第1章 概览

**本章课程内容：**

C的历史和特性。

编写程序所需步骤。

关于编译器和链接器的一些知识。

C的标准。

**程序清单：**

101CcodeExample

102Concrete

**知识点：**

C语言的起源

**总结：**

C是一种强大、简洁的编程语言。之所以流行是因为它提供了有用的编程工具和对硬件良好的控制，还因为C程序在一个系统向另一个系统移植方面比大多数程序更容易。

C是一种需要编译的语言。C编译器和链接器是将C语言源代码转换为可执行代码的程序。

用C编程可能很费力、困难并让你感到灰心，但这一工作也可能让你着迷、兴奋和感到满意。希望你也能像我们一样，沉醉于用C进行编程。

**复习题：**

1.就编程而言，可移植性表示什么？

一个程序的可移植性好，就是指它的源代码不经修改就可以在多种不同的计算机系统上编译成可以成功运行的程序。

2.解释源代码文件、目标代码文件和可执行文件之间的区别？

源代码文件中包含着程序员使用计算机编程语言编写的代码。目标代码文件包含着机器语言代码。可执行文件包含着组成可执行程序的全部机器语言代码。

3.编程的7个主要步骤是什么？

A．定义程序目标。

B．设计程序。

C．编写程序代码。

D．编译程序。

E．运行程序。

F．测试和调试程序。

G．维护和修改程序。

4.编译器的任务是什么？

编译器把源代码转换成机器语言代码，也称对象代码。

5.链接器的任务是什么？

链接器把多个来源（例如：已编译的源代码、库代码、和启动代码）的目标代码连接成一个单独的可执行程序。

**编程练习：**

1001.您刚刚被MacroMuscle有限公司聘用。该公司要进入欧洲市场，需要一个英寸转换为厘米的程序（1英寸=2.54CM）.他们希望建立的该程序可提示用户输入英寸值。你的工作是定义程序目标并设计程序（编程过程的第1步与第2步）。

第2章 C语言概述

**本章课程内容：**

运算符： = （被赋值）。

函数： main(),printf()。

编写一个简单的C程序。

创建整型变量，为其赋值，并在屏幕上显示该值。

换行字符。

如何在程序中加入注释，建立包含多个函数的程序，以及找出程序中的错误。

理解什么是关键字。

**程序清单：**

201first

202fathm\_ft

203two\_func

204nogood

205stillbad

**C语言的关键字列表**

|  |  |  |  |
| --- | --- | --- | --- |
| auto | **enum** | *restrict* | unsigned |
| break | extern | return | **void** |
| case | float | short | **volatile** |
| char | for | **signed** | while |
| **const** | goto | sizeof | *\_Bool* |
| continue | if | static | *\_Complex* |
| default | *inline* | struct | *\_Imaginary* |
| do | int | switch |  |
| double | long | typedef |  |
| else | register | union |  |

**知识点：**

头文件简介

main()函数

声明

printf()函数

return语句

多个函数声明

总结：

C语言程序是由一个或者多个函数组成。每一个C程序都必须包含一个名为main()的函数，因为当程序开始时要调用该函数。一个简单的函数结构如下：函数头后面紧跟着一个开始花括号，后面是构成函数体的语句，然后是起终止作用的结束花括号。

每个C语句都是一个针对计算机的指令，并以一个分号作为结束标志。声明语句为变量指定一个名字并指明该变量中存储的数据类型。变量名是标识符的例子。赋值语句把值赋给变量，或者更一般地说，是把值赋给存储区域。函数调用语句会导致所指定函数的执行。当被调函数执行完毕之后，程序会返回到函数调用之后的语句继续进行。

printf()函数用于输出语句和变量的值。

一门语言的语法是一套规则，用于管理这种语言中的合法语句组织在一起的方式。语句的语义就是它所表达的意思。编译器可以帮助您发现语法上的错误，但是程序里的语义错误只有在编译完成之后才能从程序的行为中表现出来。检测语义错误可能包含跟踪程序的状态，即程序每执行一步之后所有变量的值。

关键字是C语言的词汇。

复习题：

1.如何称呼C程序的基本模块？

他们被称为函数。

2.什么是语法错误？给出它的一个英语例子和C语言例子。

语法错误就是指违背了如何把语句或程序放置在一起的规则。这是英语中的一个例子：“Me speak English good.”下面是C语言中的一个例子：

printf “Where are the parentheses?”;

3.什么是语义错误？给出它的一个英语例子和C语言例子。

语义错误是指含义上的错误。这是英语中的一个例子：“This sentence is excellent Italian.”下面是C语言中的一个例子：

thrice\_n = 3 + n;

4.Indiana Sloth 已经编好了下面的程序，并想征求您的意见。请帮忙评定。

include studio.h

int main{void} /\* 该程序可显示出一年中有多少周 /\*

(

int s

s: = 56;

print(There are a weeks in a year.):

return 0;

正确如下：

#include <stdio.h>

int main(void)

{

int s;

s = 56;

printf(“There are %d weeks in a year.\n”, s);

return 0;

}

5.假设下面的每一个例子都是某个完整程序的一部分，他们每个将输出什么结果？

a.

printf(“Baa Baa Black Sheep.”);

printf(“Have you any wool?\n”);

b.

printf(“Begone!\nO creature of lard!”);

c.

printf(“What?\nNo/nBonzo?\n”);

d.

int num;

num = 2;

printf(“%d + %d = %d”, num, num, num + num);

答：

a.

Baa Baa Black Sheep.Have you any wool?

b.

Begone!

O creature of lard!

c.

What?

No/nBonzo?

d.

2 + 2 = 4

6.下面那几个是C的关键字？main, int, function, char, =

关键字是int与char。main是函数名，=是被赋值运算符。

7.如何以下面的格式输出words 和lines 的值：“There were 3030 words and 350 lines” ? 这里，3020和350代表两个变量的值。

printf(“There were %d words and %d lines.\n”, words, lines);

8.考虑下面的程序：

#include <stdio.h>

int main(void)

{

int a, b;

a = 5;

b = 2;

b = a;

a = b;

printf(“%d %d\n”, b, a);

return 0;

}

请问在第7行、第8行和第9行之后程序的状态分别是什么？

7行：b被赋值a

8行：a被赋值b

9行：打印输出 5 5

编程练习

2001.编写一个程序，调用printf（）函数在一行上输出您的名和姓，再调用一次printf（）函数在两个单独的行上输出您的名和姓，然后调用一对printf（）函数在一行上输出您的名和姓。输出应如下所示（当然里面要换成您的姓名）:

Anton Bruckner 第一句printf实现

Anton 第二句printf实现

Bruckner 第二句printf实现

Anton Bruckner 第三句printf实现

2002.编写一个程序输出您的姓名及地址。

2003.编写一个程序，把您的年龄转换成天数并显示二者值。不用考虑平年和润年的问题。

2004.编写一个能够产生下面输出的程序：

For he’s a jolly good fellow!

For he’s a jolly good fellow!

For he’s a jolly good fellow!

Which nobody can deny!

2005.编写一个程序，创建一个名为toes的整数变量。让程序吧toes设置为10.再让程序计算两个toes的和以及toes的平方。程序应该输出所有的3个值，并分别标识他们。

2006.编写一个能够产生下列输出的程序：

smile! smile! smile!

smile! smile!

smile!

在程序中定义一个能显示字符串smile!一次的函数，并在需要时使用该函数。

2007.编写一个程序，程序中要调用名为one\_three()的函数。该函数要在一行中显示单词“one”，在调用two（）函数，然后再在另一行中显示单词“three”。函数two（）应该能在一行中显示单词“two”。main（）函数应该在调用one\_three()函数之前显示短语“starting now:”，函数调用之后要显示“done!”。这样最后的输出结果应如下所示：

starting now:

one

two

three

done!

第3章 数据和C

**本章学习内容：**

关键字：int, short, long, unsigned, char, float, double, \_Bool, \_Complex, \_Imaginary

运算符：sizeof

函数：scanf()

C使用的基本数据类型。

整数类型和浮点数类型的区别。

对上述类型，如何书写常量和声明变量。

使用printf()和scanf()函数读写各种类型数据的值。

**程序清单：**

301rhodium

302print1

303bases

304print2

305charcode

306altnames

307showf\_pt

308typesize

309badcount

310escape

**知识点：**

变量与常量的区别

整数与浮点数类型的区别

C的数值数据类型

八进制、十进制、十六进制打印

头文件inttypes.h

科学计数法、指数计数法

不同类型在不同平台上的数值范围

sizeof()求不同类型字节大小

转义序列

|  |  |
| --- | --- |
| 序列 | 意义 |
| \a | 报警（ANSIC） |
| \b | 退格 |
| \f | 走纸 |
| \n | 换行 |
| \r | 回车 |
| \t | 水平制表符 |
| \v | 垂直制表符 |
| \\ | 反斜杠（\） |
| \’ | 单引号（’） |
| \” | 双引号（”） |
| \? | 问号（?） |
| \Ooo | 八进制值（o表示一个八进制数字） |
| \xhh | 十六进制值（h表示一个十六进制数字） |

**总结：**

C有多种数据类型。基本的数据类型包括两大类：整数类型和浮点类型。整数类型的两个重要特征是其类型的大小以及它是有符号还是无符号的。最小的整数类型是char，因实现不同可以是有符号或无符号的，可以使用signed char和unsigned char确定该类型的符号属性，不过这通常用于使用此类型表示小整数而非字符编码。其他的整数类型包括shor、int、long、和long long类型。对于上述类型的大小，C要求后面的类型不能小于前面的类型。上述类型都是有符号的，但可以使用unsigned关键字产生相应的无符号类型：unsigned short、unsigned int、unsigned long、unsigned long long类型，也可以使用signed修饰词明确地表示一个类型为有符号类型，最后，\_Bool类型是一种无符号类型，它只含有两个值0和1，对应于false和true。

3种浮点类型为float、double和ANSI C新增的long double，后面类型的大小至少要和前面的类型一样大。有些实现中支持复数和虚数类型，方法是把\_Complex和\_Imaginary关键字同浮点类型关键字结合使用，例如double \_Complex 和 float \_Imaginary类型。

整数可以表达为十进制、八进制、十六进制形式。前缀O指示八进制数，前缀Ox或OX指示十六进制数。例如，32、O40和Ox20分别表示十进制、八进制和十六进制的同一个值。后缀l或L指示long类型值，后缀ll或LL表示long long类型值。

字符常量表示为放在单引号种的一个字符，比如‘Q’、‘8’、‘$’.C的转义序列（例如\n）用于表示一些非打印字符。可以使用诸如‘\007’这样的形式通过字符的ASCII码表示一个字符。

浮点数可以书写为小数点固定的形式，比如9393.912；或者书写为指数形式，比如7.38E10.

printf（）函数通过对应于各种类型的转换说明符打印相应类型的数据。形式最简单的转换说明符由一个百分号和一个指示类型的字符组成，比如%d或%f。

**复习题：**

1.对下面的各种数据使用合适的数据类型：

a.East Simpleton的人口

b.DVD影碟的价格

c.本章节出现次数最多的字母

d.这个字母出现的次数

答：

a.long or unsigned int看城市大小 b.float c.char d.int（unsigned）

2.需要用long类型变量代替int类型变量的原因是什么？

一个原因是在您的系统中long可以容纳比int更大的数；另一个原因是如果您确实需要处理更大的值，那么使用一种在所有系统上都保证至少是32位的类型会使程序的可移植性更好。

3.获得一个32位的有符号整数，可以使用哪些可移植的数据类型？每种选择的原因是什么？

要获得正好是32位的数，您可以使用int32\_t（如果在您的系统中有这一定义的话）。要获得可存储至少32位的最小类型，可以使用int\_least32\_t。如果要在32位的类型中获得提供最快计算速度的类型，可以选择int\_fast32\_t。

4.指出下列常量的类型和意义（如果有的话）：

a.’\b’

b.1066

c.99.44

d.OXAA

e.2.0e30

答：

a.char常量（但以int类型存储）

b.int常量

c.float常量

d.unsigned int常量，十六进制格式

e.double常量

5.Dottie Cawm写的下面这个程序中有很多错误，找出这些错误。

include <stdio.h>

main

(

float g; h;

float tax, rate;

g = e21;

tax = rate\*g;

)

答：

#include <stdio.h>

int main(void)

{

float g,h;

float tax, rate;

g = 2.1e21

tax = rate\*g;

}

6.指出下表中各常量的数据类型（在声明语句中使用的数据类型）及其在printf（）中的格式说明符。

|  |  |  |
| --- | --- | --- |
| 常量 | 类型 | 说明符 |
| a. | 12 | %d |
| b. | Ox3 | %#x |
| c. | ‘C’ | %c |
| d. | 2.34E07 | %e |
| e. | ‘\040’ | %c |
| f. | 7.0 | %f |
| g. | 6L | %ld |
| h. | 6.0f | %f |

7.指出下表中各常量的数据类型（在声明语句中使用的数据类型）及其在printf（）中的格式说明符，假设int类型为16位长。

|  |  |  |
| --- | --- | --- |
| 常量 | 类型 | 说明符 |
| a. | 012 | %#o |
| b. | 2.9e05L | %Le |
| c. | ‘s’ | %c |
| d. | 100000 | %ld |
| e. | ‘\n’ | %c |
| f. | 20.0f | %f |
| g. | Ox44 | %#x |

8.假设一个程序开始处有如下的声明：

int imate = 2;

long shot = 53456;

char grade = ‘A’;

float log = 2.71828;

在下面printf（）语句中添上合适的类型说明符：

printf（“The odds against the %\_\_ were %\_\_ to 1.\n”,imate, shot）;

printf(“A score of %\_\_ is not an %\_\_ grade.\n”, log, grade);

printf（“The odds against the %d were %ld to 1.\n”,imate, shot）;

printf(“A score of %f is not an %c grade.\n”, log, grade);

9.假设ch为char类型变量。使用转义序列、十进制值、八进制字符常量以及十六进制字符常量等方法将其赋值为回车符（假设使用ASCII编码值）。

ch = ‘\r’;

ch = 13;

ch = ‘\015’;

ch = ‘\xd’

10.改正下面程序（在C中/表示除法）。

woid main(int)/this program is perfect /

{

cows, legs integer;

printf(“How many cow legs did you count?\n);

scanf(“%c”, legs);

cows = legs / 4;

printf(“That implies there are %f cows.\n”, cows)

}

答：

#include <stido.h>

int main(void)

{

int cows, legs;

printf(“How many cow legs did you count?\n”);

scanf(“%d”,&legs);

cows = legs / 4;

printf(“That implies there are %d cows.\n”,cows);

return 0;

}

11.指出下列转义字符的含义：

a. \n

b. \\

c. \”

d. \t

答：

a. \n 换行字符

b. \\ \字符

c. \” ”字符

d. \t 水平制表符

**编程练习：**

3001.通过实验的方法（即编写带有此类问题的程序）观察系统如何处理整数上溢、浮点数上溢和浮点数下溢的情况。

double说明符要用%lf

float精度在8个数字，double精度在15个数字

3002.编写一个程序，要求输入输入一个ASCII码值（如66），然后输出相应的字符。

3003.编写一个程序，发出警报声，并打印下列文字：

Startled by the sudden sound, Sally shouted, “By the Great Pumpkin, what was that!”

输出\007要用%c说明符。

3004.编写一个程序，读入一个浮点数，并分别以小数形式和指数形式打印。输出应如同下面格式（实际显示的指数位数也许因为系统而不同）：

Theinputis21.290000 or 2.129000e+001.

3005.一年约有3.156\*10³S（立方3改为7）。编写一个程序，要求输入您的年龄，然后显示该年龄合多少秒。

3006.1个水分子的质量约为3.0\*10-³g（立方3为23）,1夸脱水大约有950g。编写一个程序，要求输入水的夸脱数，然后显示这么多水中包含多少个水分子。

3007.1英寸等于2.54CM。编写一个程序，要求输入您的身高（以英寸为单位），然后显示该身高值等于多少厘米。如果您愿意，也可以要求以厘米为单位输入身高，然后以英寸为单位进行显示。

第4章 字符串和格式化输入\输出

**本章学习内容：**

函数：strlen()

关键字：const

字符串。

如何创建和存储字符串。

如何使用scanf()和print()读取和显示字符串。

如何使用strlen()函数获取字符串的长度。

使用C预处理器的#define指令和ANSIC的const修饰符创建符号常量。

**程序清单：**

401talkback

402praise1

403praise2

404pizza

405defines

406printout

407width

408floats

409flags

410strings

411intconv

412floatcnv

413printval

414longstrg

415input

416varwid

417skip2

**知识点：**

头文件limits.h

|  |  |
| --- | --- |
| 符号常量 | 含义 |
| CHAR\_BIT | 一个char的位数 |
| CHAR\_MAX | char类型的最大值 |
| CHAR\_MIN | char类型的最小值 |
| SCHAR\_MAX | signed char类型的最大值 |
| SCHAR\_MIN | signed char类型的最小值 |
| UCHAR\_MAX | unsigned char类型的最大值 |
| SHRT\_MAX | short 类型的最大值 |
| SHRT\_MIN | short 类型的最小值 |
| USHRT\_MAX | unsigned short类型的最大值 |
| INT\_MAX | int 类型的最大值 |
| INT\_MIN | int 类型的最小值 |
| UINT\_MAX | unsigned int 类型的最大值 |
| LONG\_MAX | long 类型的最大值 |
| LONG\_MIN | long 类型的最小值 |
| ULONG\_MAX | unsigned 类型的最大值 |
| LLONG\_MAX | long long 类型的最大值 |
| LLONG\_MIN | long long 类型的最小值 |
| ULLONG\_MAX | unsigned long long类型的最大值 |

头文件float.h

|  |  |
| --- | --- |
| 符号常量 | 含义 |
| FLT\_MANT\_DIG | float 类型的位数位 |
| FLT\_DIG | float 类型的最少有效数字位数（十进制） |
| FLT\_MIN\_10\_EXP | 带有全部有效数字的float类型的负指数的最小值（以10为低） |
| FLT\_MAX\_10\_EXP | float 类型的正指数最大值（以10为低） |
| FLT\_MIN | 保留全部精度的float类型正数的最小值 |
| FLT\_MAX | float类型正数的最大值 |
| FLT\_EPSILON | 1.00和比1.00大的最小的float类型值之间的差值 |

转换说明符及作为的打印输出

|  |  |
| --- | --- |
| 转换说明 | 输出 |
| %a | 浮点数、十六进制数字和p-记数法（C99） |
| %A | 浮点数、十六进制数字和p-记数法（C99） |
| %c | 一个字符 |
| %d | 有符号十进制整数 |
| %e | 浮点数、e-记数法 |
| %E | 浮点数、E-记数法 |
| %f | 浮点数、十进制记数法 |
| %g | 根据数值不同自动选择%f或%e。%e格式在指数小于-4或者大于等于精度时使用 |
| %G | 根据数值不同自动选择%f或%E。%E格式在指数小于-4或者大于等于精度时使用 |
| %i | 有符号十进制整数（与%d相同） |
| %o | 无符号八进制整数 |
| %p | 指针（地址） |
| %s | 字符串 |
| %u | 无符号十进制整数 |
| %x | 使用十六进制数字0f的无符号十六进制整数 |
| %X | 使用十六进制数字0F的无符号十六进制整数 |
| %% | 打印一个百分号 |

printf()修饰符

|  |  |
| --- | --- |
| 修饰符 | 意义 |
| 标志 | 五种标志（-、+、空格、#和0）都将在4.5中描述。可以使用零个或者多个标志 示例：%-10d |
| digit(s) | 字段宽度的最小值。如果该字段不能容纳要打印的数或者字符串，系统就会使用更宽的字段。示例：%4d |
| .digit（s） | 精度。对于%e、%E和%f转换，是将要在小数点的右边打印的数字的位数。对于%g和%G转换，是有效数字的最大位数。对于%s转换，是将要打印的字符的最大数目。对于整数转换，是将要打印的数字的最小位数；如果必要，要使用前导零来达到这个位数。只使用.表示其后跟随一个零，所以%.f与%.0f相同。示例：%5.2f打印一个浮点数，它的字段宽度为5个字符，小数点后有两个数字 |
| h | 和整数转换说明符一起使用，表示一个short int或unsigned short int类型数值。示例：%hu、%hx、%6.4hd |
| hh | 和整数转换说明符一起使用，表示一个signed char或unsigned char类型数值。  示例：%hhu、%hhx、%6.4hd |
| j | 和整数转换说明符一起使用，表示一个intmax\_t或uintmax\_t值。示例：%jd、%8jX |
| l | 和整数转换说明符一起使用，表示一个long int或unsigned long int类型值。示例：%ld、%8lu |
| ll | 和整数转换说明符一起使用，表示一个long long int或unsigned long long int类型值（C99）  示例：%lld、%8llu |
| L | 和浮点转换说明符一起使用，表示一个long double值  示例：%Lf、%10.4Le |
| t | 和整数转换说明符一起使用，表示一个ptrdiff\_t值（与两个指针之间的差相对应的类型）（C99）  示例：%td、%12ti |
| z | 和整数转换说明符一起使用，表示一个size\_t值（sizeof返回的类型）（C99）  示例：%zd、%12zx |

printf（）的标志

|  |  |
| --- | --- |
| 标志 | 意义 |
| - | 项目是左对齐的；也就是说，会把项目打印在字段的左侧开始处。示例：%-20s |
| + | 有符号的值若为正，则显示带加号的符号；若为负，则带减号的符号。示例：%+6.2f |
| 空格 | 有符号的值若为正，则显示时带前导空格（但是不显示符号）；若为负，则带减号符号。+标志会覆盖空格标志。  示例：% 6.2f |
| # | 使用转换说明的可选形式。若为%o格式，则以0开始；若为%x和%X格式，则以0x或0X开始。对于所有的浮点形式，#保证了即使不跟任何数字，也打印一个小数点字符。对于%g和%G格式，它防止尾随零被删除。  示例：%#o、%#8.0f、%+#10.3E |
| 0 | 对于所有的数字格式，用前导零而不是用空格填充字段宽度。如果出现-标志或者指定了精度（对于整数）则忽略该标志。示例：%010d、%08.3f |

ANSI C中scanf()的转换说明符

|  |  |
| --- | --- |
| 转换说明符 | 意义 |
| %c | 把输入解释成一个字符 |
| %d | 把输入解释成一个有符号十进制整数 |
| %e %f %g %a | 把输入解释成一个浮点数（%a是C99） |
| %E %F %G %A | 把输入解释成一个浮点数（%A是C99） |
| %i | 把输入解释成一个有符号十进制整数 |
| %o | 把输入解释成一个有符号八进制整数 |
| %p | 把输入解释成一个指针（一个地址） |
| %s | 把输入解释成一个字符串；输入的内容以第一个非空白字符作为开始，并且包含直到下一个空白字符的全部字符 |
| %u | 把输入解释成一个无符号十进制整数 |
| %x %X | 把输入解释成一个有符号十六进制整数 |

scanf()的转换修饰符

|  |  |
| --- | --- |
| 修饰符 | 意义 |
| \* | 滞后赋值（请参见书中的文字部分）  示例：%\*d |
| digit（s） | 最大字段宽度；在达到最大字段宽度或者遇到第一个空白字符时（不管哪个先发生都一样）停止对输入项的读取  示例：%10s |
| hh | 把整数读作signed char 或 unsigned char  示例：%hhd、%hhu |
| ll | 把整数读作long long或者unsigned long long（C99）  示例：%lld、%llu |
| h l L | %hd和%hi指示该值将会存储在一个short int中。  %ho和%hu指示该值将会存储在一个unsigned short int中。  %ld和%li指示该值将会存储在一个long中。  %lo、%lx、%lu指示该值将会存储在一个unsigned long中。  %le、%lf、%lg指示该值以double类型存储。  将L（而非l）与e、f和g一起使用指示该值以long double类型存储。  如果没有这些修饰符，d、i、o、x指示int类型，而e、f、g指示float类型。 |

**总结：**

字符串是作为一个单位处理的一系列字符。在C中，以空白符结束的一系列字符代表一个字符串，空字符就是ASCII码为0的字符。字符串可以存储在字符数组中。一个数组就是一系列项目或元素，并且所有这些项目或元素的类型都相同。要声明为name并有30个char类型的元素的数组，请使用以下语句：

char name[30];

请确保分配足够多的元素来存放整个字符串（包括空字符）。

字符串常量用双引号引起来的字符串表示：

“This is an example of a string”

strlen（）函数（在string.h头文件中定义）可以用于获得一个字符串的长度（不包括标示终止的空白符）。scanf（）函数在使用%s说明符时，可以用于读取包含一个单词的字符串。

C预处理器在源代码程序中搜索预处理器指令（预处理器指令以#符号开始），并在程序开始编译前处理它们。#include指令使处理器把另一个文件的内容添加到文件中该指令所在的位置。使用#define指令可以创建明显常量，也就是代表常量的符号。limits.h和float.h头文件使用#define定义了一套表示整数和浮点类型的各个属性的常量。也可以使用const修饰符来创建符号常量。

printf（）和scanf（）函数对输入和输出提供多种支持。二者都使用一个包含内嵌转换说明符的控制字符串来指示将要读取或打印的数据项的类型和数目。还可以使用转换说明符来控制输出的外观：字符宽度、小数点位置和字段内的布局。

**复习题：**

1.再次运行程序清单401，但是在要求您输入名字是，请输入您的名字和姓氏。发生了什么？为什么？

第二个scanf（）会工作不正常。第一个scanf（）语句只是读入了您的名而没有读入您的姓，您的姓依然存储在输入缓冲区（缓冲区只是一块用来存放输入的临时存储区域）中。当下一个scanf（）语句想要读入您的体重时，它从上次读入结束的地方开始，这样就试图把您的姓作为体重来读取。这会使scanf（）失败。另一方面如果您对姓名请求作出像Lasha 144这样的响应，程序会使用144作为您的体重，虽然您是在程序请求体重之前输入144的。

2.假定下列每个示例都是某个完整程序的一部分。他们的打印结果分别是什么？

a. printf(“He sole the painting for $%2.2f.\n”, 2.345e2);

b. printf(“%c%c%c\n”, ‘H’, 105, ‘\41’);

c. #define Q “His Hamlet was funny without being vulgar.”

printf(“%s\nhas %d characters.\n”, Q, strlen(Q));

d. printf(“Is %2.2e the same as %2.2f?\n”, 1201.0, 1201.0);

解：

a. He sole the painting for $234.50.

b. Hi!

c. His Hamlet was funny without being vulgar.

has 42 characters.

d. Is 1.20e+003 the same as 1201.00?

3.在问题2c中，应进行哪些更改以使字符串Q引在双引号中输出？

#define Q “\”His Hamlet was funny without being vulgar.\””

或printf(“\”%s\”\nhas %d characters.\n”, Q, strlen(Q));

4.找出下列程序中的错误。

define B booboo

define X 10

main(int)

{

int age;

char name;

int xp;

printf(“Please enter your first name. “);

scanf(“%s”,name);

printf(“All right, %c, what’s your age?\n”, name);

scanf(“%f”, age);

xp = age + X;

printf(“That’s a %s! You must be at least %d.\n”, B, xp);

rerun 0;

}

解：

#define B “booboo”

#define X 10

int main(void)

{

int age;

char name[40];

printf(“Please enter your first name.”);

scanf(“%s”,name);

printf(“All right, %s, what’s your age?\n”,name);

scanf(“%d”,age);

xp = age + X;

printf(“That’s a %s!You must be at least %d.\n”,B, xp);

return 0;

}

5.假设一个程序这样开始：

#define BOOK “War and Peace”

int main(void)

{

float cost =12.99;

float percent = 80.0;

请构造一个printf()语句，使用BOOK、cost和percent打印下列内容：

This copy of “War and Peace” sells for $12.99.

That is 80% of list.

答:

printf(“This copy \”%s\” sells for $%.2f.\nThat is %.f%% of list.\n”, BOOK, cost, percent);

6.您会使用什么转换说明来打印下列各项内容？

a. 一个字段宽度等于数字位数的十进制整数。

b. 一个形如8A、字段宽度为4的十六进制整数

c. 一个形如232.346、字段宽度为10的浮点数

d. 一个形如2.33e+002、字段宽度为12的浮点数

e. 一个字段宽度为30、左对齐的字符串

答：

a.%d

b.%4X

c.%10.3f

d.%12.2e

e.%-30s

7.您会使用哪个转换说明来打印下列各项内容？

a. 一个字段宽度为15的unsigned long 整数

b. 一个形如0x8a、字段宽度为4的十六进制整数

c. 一个形如2.33E+02、字段宽度为12、左对齐的浮点数

d. 一个形如+232.346、字段宽度为10的浮点数

e. 一个字符串的前8个字符，字段宽度为8字符

答：

a.%15lu

b.%#4x

c.%-12.2E

d.%+10.3f

e.%8.8s

8.您会使用什么转换说明来打印下列各项内容？

a. 一个字段宽度为6、最少有4位数字的十进制整数

b. 一个字段宽度在参数列表中给定的八进制整数

c. 一个字段宽度为2的字符

d. 一个形如+3.13、字段宽度等于数字中字符个数的浮点数

e. 一个字符串的前5个字符，字段宽度为7、左对齐

答：

a.%6.4d

b.%\*x

c.%2c

d.%+0.2f

e.%-7.5s

9.为下列每个输入行提供一个对其进行读取的scanf（）语句，并声明语句中用到的所有变量或数组。

a. 101

b. 22.32 8.34E-09

c.linguini

d. catch 22

e. catch 22（但是跳过catch）

答：

a.

int a;

scanf(“%d”,a);

b.

float b;float b2;

scanf(“%f%f”,&b, &b2);

c.

char c[10];

scanf(“%s”,c);

d.

char d[10];int d2;

scanf(“%s %d”, d, &d2);

e.

int e2;

scanf(“%\*s %d”, &e2);

10.什么是空字符？

空白字符包括空格、制表符和换行符。C使用空白字符分隔各个语言符号；scanf()使用空白字符分隔相邻的输入项。

11.假设您想在程序中使用圆括号代替花括号。以下方法可以吗？

#define ( {

#define ) }

原则上可以这样定义，但不幸的是，预处理器不能区别哪些圆括号应该被替换成花括号，哪些圆括号不应该被替换成花括号。

**编程练习：**

4001.编写一个程序，要求输入名字和姓氏，然后以“名字，姓氏”的格式打印。

4002.编写一个程序，要求输入名字，并执行以下操作：

a. 把名字引在双引号中打印出来。

b. 在宽度为20个字符的字段内打印名字，并且整个字段引在引号内。

c. 在宽度为20个字符的字段的左端打印名字，并且整个字段引在引号内。

d. 在比名字宽3个字符的字段内打印它。

4003.编写一个程序，读取一个浮点数，并且首先以小数点记数法，然后以指数记数法打印之。输出使用下列形式（在指数位置显示的数字的位数可能会随系统而不同）：

a. The input is 21.3 or 2.1e+001

b. The input is +21.290 or 2.129E+001

4004.编写一个程序，要求输入身高（以英寸为单位）和名字，然后以如下形式显示：

Dabney, you are 6.208 feet tall

使用float类型，使用/作为除号。如果您愿意，可以要求以厘米为单位输入身高，并以米为单位进行显示。

4005.编写一个程序，首先要求用户输入名字，然后要求用户输入姓氏。在一行打印输入的姓名，在下一行打印每个名字中字母的个数。把字母个数与相应名字的结尾对齐，如下所示：

Melissa Honeybee

7 8

然后打印相同的信息，但是字母个数与相应单词的开始对齐。

Melissa Honeybee

7 8

4006.编写一个程序，设置一个值为1.0/3.0的double类型变量和一个值为1.0/3.0的float类型变量。每个变量的值显示三次：一次在小数点右侧显示4个数字，一次在小数点右侧显示12个数字，另一次在小数点右侧显示16个数字。同事要让程序包括float.h文件，并显示FLT\_DIG和DBL\_DIG的值。1.0/3.0的显示值与这些值一致吗？

4007.编写一个程序，要求用户输入行驶的英里数和消耗汽油的加仑数。接着应该计算和显示消耗每加仑汽油行驶的英里数，显示方式是在小数点右侧显示一个数字。然后根据1加仑约等于3.785升，1英里约等于1.609公里的规则，它应该把每加仑英里数转换成每100公里的升数（欧洲通用的燃料消耗表示法），并显示结果，显示方式是在小数点右侧显示一个数字（请注意，美国方案测量每单位距离消耗的燃料数，而欧洲方案测量每单位燃料的行驶距离）。用符号常量表示两个转换系数（使用const或#define）。

第5章 运算符、表达式和语句

**本章学习内容：**

关键字：while, typedef

运算符：= - \* / % ++ --(type)

C的各种各样的运算符，其中包括用于普通数学运算的运算符。

运算符的优先级以及术语“语句”和“表达式”的含义。

简单的while循环。

复合语句、自动类型转换和类型指派。

如何编写带有参数的函数。

**程序清单：**

501shoes1

502shoes2

503golf

504squares

505wheat

506divide

507rules

508sizeof

509min\_sec

510add\_one

511post\_pre

512bottles

513addemup

514convert

515pound

516running

**知识点：**

类型转换：类型提升

运算符

|  |  |
| --- | --- |
| 赋值运算符： |  |
| = | 将它右边的值赋给它左边的变量 |
| 算数运算符： |  |
| + | 将它右边的值和它左边的值相加 |
| - | 从它左边的值里减掉它右边的值 |
| - | 作为一元运算符，改变它右边值的符号 |
| \* | 用它左边的值乘以它右边的值 |
| / | 用它右边的值去除它左边的值。如果两个操作数都是整数，那么结果被截尾（截尾为整数类型时） |
| % | 当它左边的值被它右边的值除时，得到的余数（整型） |
| ++ | 对它右边的值先加1（前缀模式），语句结束以后对它左边的值加1（后缀模式） |
| -- | 与++类似，只不过是减1 |
| 其他运算符： |  |
| sizeof | 给出它右边的操作数的字节大小。操作数可以是在圆括号里的一个类型说明符，例如sizeof(float);或者是一个具体的变量、数组等的名字，例如：sizeof foo |
| (type) | 作为指派运算符，它将跟在它后面的值转换成由圆括号中的关键字所指定的类型。例如，（float）9将整数9转换成浮点数9.0 |

**总结：**

C有多种运算符，例如在本章中讨论的赋值和算术运算符。总的来说，一个运算符作用于一个或多个操作数来产生一个值。带一个操作数的运算符（例如负号和sizeof）称为一元运算符。要求两个操作数的运算符（例如加法和乘法运算符）称为二元运算符。

表达式是运算符合操作数的组合。在C里，每一个表达式都有一个值，其中包括赋值表达式和比较表达式。运算符优先级的规则帮助决定当对表达式进行求值时，如何组合表达式里的各项。当两个运算符共享一个操作数时，具有较高优先级的运算符先被运算。如果运算符有相同的优先级，结合性（从左到右或者从右到左）决定了哪个运算符先被应用。

语句是对计算机的完整指示，在C中通过一个分号来标识。到目前，您已经使用了声明语句、赋值语句、函数调用语句和控制语句。包含在一对花括号里的语句构成了一个复合语句或者代码块。一个特殊的控制语句是while循环，只要判断条件保持为真，该循环就重复执行循环体里的语句。

在C里，许多类型转换会自动发生。当char和short类型出现在表达式里或者作为函数的参数时，它们都将被提升为int类型。当float类型作为一个函数参数时被提升为double类型。在K&RC（而不是ANSIC）下，当float用于表达式里时也被提升为double类型。当把一种类型的值赋给另一种类型的变量时，该值被转换成和那个变量相同的类型。当较大类型的值被转换成较小类型的值（例如，long变成short，或者double变成float）时，它们可能丢失数据。根据本章概括的规则，在混合类型的算术运算的情况下，较小的类型被转换成较大的类型。

当您定义了一个接受一个参数的函数时，您在函数定义里声明了一个变量，或称形式参数。然后再函数调用中传入的值会赋给这个变量，就可以在函数里使用该值了。

**复习题：**

1.假定所有的变量都是int类型。找出下面每一个变量的值：

a. x = (2 + 3) \* 6;

b. x = (12 +6) /2\*3;

c. y = x =(2 + 3) /4;

d. y = 3 + 2\*(x = 7/2);

答：

a. 30

b. 27

c. 1

d. 9

2.假定所有变量都是int类型。找出下面每一个变量的值：

a. x = (int) 3.8 + 3.3;

b. x = (2+3) \* 10.5;

c. x = 3 / 5 \* 22.0;

d. x = 22.0 \* 3 / 5;

答：

a. 6

b. 52

c. 0

d. 13

3.您怀疑下面的程序里有一些错误。您能找出这些错误吗？

int main(void)

{

int i = 1,

float n;

printf(“Watch out! Here come a bunch of fractions!\n”);

while(i++ < 30)

n = 1/i;

printf(“%f”, n);

printf(“That’s all, folks!\n”);

return;

}

答：

#include <stdio.h>

int main(void)

{

int i = 1;

float n;

printf(“Watch out! Here come a bunch of fractions!\n”);

while(i < 30)

{

n = 1.0/i;

printf(“%f\n”,n);

}

printf(“That’s all, folks!\n”);

return 0;

}

4.这是程序清单5.9的另一种设计方法。表面上看，它使用一个scanf()函数替代了程序清单5.9中的两个scanf（）。但是该程序不令人满意。和程序清单5.9相比，它有什么缺点？

#include <stdio.h>

#define S\_TO\_M 60

int main(void)

{

int sec, min, left;

printf(“This program converts seconds to minutes and”);

printf(“seconds.\n”);

printf(“Just enter the number of seconds.\n”);

printf(“Enter 0 to end the program.\n”);

while (sec >0)

{

scanf(“%d”,&sec);

min = sec/S\_TO\_M;

left = sec % S\_TO\_M;

printf(“%d sec is %d min, %d sec. \n”, sec, min, left);

printf(“Next input?\n”);

}

printf(“Bye!\n”);

return 0;

}

答：英文看不懂本题不作答

5.下面的程序将打印出什么？

#include <stdio.h>

#define FORMAT “%s! C is cool!\n”

int main(void)

{

int num = 10;

printf(FORMAT,FORMAT);

printf(“%d\n”,++num);

printf(“%d\n”, num++);

printf(“%d\n”, num--);

printf(“%d\n”,num);

return 0;

}

答：

%s! C is cool!

! C is cool!

11

11

12

11

6.下面的程序将打印出什么？

#include <stdio.h>

int main(void)

{

char c1, c2;

int diff;

float num;

c1 = ‘S’;

c2 = ‘O’;

diff = c1 - c2;

num = diff;

printf(“%c%c%c: %d %3.2f\n”, c1, c2, c1, diff, num);

return 0;

}

答：

SOS: 4 4.00

7.下面的程序将打印出什么？

#include <stdio.h>

#define TEN 10

int main(void)

{

int n = 0;

while (n++ < TEN)

printf(“%5d”,n);

printf(“\n”);

return 0;

答：

1 2 3 4 5 6 7 8 9 10

8.修改上一个程序，让它打印从a到g的字母。

#include <stdio.h>

const char TEN = 'g';

int main(void)

{

char n = 'a'-1;

while (n++ < TEN)

printf("%5c",n);

printf("\n");

return 0;

}

9.如果下面的片段是一个完整程序的一部分，它们将打印出什么？

a.

int x = 0;

while(++x < 3)

printf(“%4d”, x);

b.

int x = 100;

while(x++ < 103) //后加加在这里匹配条件后向下执行前就加1了，并不是在printf结束才加1

printf(“%4d\n”, x);

printf(“%4d\n”, x);

c.

char ch = ‘s’;

while(ch< ‘w’)

{

printf(“%c”,ch);

ch++;

}

printf(“%c\n”,ch);

答：

a. 1 2

b.

101

102

103

104

c.

stuvw

10.下面的程序将打印出什么？

#define MESG “COMPUTER BYTES DOG”

#include <stdio.h>

int main(void)

{

int n = 0;

while(n<5)

printf(“%s\n”, MESG);

n++;

printf(“That’s all.\n”);

return 0;

}

答：

COMPUTER BYTES DOG一直打印，进入死循环。

11.构造完成下面功能（或者用一个术语来说，有下面的副作用）的语句：

a.把变量x的值增加10

b.把变量x的值增加1

c.将a与b之和的两倍赋给c

d.将a与两倍的b之和赋给c

答：下面片段是一个完整程序的一部分

a. x = x + 10;

b. x++

c. c = (a+b)\*2

d. c = a+b\*2

12.构造具有下面功能的语句：

a. 把变量x的值减1

b. 把n除以k所得的余数赋给m

c. 用b减去a的差去除q，并将结果赋给p

d. 用a与b的和除以c与d的乘积，并将结果赋给x

答：

a. x—

b. m = n%k

c. p = q/(b-a)

d. x = (a + b)/(c\*d)

**编程练习：**

5001.编写一个程序。将用分钟表示的时间转换成以小时和分钟表示的时间。使用#define或者const来创建一个代表60的符号常量。使用while循环来允许用户重复键入值，并且当键入一个小于等于0的时间时终止循环。

5002.编写一个程序，此程序要求输入一个整数，然后打印出从（包括）输入的值到（包括）比输入的值大10的所有整数值（也就是说，如果输入为5，那么输出就从5到15）。要求在各个输出值之间用空格、制表符或换行符分开。

5003.编写一个程序，该程序要求用户输入天数，然后将该值转换为周数和天数。例如，此程序将把18天转换成2周4天。用下面的格式显示结果：

18 days are 2 weeks, 4 days.

使用一个while循环让用户重复输入天数；当用户输入一个非正数（如0或-20）时，程序将终止循环。

5004.编写一个程序让用户按厘米输入一个高度值，然后，程序按照厘米和英尺英寸显示这个高度值。允许厘米和英寸的值出现小数部分。程序允许用户继续输入，直到用户输入一个非正的数值。程序运行的示例如下所示：

Enter a height in centimeters:182

182.0 cm = 5feet, 11.7 inches

Enter a height in centimeters (<=0 to quit):168

168.0 cm = 5 feet, 6.1 inches

Enter a height in centimeters (<=0 to quit):0

bye

5005.改写用来找到前20个整数之和的程序addemup.c（程序清单5.13）（如果您愿意，可以把addemup.c程序看成是一个计算如果您第一天得到$1，第二天得到$2，第三天得到$3，以此类推，您在20天里会挣多少钱的程序）.修改该程序，目的是您能交互地告诉程序计算将进行到哪里。也就是说，用一个读入的变量来代替20.

5006.现在修改编程练习5中的程序，使它能够计算整数平方的和（如果您喜欢，可以这样认为：如果您第一天得到$1，第二天得到$4，第三天得到$9，以此类推您将得到多少钱。这看起来像一个很好的买卖）。C没有平方函数，但是您可以利用n的平方是n\*n的事实。

5007.编写一个程序，该程序要求输入一个float型数并打印该数的立方值。使用您自己设计的函数来计算该值的立方并且将它打印出来。main（）程序把输入的值传递给该函数。

5008.编写一个程序，该程序要求用户输入一个华氏温度。程序以double类型读入温度值，并将它作为一个参数传递给用户提供的函数Temperatures（）。该函数将计算相应的摄氏温度和绝对温度，并以小数点右边有两位数字的精度显示这三种温度。它应该用每个值所代表的温度刻度来标识这3个值。下面是将华氏温度转化成摄氏温度的方程：

Celsius = 1.8 \* Fahrenheit + 32.0

通常用在科学上的绝对温度的刻度是0代表绝对零，是可能温度的下界。下面是将摄氏温度转换为绝对温度的方程：

Kelvin = Celsius + 273.16

Temperatures()函数使用const来创建代表该转换里的3个常量的符号。main（）函数将使用一个循环来允许用户重复地输入温度，当用户输入q或其他非数值时，循环结束。

第6章 C控制语句：循环

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第7章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第8章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第9章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第10章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第11章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第12章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第13章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第14章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第15章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第16章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**

第17章

**本章学习内容：**

**程序清单：**

**知识点：**

**总结：**

**复习题：**

**编程练习：**