门程序设计语言或者相关课程。各门课程设置比较孤立,培养的学生综合运用各方面的知识能力方面有欠缺。传统的教学模式以传授知识为主要目的,能力培养没有得到充分的重视。很多教材受教学模式的影响,在编写过程中,偏重概念讲解比较多,而忽略了能力培养。为了突出内容的案例性、解惑性、可读性、自学性,本套书努力在以下方面做好工作。

## 1. 案例性

所举案例突出与本课程的关系,并且能恰当反映当前知识点。例如,在计算机专业中,很多高校都开设了高等数学、线性代数、概率论,不言而喻,这些课程对于计算机专业的学生来说是非常重要的,但就目前对不少高校而言,这些课程都是由数学系的老师讲授,教材也是由数学系的老师编写,由于学科背景不同和看待问题的角度不同,在这些教材中基本都是纯数学方面的案例,作为计算机系的学生来说,学习这样的教材缺少源动力并且比较乏味,究其原因,很多学生不清楚这些课程与计算机专业的关系是什么。基于此,在编写这方面的教材时,可以把计算机上的案例加入其中,例如,可以把计算机图形学中的三维空间物体图像在屏幕上的伸缩变换、平移变换和旋转变换在矩阵运算中进行举例;可以把双机热备份的案例融入到马尔科夫链的讲解;把密码学的案例融入到大数分解中等。

## 2. 解惑性

很多教材中的知识讲解注重定义的介绍,而忽略因果性、解释性介绍,往往造成知其然而不知其所以然。下面列举两个例子。

(1) 读者可能对 OSI 参考模型与 TCP/IP 参考模型的概念产生混淆,因为两种模型之

很多,并且会激发学生的学习兴趣。

#### 4. 自学性

一本书如果适合自学学习,对其语言要求比较高。写作风格不能枯燥无味,让人看一眼就拒人千里之外,而应该是风趣、幽默,重要知识点多举实际应用的案例,说明它们在实际生活中的应用,应该有画龙点睛的说明和知识背景介绍,对其应用需要注意哪些问题等都要有提示等。

一书在手,从第一页开始的起点到最后一页的终点,如何使读者能快乐地阅读下去并获得知识?这是非常重要的问题。在数学上,两点之间的最短距离是直线。但在知识的传播中,使读者感到“阻力最小”的书才是好书。如同自然界中没有直流的河流一样,河水在重力的作用下一定沿着阻力最小的路径向前进。知识的传播与此相同,最有效的传播方式是传播起来损耗最小,阅读起来没有阻力。

是为序。

欢迎老师投稿: [bailj@tup.tsinghua.edu.cn](mailto:bailj@tup.tsinghua.edu.cn)。

2014年12月15日

# 前 言



“C 语言程序设计”是计算机专业及各理工科专业的一门重要的基础课程。学习和掌握 C 语言,既可以增进对于计算机底层工作机制的了解,又可以为进一步学习其他高级语言打下坚实基础。

在计算机编程语言的学习过程中,最重要的是勤动手、勤动脑,只有当代码编写达到一定量时,才能真正掌握一门语言。要想学好 C 语言,深刻理解教材中提到的每个知识点,做一定数量的习题是必不可少的。

本书作者编写的《深入浅出 C 语言程序设计(第 2 版)》(书号: 9787302391791)一书于 2015 年 3 月由清华大学出版社出版,为了更好地帮助读者解决学习中的疑点和难点,进一步吃透教材中的内容,故编写此书作为 C 语言课程的辅助教材。

本书由两大部分内容组成,第一部分包括第 1~14 章,其内容与主教材各章相对应,首先给出每章的知识点总结,然后是对主教材各章课后习题进行精讲,并配有同步练习题。第二部分是第 15 章,给出大量编程题的题库,题目类型包括程序设计题、程序修改题、程序填空题,这些题均选自全国计算机等级考试二级 C 语言的考题,方便读者在掌握 C 语言基础知识的前提下,备考全国计算机等级考试。

尽管本书给出所有习题的答案,但读者切莫盲目依赖答案。正确的学习方法是遇到难题首先独立思考,复习相关知识,找出解题思路,若确实无法解答,再参考本书给出的答案。总之,应将答案作为检验自己掌握课程内容深浅的标准,切不可死记硬背。

由于作者水平有限,成书仓促,错误和不足之处在所难免,恳请读者和同行专家批评指正。为了统一风格,全书程序、代码、函数等用正体。

如果需要书中的程序源代码,可与作者联系: [li\\_junqiu@sohu.com](mailto:li_junqiu@sohu.com)。

作 者

2015 年 10 月

间有很多相似之处。其实,OSI 参考模型是在其协议开发之前设计出来的,也就是说,它不是针对某个协议族设计的,因而更具有通用性。而 TCP/IP 模型是在 TCP/IP 协议栈出现后出现的,也就是说,TCP/IP 模型是针对 TCP/IP 协议栈的,并且与 TCP/IP 协议栈非常吻合。但是必须注意,TCP/IP 模型描述其他协议栈并不合适,因为它具有很强的针对性。说到这里读者可能更迷惑了,既然 OSI 参考模型没有在数据通信中占有主导地位,那为什么还花费这么大的篇幅来描述它呢?其实,虽然 OSI 参考模型在协议实现方面存在很多不足,但是,OSI 参考模型在计算机网络的发展过程中起到了非常重要的作用,并且,它对未来计算机网络的标准化、规范化的发展有很重要的指导意义。

(2) 再例如,在介绍原码、反码和补码时,往往只给出其定义和举例表示,而对最后为什么在计算机中采取补码表示数值?浮点数在计算机中是如何表示的?字节类型、短整型、整型、长整型、浮点数的范围是如何确定的?下面我们来回答这些问题(以 8 位数为例),原码不能直接运算,并且 0 的原码有+0 和-0 两种形式,即 00000000 和 10000000,这样肯定是不行的,如果根据原码计算设计相应的门电路,由于要判断符号位,设计的复杂度会大大增加,不合算;为了解决原码不能直接运算的缺点,人们提出了反码的概念,但是 0 的反码还是有+0 和-0 两种形式,即 00000000 和 11111111,这样是不行的,因为计算机在计算过程中,不能判断遇到 0 是+0 还是-0;而补码解决了 0 表示的唯一性问题,即不会存在+0 和-0,因为+0 是 00000000,它的补码是 00000000,-0 是 10000000,它的反码是 11111111,再加 1 就得到其补码是 10000000,舍去溢出量就是 00000000。知道了计算机中数用补码表示和 0 的唯一性问题后,就可以确定数据类型表示的取值范围了,仍以字节类型为例,一个字节共 8 位,有 00000000~11111111 共 256 种结果,由于 1 位表示符号位,7 位表示数据位,正数的补码好说,其范围从 00000000~01111111,即 0~127;负数的补码为 10000000~11111111,其中,11111111 为-1 的补码,10000001 为-127 的补码,那么到底 10000000 表示什么最合适呢?8 位二进制数中,最小数的补码形式为 10000000;它的数值绝对值应该是各位取反再加 1,即为  $01111111+1=10000000=128$ ,又因为是负数,所以是-128,即其取值范围是-128~127。

### 3. 可读性

图书的内容要深入浅出,使人爱看、易懂。一本书要做到可读性好,必须做到“善用比喻,实例为王”。什么是深入浅出?就是把复杂的事物简单地描述明白。把简单事情复杂化的是哲学家,而把复杂的问题简单化的是科学家。编写教材时要以科学家的眼光去编写,把难懂的定义,要通过图形或者举例进行解释,这样能达到事半功倍的效果。例如,在数据库中,第一范式、第二范式、第三范式、BC 范式的概念非常抽象,很难理解,但是,如果以一个教务系统中的学生表、课程表、教师表之间的关系为例进行讲解,从而引出范式的概念,学生会比较容易接受。再例如,在生物学中,如果纯粹地讲解各个器官的功能会比较乏味,但是如果提出一个问题,如人的体温为什么是 37℃?以此为引子引出各个器官的功能效果要好得多。再例如,在讲解数据结构课程时,由于定义多,表示抽象,这样达不到很好的教学效果,可以考虑在讲解数据结构及其操作时用程序给予实现,让学生看到直接的操作结果,如压栈和出栈操作,可以把 PUSH()和 POP()操作实现,这样效果会好



<b>第 1 章</b>	<b>程序设计入门</b> .....	<b>1</b>
1.1	知识点总结 .....	1
1.2	教材习题精讲 .....	1
1.2.1	选择题 .....	1
1.2.2	填空题 .....	3
1.2.3	编程题 .....	3
<b>第 2 章</b>	<b>程序结构描述</b> .....	<b>4</b>
2.1	知识点总结 .....	4
2.2	教材习题精讲 .....	4
<b>第 3 章</b>	<b>C 语言预备知识</b> .....	<b>8</b>
3.1	知识点总结 .....	8
3.1.1	C 程序的基本结构 .....	8
3.1.2	C 语言的标识符 .....	8
3.1.3	C 语言的数据类型 .....	9
3.1.4	常量 .....	9
3.1.5	变量 .....	11
3.1.6	几种运算符和表达式 .....	12
3.2	教材习题精讲 .....	14
3.2.1	选择题 .....	14
3.2.2	填空题 .....	18
3.2.3	程序改错题 .....	19
3.3	同步练习题 .....	21
<b>第 4 章</b>	<b>顺序结构程序设计</b> .....	<b>23</b>
4.1	知识点总结 .....	23
4.1.1	C 语言语句的分类 .....	23
4.1.2	格式化屏幕输出函数 printf() .....	24



4.1.3	格式化键盘输入函数 scanf()	25
4.1.4	字符输入输出函数	27
4.2	教材习题精讲	27
4.2.1	选择题	27
4.2.2	填空题	30
4.2.3	编程题	31
4.3	同步练习题	35
<b>第 5 章</b>	<b>选择结构程序设计</b>	<b>38</b>
5.1	知识点总结	38
5.1.1	关系运算符	38
5.1.2	逻辑运算符	38
5.1.3	条件运算符	39
5.1.4	if 语句	39
5.1.5	switch 语句	41
5.2	教材习题精讲	42
5.2.1	选择题	42
5.2.2	编程题	47
5.3	同步练习题	54
<b>第 6 章</b>	<b>循环结构程序设计</b>	<b>57</b>
6.1	知识点总结	57
6.1.1	while 语句	57
6.1.2	do...while 语句	58
6.1.3	for 语句	58
6.1.4	break 语句	59
6.1.5	continue 语句	59
6.1.6	goto 语句	60
6.2	教材习题精讲	60
6.2.1	选择题	60
6.2.2	填空题	62
6.2.3	编程题	64
6.3	同步练习题	74
<b>第 7 章</b>	<b>数组</b>	<b>77</b>
7.1	知识点总结	77
7.1.1	一维数组	77
7.1.2	二维数组	78

7.2	教材习题精讲	80
7.2.1	选择题	80
7.2.2	填空题	83
7.2.3	编程题	84
7.3	同步练习题	95
<b>第8章</b>	<b>指针基础</b>	<b>99</b>
8.1	知识点总结	99
8.1.1	指针的基本概念	99
8.1.2	指针的运算	100
8.1.3	指针与一维数组	102
8.2	教材习题精讲	104
8.2.1	选择题	104
8.2.2	填空题	106
8.2.3	编程题	107
8.3	同步练习题	113
<b>第9章</b>	<b>函数</b>	<b>116</b>
9.1	知识点总结	116
9.1.1	函数基础	116
9.1.2	参数的传值方式与传地址方式	119
9.1.3	变量的作用范围与存储类别	120
9.1.4	函数的递归调用	121
9.1.5	编译预处理命令	122
9.2	教材习题精讲	123
9.2.1	选择题	123
9.2.2	填空题	127
9.2.3	编程题	128
9.3	同步练习题	142
<b>第10章</b>	<b>指针提高篇</b>	<b>147</b>
10.1	知识点总结	147
10.1.1	二级指针	147
10.1.2	指针与二维数组	148
10.1.3	指针的动态存储分配	150
10.1.4	函数指针	151
10.1.5	main 函数的参数	153
10.2	教材习题精讲	153

10.2.1	选择题	153
10.2.2	编程题	158
10.3	同步练习题	162
<b>第 11 章</b>	<b>字符串</b>	<b>165</b>
11.1	知识点总结	165
11.1.1	字符串的概念	165
11.1.2	字符数组与字符串	165
11.1.3	字符指针与字符串	167
11.1.4	字符串处理函数	169
11.1.5	字符串数组	171
11.2	教材习题精讲	172
11.2.1	选择题	172
11.2.2	填空题	174
11.2.3	编程题	177
11.3	同步练习题	186
<b>第 12 章</b>	<b>结构体和共用体</b>	<b>190</b>
12.1	知识点总结	190
12.1.1	声明结构体类型	190
12.1.2	结构体变量和结构体指针	192
12.1.3	结构体数组	194
12.1.4	链表	195
12.1.5	共用体	197
12.2	教材习题精讲	198
12.2.1	选择题	198
12.2.2	填空题	200
12.2.3	编程题	202
12.3	同步练习题	212
<b>第 13 章</b>	<b>文件</b>	<b>217</b>
13.1	知识点总结	217
13.1.1	文件的基本概念	217
13.1.2	文件操作库函数	218
13.2	教材习题精讲	222
13.2.1	选择题	222
13.2.2	编程题	225
13.3	同步练习题	228



<b>第 14 章</b>	<b>位运算</b> .....	<b>232</b>
14.1	知识点总结 .....	232
14.2	教材习题精讲 .....	235
14.2.1	选择题 .....	235
14.2.2	编程题 .....	237
14.3	同步练习题 .....	240
<b>第 15 章</b>	<b>编程题题库</b> .....	<b>242</b>
15.1	程序设计题 .....	242
15.1.1	数值处理 .....	242
15.1.2	一维数组 .....	250
15.1.3	二维数组 .....	255
15.1.4	字符串 .....	260
15.1.5	结构体 .....	273
15.1.6	链表 .....	277
15.2	程序修改题 .....	279
15.2.1	数值处理 .....	279
15.2.2	一维数组 .....	289
15.2.3	二维数组 .....	294
15.2.4	字符串 .....	295
15.2.5	结构体、链表 .....	303
15.3	程序填空题 .....	306
15.3.1	数值处理 .....	306
15.3.2	字符处理 .....	311
15.3.3	函数 .....	312
15.3.4	一维数组 .....	312
15.3.5	二维数组 .....	315
15.3.6	字符串 .....	319
15.3.7	结构体 .....	326
15.3.8	链表 .....	328
15.3.9	文件 .....	334
<b>附录 A</b>	<b>常用字符及 ASCII 码表</b> .....	<b>340</b>
<b>附录 B</b>	<b>C 语言关键字</b> .....	<b>342</b>
<b>附录 C</b>	<b>C 语言运算符优先级和结合性</b> .....	<b>344</b>
	<b>参考文献</b> .....	<b>346</b>



### 1.1 知识点总结

#### 1. 现代计算机的工作原理

电子计算机采用了“存储程序控制”原理。这一原理是 1946 年由美籍匈牙利数学家冯·诺依曼提出的,所以又称为“冯·诺依曼原理”。这一原理在计算机的发展过程中,始终发挥着重要影响,确立了现代计算机的基本组成和工作方式,直到现在,各类计算机的工作原理还是采用冯·诺依曼原理思想。

#### 2. 计算机语言的发展

计算机语言的发展,经历了机器语言、汇编语言到高级语言的历程。

C 语言是一种广泛使用的计算机程序设计语言,它既具有高级语言的特点,又具有低级语言的特点。凭借其功能强大、结构优雅、移植性好等特点,深受广大编程者喜爱。

#### 3. C 语言程序设计的工作原理

C 语言程序的执行过程称为“编译运行”,编译运行过程是最经典、最高效的一种执行方式。一般分为 4 个步骤:编辑(产生后缀名为.c 的源文件)、编译(产生后缀名为.obj 的目标文件)、链接(产生后缀名为.exe 的可执行文件)、运行。

### 1.2 教材习题精讲

#### 1.2.1 选择题

1. 下列对 C 语言特点的描述中,错误的是( )。
- A. C 语言不是结构化程序设计语言
  - B. C 语言编程简洁明了
  - C. C 语言功能较强
  - D. C 语言移植性好

**【答案】** A

**【解析】** 计算机高级语言分为结构化程序设计语言和面向对象程序设计语言,C 语言属于结构化程序设计语言。

2. 最早开发 C 语言是为了编写( )操作系统。  
A. Windows      B. DOS      C. UNIX      D. Linux

**【答案】** C

**【解析】** C 语言是贝尔实验室开发 UNIX 操作系统的“副产品”。

3. 下面的说法正确的是( )。  
A. C 程序由符号构成      B. C 程序由标识符构成  
C. C 程序由函数构成      D. C 程序由 C 语句构成

**【答案】** C

**【解析】** C 程序由若干函数组成,其中有且仅有一个主函数。

4. 一个 C 程序的执行是从( )。  
A. 本程序的 main 函数开始,到 main 函数结束  
B. 本程序文件的第一个函数开始,到本程序文件的最后一个函数结束  
C. 本程序的 main 函数开始,到本程序文件的最后一个函数结束  
D. 本程序文件的第一个函数开始,到本程序 main 函数结束

**【答案】** A

**【解析】** main 函数也称为主函数,它是一个 C 程序的入口和正常出口,而不论其放在程序的任何位置。

5. 以下叙述不正确的是( )。  
A. 一个 C 源程序可由一个或多个函数组成  
B. 一个 C 源程序必须包含一个 main 函数  
C. C 程序的基本组成单位是函数  
D. 在 C 程序中,注释说明只能位于一条语句的后面

**【答案】** D

**【解析】** 注释说明可以放在语句前面、后面,甚至中间,只要不构成语法错误即可。

6. 编写 C 程序一般需要经过的几个步骤依次是( )。  
A. 编译、编辑、链接、调试、运行  
B. 编辑、编译、链接、运行、调试  
C. 编译、运行、调试、编辑、链接  
D. 编辑、调试、编辑、链接、运行

**【答案】** B

**【解析】** C 程序设计一般要经历 4 个基本步骤:编辑(产生.c 源文件)、编译(产生.obj 目标文件)、链接(产生.exe 可执行文件)、运行。若程序有错误,还应对程序进行调试。

7. Windows 系统里由 C 源程序文件编译而成的可执行文件的默认扩展名为( )。  
A. cpp      B. exe      C. obj      D. lik

**【答案】** B

**【解析】** cpp 是用 C++ 语言编写的源程序;obj 是目标文件。

8. C 程序中主函数的个数是( )。

- A. 2                      B. 3                      C. 任意多个                      D. 1

**【答案】** D

**【解析】** 在一个C程序中,主函数有且仅有一个,它是一个程序的入口和正常出口。

9. 下面叙述中不属于C语言的特点的是( )。

- A. 是一种面向对象的程序设计语言  
B. 数据类型丰富,表达力强  
C. 允许直接访问物理地址,对硬件进行操作  
D. 运算符丰富

**【答案】** A

**【解析】** C语言是面向过程的程序设计语言,也称为结构化程序设计语言。

## 1.2.2 填空题

1. 计算机语言的种类非常多,从其发展过程,经历了 ①、② 到 ③ 的历程。

**【答案】** ① 机器语言    ② 低级语言    ③ 高级语言

2. 在C程序的编辑、编译、链接、运行和调试过程中,编译是指\_\_\_\_\_。

**【答案】** 把高级语言编写的源程序变成计算机可以识别的二进制语言的过程。

3. C语言源文件名的后缀是 ①;经过编译后,生成目标文件的后缀是 ②;再经过链接后,生成可执行文件的后缀是 ③。

**【答案】** ① .c    ② .obj    ③ .exe

4. 程序中的错误大致可分为三类,具体为编译错误、① 和 ②。

**【答案】** ① 链接错误    ② 运行错误

## 1.2.3 编程题

编写程序,输出如下信息(对应主教材 1.6.3 编辑题中的第2题)。

```
*****
```

```
C is quirky,flawed,and an enormous success.
```

```
-----
```

**【参考程序】**

```
#include<stdio.h>
int main()
{
    printf("*****\n");
    printf("    C is quirky, flawed, and an enormous success.\n");
    printf("----- \n");
    return 0;
}
```

## 2.1 知识点总结

### 1. 算法

“算法”是指为解决某个特定问题而采取的确定的且有限的步骤的一种描述,它是指令的有限序列,使得给定类型的问题通过有限的指令序列,在有限的时间内被求解。

### 2. 结构化程序设计语言

计算机语言的发展过程,最初为面向机器的语言,然后到结构化语言,再到面向对象的语言。C语言是结构化语言的代表。

结构化程序设计的基本原则包括自顶向下、逐步求精,模块化,限制使用 goto 语句(无条件跳转)。

结构化程序设计使用 3 种基本控制结构构造程序,即任何程序都可用顺序、选择、循环 3 种基本控制结构构造。

### 3. 流程图的基本组成元素

结构化程序设计的思路一般采用流程图描述。一般的流程图由图 2-1 所示的几种基本图形组成。



图 2-1 一般流程图中的基本图形元素

## 2.2 教材习题精讲

绘制以下各问题的流程图。

1. 输入 3 个整数给 a、b、c,然后交换它们中的数,把 a 中的值给 b,b 中的值给 c,c 中的值给 a,然后输出 a、b、c。



**【解题思路】** 本题是顺序结构。第一步将 a 的值给中间变量 t, 第二步将 b 的值给 a, 第三步将 c 的值给 b, 第四步将 t 的值给 c。流程图如图 2-2 所示。

2. 输入两个整数 1200 和 370, 求出它们的商和余数, 然后输出。

**【解题思路】** 本题是顺序结构。流程图如图 2-3 所示。

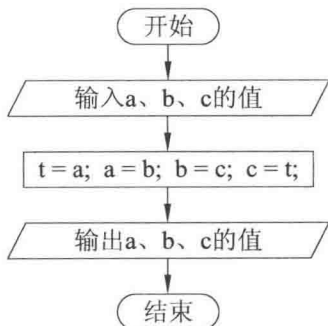


图 2-2 第 1 题的流程图

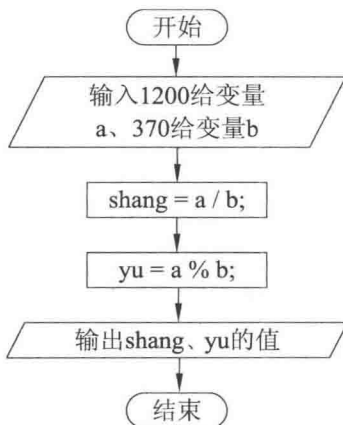


图 2-3 第 2 题的流程图

3. 输入一个实数 1.245678, 将该数进行四舍五入运算后, 保留两位小数, 输出 1.25。

**【解题思路】** 本题是顺序结构。第一步将 1.245678 乘以 100 得 124.5678; 第二步将 124.5678 加 0.5 得 125.0678; 第三步将 125.0678 强制转换为整数得 125; 第四步将 125 除以 100.0 得 1.25。流程图如图 2-4 所示。

4. 输入一个整数, 判断它是奇数还是偶数, 并输出结果。

**【解题思路】** 本题是选择结构, 有两个分支。判断一个数是奇数还是偶数的方法是用该数对 2 求余, 如果余数为 1 是奇数, 余数为 0 是偶数。流程图如图 2-5 所示。

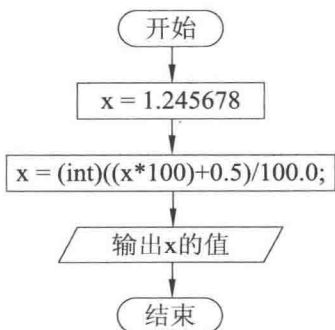


图 2-4 第 3 题的流程图

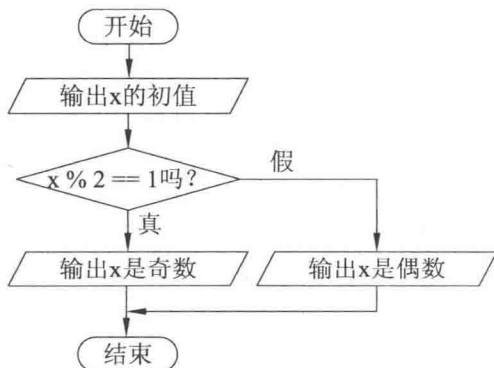


图 2-5 第 4 题的流程图

5. 输入 3 个整数到 a、b、c 中, 输出其中的最大值。

**【解题思路】** 本题是选择结构。首先选出 a、b 中的较大值存入 t 中, 再用 t 和 c 进行