



21st CENTURY
规划教材

面向21世纪高等院校计算机系列规划教材
COMPUTER COURSES FOR UNDERGRADUATE EDUCATION

C语言程序设计学习指导与练习 (二级)

魏 文 主 编
陈燕敏 汪志勤 李文锦 副主编



科学出版社
www.sciencep.com



面向21世纪高等院校计算机系列规划教材
COMPUTER COURSES FOR UNDERGRADUATE EDUCATION

C 语言程序设计学习指导与练习

(二级)

魏 文 主编

陈燕敏 汪志勤 李文锦 副主编

科 学 出 版 社

北 京

内 容 简 介

本书针对浙江省计算机等级考试的大纲,结合作者多年的实际教学经验,总结归纳出各章节的知识点,对重点、难点等内容进行解释与剖析,并给出实例。书中提供了四套笔试综合练习题和详解,分析解题思路,给出应试技巧。针对上机考试,编写了上机练习精解,同时提供了一套上机考试模拟练习题及解答,使考生能够有的放矢地进行复习备考。书中还收集了历年浙江省计算机等级考试二级C的试卷,并给出参考答案。本书的所有例题和练习都在 Visual C++ 6.0 集成开发环境中通过调试运行。

本书非常适用于参加计算机等级考试二级C考生的考前辅导,也可作为各大专院校“C语言程序设计”课程的教学参考书。

图书在版编目(CIP)数据

C语言程序设计学习指导与练习. 二级/魏文主编. —北京:科学出版社, 2010
(面向21世纪高等院校计算机系列规划教材)

ISBN 978-7-03-028358-0

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

中国版本图书馆CIP数据核字(2010)第139799号

责任编辑:陈晓萍/责任校对:刘玉清

责任印制:吕春珉/封面设计:北大彩印

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

骏杰印刷厂印刷

科学出版社发行 各地新华书店经销

*

2010年8月第一版 开本:787×1092 1/16

2010年8月第一次印刷 印张:14

印数:1—4 500 字数:329 000

定价:24.00元

(如有印装质量问题,我社负责调换(环伟))

销售部电话 010-62134988 编辑部电话 010-62138978-8003

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

C 语言作为一种结构化、模块化的程序设计语言，具有简洁、灵活、实用、高效、移植性好等特点，是目前全国各高等院校理工科学生的主要必修课程，而且在程序员考试和各类计算机等级考试中也占有重要地位。

为了帮助学生尽快掌握 C 语言，提高 C 语言编写程序、解决实际问题的能力，同时为配合浙江省计算机等级考试的复习，我们组织了具有多年本科教学经验的老师编写了本书。全书主要包括以下几个章节。

第 1 章综合复习 C 语言程序设计中各教学章节的知识点，并归纳了教学重点、难点以及初学者易犯的错误，用实例进行分析；整理总结出计算机等级考试中常见的一些算法。第 2 章包含四套综合练习的详细解答。针对一些基本概念、常见题型和算法，给出解题思路和详细的解题过程，有助于提高考生的解题能力。第 3 章精选上机试题，给出解题步骤，包含上机考试的常见考点，同时对上机考试系统的使用进行了简要的说明。从第 4 章开始，给出了上机考试的模拟练习题及答案、历年浙江省计算机等级考试二级 C 笔试试题及参考答案，供考生进行练习，提高应试能力。

本书内容精练，重点突出，习题丰富，不仅可以作为计算机等级考试复习用书，也可以作为各类高校非计算机专业“C 语言程序设计”课程的教学辅导书。

本书由魏文任主编，陈燕敏、汪志勤、李文锦任副主编。

由于时间仓促，编者水平有限，书中难免有错误或不当之处，敬请广大读者批评指正，如需联系编者可发邮件至 cprogram2000@hotmail.com。

魏 文

2010 年 6 月

目 录

第 1 章 C 语言要点、重点、难点分析	1
1.1 C 语言程序设计基础知识	1
1.2 数据类型及其运算	2
1.3 控制结构与语句	8
1.4 数组	20
1.5 指针	37
1.6 函数	45
1.7 编译预处理	51
1.8 结构体	54
1.9 共用体、枚举和位运算	60
1.10 文件	63
第 2 章 综合练习详解	69
2.1 综合练习一详解	69
2.2 综合练习二详解	81
2.3 综合练习三详解	94
2.4 综合练习四详解	107
第 3 章 上机练习精解	124
3.1 程序填空题	124
3.2 程序改错题	129
3.3 程序设计题	136
3.4 浙江省高校计算机等级考试二级 C 上机考试说明	144
第 4 章 上机考试模拟练习	147
4.1 程序填空题	147
4.2 程序改错题	151
4.3 程序设计题	156
第 5 章 历年浙江省计算机等级考试二级 C 笔试试题	161
5.1 2007 年春浙江省高等学校计算机等级考试试卷 (二级 C)	161
5.2 2007 年秋浙江省高等学校计算机等级考试试卷 (二级 C)	167
5.3 2008 年春浙江省高等学校计算机等级考试试卷 (二级 C)	173
5.4 2008 年秋浙江省高等学校计算机等级考试试卷 (二级 C)	178
5.5 2009 年春浙江省高等学校计算机等级考试试卷 (二级 C)	184
5.6 2009 年秋浙江省高等学校计算机等级考试试卷 (二级 C)	191
5.7 2010 年春浙江省高等学校计算机等级考试试卷 (二级 C)	196

第 6 章 上机考试模拟练习参考答案	202
6.1 程序填空题	202
6.2 程序改错题	202
6.3 程序设计题	203
第 7 章 历年浙江省计算机等级考试二级 C 笔试题参考答案	206
7.1 2007 年春浙江省高校计算机等级考试参考答案 (二级 C)	206
7.2 2007 年秋浙江省高校计算机等级考试参考答案 (二级 C)	206
7.3 2008 年春浙江省高校计算机等级考试参考答案 (二级 C)	207
7.4 2008 年秋浙江省高校计算机等级考试参考答案 (二级 C)	208
7.5 2009 年春浙江省高校计算机等级考试参考答案 (二级 C)	209
7.6 2009 年秋浙江省高等学校计算机等级考试参考答案 (二级 C)	210
7.7 2010 年春浙江省高等学校计算机等级考试参考答案 (二级 C)	211
附录	212
附录 1 C 语言的关键字	212
附录 2 常用字符的 ASCII 码对照表	212
附录 3 C 语言运算符的优先级与结合性	213
附录 4 常用库函数	214
参考文献	216

第 1 章 C 语言要点、重点、难点分析

1.1 C 语言程序设计基础知识

1. 知识要点

(1) 计算机程序设计语言

程序设计语言主要分成三大类：机器语言、汇编语言和高级语言。只有用机器语言编写的程序，计算机才能够直接执行。汇编语言编写的程序需要经过汇编程序，高级语言编写的程序需要经过编译或解释程序，最后才能生成机器语言程序，在计算机上运行。

C 语言属于高级语言，由于其特有的优点，它既可用于编写系统软件，也可用于编写应用软件。

(2) C 语言程序设计的基本知识

C 语言具有语言简洁、使用灵活、运算符丰富、允许直接访问物理地址、生成目标代码质量高以及编写的程序移植性好等特点。

用 C 语言编写的程序由函数构成。一个 C 源程序有且只有一个 `main` 函数和若干个（0、1 或多个）其他函数。函数包括函数头部和函数体两部分。函数的头部包括函数名、函数类型、函数属性、函数参数及其类型等组成。函数体放在一对大括弧中，分成声明部分（变量、函数声明等）和执行部分（若干执行语句）。一个 C 语言的程序可以由多个文件组成。

一个 C 程序总是从 `main` 函数开始执行，并在 `main` 函数中结束运行。

(3) C 语言程序在计算机上的执行过程

首先编辑 C 的源程序，并以扩展名为 `.c` 的文件名保存。然后必须经过编译程序对源程序进行编译，生成扩展名为 `.obj` 的二进制目标程序。最后，还需要将该程序包括的所有目标程序、系统库文件进行连接，生成一个扩展名为 `.exe` 的可执行程序。

在 Microsoft Visual C++ 集成开发环境中可以方便地进行 C 源程序的编辑、编译、连接和运行。

2. 重点及难点

(1) 重点

C 源程序的构成和运行过程。

(2) 难点

C 源程序在计算机上的执行。

C 源程序在计算机上调试的过程中，可能出现两种错误：语法错误和逻辑错误。没

有语法错误的程序不一定就是正确的程序,可能还存在逻辑错误。因此,还必须经过运行测试,根据程序运行结果的正确与否来判断程序是否正确。

3. 例题分析

例 1.1 以下叙述中正确的是_____。

- A. 高级语言就是机器语言
- B. 汇编语言程序、高级语言程序都是计算机程序,但只有机器语言程序才是计算机可以直接识别并执行的程序
- C. C 语言因为具有汇编语言的一些特性,所以是汇编语言的一种
- D. C 源程序经过编译、连接,若正确,执行后就能得到正确的运行结果

答:高级语言接近于自然语言和数学语言,计算机不能直接识别,但使用上比机器语言和汇编语言更方便、简单。C 语言属于高级语言。C 源程序如果编译、连接正确后,运行不一定能得到正确的运行结果,因为还可能存在逻辑错误。本题答案为 B。

例 1.2 以下叙述中错误的是_____。

- A. main 函数在 C 程序中必须有且只有一个
- B. C 程序的基本组成单位是函数
- C. C 程序的执行从 main 函数开始,所以 main 函数必须放在程序最前面
- D. C 程序中各个函数位置可以任意

答:一个 C 程序由一个或多个函数构成,其中 main 函数有且只能有一个。不论 main 函数放在什么位置,一个 C 程序总是从 main 函数开始执行。本题答案为 C。

例 1.3 以下叙述中正确的是_____。

- A. C 语言编写的源程序经过编译后生成可执行文件
- B. C 的源程序可以直接在计算机上运行
- C. C 语言的函数不可以单独进行编译
- D. C 语言中的每条可执行语句最后都要被转换成二进制的机器指令

答:C 的源程序不能在计算机上直接运行,首先必须经过编译,可以生成一个二进制的.obj 目标程序,然后与系统库文件等其他程序连接后,才能生成一个可直接运行的.exe 可执行文件。C 的函数可以单独进行编译,生成二进制的.obj 目标程序。本题答案为 D。

1.2 数据类型及其运算

1. 知识要点

(1) C 语言标识符

C 语言源程序中可以用到的字符集包括英文大小写字母(共 52 个),数字 0~9(共 10 个),还有+、-等其他 30 个字符。

标识符是一个名字,是给程序中的变量、符号常量、数组、结构体、函数或文件等所起的名字。C 语言的标识符包括关键字、预定义标识符和用户标识符。关键字是系统指定的标识符,代表固定的含义,不能另作它用。预定义标识符在 C 语言中也是有特定

的含义,如C语言提供的库函数名称。C语言语法允许将这类标识符另作它用,但为了避免误解,建议大家不要将这些标识符另作它用。用户标识符是由用户根据需要自己定义的,必须以字母或下划线开头,由字母、数字、下划线组成的字符序列。

在定义标识符时,尽量做到“见名知义”。需要注意的是,C语言区分大小写字母,即大写字母和小写字母被认为是两个不同的字符。

(2) 数据类型

C语言有丰富的数据类型,不同数据类型的数据在存储(占内存字节数)、表示(取值范围),和对其可进行的操作上都各不相同。C语言的数据类型包括基本类型(整型、字符类型、实型、枚举类型)、构造类型(数组、结构体类型、共用体类型)、指针类型和空类型。

基本数据类型中,整型又可细分成短整型(short)、基本整型(int)和长整型(long),实型可分成单精度(float)类型和双精度(double)类型。整型数据以补码形式存放在内存中,实型数据按指数形式存储,字符(char)类型在内存中则以其对应的ASCII码值来存放。

不同的数据类型在内存中存放所占的长度(即字节数)不同。以Visual C++ 6.0系统为例,短整型数据占2个字节,基本整型和长整型数据都是占4个字节。单精度类型数据占4个字节,双精度类型数据占8个字节。

(3) 常量

在程序运行过程中其值不变的量称为常量。常量可以分为整型常量、实型常量、字符常量、字符串常量和符号常量。

整型常量可以是十进制、八进制及十六进制表示的整数。十进制整数是带符号(+、-)的包含0~9的一位或多位十进制数,但第一位不能是0。八进制整数是带符号(+、-)的以0开头、包含0~7的一位或多位八进制数。十六进制整数是带符号(+、-)的以0x或0X开头、包含0~9、a~f的一位或多位十六进制数。例如,1、-30、0是整型常量,0127、-0254是八进制常量,0x2af、0Xc2、-0x12是十六进制常量。如果在整型常量后面加上字母l或L,则表示它是一个长整型常量。

实型常量有两种表示方式:小数形式和指数形式。小数形式由符号(+、-)、整数部分、小数点和小数部分组成,其中整数部分和小数部分不可同时省略,小数点不可省略。例如,-1.2、.23、12.、45.232等都是合法的实型常量。指数形式由符号(+、-)、小数部分、指数符号e或E和指数部分组成,其中小数部分不可省略,指数部分必须为整数。例如,1.23e-5、2E3、-4.213e2等都是合法的实型常量。在Visual C++ 6.0中,实型常量都是double类型,在实型常量后面加f,则为float类型,如3.14f。

字符常量是用单引号括起来的一个字符,其中单引号只是起到定界作用并不表示字符本身。C语言中还有一些字符无法表示(如换页、退格等),则采用转义字符来表示。转义字符以反斜杠(\)开头,后面跟一些特殊字符或数字,将反斜杠(\)后面的字符或数字转换成其他意义的字符。常用的转义字符如表1.1所示。

对于字符A(ASCII码值为65),可以有不同的表示方法。'A'、'\101'、'\x41'都代表ASCII码值为65的字符常量。

需要掌握以下一些常用字符之间的转换。

① 英文大小写字母之间的转换,对应大小写字母之间 ASCII 码值相差 32(即'a'-'A')。

② 数字字符'0'~'9'、大写字母'A'~'Z'、小写字母'a'~'z'互相之间 ASCII 码值依次递增 1。

③ 数字 0~9 加上 48 (即字符'0'),转换成对应的数字字符'0'~'9'。例如,字母'G'加 32 即转换成字母'g';字母'a'加 4 即为字母'e';数字 1 加上'0'字符就转换成数字字符'1';数字字符'1'加上 8 就转换成了数字字符'9'。

表 1.1 转义字符

转义字符	意 义	ASCII 码值 (十进制)
\a	响铃 (BEL)	7
\b	退格,即输出位置移到前一列 (BS)	8
\n	换行,即输出位置移到下一行开头 (LF)	10
\r	回车,即输出位置移到本行开头 (CR)	13
\t	水平制表,即输出位置移到下一个输出区 (HT)	9
\\	反斜杠字符	92
\?	问号字符	63
\'	单引号字符	39
\"	双引号字符	34
\0	空字符 (NULL)	0
\ddd	ASCII 码值为 1~3 位八进制数所代表的字符	
\xhh	ASCII 码值为 1~2 位十六进制数所代表的字符	

字符串常量是用双引号括起来的一串字符,其中双引号只是起到定界作用。C 语言中,字符串常量在内存中存放时,系统自动在字符串的末尾加一个“串结束标志”,即 ASCII 码值为 0 的空字符'\0'。因此,包含 n 个字符的字符串常量在内存中要占 (n+1) 个字节的存储空间。空串""表示字符串中只有一个空字符'\0'。注意区分'a'和"a"的不同之处。

符号常量是指在程序中用一个标识符代表一个常量。定义的形式如下:

```
#define 符号常量名 常量
```

系统进行预处理时,将程序中的符号常量名替换成对应的常量(即宏展开)。

(4) 变量

在程序运行过程中其值可变的量称为变量。C 语言规定,变量必须先定义后使用。变量定义的一般形式如下:

```
数据类型 变量名;
```

在变量定义的同时给变量赋值,即为变量的初始化。例如, `int a=3, b=1;`。

需要注意的是,一般变量定义后若未初始化,则此时变量的值为不确定。可以通过变量初始化、赋值语句和调用输入函数等三种方式给变量赋值。

(5) 数据类型的转换

不同数据类型的数据混合运算时,需要转换数据类型。C 语言数据类型的转换分自动转换和强制转换两种。

当不同数据类型混合运算时由系统进行自动转换。

① 级别低的数据类型自动向高级别的类型转换。字符型 (char) 和短整型 (short) 必须先转换成整型 (int)，单精度型 float 必须先转换成双精度型 double。类型转换的优先级为 $\text{int} \rightarrow \text{unsigned} \rightarrow \text{long} \rightarrow \text{double}$ 。

② 在赋值运算中，赋值号右边的表达式类型自动转换为赋值号左边变量的类型。

③ 用户自定义函数中，由 return 返回的表达式类型自动转换为函数头部的函数类型。

C 语言中还有强制类型转换运算符，其一般形式如下：

(类型说明符)(表达式)

将表达式的值强制转换成所需的数据类型。

(6) 运算符、优先级、结合性和表达式

C 语言中提供了丰富的运算符，分为算术运算符、关系运算符、逻辑运算符、条件运算符、赋值运算符、逗号运算符、移位运算符和位逻辑运算符等。

对于运算符，需要掌握参与运算的操作数个数 (单/双目)、运算符的优先级和结合性。C 语言的运算符中，优先级为 2 级的都是单目运算符，条件运算符是三目运算符，其余都是双目运算符。所有运算符分成 15 个优先级。常用运算符 (除其中的单目运算符以外) 的优先级按算术运算符 > 关系运算符 > 逻辑运算符排列。C 语言中运算符的结合性分为左结合性 (从左到右计算)、右结合性 (从右到左计算) 两种，其中单目、三目运算符都是右结合性，除赋值及复合赋值运算符之外的所有双目运算符都是左结合性。

由常量、变量、函数调用以及按 C 语言语法规则用运算符把运算数据连起来的式子都是 C 的合法表达式。在计算表达式时，首先计算优先级高的运算符，对于同一优先级的运算符则根据其结合性决定计算次序。在表达式的计算过程中，参加运算的两个操作数在计算前先自动转换成同一数据类型，再运算求值。

对于常用的运算符和表达式需要注意以下问题。

① 自增、自减运算符 (++ 和 --) 只能用于单个变量 (或数组元素)。前缀自增 (减) 运算先对变量加 (减) 1，然后此时变量的值 (加/减 1 之后的值) 即为表达式的值；后缀自增 (减) 运算先将此时变量的值 (未加/减 1 之前的值) 作为表达式的值，然后对变量加 (减) 1。由此可见，不论前缀还是后缀自增 (减) 运算，计算后只有表达式的值不同，变量的值都是一样的。

② 求余运算符 (%) 要求两个操作数都是整型数据。如果实型数据要进行求余运算，则必须对其进行强制类型转换，转换成整型。

③ 如果除法运算 (/) 的两个操作数都是整数，则其结果取整。例如，表达式 $2/5$ 的结果为 0。如果不希望结果取整 (即需要结果为实数)，则需要将其中一个操作数转换成实型。

④ 使用求余 % 运算符，根据余数是否是零可以判断该整数是否能被另一个整数整除。对于整数 n，计算 $n\%10$ 即可得到 n 的个位数字；将 $n\%10$ 和 $n/10$ 配合起来使用，可以得到整数 n 各个位上的数字；采用 $n\%2$ 和 $n/2$ ，可以将十进制正整数 n 转换成二进制；同理，采用 $n\%8$ 和 $n/8$ (或 $n\%16$ 和 $n/16$)，可以将十进制正整数 n 转换成八进制 (或十六进制)。

⑤ 关系运算符为左结合性，运算结果为 1 (真) 或者 0 (假)。需要注意的是数学上

成立的不等式作为 C 语言的表达式时,却可能出现相反的运行结果。例如,数学上 $6 > 4 > 2$ 是成立的,但作为 C 语言的关系表达式 $6 > 4 > 2$ 的计算结果却是 0 (假)。根据关系运算符的左结合性,先计算 $6 > 4$,结果为 1 (真),然后再计算 $1 > 2$,结果为 0 (假)。因此,在写类似的关系表达式时,一定要将两个关系运算分开写,并通过逻辑与运算连接起来。例如, $6 > x > 2$ 要写成 $6 > x \&\& x > 2$ 。

⑥ 在 $\&\&$ 和 $\|\|$ 运算符组成的逻辑表达式中,为提高程序的执行效率, C 语言规定,只对能够确定整个表达式值的最少数目的子表达式进行计算,即当计算出某个子表达式后就可以确定整个逻辑表达式的值,后面的子表达式就不再计算了。

在计算 (表达式 A) $\&\&$ (表达式 B) 时,先计算表达式 A 的值。若表达式 A 的值为零 (假),则不必进行表达式 B 的计算,就可得出整个逻辑表达式值为零。在计算 (表达式 A) $\|\|$ (表达式 B) 时,先计算表达式 A 的值。若表达式 A 的值为非零 (真),则不必进行表达式 B 的计算,就可得出整个逻辑表达式值为 1。

⑦ 在进行移位运算或位运算时,先要将两个操作数都转化成二进制,然后再按位进行计算。

(7) 数据的输入/输出

C 语言中没有输入/输出语句,所有的输入/输出都由系统提供的库函数完成。字符型数据输入/输出常用 `getchar()`/`putchar()` 函数,格式输入/输出函数 (`scanf()`/`printf()`) 可以输入/输出各种类型的数据。

getchar()函数:从标准输入设备读取一个字符。使用方式: `char c; c=getchar()`。

putchar()函数:向标准输出设备输出一个字符。使用方式: `putchar(c)`,其中 `c` 表示任意一字符常量或字符变量。

scanf()函数:按照格式控制字符串所给定的输入格式,将输入数据按地址列表存入指定的存储单元。使用方式: `scanf (格式控制字符串, 地址列表)`,其中地址列表可以是变量 (或数组元素) 的地址、字符串的首地址等。

printf()函数:按照格式控制字符串所给定的输出格式,把输出项表中各表达式的值输出到显示器。使用方式: `printf (格式控制字符串, 输出项表)`,其中输出列表是若干个 C 的表达式。

格式控制字符串由两部分组成:格式说明符和普通字符。格式说明符一般格式如下:

`%[宽度]格式说明符号`

`scanf()`函数中的普通字符必须按原样输入, `printf()`函数中的普通字符要按原样输出。

2. 重点及难点

(1) 重点

各种常量的表示、运算符的优先级和结合性、表达式的求值。整除 (`/`) 和求余 (`%`) 运算符的结合,解决十进制与二进制、十六进制之间的转换,求整数的各位数等问题。

注意:“`=`”与“`==`”、“`++a`”与“`a++`”的区别;逻辑表达式的计算特点; C 语言关系表达式与数学中的关系运算的不同。

(2) 难点

根据运算符的优先级、结合性和不同数据类型,计算 C 表达式的值。

3. 例题分析

例 1.4 设 m 、 n 、 a 均为 int 型变量, 则执行语句 $m=n=0; a=++m||++n;$ 后, m 、 n 和 a 的值为_____。

- A. 1 1 1 B. 1 0 1 C. 0 0 0 D. 0 1 1

答: 根据逻辑表达式的计算特点 (参见本节的知识要点), 先计算 $++m||++n$, 然后将计算结果赋给变量 a 。 $++m$ 为自增前缀, 先做 $m=m+1$, $++m$ 表达式的值为 1。在进行逻辑或运算时, 由于 $++m$ 的值为真, $++n$ 就不执行, 可以得出表达式 $++m||++n$ 的值为 1。本题答案为 B。

例 1.5 执行下面程序段后, 变量 i 和 k 的值为_____。

```
int i=0, k=10;
if (i=8) k--;
else k++;
```

- A. 8 10 B. 0 9 C. 8 9 D. 0 11

答: 根据双分支 if 语句的执行过程 (参见下节的知识要点), 先计算表达式 $(i=8)$ 的值。由于 $i=8$ 是赋值表达式, 因此 i 的值为 8, 赋值表达式的值也是 8, 即为真, 接下来执行 $k--$ 。本题答案为 C。

思考: 如果将 “ $i=8$ ” 改为 “ $i==8$ ”, 那么结果是多少?

例 1.6 判断字符变量 c 是数字字符的逻辑表达式为_____。

- A. $0 \leq c \leq 9$ B. $"0" \leq c \leq "9"$
C. $'0' \leq c \leq '9'$ D. $'0' \leq c \&\& c \leq '9'$

答: 0、9 是整型常量, “0”、“9”是字符串常量, ‘0’、‘9’是字符常量 (参见本节关于常量的知识要点)。根据关系表达式的运算规则 (参见本节的知识要点), 不论 c 取任何值, 表达式 $'0' \leq c \leq '9'$ 的值均为真。本题答案为 D。

例 1.7 程序填空。输出所有的水仙花数。水仙花数是指一个三位数, 其各位数字的立方和等于其本身, 如 $153=1*1*1+5*5*5+3*3*3$ 。

```
#include <stdio.h>
#include<math.h>
void main()
{ int k,a,b,c;
  for(k=100;k<1000;k++)
  { a=k/100;
    _____【1】_____;
    c=k%10;
    if ( _____【2】_____)
      printf("%d is a Armstrong number!\n",k);
  }
}
```

答: 通过整除/和求余%运算符得到整数的各位数字 (参见本节关于常量的知识要点)。本题【1】的答案为 $b=i\%100/10$ 或者 $b=i/10\%10$ 。本题【2】的答案为 $a*a*a+b*b*b+$

$c*c=c=k$ 。

例 1.8 程序填空。将输入的十进制整数 n 通过函数 `Conversion` 转换为十六进制数，并将转换结果以字符串形式输出。（例如，输入十进制数 125，将输出十六进制 7d。）

```
# include<stdio.h>
char ToChar(int n)
{
    if(n<10) return '0'+n;
    else _____ 【1】 _____ ;
}
int Conversion(int x,char *s)
{ int i=0;
  while(x!=0)
  { _____ 【2】 _____ ;
    x/=16;    i++;
  }
  return i-1;
}
void main()
{
    int i,k,n;
    char str[80];
    scanf("%d",&n);
    k=Conversion(n,str);
    for (i=0;i<=k;i++) printf("%c",str[k-i]);
}
```

答：本题涉及两个知识点：整数 10~15 转换成是字符 'a'~'f'、通过整除/和求余 % 运算符进行进制的转换（参见本节关于常量的知识要点）。本题【1】的答案为 `return 'a'+n-10`。本题【2】的答案为 `s[i]=ToChar(x%16)`。

1.3 控制结构与语句

1. 知识要点

(1) C 语言的基本语句

1) 表达式语句：由表达式加上分号 “;” 组成。其一般形式为

表达式;

执行表达式语句就是计算表达式的值。例如，

`x=y+z; a=520;`

2) 函数调用语句：由函数名、实际参数加上分号 “;” 组成。其一般形式为

函数名(实际参数表);

执行函数语句就是调用函数并把实际参数赋予函数定义中的形式参数，然后执行被

调函数体中的语句。例如，

```
printf("helloworld");
```

3) 控制语句：用于控制程序的流程，以实现程序的各种结构方式。它们由特定的语句定义符组成。C语言有九种控制语句，可分成以下三类。

① 条件判断语句：if 语句、switch 语句。

② 循环执行语句：do while 语句、while 语句、for 语句。

③ 转向语句：break 语句、continue 语句、goto 语句（此语句尽量少用，因为这不利结构化程序设计）、return 语句。

上述语句的基本语法和语句的执行过程将在选择结构和循环结构中进行说明。

4) 复合语句：把多个语句用括号{}括起来组成的一个语句称复合语句。复合语句在语法上相当于一语句的作用，常用于 if 的分支语句、循环结构的循环体语句等。例如，

```
{
    x=y+z;
    a=b-c;
}
```

是一条复合语句。复合语句内的各条语句都必须以分号“;”结尾；此外，在括号“}”外不能加分号。

5) 空语句：只有分号“;”组成的语句称为空语句。空语句是什么也不执行的语句。

(2) C语言程序的基本结构

1) 顺序结构：顺序结构是程序中最简单的一种结构。顺序结构的程序主要由赋值语句及数据输入、输出等函数调用语句共同组成，程序执行时，按照语句的排列次序依次执行。

2) 选择结构：在程序设计时，如果需要根据某些条件作出判断，决定不同的处理方式，可以使用选择结构完成。if 语句和 switch 语句可以实现选择结构。

① if 语句：if 语句的三种形式如下。

(a) 单分支。

```
if(表达式) 语句1;
```

执行过程：先计算表达式的值，当表达式的值不等于零时，则执行分支语句，否则直接执行 if 语句的后续语句。

(b) 双分支。

```
if(表达式) 语句1;
else 语句2;
```

执行过程：先计算表达式的值，当表达式的值不等于零时，则执行分支语句 1，否则执行分支语句 2。

(c) if 语句的嵌套形式有如下几种。

● else-if 嵌套的一般形式：

```
if(表达式1) 语句1
else if(表达式2) 语句2
:
else if(表达式n-1) 语句n-1
```

else 语句 n

执行过程：先计算表达式 1 的值，当表达式 1 的值不等于零时，则执行分支语句 1，整个 if 语句执行结束；否则计算表达式 2，当表达式 2 的值不等于零时，则执行分支语句 2……，只有当表达式 1、表达式 2……表达式 n-1 的值都为零时，执行 else 的分支语句 n。

- 双分支嵌套：

```
if(表达式 1)
    if(表达式 2) 语句 1
    else 语句 2
else
    if(表达式 3) 语句 3
    else 语句 4
```

执行过程：先计算表达式 1 的值，当表达式 1 的值不等于零时，则计算表达式 2；当表达式 2 的值不等于零时，则执行分支语句 1，否则执行 else 的分支语句 2；当表达式 1 的值等于零时，则计算表达式 3；当表达式 3 的值不等于零时，则执行分支语句 3，否则执行 else 的分支语句 4。

使用 if 语句编写选择结构的程序时，需要注意以下几点。

- if (表达式) 后面不要随意加分号，因为分号也是一个语句，空语句。这样写意味着如果表达式为真，则执行空语句，导致语句的逻辑发生错误。

- 复合语句的使用：即如果在条件成立的情况下，需要完成的工作都由多个语句共同组成，这时需要把这些语句组成一个复合语句，即用 { } 将这些语句括在一起。条件成立则都执行，否则都不执行。条件不成立的情况下亦同。

- if 与 else 的匹配：if 语句有多层嵌套时，可能出现多个 if 与 else，else 与 if 的匹配规则是：else 与前面最靠近它的、可见的、还没有与其他 else 匹配过的 if 匹配。

- 在分支条件较多的情况下，if 的嵌套层次会比较多，程序可读性比较差。

② switch 语句。switch 语句也可以用来处理多分支结构的情况。

switch 语句的一般形式：

```
switch(表达式) {
    case 常量表达式 1: 语句组 1
    case 常量表达式 2: 语句组 2
        :
    case 常量表达式 n: 语句组 n
    default: 语句组 n+1
}
```

执行过程：先计算表达式的值，若表达式值与 case 的常量表达式值 i 相等时，就从该 case 进入，执行完后面的所有语句组，直到遇到 break 语句或者 switch 的结束括号跳出 switch 语句。若表达式值与所有 case 的常量表达式值都不相等，则从 default 进入，执行语句组 n+1。

使用 switch 语句编写选择结构的程序时，需要注意以下几点。

- switch 语句中判断的表达式的值必须是整型或者字符型的。

- **case** 后面的表达式必须是常量（即表达式中不含变量），而且必须是整型或者字符型的。各个 **case** 相互之间没有先后次序，各个常量表达式的值必须唯一。**case** 的语句组可以为空或者多条语句。多个 **case** 可共用一组语句。

- **default** 部分可以省略，也可以出现在其他位置。**default** 语句的作用：当表达式与所有的 **case** 表达式的值都不匹配时，才执行 **default** 后面的语句组。

- **break** 语句在 **switch** 中的作用：强行跳出 **switch** 语句，转到 **switch** 语句的后续语句执行。

3) 循环结构。编程时，遇到有规律的重复计算或操作，可以将这些过程写成循环结构。C语言中可用于实现循环结构的语句有 **while**、**do-while** 和 **for**。

① **while** 语句的一般形式：

```
while(表达式)
    语句 s
```

执行过程：计算表达式的值，若表达式的值为真（非零）时，重复执行语句 **s**，否则结束循环，转到 **while** 语句的后续语句执行。

② **do-while** 语句的一般形式：

```
do
    语句 s
while (表达式);
```

执行过程：先执行循环体语句 **s**，计算表达式的值，若表达式值非零，继续执行循环，否则循环执行结束，转到 **do-while** 语句的后续语句执行。

使用 **while** 语句和 **do-while** 语句编写循环结构的程序时需要注意以下几点。

- （表达式）是循环条件，也就是在表达式为真时进行循环。语句 **s** 是循环体，也就是循环做什么，即需要重复执行的语句。

- 循环体中必须有使循环条件向不满足的方向变化的语句，否则将陷入死循环。

- 复合语句的使用：即如果在循环条件成立的情况下，需要重复完成的工作由多个语句共同组成，这时需要把这些语句组成一个复合语句。

- **do-while** 语句和 **while** 语句一般情况下可以通用。它们不同之处在于：**do-while** 语句是先执行后判断，循环体至少执行一次。**while** 语句是先判断后执行，因此，循环体有可能一次也不执行。

③ **for** 语句的一般形式：

```
for( 表达式 1; 表达式 2; 表达式 3 )
    语句 s
```

执行过程如下：

(a) 先计算表达式 1。

(b) 计算表达式 2，当其值为真（非零），执行循环体语句 **s** 后转 (c)，否则循环结束，转到 **for** 的后续语句执行。

(c) 计算表达式 3，然后转 (b) 继续。

使用 **for** 语句编写循环结构的程序时需要注意以下几点。

- 执行 **for** 语句时，先执行表达式 1，且只执行一次，一般用于循环的准备，为有关变量赋初值。