



21世纪高校计算机系列规划教材

C语言程序设计同步训练 与上机指导 (第二版)

时景荣 张晓东 主 编
于 波 赵 岩 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

 21 世纪高校计算机系列规划教材

C 语言程序设计同步训练与上机指导

(第二版)

主 编 时景荣 张晓东

副主编 于 波 赵 岩

参 编 罗传义 李立春 林 琳

内 容 简 介

本书是《C 语言程序设计》的配套教材。编写时突出了“基本概念、基本方法、基本技能”，尽量做到少而精，重点放在“常用、实用”上，习题和实验内容丰富，具有启发性和综合性，加强了程序设计能力的训练。

全书共分三个部分。第一部分为 C 语言程序设计同步训练，除第 1 章外，其余每章都包含“要点、难点阐述”、“例题分析”、“同步练习”、“参考答案”四节。第二部分为 C 语言程序设计上机指南，详细介绍了 Turbo C 2.0 集成环境的上机过程及程序调试和测试的基本知识和常用方法；介绍了 Visual C++ 6.0 的使用方法。第三部分为 C 语言程序设计上机实验，结合理论教学精选内容，具体安排了 15 个实验（每个实验都包含实验目的、预习内容、实验内容、课外任务）；特别安排了综合程序设计，便于学生综合应用所学程序设计知识，进一步提高程序设计的能力。书中所有程序均在 Turbo C 2.0 上调试通过。

本书适合作为高等学校各专业 C 语言程序设计课程的配套教材，也可作为计算机等级考试培训辅导用书，还可作为自学 C 语言程序设计的参考书。

图书在版编目（CIP）数据

C 语言程序设计同步训练与上机指导 / 时景荣，张晓东
主编. —2 版. —北京：中国铁道出版社，2008.12
（21 世纪高校计算机系列规划教材）
ISBN 978-7-113-09394-5

I. C… II. ①时…②张… III. C 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字（2008）第 194128 号

书 名：C 语言程序设计同步训练与上机指导（第二版）
作 者：时景荣 张晓东 主编

策划编辑：严晓舟 唐 凯

责任编辑：李小军

编辑助理：吴媛媛

责任印制：李 佳

编辑部电话：(010) 63583215

封面设计：付 巍

封面制作：白 雪

出版发行：中国铁道出版社（北京市宣武区右安门西街 8 号 邮政编码：100054）

印 刷：三河市华丰印刷厂

版 次：2009 年 2 月第 2 版 2009 年 2 月第 1 次印刷

开 本：787mm×1092mm 1/16 印张：15 字数：351 千

印 数：3 000 册

书 号：ISBN 978-7-113-09394-5/TP·3033

定 价：25.00 元

版权所有 侵权必究

凡购买铁道版的图书，如有缺页、倒页、脱页者，请与本社计算机图书批销部调换。

第二版前言

本书是《C 语言程序设计》(时景荣主编,中国铁道出版社出版)的配套教材。第二版突出了“基本概念、基本方法、基本技能”,尽量做到少而精,重点放在“常用、实用”上,习题和实验内容丰富,具有启发性和综合性,加强了程序设计能力的训练。

全书分三个部分:

第一部分为 C 语言程序设计同步训练。除第 1 章外,每章都包含“要点、难点阐述”、“例题分析”、“同步练习”、“参考答案”四节。“要点、难点阐述”力图用简洁准确的语言给出本章的脉络,突出重点、难点,便于读者把握。“例题分析”以实例进一步解析本章要点、难点,分析可能的出错及原因,读者可以从中学习阅读程序和分析问题的方法。“同步练习”以选择题、填空题、编程题三种形式帮助读者消理解解和应用本章内容。第二版加强了编程训练,编程题程序前有算法分析,程序中有注释,程序后有测试;有的题目给出多种方法,以开拓读者思维,培养程序设计的能力。习题的选择既考虑了知识点的覆盖面,以培养程序设计能力为主线,又兼顾了计算机等级考试的能力训练,旨在培养综合能力。

第二部分为 C 语言程序设计上机指南。详细介绍了 Turbo C 2.0 集成环境的上机过程以及程序调试和测试的基本知识和常用方法;介绍了 Visual C++ 6.0 的使用方法。

第三部分为 C 语言程序设计上机实验。给出了上机实验的目的和要求,结合理论教学精选内容,具体安排了 15 个实验(每个实验都包含实验目的、预习内容、实验内容、课外任务),对于有一定难度的实验题目给出了提示,便于进行实验教学。特别在实验 15 中安排了综合程序设计,便于学生综合应用所学程序设计知识,独立完成相对完整且有一定难度的题目,进一步提高程序设计的能力。

此外,在附录中给出了 Turbo C 2.0 常见的编译出错信息和各菜单的详细功能,便于读者自学查阅。

书中所有程序均在 Turbo C 2.0 上调试通过。

本书适合作为高等学校各专业 C 语言程序设计课程的配套教材,也可作为计算机等级考试培训辅导用书,还可作为自学 C 语言程序设计的参考书。

本书由时景荣、张晓东任主编,于波、赵岩任副主编。第 1~4 章由于波编写,第 5~8 章由罗传义编写,第 10~11 章由赵岩编写,第 12~15 章由张晓东编写,其余部分由时景荣编写,李立春、林琳参与了书中部分程序的调试。全书由时景荣统稿,王立国主审。

书中难免有疏漏和不妥之处,敬请读者给予指正。

编 者

2008 年 11 月

第一版前言

C 语言以其语言简洁紧凑、使用灵活方便、功能强、应用面广等诸多优点成为学习计算机程序设计语言的首选语言。《C 语言程序设计》是大、中专院校普遍开设的计算机基础课程。然而，正是由于其功能强，编程限制少，灵活性大，也意味着不好把握，易出错，难检错，调试困难。所以对使用者要求较高，尤其是初学者会感到很难学。本书的编写就是针对这些问题，力图概念叙述简明清晰、通俗易懂，例题习题选择针对性强，希望成为读者 C 程序设计解惑的工具、能力培养的助手。

全书分两个部分：

第一部分为 C 程序设计同步训练。每一章中包含“要点、难点阐述”、“例题分析”、“同步练习”、“参考答案”四部分。“要点、难点阐述”力图用简洁准确的语言给出本章的脉络，突出重点、难点，便于读者把握。“例题分析”以实例进一步解析本章要点、难点，分析可能的出错及原因，读者可以从中学习读程序和分析问题的方法。“同步练习”以选择题、填空题、编程题三种形式，帮助读者消化理解和应用本章内容。编程题程序前有算法分析，程序中有注释，程序后有测试；有的题目给出多种方法，以开拓读者思维，培养程序设计的能力；若用后续内容编程效果更好，则给出相应的程序，以增加学习后续课程的兴趣，同时在方法前加“*”以示区别，可以在学完相应内容后再回头理解这些程序。习题的选择既考虑了知识点的覆盖面，以培养程序设计的能力为主线，又重点兼顾了计算机等级考试的能力训练，旨在培养综合能力。

第二部分为 C 程序设计上机指导。详细介绍了目前广泛使用的 Turbo C 集成环境的上机过程，介绍了程序调试和测试的基本知识和常用方法；简要介绍了 Visual C++ 6.0 的使用方法；给出了上机实验的目的和要求，结合理论教学精选内容，具体安排了十五个实验（每个实验都包含：实验目的、预习内容、实验内容、课外任务），便于进行实验教学。

此外，在附录中给出了 Turbo C 常见的编译出错信息和各菜单的详细功能，便于读者自学查阅。

本书既可以用作本科、专科教学的辅助教材，又可以作为计算机等级考试培训辅导用书，也可以作为自学 C 语言程序设计的参考书。

本书由时景荣、童少为主编。第十五章、各章“同步练习”中的选择题、填空题由童少为编写，其余部分由时景荣编写，罗传义教授主审全书。

本书难免有错误和不足之处，恳请读者不吝赐教，给予指正。

编 者

2005 年 11 月

目 录

第一部分 C 语言程序设计同步训练

第 1 章 程序设计概述	1
1.1 要点、难点阐述	1
1.2 同步练习	3
1.3 参考答案	4
第 2 章 数据类型与数据运算	5
2.1 要点、难点阐述	5
2.2 例题分析	8
2.3 同步练习	11
2.4 参考答案	14
第 3 章 顺序结构的程序设计	16
3.1 要点、难点阐述	16
3.2 例题分析	17
3.3 同步练习	21
3.4 参考答案	28
第 4 章 选择结构的程序设计	29
4.1 要点、难点阐述	29
4.2 例题分析	29
4.3 同步练习	32
4.4 参考答案	38
第 5 章 循环结构的程序设计	40
5.1 要点、难点阐述	40
5.2 例题分析	42
5.3 同步练习	44
5.4 参考答案	53
第 6 章 编译预处理	55
6.1 要点、难点阐述	55
6.2 例题分析	56
6.3 同步练习	58
6.4 参考答案	62

第 7 章 函数	63
7.1 要点、难点阐述	63
7.2 例题分析	66
7.3 同步练习	71
7.4 参考答案	81
第 8 章 数组	82
8.1 要点、难点阐述	82
8.2 例题分析	85
8.3 同步练习	87
8.4 参考答案	100
第 9 章 指针	101
9.1 要点、难点阐述	101
9.2 例题分析	105
9.3 同步练习	112
9.4 参考答案	124
第 10 章 结构体与共用体	126
10.1 要点、难点阐述	126
10.2 例题分析	128
10.3 同步练习	132
10.4 参考答案	143
第 11 章 数据文件	144
11.1 要点、难点阐述	144
11.2 例题分析	146
11.3 同步练习	147
11.4 参考答案	156

第二部分 C 语言程序设计上机指南

第 12 章 Turbo C 2.0 使用指南	157
12.1 进入和退出 Turbo C 环境的方法	157
12.2 Turbo C 的工作窗口	157
12.3 编辑一个 C 语言源文件	159
12.4 确定 Turbo C 文件存储路径	162
12.5 C 程序的编译、连接和运行	163
12.5.1 编译、连接和运行单文件 C 程序	163
12.5.2 编译、连接和运行多文件 C 程序	165
12.6 程序的测试与调试	166
12.6.1 关于程序的测试与调试	166

12.6.2	一般的测试与调试方法	167
12.6.3	程序动态调试方法	170
第 13 章	Visual C++使用简介	176
13.1	Visual C++ 6.0 集成开发环境	176
13.2	项目和工作区	178
13.3	如何创建并组织文件、项目和工作区	179
13.4	程序的编译、连接和运行	181
13.5	程序的跟踪调试	182
13.6	开发 C 语言程序的方法	184
 第三部分 C 语言程序设计上机实验		
第 14 章	实验目的与要求	189
14.1	上机实验的目的	189
14.2	上机实验的基本要求	190
第 15 章	上机实验的内容	191
实验 1	编辑、编译、连接并运行一个 C 程序的方法	191
实验 2	顺序结构程序设计	193
实验 3	选择结构程序设计	194
实验 4	循环结构程序设计 (1)	195
实验 5	循环结构程序设计 (2)	196
实验 6	编译预处理	197
实验 7	函数应用 (1)	198
实验 8	函数应用 (2)	199
实验 9	数组应用 (1)	199
实验 10	数组应用 (2)	200
实验 11	指针应用 (1)	201
实验 12	指针应用 (2)	203
实验 13	结构体和共用体类型的应用	205
实验 14	数据文件的应用	205
实验 15	综合程序设计	206
附录 A	Turbo C 2.0 各菜单功能	216
附录 B	Turbo C 2.0 编译错误信息	221
参考文献		231

第一部分 C 语言程序设计同步训练

第 1 章 程序设计概述

1.1 要点、难点阐述

1. 程序设计的过程

(1) 任务分析

任务分析即弄清楚任务中数据与数据之间的逻辑关系、具体的操作要求（如需要输入哪些数据，要对数据进行哪些处理，要求输出哪些数据等），也就是弄清楚要计算机“做什么”。

(2) 算法设计

算法设计是设计解决问题的方法和步骤，对问题处理过程进行细化，即明确要计算机“怎么做”。

(3) 程序编制

首先编码，即选择一种程序设计语言，根据算法写出源程序；然后编辑，即将编写好的源程序通过编辑器输入到计算机内，并以纯文本文件的形式保存。

(4) 调试运行

计算机不能直接执行源程序，需通过编译和连接生成计算机能够执行的可执行文件。调试运行过程如下：

① 编译：将源程序翻译成目标程序，翻译时编译器对源程序进行语法检查，给出编译信息。若有语法错误，通过编辑器修改、再编译直到编译成功，生成目标程序。

② 连接：将目标程序和程序所需的目标程序模块（如调用的标准函数、执行的输入/输出模块等）连接后生成可执行文件。

③ 运行并分析结果：如需输入数据，应设计能涵盖各种情况的测试数据，然后运行程序，检查结果是否符合问题要求，是否正确。即使程序能正常运行并得到了运行结果，也可能存在逻辑错误，而计算机却无法检查出这些错误。若存在逻辑错误，如果是算法有错，则应先修改算法，再修改程序；如果算法正确而程序写得不对，则直接修改程序。

(5) 编写程序文档

程序文档就是程序的使用说明书和技术说明书，它记录了程序设计的全过程。程序文档对于开发、维护时间较长的软件来说至关重要，尤其是对软件进行二次开发更离不开程序文档。

2. 数据结构与算法

(1) 数据结构

数据结构包括 3 方面的内容：

① 数据的逻辑结构: 描述数据与数据之间的逻辑关系。

② 数据的存储结构: 描述数据和数据之间的关系在计算机中存储的方式。

③ 数据的运算集合: 即对数据进行的所有操作 (如输入、查找、更新、排序、输出等)。

确定了数据的逻辑结构, 就明确了数据运算的集合。确定了如何存储数据, 就可以设计算法了。

(2) 算法

算法是为了解决一个问题而采取的方法和步骤。算法定义一个操作序列, 描述怎样从给定的数据经过有限步骤的处理后产生所求的输出结果。算法一般包括 3 个部分:

① 初始化 (包括输入原始数据和为数据处理所做的准备)。

② 数据处理 (实现具体的功能)。

③ 输出处理结果。

描述算法有多种方法, 常用的有流程图和 N-S 图。

3. 结构化程序设计方法

结构化程序设计也称面向过程的程序设计, 如 C 语言就是结构化的程序设计语言。

结构化程序设计方法的基本思想是把一个复杂的问题分解成若干个功能独立的模块, 分而治之。具体地说就是:

① 在软件设计和实现的过程中, 采用自顶向下、逐步细化的模块化设计原则。

② 在代码编写时, 每一个模块内采用 3 种基本结构。

3 种基本结构即顺序结构、选择结构和循环结构。

① 顺序结构: 按顺序依次执行。

② 选择结构 (又称分支结构): 根据条件判断, 选择某分支执行。

③ 循环结构: 只要循环条件成立, 就重复执行一组语句 (这组语句通常称为循环体)。循环结构分为两种类型:

- 当型循环 (先判断): 当循环条件为真时重复执行循环体, 为假时循环结束。

- 直到型循环 (后判断): 重复执行循环体, 直到循环条件为假时, 循环结束。

4. C 语言概述

(1) C 语言程序的结构

① C 语言程序是由函数组成的, 函数是 C 语言程序的基本单位。C 语言中有 3 种函数: `main()` 函数、系统提供的库函数、用户自定义的函数。

② 一个函数由两部分组成:

- 函数首部: 包括函数类型、函数名、函数形式参数及类型说明等。

- 函数体: 即函数首部下面最外层大括号内的部分。函数体又可以分为声明部分和执行部分。

③ 一个 C 语言的源程序有且只有一个 `main()` 函数, 有若干个 (包括零个) 其他函数。

④ `main()` 函数的位置没有限制, 可以位于程序的任何地方, 但是 C 程序总是从 `main()` 函数开始执行, 并且结束于 `main()` 函数。

⑤ C 语言程序书写格式自由。一行可以写多个语句, 一个语句也可以写在多行上, 用分号 “;” 标识语句结束。

⑥ C语言源程序中可以在任何可以插入空格的地方插入注释，格式为：

/* 注释内容 */

注释内容可以是中文或英文，也可以是任何可显示的符号。

⑦ C语言中没有输入和输出语句。C语言程序中输入和输出操作是通过调用库函数 `scanf()`、`printf()` 和其他输入/输出函数来完成的。

(2) C语言的标识符

标识符是一个作为名字的字符序列，用来标识变量名、类型名、数组名、函数名和文件名等。C语言的标识符可分为用户标识符、保留字和预定义标识符3类。

① 用户标识符：就是程序设计者根据编程需要自己定义的名字，用来作为变量名、符号常量名、数组名、函数名、类型名和文件名等。

标识符命名规则：可以是单个字母；也可以由字母、数字和下画线组成，但必须是以字母或下画线开头。

注意：在C语言中大小写字母是不同的字符。例如，`SUM`、`Sum`、`sum`是3个不同的标识符。

② 保留字（又称关键字）：C语言专门用来描述类型和语句的标识符，共有32个。

注意：在C语言中，每个保留字都用小写英文字母表示，在C语言中都代表固定的含义，不允许作为用户标识符使用。

③ 预定义标识符：除了上述保留字外，还有一类具有特殊意义的标识符，它们被用做编译预处理命令或库函数的名字。如 `define`、`include`、`scanf`、`printf` 等，这类标识符称为预定义标识符。最好不要用它们作为用户标识符。

1.2 同步练习

一、选择题

1. 编辑程序就是_____。
A. 调试程序
B. 建立并修改源程序文件
C. 将源程序变成目标程序
D. 命令计算机执行程序
2. C语言程序的基本单位是_____。
A. 函数
B. 语句
C. 字符
D. 程序行
3. 在一个源程序中，`main()`函数的位置_____。
A. 必须在最前面
B. 可以在程序的任何位置
C. 必须在最后面
D. 必须在系统提供的库函数调用之后
4. 系统默认的C语言源程序的扩展名是_____。
A. `.exe`
B. `.c`
C. `.obj`
D. `.doc`

5. C 语言用_____标志语句结束。
A. 逗号 B. 分号
C. 句号 D. 冒号
6. 下面可以用做 C 语言用户标识符的一组标识符是_____。
A. void word FOR B. a1_b1 _123 IF
C. Case -abc xyz D. case5 liti 2ab
7. 下面的标识符中, 不合法的用户标识符为_____。
A. Pad B. _int C. CHAR D. a#b
8. 下面的标识符中, 合法的用户标识符为_____。
A. day1 B. long C. 3AB D. a+b

二、填空题

1. 软件包括程序和_____。
2. 程序的错误一般分为两种: (1) 和 (2), 前者是编译器可以发现的, 而后者编译器无法发现。
3. C 语言程序是由函数构成的, 其中_____一个 main() 函数。
4. C 语言程序的执行总是由_____函数开始, 并且在_____函数中结束。
5. C 语言程序的注释可以出现在程序中的_____。
6. 计算机不能直接执行 C 语言的源程序, 必须经过 (1) 和 (2), 形成可执行文件。

1.3 参考答案

一、选择题

1. B 2. A 3. B 4. B 5. B
6. B 7. D 8. A

二、填空题

1. 程序文档
2. (1) 语法错误 (2) 逻辑错误
3. 有且只有
4. main(), main()
5. 任何空格可以出现的位置
6. (1) 编译 (2) 连接

第2章 数据类型与数据运算

2.1 要点、难点阐述

1. 数据类型

(1) 基本类型

- 整型：又可分为 int、short、long、unsigned。
 - 字符型：char。
 - 实型：又可分为 float、double。
 - 枚举型。（用户自定义的）
- } （系统提供的）

(2) 构造类型

- 数组：同类型数据的一个组合类型。
 - 结构体：同类型或不同类型数据的一个组合类型。
 - 共用体：与结构体相似，但组合中数据从同一内存地址开始存储。
 - 文件：当输入/输出的对象是磁盘时，使用文件类型（FILE）。（系统提供的）
- } （用户自定义的）

(3) 指针类型

指针类型是某种类型数据的地址类型。（用户自定义的）

(4) 空类型

不需要类型时，使用空类型（void）。（系统提供的）

2. 常量

常量是指在程序运行过程中其值不能改变的量。基本类型常量的情况如表 2-1 所示。

表 2-1 基本类型常量的情况

常量类型	常量形式	实 例	备 注
整型常量	十进制形式	12, +123, -16, 20, 0L	常量后加 L, 表示长整型
	八进制形式	014, +0173, -020, _024	以 0 开头
	十六进制形式	0xe, +0x7b, -0x10, 0x14	以 0x (或 0X) 开头
实型常量	十进制小数形式	123.5, .123, -0.5, +12.	点前 0、点后 0 都可以省略
	指数形式	-1.23e2, 1.23E2, +1.23e+2, 1E-2	e (E) 前必须有数, 其后必须是整数
字符型常量	字符形式	'a', 'A', '*', '3', '')	大多数可以在屏幕上显示的字符
	转义字符形式	'\n', '\w', '\141', '\x41'	多用于控制字符
字符串常量	用双引号括起的字符序列	"a", "I love China"	n 个字符的字符串在内存中占 n+1 个字节
符号常量	用#define 定义	#define PI 3.1416	在程序中用 PI 代替 3.1416

3. 变量

变量是内存的一个存储单元,存储单元的当前值就是变量的值。其值在程序运行过程中可以改变,所以称为变量。变量必须先定义其类型,然后才能使用。基本数据类型变量的情况如表 2-2 所示。

表 2-2 基本数据类型变量的情况

变 量 类 型	字 节 数	取 值 范 围	有 效 数 字
int	2	-32 768 ~ 32 767, 即 $-2^{15} \sim (2^{15}-1)$	
long [int]	4	-2 147 483 648 ~ 2 147 483 647, 即 $-2^{31} \sim (2^{31}-1)$	
unsigned [int]	2	0 ~ 65535, 即 $0 \sim (2^{16}-1)$	
float	4	$-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$	6 ~ 7
double	8	$-1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$	15 ~ 16
char	1	0 ~ 255	

变量定义的一般形式为:

数据类型名 变量表;

其中,变量表中变量之间用逗号分隔,分号标志着定义结束。例如:

```
int a,b,c;
```

变量可以在定义的同时初始化(即赋初值)。例如:

```
int a=1,b=2,c=3;
```

也可以对部分变量初始化。例如:

```
int a,b,c=5;
```

注意:在程序设计时,要根据使用数据的范围定义不同类型的变量。定义过大浪费计算机资源,定义过小可能产生“溢出”或导致数据不精确。

4. 数据运算

C 语言规定了运算符的优先级和结合性。在表达式求值时,先按运算符的优先级别高低次序运算,同级则按规定的“结合方向”处理。

(1) 赋值运算

赋值运算符“=”的左侧必须是变量名,右侧是任意类型的表达式。

赋值表达式的求解过程:将赋值运算符右侧表达式的值送到左边变量所代表的存储单元中。

结合方向是右结合,优先级别较低,只高于逗号运算符。

做赋值运算时,系统对不同类型的数据自动进行类型转换。

① 把实型数据赋给整型变量时,小数部分截去。

② 长的数据给短的变量赋值,低位对齐赋值,高位丢失,有可能造成溢出。

③ 短的数据赋给长的变量,低位对齐赋值。对无符号数高位补 0;对有符号数要进行“符号扩展”:即符号位为“0”,高位全补“0”;符号位为“1”,高位全补“1”。

(2) 算术运算

① 除法运算中,当两个操作数都是整型时,结果为整型,如 $1/4$ 结果为 0。

② 求余运算中，两个操作数必须都是整型。

③ 自增/自减运算符的操作数只能是变量。在 Turbo C 中，若表达式中有自增/自减运算符，则应先扫描每一个变量的当前值，然后再计算表达式的值。例如：

```
int a=2,b=3,c;c=(++a)*(b--);
```

先扫描表达式 $(++a)*(b--)$ ，变量 a 先自增 1，即当前值为 3， b 的当前值为 3，所以 c 的值为 $3*3=9$ ，然后 b 自减为 2。所以，语句执行结束时 $a=3$ ， $b=2$ ， $c=9$ 。

④ 数学表达式转换成 C 语言表达式要注意两点：一是乘法的乘号不能省略；二是恰当地使用圆括号以保证原来的逻辑关系不变。

(3) 关系运算和逻辑运算

① 关系运算和逻辑运算的结果都是一个逻辑值，即“真”或“假”。C 语言没有逻辑数据，用“1”代表“真”，用“0”代表“假”。当进行运算时，非 0 即“真”，0 即“假”。

② 关系表达式、逻辑表达式的书写要注意，如数学中的不等式 $0 < x < 10$ ，C 语言中应写成： $x > 0 \ \&\& \ x < 10$ 。

③ 在逻辑表达式的求解中，并不是所有的逻辑运算符都被执行，只是在必须执行下一个逻辑运算符才能求出表达式的解时，才执行该运算符。

如 $a=0; b=1; c=a++\&\&b++$ ；则结果为 $a=1$ ， $b=1$ ， $c=0$ 。后一个赋值语句中赋值号右侧逻辑表达式的求解过程：逻辑与运算符左侧的表达式 $a++$ 的当前值为 0，整个逻辑表达式值即为 0，无须继续计算， a 自身增 1，0 赋给 c ，运算结束。

再如 $a=0; b=1; c=++a\||b++$ ；则结果为 $a=1$ ， $b=1$ ， $c=1$ 。后一个赋值语句中赋值号右侧逻辑表达式的求解过程：逻辑或运算符左侧的表达式 $++a$ 的值为 1，整个逻辑表达式值即为 1，无须继续计算，1 赋给 c ，运算结束。

注意：当需要判断两个量是否相等时，运算符是由两个等号“==”组成的“等于”，而一个等号“=”表示赋值。

(4) 位运算

所谓位运算是指二进制位的运算，即运算的单位是二进制的位 (bit)。

在程序中，若进行位运算的数据不是用二进制表示的，要先转换成二进制补码形式再运算，运算后的结果再转换成要求的进制。位运算有如下几种：

① 按位与运算“&”：两个对应位均为 1 时该位结果为 1，否则为 0。运算的特点：与 0 相与对应位清零；与 1 相与对应位保留原值。

② 按位或运算“|”：两个对应位均为 0 时该位结果为 0，否则为 1。运算的特点：与 1 相或对位位置 1；与 0 相或对位保留原值。

③ 按位异或运算“^”：两个对应位不同为 1，相同为 0。运算的特点：与 1 相异或对位翻转；与 0 相异或对位保留原值。

④ 按位取反运算“~”：就是 0 变 1，1 变 0。

⑤ 左移运算“<<”：用来将一个数的各二进制位全部左移若干位，高位溢出舍弃，低位补 0。操作数每左移一位，相当于乘以 2。

⑥ 右移运算“>>”：用来将一个数的各二进制位全部右移若干位，低位舍弃。若无论操作

数是正还是负都高位补 0, 称为“逻辑右移”; 若操作数为正高位补 0, 为负高位补 1, 称为“算术右移”。操作数每右移一位, 相当于除以 2。

2.2 例题分析

【例 2.1】有变量定义语句 `int a=3;b=4;` 则_____。

- A. 定义了 a、b 两个变量
- B. 定义了 a、b 两个变量, 并均已初始化
- C. 程序编译时出错
- D. 程序运行时出错

解题知识点: 变量定义语句规则。

解: 答案为 C。本题的解题要点是: 变量定义语句以分号表示结束。b 前面的分号标志着 int 型变量定义结束, 而变量 b 则未被定义。所以编译时出错: “Undefined symbol 'b' in function…”。

【例 2.2】表达式 `3.5+5%2*(int)(1.5+1.3)/4` 的值为_____。

- A. 4.5
- B. 4.0
- C. 3.5
- D. 4.2

解题知识点: 强制类型转换; 算术运算。

解: 答案为 C。本题的解题要点是: 做除法运算, 当两个操作数都是整型时, 结果为整型。因为 `5%2` 为 1, `(int)(1.5+1.3)` 为 2, `2/4` 为 0, 所以结果为 3.5。

【例 2.3】有变量说明语句 `float a,b;int k=0;` 合法的 C 语言赋值语句是_____。

- A. `a=b=8.5`
- B. `a=8.5,b=8.5`
- C. `k=int(a+b);`
- D. `k++;`

解题知识点: 强制类型转换; 赋值语句的书写规则。

解: 答案为 D。赋值语句的一般格式为: 变量名=表达式; , 语句以分号作为结束标志。选项 A 是赋值表达式, 不是语句; 选项 B 是逗号表达式; 选项 C 强制类型转换应将类型符用圆括号括起来, 即 `k=(int)(a+b);`; 选项 D 相当于 `k=k+1;`, 所以是合法的赋值语句。

【例 2.4】有变量说明语句 `int a=9;` 则执行完语句 `a+=a-=a*a;` 后, a 的值是_____。

- A. 9
- B. 144
- C. 0
- D. -144

解题知识点: 复合赋值语句的运算规则。

解: 答案为 D。赋值语句的结合性为“自右至左”, 复合赋值语句可以分解进行。可以将语句 `a+=a-=a*a;` 分解成如下两条语句: `a-=a*a;` `a+=a;`, 由第一条语句可以算出 a 的值为 -72, 由第二条语句算出 a 的值为 -144。

【例 2.5】有下面程序段:

```
int i,j;  
j=(i=32767,i+1);  
printf("i=%d,j=%d\n",i,j);
```

则运行结果是_____。

- A. `i=32767,j=32768`
- B. `i=32768,j=32768`
- C. `i=32767,j=-32768`
- D. `i=32767,j=32767`

解题知识点：逗号表达式；整型变量的取值范围。

解：答案为 C。题中赋值语句 $j=(i=32767,i+1)$ ；赋值号右侧的括号内是逗号表达式，执行过程：先用 32767 为 i 赋值，再计算 $i+1$ 的值，并用 $i+1$ 的值作为逗号表达式的值为 j 赋值。在内存中的表示形式为：

i:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
i+1:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

可见， $i+1$ 的值为 -32768。原因是整型变量在内存中占 2 个字节，最大可表示 32 767，再加 1 则产生溢出。

【例 2.6】下面程序段的运行结果是_____。

```
int x=5,y=5,z=5;
printf("x=%d,y=%d\n",x++,++y);
z++;++z;
printf("x=%d,y=%d,z=%d\n",x,y,z);
```

A. $x=6,y=6$

B. $x=5,y=5$

$x=6,y=6,z=6$

$x=6,y=6,z=7$

C. $x=6,y=5$

D. $x=5,y=6$

$x=6,y=6,z=6$

$x=6,y=6,z=7$

解题知识点：自增/自减运算符。

解：答案为 D。第一个输出语句中的两个输出项都是表达式。 x 是后加操作，是先用变量的当前值作为表达式的值，然后变量的值加 1； y 是前加操作，是先为变量的值加 1，加 1 后的值作为当前表达式的值；所以输出为 $x=5,y=6$ 。 $z++$ 和 $++z$ 都是独立的语句，都是完成加 1 的操作，所以在第二个输出语句中输出时 x 、 y 各做一次加 1 操作， z 做了两次加 1 操作。

【例 2.7】若有变量定义 $\text{int } x=6$ ；则下面语句的运行结果是_____。

```
printf("x1=%d,x2=%d\n",++x,x++);
```

A. $x1=7,x2=7$

B. $x1=8,x2=6$

C. $x1=8,x2=8$

D. $x1=7,x2=8$

解题知识点：自增/自减运算符；系统对函数参数的求值顺序。

解：答案为 B。本题的解题要点是：在 Turbo C 中系统对函数参数的求值顺序是自右而左。 printf 中要输出两个表达式的值 ($++x$ 和 $x++$)，先求右边的表达式 $x++$ 的值为 6 (先使用当前值，后加 1)， x 自增 1 为 7；再求左边的表达式 $++x$ 的值为 8 (先加 1，再使用)，所以输出 $x1=8,x2=6$ 。可能出现的错误是从左到右计算表达式而选择选项 A。

【例 2.8】在下列选项中，不合法的赋值语句是_____。

A. $n1=n2=n3=0$;

B. $k=i==j$;

C. $k=i\&\&j$;

D. $a=b+c=1$;

解题知识点：赋值语句的书写规则。

解：答案为 D。选项 A 相当于 $n3=0;n2=n3;n1=n2$ ；所以合法；选项 B 相当于 $k=(i==j)$ ；即赋值号右侧是关系表达式，当 i 和 j 相等时， k 赋值为 1，否则 k 赋值为 0，合法；选项 C 中赋值号右侧是逻辑表达式，当 i 和 j 均非零时 k 赋值为 1，否则 k 赋值为 0，也合法；选项 D 相当于