

第3章

C语言程序设计基础

学习一门语言,要遵循从字、词、短语、句子、段落、篇章的顺序,学习 C 语言也不例外。什么是 C 语言中的“单词”呢? 各种数据类型、运算符、关键字就是 C 语言中的“单词”,由它们组成的算式表达式就是 C 语言中的“短语”,由关键字、短语构成 C 语言中的“句子”,用户自定义的函数就是 C 语言中的“段落”,若干个函数构成一个程序就构成了 C 语言中的“一篇文章”了。本章介绍 C 语言基础知识,让读者了解 C 语言程序的基本结构、组成及运行过程,掌握 C 语言中的数据类型、运算符与表达式,为后续的学习打下坚实的基础。

为方便读者了解、学习本章内容,绘制本章思维导图,如图 3-1 所示。

3.1 C 语言程序结构特点

用 C 语言进行编程,必须了解 C 语言程序结构特点,掌握 C 语言程序的组成、执行过程及规范的书写格式。

3.1.1 知识点介绍

1. C 语言程序的组成

一个 C 语言源程序可以由一个或多个源文件组成,每个源文件可由一个或多个函数组成。但一个源程序无论由多少个源文件、多少个函数组成,有且只有一个名为 main 的主函数。

2. C 语言程序的基本结构

(1) 预处理部分

`#include <stdio.h>` 是一条预处理命令,用“#”开头,后面不能加“;”,`stdio.h` 是系统提供的头文件,其中包含输入输出标准库函数的信息。预处理命令内容详见本书第 8 章编译预处理。

(2) 全局声明

在函数之外进行的数据或函数的声明,称为全局声明,其作用范围从定义开始,到整个程序结束。

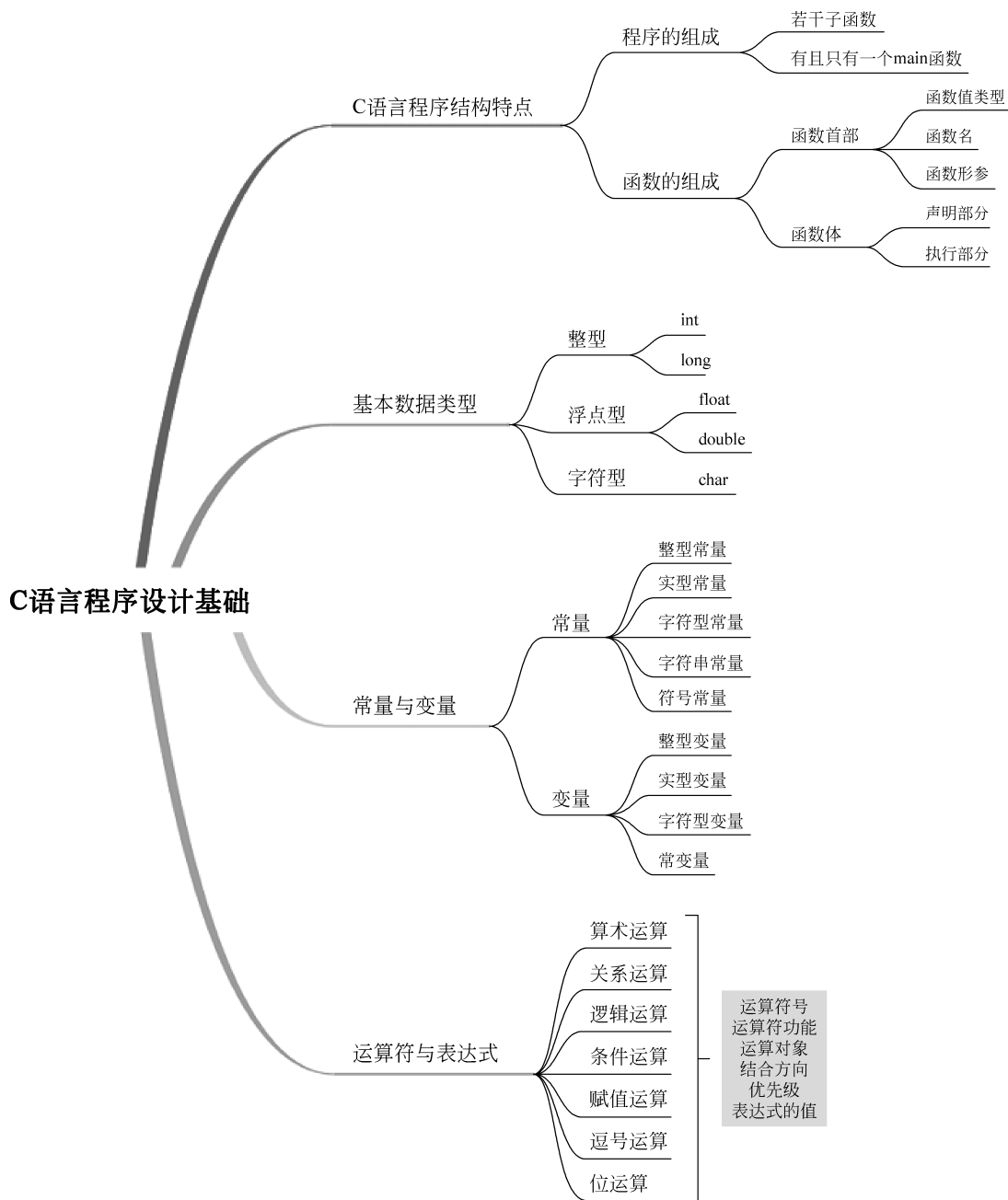


图 3-1 “C 语言程序设计基础”思维导图

(3) 函数

函数是 C 语言程序的基本单位,在设计良好的程序中,每个函数都用来实现一个或几个特定的功能。编写 C 语言程序的主要工作是编写一个又一个的函数。

(4) 注释

为了增加程序的可读性,便于用户阅读理解程序,可以对程序代码添加注释,说明变量的含义、程序段的功能等。注释不参与编译,对程序运行不起作用。注释的表示形式有两

种,如表 3-1 所示。

表 3-1 注释的表示形式

注释形式	注释的表示形式	说 明
单行注释	//.....	从“//”开始到本行结束均为注释内容
块注释	/* */	“/ * ”与“ * /”之间的内容均为注释内容,可以是一行也可以是多行

3. 函数的组成

函数包含两部分,函数首部和函数体。函数定义形式如表 3-2 所示。

表 3-2 函数的组成

函数的组成		示 例
函数首部	数据类型 函数名(形参 1,形参 2,...)	int max(int x,int y)
函数体	{ 声明语句 执行语句 }	{//函数体开始的标志 int z; //声明部分 if(x>y) //执行部分 z = x; else z = y; return (z); //将函数值返 //回到调用之处 } //函数体结束标志

4. C 语言程序的执行

C 语言程序的执行从 main 函数的第一条语句开始执行,称为程序的入口;当 main 函数的最后一条语句执行完毕,整个程序执行结束,称为程序的出口。其他函数在 main 函数执行期间,由 main 函数调用执行。

5. C 语言程序的书写格式

为了便于阅读、理解和维护程序,形成良好的编程风格,在书写程序时应遵循以下规则。

- (1) C 语言程序语句末尾需要加上分号。
- (2) 一个说明或一个语句占一行。
- (3) 程序中相匹配的一组左、右花括号,一般单独占一行,并上下对齐。
- (4) 正确使用缩进的方式体现语句块的层次关系,嵌套的语句块向后缩进,一般用 Tab 键实现。

3.1.2 实验部分

1. 实验目的

- (1) 熟练掌握 Microsoft Visual Studio 2010 编译系统的常用功能。
- (2) 学会使用在 Microsoft Visual Studio 2010 环境下创建、打开、编辑、保存、运行 C 语言程序。

(3) 了解 C 语言程序结构、基本语法规则及程序书写格式。

2. 实验内容

(1) 输入并运行程序一

功能：要求在屏幕上输出一行“This is a C Program.”信息。代码如下：

```
1  #include <stdio.h>           // 编译预处理命令
2  int main()                   // main 函数首部,函数值类型为 int 整型
3  {                             // 函数体开始的标志
4      printf("This is a C Program.\n"); // 输出一行指定的信息
5      return 0;                // 返回函数值 0,函数执行完毕
6  }                             // 函数体结束的标志
```

【实验提示】

① 按照第 2 章操作步骤创建项目,如 Pro3-1,添加源程序文件,如 3-1.cpp。在源程序文件编辑窗口输入源程序。

② 程序第 1 行 stdio.h 是 C 编译系统提供的一个有关标准输入输出信息的头文件,用 #include 预处理命令,将此文件写入本程序中。任何一个程序都要有结果的输出,因此任何一个程序的开始都需要“#include <stdio.h>”。

③ 程序只有一个 main 函数,C99 标准(ISO/IEC 9899: 1999-Programming languages-C 的简称)建议把 main 函数值类型指定为 int 整型类型,通过函数体中的“return 0;”语句,将函数值 0 返回给调用 main 函数的操作系统。

④ 函数体用花括号{ }括起来,两个花括号分别占用一行,与函数首部第一个字符对齐,函数体内部语句自动向后缩进一个 Tab 键的位置。

⑤ 函数体内调用标准输出 printf 函数,实现一行字符的输出,printf 函数中双引号内的字符序列原样输出。

⑥ “\n”是换行转义字符,即在输出“This is a C Program.”后,光标换至下一行。

⑦ 程序中“//”开始到本行结束为注释内容,不参与编译,可以不输入。

⑧ 去掉 printf 函数中的“\n”,观察程序运行结果有何不同。

(2) 程序填空

功能：要求在屏幕上分两行输出如下信息。

Strive to run and be the Dreamer of the new era.

Happiness comes from struggle!

请在程序的下画线处填入正确的内容,并把下画线删除,使程序得出正确的结果。

注意：不要增行或删行,也不得更改程序结构。

代码如下：

```
1  #include <stdio.h>           // 编译预处理命令
2  int main()                   // 主函数首部
3  {                             // 函数体开始的标志
4      / ***** FILL ***** /
5      _____
```

```
6      / ***** FILL ***** /
7      _____
8      return 0;                // 返回函数值 0, 函数执行完毕
9  }                            // 函数体结束的标志
```

【实验提示】

① 创建项目,如 Pro3-2,添加源程序文件,如 3-2. cpp。在源程序文件编辑窗口输入源程序。

② 模仿第 1 个程序实现分两行输出指定信息。

③ 注意使用“\n”换行。

(3) 输入并运行程序二

功能: 求两个整数之和。代码如下:

```
1  #include <stdio.h>           // 编译预处理命令
2  int main()                   // main 函数首部
3  {                             // 函数体开始的标志
4      int a,b,sum;             // 声明部分,定义 a,b,sum 为整型变量
5      a = 123;                 // 对整型变量 a 赋值为 123
6      b = 456;                 // 对整型变量 b 赋值为 456
7      sum = a + b;             // 将 a+b 的和 579 赋值给整型变量 sum
8      printf("sum is %d\n",sum); // 输出结果
9      return 0;               // 返回函数值 0, 函数执行完毕
10 }                            // 函数体结束的标志
```

【实验提示】

① 创建项目,如 Pro3-3,添加源程序文件,如 3-3. cpp。在源程序文件编辑窗口输入源程序。

② 第 4 行声明部分,定义 a、b、sum 为 int 整型变量。

③ 第 5~7 行为赋值语句,5~6 行将常量赋值给变量,7 行将求和表达式的值赋值给变量 sum。

④ 第 8 行调用 printf 标准库函数输出结果,双引号内 sum is 原样输出,%d 指定输出项整型变量 sum 对应的输出格式,输出时%d 由 sum 值代替。

⑤ 程序中“//”开始到本行结束为注释内容,不参与编译,可以不输入。

⑥ 修改程序求两个数的差、和、积。

(4) 输入并运行程序三

功能: 从键盘输入两个整数,输出两个整数中的较大者。代码如下:

```
1  #include <stdio.h>           // 编译预处理命令
2  int main()                   // main 函数首部
3  {                             // 函数体开始的标志
4      int max(int x,int y);     // 对被调用 max 函数的声明
5      int a,b,c;               // 定义变量 a,b,c
```

```

6    scanf("%d, %d", &a, &b);           // 输入变量 a 和 b 的值
7    c = max(a,b);                     // 调用 max 函数,将得到的函数值赋给 c
8    printf("max = %d\n", c);          // 输出最大值
9    return 0;                         // 返回函数值 0 到系统,函数执行完毕
10 }                                  // 函数体结束的标志
11
12 int max(int x, int y)                // max 函数首部
13 {                                    // max 函数体开始的标志
14     int z;                           // max 函数中的声明部分
15     if (x > y)                        // max 函数中的执行部分
16         z = x;
17     else
18         z = y;
19     return(z);                       // 将函数值 z 返回到 main 函数调用之处
20 }                                  // 函数体结束的标志

```

【实验提示】

① 按照第 2 章操作步骤创建项目,如 Pro3-4,添加源程序文件,如 3-4.cpp。在源程序文件编辑窗口输入源程序。

② 该程序由 main 函数和 max 子函数组成,其功能是从键盘上输入两个数,求出这两个数中的最大值并输出。

③ 程序执行过程如图 3-2 所示。

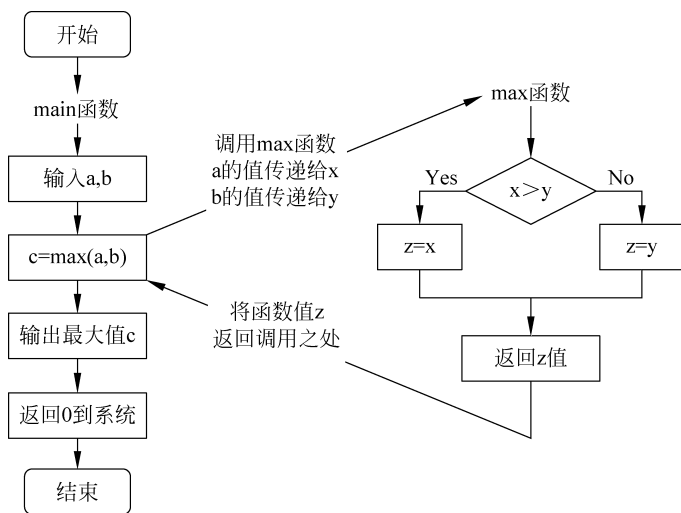


图 3-2 程序的执行过程

④ 第 4 行为对 max 子函数的声明,若子函数写在调用 main 函数之后,需要在 main 函数之前或之内对子函数进行声明,声明格式是函数首部加上分号,形参变量可以省略,但形参类型不可以省略。

⑤ 第 6 行,scanf 函数是标准的输入函数,scanf 函数中双引号内的格式控制串“%d, %d”,表示输入地址表列“&a, &b”对应的输入格式。编译链接成功后,运行窗口出现黑屏

状态,此时应在光标闪烁处从键盘任意键入两个整型数,例如在键盘上输入 4,9,则系统将 4 赋值给 a,将 9 赋值给 b。注意,因格式控制串两个 %d 之间有一个英文逗号,因此输入两个整型数据时一定要用英文逗号分隔。

⑥ 第 7 行, `c=max(a,b);`调用子函数,圆括号中的 a 和 b 是实际参数,相当于数学函数自变量的值,其值由 `scanf` 函数输入得到。

⑦ 第 12 行, `max` 函数首部, x 与 y 是形式参数,相当于数学函数的自变量。

⑧ 第 14 行, `max` 函数体内定义的变量 z 用于存放函数值相当于数学函数的因变量。

⑨ 第 15~18 行,利用“打擂台”算法求 x,y 的最大值,将 x 与 y 中的最大值存放在变量 z 中。

⑩ 第 19 行,将函数值 z 返回到 main 函数第 7 行调用之处赋值给变量 c。

3.1.3 练习与思考

1. 单选题

- (1) 以下叙述中不正确的一项是()。
 - A. C 语言程序的基本组成单位是函数
 - B. 在 C 语言程序中,注释说明只能位于一条语句的后面
 - C. 一个 C 语言源程序必须包含一个 main 函数
 - D. 一个 C 语言源程序可由一个或多个函数组成
- (2) 对于用 C 语言编写的代码程序,以下叙述中正确的一项是()。
 - A. 是一个源程序
 - B. 可立即执行
 - C. 经过编译解释才能执行
 - D. 经过编译即可执行
- (3) 以下叙述中正确的一项是()。
 - A. C 语言可以不用编译就能被计算机识别执行
 - B. C 语言出现的最晚,具有其他语言的一切优点
 - C. C 语言比其他语言高级
 - D. C 语言以接近英语国家的自然语言和数学语言作为语言的表达形式
- (4) 有一个命名为 C001.cpp 的 C 语言源程序,编译后得到的文件是()。
 - A. C001.obj
 - B. C001.cpp
 - C. C001.exe
 - D. C001.dat
- (5) C 语言规定,在一个源程序中,main 函数的位置()。
 - A. 必须在最开始
 - B. 必须在系统调用的库函数的后面
 - C. 必须在最后
 - D. 可以任意

2. 填空题

- (1) C 语言程序中有且只有一个_____,它的名称必须是_____。
- (2) /* 和 */ 之间的内容称为_____,它的作用是_____。
- (3) 组成 C 语言程序的基本单位是_____,其组成部分包括_____和_____。
- (4) 利用 Microsoft Visual Studio 2010 创建的项目文件的扩展名是_____。
- (5) 一个 C 语言程序的执行是从_____开始,到_____函数结束。

3.1.4 综合应用

1. 程序设计一

功能：编写程序，分 3 行输出如下信息：

```
*****
***** Very Good! *****
*****
```

【实验提示】

- (1) 参照 3.1.2 节中的实验内容输入并运行程序及程序填空和编写程序。
- (2) 注意正确使用“\n”回车换行符。

2. 程序设计二

功能：编写程序计算并输出算式 $(123+345+567) \div 3$ 的结果。

【实验提示】

- (1) 参照 3.1.2 节中的实验内容(3)输入并运行程序。
- (2) 算式的功能是计算 3 个数的平均值，需要定义 3 个整型变量，如 a,b,c 分别存放 3 个数，定义 aver 整型变量存放平均值。
- (3) C 语言用符号“/”实现除法运算。
- (4) 请读者自行拟定一个算式，编程计算并输出其运算结果。

3. 输入并运行程序

功能：实现两个整型数据的交换，代码如下：

```
1  #include <stdio.h>
2  void swap(int i,int j)           // swap 函数首部,函数值类型为 void 空类型
3  {                                // 函数体开始的标志
4      int  temp;                   // swap 函数中的声明部分
5      temp = i;                    // 3 个赋值语句,实现 i 与 j 的交换
6      i = j;
7      j = temp;
8      printf("x = % - 4dy = % - 4d\n",i,j); // 输出交换后的数据
9  }                                // 函数体结束的标志
10
11 int main()                       // main 函数首部
12 {                                // 函数体开始的标志
13     int  x=3,y=10;                // 定义变量时分别对变量初始化
14     printf("x = % - 4dy = % - 4d\n",x,y); // 输出交换前的数据
15     swap(x,y);                    // 调用 swap 子函数
16     return 0;                     // 返回函数值 0 到系统,函数执行完毕
17 }                                // 函数体结束的标志
```

【实验提示】

(1) 该程序由 main 函数和 swap 子函数组成,swap 子函数写在 main 函数之前,不需要对 swap 子函数进行声明。swap 子函数的功能是借助中间变量 temp,实现 i 与 j 两个数的交换。

(2) 程序的执行过程如图 3-3 所示。

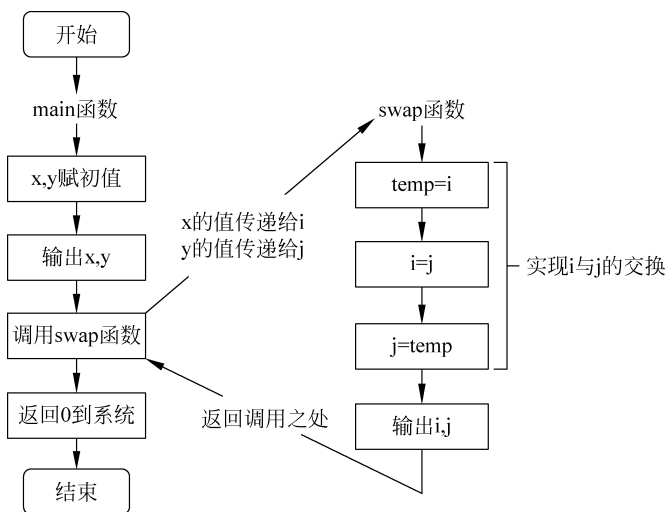


图 3-3 程序的执行过程

(3) printf 函数中的“%-4d”表示整数的输入与输出格式,占 4 字符宽度,负号“-”表示左对齐,右边不足 4 位补空格使之占 4 字符宽度。

(4) 程序中“//”开始到本行结束为注释内容,不参与编译,可以不输入。

(5) 程序中两次调用 printf 函数,请将程序运行结果与之对应。

(6) 根据程序运行结果,理解程序的执行过程。

(7) 将程序中的“%-4d”分别修改为“%4d”与“%d”,比较结果有何不同。

4. 程序填空

功能: 从键盘上输入 3 个整数,输出其中最大者。

请在程序的下画线处填入正确的内容,并把下画线删除,使程序得出正确的结果。

注意: 不要增行或删行,也不得更改程序结构。

代码如下:

```
1  #include <stdio.h>
2  int main()                                // main 函数首部
3  {
4      int max(int,int,int);                // 对被调用 max 函数的声明
5      int a,b,c,d;
6      scanf("%d,%d,%d",&a,&b,&c);          // 输入变量 a,b,c 的值
7      /***** FILL *****/
8      d = _____;                      // 调用 max 函数求最大值
```

```
9      / ***** FILL ***** /
10     printf("max = %d\n", _____);          // 输出最大值
11     return 0;
12 }
13
14 int max(int x, int y, int z)                    // max 函数首部
15 {
16     int t;                                     // 定义变量 t 用于存放 3 个数的最大值
17     if (x > y)                                  // 将 x, y 中的最大值赋给 t
18         t = x;
19     else
20         t = y;
21     if (z > t)                                  // 将 x, y, z 中的最大值赋给 t
22         t = z;
23     / ***** FILL ***** /
24     return _____;                         // 将函数值返回到 main 函数调用之处
25 }
```

【实验提示】

- (1) 参照 3.1.2 节中实验内容(4)输入并运行程序,实现求 3 个数中的最大值。
- (2) 本程序中 max 子函数有 3 个形式参数 x, y, z, 其值来源于 main 函数中的 a, b, c 3 个实际参数。
- (3) 第 15~20 行利用“打擂台”算法求 x, y, z 的最大值,先将 x 与 y 中的最大值存放在变量 t 中,再将变量 t 与变量 z 的最大值存放在变量 t 中。
- (4) 第 24 行将所求最大值返回 main 函数调用之处。
- (5) 程序中“//”开始到本行结束为注释内容,不参与编译,可以不输入。
- (6) 修改程序实现求 3 个数的最小值。

3.2 数据类型、运算符与表达式

计算机程序的主要任务就是对数据进行处理,计算机中的数据不单是简单的数字,计算机处理的信息,包括文字、声音、图像等都是以一定的数据形式存储的。数据在内存中存放的情况由数据类型决定。本节着重介绍 C 语言的基本数据类型、运算符和表达式。

3.2.1 知识点介绍

1. 数据类型

C 语言一个很重要的特点是数据类型十分丰富,因此,C 语言程序的数据处理能力很强。C 语言的数据类型,如图 3-4 所示。

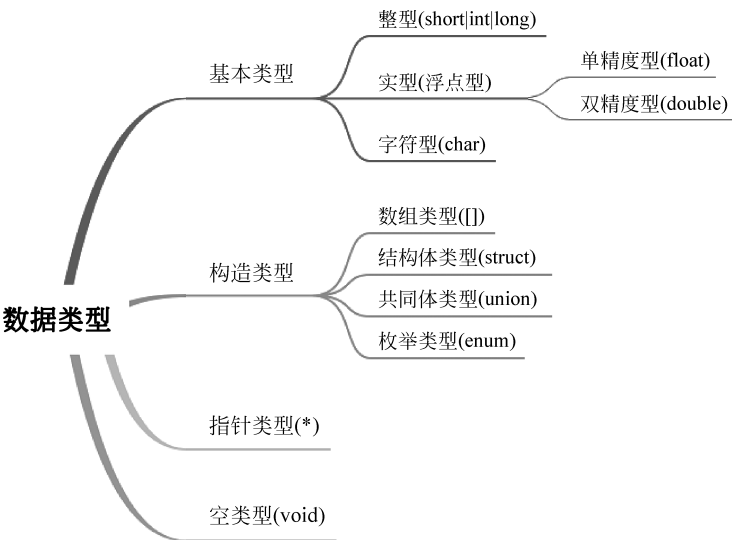


图 3-4 数据类型

2. 常量

在程序运行过程中，其值不能被改变的量称为常量，按数据类型分为以下 5 种整型常量。

(1) 整型常量

整型常量的表示形式有十进制整型常量、八进制整型常量和十六进制整型常量，详见表 3-3。

表 3-3 整型常量表示形式

整型常量表示形式		示 例
十进制整数	如 123，-456	
八进制整数	以 0 开头的数是八进制数。如 0123(相当于十进制数 83)，-011(相当于十进制数-9)	
十六进制整数	以 0x 开头的数是十六进制数。如 0x123(相当于十进制数 291)，-0x12(相当于十进制数-18)	

(2) 实型常量

实型常量的表示形式有小数点形式和指数形式，详见表 3-4。

表 3-4 实型常量表示形式

实型常量表示形式		示 例
十进制小数	如 0.34，-56.79,0.0	
指数形式	格式：尾数+字母 e 或 E+指数，三者缺一不可，如 12.34e3（代表 12.34×10 ³ ），尾数为小数形式，字母 e 或 E 表示底数为 10，指数必须为整数	

(3) 字符常量

字符常量有两种形式，即普通字符和转义字符，详见表 3-5。

表 3-5 字符常量表示形式

字符常量表示形式	示 例
普通字符	用英文单引号括起来的一个字符,如'a','A','0','>'
转义字符	以字符\开头的字符序列,如'\n','\t','\\','\b','\','\"','\a'

(4) 字符串常量

字符串常量是用英文双引号括起来的若干字符序列,如"China","123","I am a boy."

(5) 符号常量

用# define 预编译命令,指定用一个符号名称代表程序中需要反复使用的常量。如:

```
#define PI 3.14159 //注意行末没有分号
```

经上述指定后,符号常量 PI 表示常量 3.141 59。在对程序编译前,预处理器将程序中自本行开始,所有符号常量 PI 替换为 3.141 59。

3. 变量

程序执行过程中值可以发生变化的量,称为变量。变量必须先定义才能使用。在定义变量时需要指定变量的名字,便于程序中引用变量;还需要指定变量的类型,便于在内存中开辟相应存储空间存放变量的值。不同数据类型的变量,在内存中的存储空间和取值范围是不一样的,详见表 3-6。

表 3-6 VS2010 环境下数据类型的存储空间和取值范围

类 型	符 号	关 键 字	字 节 数	数的表示范围
整型	有	(signed)short	2	-32 768~32 767
		(signed)int	4	-2 147 483 648~2 147 483 647
		(signed)long	4	-2 147 483 648~2 147 483 647
	无	unsigned int	2	0~65 535
		unsigned short	4	0~4 294 967 295
		unsigned long	4	0~4 294 967 295
实型	有	float	4	1.2E-38~3.4E38
		double	8	2.3E-308~1.7E308
字符型	有	char	1	-128~127
	无	unsigned char	1	0~255

4. 标识符

标识符是用来标识常量名、变量名、函数名、数组名、类型名、文件名及标号名的有效字符序列,它是一个名字。C 语言规定标识符只能由字母、数字和下画线 3 种字符组成,且第一个字符必须是字母或下画线。标识符分类详见表 3-7。

表 3-7 标识符分类

分 类	说 明
关键字标识符	auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while, _bool, _Complex, _Imaginary 注意: include 与 define 不是 C 语言关键字
用户定义标识符	“见名知意”,关键字不能作为用户定义标识符,区分大小写,如 sum, Sum, aver

5. 运算符与表达式

运算符是用来表示某种运算的符号。表达式是用运算符和括号将运算对象(操作数)连接起来的符合 C 语言语法规则的式子。学习运算符与表达式要从 6 个方面学习,即运算符号、运算符的功能、运算对象、结合方向、优先级、表达式的值。本部分主要介绍算术运算、赋值运算、逗号运算。

(1) 算术运算

算术运算即加(+)、减(-)、乘(*)、除(/)、求余(%)运算,基本运算规则同数学中的算术运算,但需要注意 C 语言中除法与求余运算的运算规则,示例如表 3-8 所示。

表 3-8 除法与求余运算规则

运 算 符	运 算 规 则	示 例
除法(/)运算	若参与除法运算的运算对象均为整型,则商也为整型,直接舍去小数部分;如果运算对象中有一个是实型,则商为实型	5/3=1 5.0/3=1.666667
求余(%)运算	参与运算的运算对象均为整型数,不能对实型数进行求余运算	5%3=2 5.0%3 则不合法

(2) 自增、自减运算

自增(++)、自减(--)运算符均为单目运算符,即一个运算对象,从右到左结合。运算对象只能为变量,不能为常量或表达式,自增(++使变量增加 1,自减(--)使变量减少 1。自增与自减运算示例详见表 3-9。

表 3-9 自增、自减运算示例

语 句	等价的语句	执行该语句后 m 的值	执行该语句后 n 的值	说 明
int m,n=3; m=n++;	m=n; n=n+1;	3	4	先将 n 的值赋给 m,再使 n 自增 1
int m,n=3; m=n--;	m=n; n=n-1;	3	2	先将 n 的值赋给 m,再使 n 自减 1
int m,n=3; m=++n;	n=n+1; m=n;	4	4	先使 n 自增 1,再将 n 的值赋给 m
int m,n=3; m=--n;	n=n-1; m=n;	2	2	先使 n 自减 1,再将 n 的值赋给 m

续表

语 句	等价的语句	执行该语句后 m 的值	执行该语句后 n 的值	说 明
int m=3; m++;	m=m+1;	4	—	m++与++m 作为独立的表达式时,二者等价
int m=3; --m;	m=m-1;	2	—	m--与--m 作为独立的表达式时,二者等价

注意:两个“++”和两个“--”之间不能有空格。

(3) 赋值运算

赋值(=)运算符为双目运算,即两个运算对象,从右到左结合,优先级仅高于逗号运算符。用于给变量赋值,由赋值运算符连接的式子称为赋值表达式,其一般形式及示例如表 3-10 所示。

表 3-10 赋值表达式一般形式及示例

一般形式	变量=常量或变量或表达式或函数调用	
功 能	计算出“=”右边的值,再将值赋给左边的“变量”,赋值表达式的值即为“=”左边变量的值	
示 例	a=3	把常量 3 赋值给变量 a
	x=(-b+sqrt(b*b-4*a*c))/(2*a)	把算术表达式的值赋值给变量 x
	c=max(a,b)	把函数 max 的值赋值给变量 c

赋值运算符具有右结合性,因此可以多重赋值,例如:a=b=c=5 可理解为 a=(b=(c=5))。

(4) 复合赋值运算

在赋值运算符“=”前面加上其他运算符可构成复合赋值运算符,如+=、-=、*=、/=、%=,两个运算符之间不能有空格。复合赋值运算符可以简化赋值表达式。复合赋值运算示例详见表 3-11。

表 3-11 复合赋值运算示例

运 算 符	示 例	等 价 形 式
+=	a += 10; a += b - 1;	a = a + 10; a = a + (b - 1);
-=	a -= 10; a -= b - 1;	a = a - 10; a = a - (b - 1);
*=	a *= 10; a *= b - 1;	a = a * 10; a = a * (b - 1);
/=	a /= 10; a /= b - 1;	a = a / 10; a = a / (b - 1);
%=	a %= 10; a %= b - 1;	a = a % 10; a = a % (b - 1);

赋值运算符可以多重赋值,复合赋值运算也可以实现多重赋值,在进行多重复合赋值运算时,不仅要考虑运算符的结合性,还要考虑运算符的优先级,多重复合赋值运算示例详见表 3-12。

表 3-12 多重复合赋值运算示例

运算步骤	<code>int a = 3;</code> <code>a += a -= a * a;</code>	<code>int a = 3;</code> <code>a += a -= a * a;</code>
第 1 步	计算 <code>a * a</code> , 值为 9, 原 <code>a</code> 的值不变, 仍为 3	计算 <code>a * a</code> , 等价于 <code>a = a * a</code> , 计算得 <code>a = 9</code> , 原 <code>a</code> 的值由 3 变为 9
第 2 步	计算 <code>a -= 9</code> , 等价于 <code>a = a - 9</code> , 计算得 <code>a = -6</code>	计算 <code>a -= 9</code> , 等价于 <code>a = a - 9</code> , 计算得 <code>a = 0</code>
第 3 步	计算 <code>a += -6</code> , 等价于 <code>a = a + (-6)</code> , 计算得 <code>a = -12</code>	计算 <code>a += 0</code> , 等价于 <code>a = a + 0</code> , 计算得 <code>a = 0</code>
最后结果	<code>a = -12</code>	<code>a = 0</code>

(5) 逗号运算

C 语言中的运算符类型丰富,逗号(,)也是一种运算符,称为逗号运算符。逗号运算符从左向右结合,是所有运算符中级别最低的运算符。其功能是把 *n* 个表达式连接起来组成一个表达式,称为逗号表达式。其一般形式及示例如表 3-13 所示。

表 3-13 逗号表达式一般形式及示例

一般形式	表达式 1, 表达式 2, ..., 表达式 <i>n</i>	
功 能	先求解表达式 1, 求解表达式 2, ..., 最后求解表达式 <i>n</i> , 表达式 <i>n</i> 的值是整个逗号表达式的值	
示 例	<code>a = 2, b = 5, a ++,</code> <code>b ++, a + b</code>	表达式 1: <code>a = 2</code> 表达式 2: <code>b = 5</code> 表达式 3: <code>a ++</code> 等价于 <code>++a</code> 等价于 <code>a = a + 1</code> 计算后 <code>a = 3</code> 表达式 4: <code>b ++</code> 等价于 <code>++b</code> 等价于 <code>b = b + 1</code> 计算后 <code>b = 6</code> 表达式 5: <code>a + b</code> 计算得 9 整个逗号表达式的值即为 9

6. 自动类型转换与强制类型转换

(1) 表达式中的自动类型转换

在 C 语言表达式中,若运算对象数据类型不一致时,C 语言编译器在运算之前将所有运算对象的数据类型自动转换成取值范围较大的运算对象的数据类型,称为类型提升。类型提升规则如图 3-5 所示。

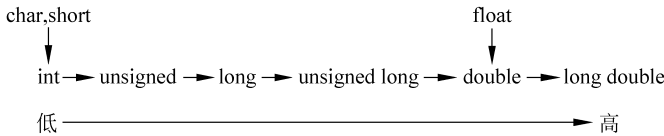


图 3-5 表达式中数据类型自动转换规则

数据类型级别的高低由数据类型在内存中所占字节数大小确定,所占字节数越大,级别越高,表示数的范围越大。图 3-5 中 ↓ 表示必然转换,即所有 `char` 和 `short` 都必须转换成 `int` 数据类型。→ 表示数据类型由低向高的方向自动转换。

(2) 赋值中的自动类型转换

在一个赋值表达式中,若赋值运算符左侧变量的类型与右侧表达式的数据类型不一致,系统编译时会出现警告信息,警告信息不影响程序的执行,系统将自动把右侧表达式值的数据类型转换成左侧变量的数据类型。一般情况下应尽量避免赋值运算符“=”左右两侧数据类型不一致的情况。

(3) 强制类型转换

使用强制类型转换,可以将一个表达式的值的类型,强制转换为用户指定的类型参与运算,有助于消除前面所述的两种自动类型转换而导致的警告信息。强制类型转换一般形式及示例如表 3-14 所示。

表 3-14 强制类型转换一般形式及示例

一般形式	(数据类型)(表达式)	
功 能	将表达式的值强制转换成指定数据类型参与运算,但并不改变表达式原有的数据类型。如果表达式为一个常量或变量时,表达式的括号可以省略	
示 例	float a=3.57; int b; b=(int)a;	把变量 a 的值 3.57 强制转换成 int 类型 3 赋值给 b,但变量 a 仍为 float 类型
	int a=4,b=3; float x,y; x=(float)a/b; y=(float)(a/b);	x=4.0/3=1.333333 把 a 的值强制转换成 4.0 参与运算,避免了整数除以整数 y=(float)(4/3)=1.000000 表达式 4/3=1,然后再将 1 强制转化成 float 型 1.000000

3.2.2 实验部分

1. 实验目的

- (1) 进一步熟悉 VS2010 环境下 C 语言程序编辑、调试、运行的基本操作方法。
- (2) 掌握 C 语言基本数据类型及各种类型常量、变量的表示方法。
- (3) 能够正确定义变量并对其赋值,掌握不同类型数据之间赋值规律。
- (4) 学会使用 C 语言的各种运算符和表达式。
- (5) 了解 C 语言程序结构、基本语法规则及程序书写格式。
- (6) 了解结构化程序设计中最基本的顺序结构程序。

2. 实验内容

(1) 输入并运行程序一

功能: 使用符号常量计算圆的周长和面积。代码如下:

```
1  #include <stdio.h>
2  #define PI 3.14159                //定义符号常量 PI
3  int main()
4  {
5      float r,c,s;                  //定义 3 个变量分别为 float 数据类型
```

```
6    printf("请输入圆的半径:");
7    scanf("%f",&r);                //float 数据类型对应的格式控制符为 %f
8    c = 2 * PI * r;                //计算周长
9    s = PI * r * r;                //计算面积
10   printf("圆的周长: %f\n",c);    //输出周长
11   printf("圆的面积: %f\n",s);    //输出面积
12   return 0;
13 }
```

【实验提示】

① 计算圆的周长与面积需要用到常量 3.141 59,第 2 行定义符号常量 PI 表示 3.141 59。在编译之前,系统会自动把程序中所有 PI 替换成 3.141 59。

② 为了与变量名、函数名、关键字相区别,符号常量一般用大写字母表示。

③ 第 5 行定义 float 型变量 r、c、s 分别表示半径、周长、面积,float 数据类型对应的输入输出格式为 %f。

④ 编译时会提示第 8、9 行有如下警告信息:

warning C4244: “=”表示从 double 转换到 float,可能丢失数据。

原因是系统默认实型常量为 double 双精度数据类型,而赋值运算“=”的左边为 float 数据类型,因左右两边数据类型不一致而出现的警告信息,不影响程序的运行。

为避免警告信息的产生,可以在实型常量末尾加上专用字符“f”或“F”,强制指定实型常量为 float 类型。也可以将 r、c、s 这 3 个变量直接定义为 double 类型。

⑤ 观察程序运行结果,%f 输出的数据小数位数是多少位?

(2) 输入并运行程序二

功能: 字符型数据的定义与使用。代码如下:

```
1  #include <stdio.h>
2  int main()
3  {
4      char c1,c2;                //定义字符变量
5      int x;
6      c1 = 49;                    //ASCII 码 49 对应的字符'1'赋给字符变量 c1
7      c2 = 'A';                  //字符常量'A'赋给字符变量 c2
8      x = c1 + c2;
9      printf("c1 = %c,c2 = %d\n",c1,c2);
10     printf(" %c, %d\n",x,x);
11     printf(" %c\t%c\t%c\t%c\n", 'B', 'B', '\102', '\x42');
12     printf(" %s\t%s\n", "China", "\"China\"" );
13     printf(" %c%c%c\n", '\a', '\a', '\007' );
14     return 0;
15 }
```

【实验提示】

① 字符变量在内存中存放的不是字符本身,而是字符的 ASCII 码。

② 第 8 行,变量 c1 与 c2 存放的字符的 ASCII 码相加,将其和赋给整型变量 x。

③ 第9行,字符型数据按“%c”格式输出字符本身,按“%d”格式输出字符对应的ASCII码。

④ 第10行,值在0~127的整型数据按“%c”格式输出整型数据ASCII码对应的字符。若ASCII码在128~255,按“%c”格式输出,在VS2010环境下会显示为“?”字符,这是因为128~255的ASCII码是扩展的编码,一般不显示。

⑤ 第11~13行,涉及转义字符的应用。'\t'相当于Tab键, '\102'表示八进制数102对应的ASCII码66对应的字符'B'; '\x42'表示十六进制42对应的ASCII码66对应的字符'B'; '\"'表示双引号字符常量; '\a', '\007'代表蜂鸣声、警告声。

⑥ 第9~12行,程序运行会有对应字符信息显示输出,而第13行没有信息显示,输出的是声音,程序运行时戴上耳机可以听到3声蜂鸣声音。如果想听到更多的蜂鸣声,请问应如何修改程序?

(3) 输入并运行程序三

功能: 理解除法与求余运算。代码如下:

```
1  #include <stdio.h>
2  int main()
3  {
4      int  a=14,b=3;
5      float x,y;
6      printf("整数相除:\n");
7      printf("a/b= %d,b/a= %d\n",a/b,b/a);
8      x=a/b;
9      y=b/a;
10     printf("x= %f,y= %f\n",x,y);
11     printf("强制转换:\n");
12     printf("forced transform: %f, %f\n", (float)a/b, (float)(a/b));
13     printf("求余运算:\n");
14     printf("a%%b= %d,b%%a= %d\n", a%b,b%a);
15     return 0;
16 }
```

【实验提示】

① 除法运算,整数除以整数,商为整数。如果不能整除,舍去小数部分,只取整数。

② 求余运算的运算对象为整型数据。

③ 第8~9行赋值运算“=”左右两边类型不一致,编译时出现如下警告信息:

warning C4244: “=”表示从 int 转换到 float,可能丢失数据。

原因是赋值运算“=”左右两边类型不一致,int 型数据赋值给 float 型变量时,直接在 int 型数据后加上“.000000”,使之成为实型数据赋值给左边的 float 型变量。警告信息虽不影响程序运行,但也要尽量避免,在进行赋值运算时尽量使“=”左右两边数据类型一致。

④ 第12行,强制类型转换只是将变量或表达式的值强制转换成某种数据类型参与运算,但并不改变它们实际的数据类型。有时为避免整型数除以整型数,需要将被除数或除数强制类型转换。

⑤ 如果要对表达式的值进行强制类型转换时,需要将表达式用括号括起来。请注意

(float) a/b 只是将变量 a 强制转换成 float 数据类型, (float)(a/b) 是将表达式 a/b 的值强制转换成 float 数据类型, 详见表 3-14。

⑥ 第 14 行 printf 函数的格式控制中连续两个“%%”的作用是输出“%”字符本身。

(4) 程序改错

功能: 学习使用运算符与表达式。修改注释行下面一程序的错误, 不要增行或删行, 也不得更改程序结构。代码如下:

```
1  #include <stdio.h>
2  int main()
3  {
4      int  a,b,c,x,y;
5      / ***** ERROR ***** /
6      int  i=j=0,a;
7      float f,g,h;
8      f=g=h=1.6;
9      / ***** ERROR ***** /
10     x=(a=10,b=100,c=50)
11     y=h;
12     / ***** ERROR ***** /
13     f=+(float)(x+y);
14     c=++a+b--;
15     c--;
16     j=b++%--a;
17     i=g;
18     printf("x=%d,y=%d,f=%f,c=%d,j=%d,i=%d\n",x,y,f,c++,j,i);
19     printf("float=%d,int=%d\n",sizeof(f),sizeof(a+b));
20     return 0;
21 }
```

【实验提示】

① 程序中的 3 处错误均为语法错误, 根据编译提示的错误信息修改程序中的错误。

② 第 10 行将逗号表达式(a=10,b=100,c=50)的值赋给变量 x, 逗号表达式由 3 个赋值表达式组成, 第三个赋值表达式 c=50 的值 50 为整个逗号表达式的值, 因此 x=50。

注意: 赋值运算符的优先级别高于逗号表达式, 因此当把逗号表达式的值赋给某个变量时, 一定要用括号括起来。

③ 第 13 行复合赋值运算: +=、-=、*=、/=、%=, 复合赋值运算由两个运算符构成, 中间不能有空格。

④ 第 13 行(float)(x+y)表示将表达式 x+y 的值强制转换成 float 类型。若对表达式的值强制类型转换时, 表达式一定要用括号括起来, 详见表 3-14。

⑤ 第 14~16 行关于自增(++)与自减(--)运算的使用, 自增运算符(++)和自减运算符(--)只能用于整型变量, 而不能用于常量或表达式, 运算规则详见表 3-9。

⑥ 第 19 行 sizeof 是单目运算符, 用于计算数据类型在内存中所占的字节数。

⑦ 程序编译时会出现如下警告信息, 请思考警告信息产生的原因, 应如何避免警告信息的产生?

warning C4305: “=”表示从 double 到 float 截断。

warning C4244: “=”表示从 float 转换到 int,可能丢失数据。

3.2.3 练习与思考

1. 单选题

- (1) 以下标识符中,不能作为合法的 C 用户定义标识符的是()。
- A. _123 B. void C. a3_b3 D. IF
- (2) 下列选项中,均是 C 语言关键字的选项是()。
- A. if struct type B. switch typedef continue
C. signed union scanf D. auto enum include
- (3) 以下数据中,不正确的数值或字符常量是()。
- A. 0 B. o13 C. 5L D. 9861
- (4) 设变量 a 是整型,f 是实型,i 是双精度型,则表达式 $10 + 'a' + i * f$ 值的数据类型为()。
- A. 不确定 B. double C. int D. float
- (5) `printf("%d \n", (int) (2.5 + 3.0) / 3);` 的输出结果是()。
- A. 2 B. 1
C. 有语法错误不能通过编译 D. 0
- (6) 若以下变量均是整型,且 `num = 7`; 则计算表达式 `sum = num ++, sum ++, ++ num` 后 sum 的值为()。
- A. 10 B. 8 C. 7 D. 9
- (7) 执行下列语句后,a 和 b 的值分别为()。
- ```
int a, b;
a = 1 + 'a';
b = 2 + 7 % (- 4) - 'A';
```
- A. -63   -64                      B. 79   78                      C. 98   -60                      D. 1   -60
- (8) 若 a、b、c、d 都是 int 类型变量且初值为 0,以下选项中不正确的赋值表达式是( )。
- A. `a = b = c = 100`                      B. `d = (c = 22) - (b++)`  
C. `d ++`                      D. `c + b`
- (9) 以下选项中,与 `k = n ++` 完全等价的表达式是( )。
- A. `k = n, n = n + 1`                      B. `n = n + 1, k = n`  
C. `k += n + 1`                      D. `k = ++ n`
- (10) 若有说明语句: `char c = '\72';` 则变量 c( )。
- A. 不合法,c 的值不确定                      B. 包含 3 个字符  
C. 包含 1 个字符                      D. 包含 2 个字符

#### 2. 填空题

- (1) 在 C 语言程序中,用关键字\_\_\_\_\_定义基本整型变量,用关键字\_\_\_\_\_定义单精度实型变量,用关键字\_\_\_\_\_定义双精度实型变量。

- (2) C 语言中,双精度实型类型数据在内存中占\_\_\_\_\_字节。
- (3) 若有定义: `char c = '\x42'`; 则变量 `c` 所表示的字符是\_\_\_\_\_。
- (4) 表达式 `18/4 * sqrt(4.0)/8` 值的数据类型为\_\_\_\_\_。
- (5) 表达式 `(int)((double)(5/2)+2.5)` 的值是\_\_\_\_\_。
- (6) 若有定义: `int a=7; float x=2.5,y=4.7`; 则表达式 `x+a%3*(int)(x+y)%2/4` 的值是\_\_\_\_\_。
- (7) 若有定义: `double a=5.5,b=2.5`; 则表达式 `(int)a+b/b` 的值是\_\_\_\_\_。
- (8) 若有定义: `int a=3`; 则执行完表达式 `a+=a-=a*a`; 后 `a` 的值是\_\_\_\_\_。
- (9) 若有定义: `int k=4`; 计算表达式 `(j=4,k--)` 后, `j` 的值为\_\_\_\_\_和 `k` 的值为\_\_\_\_\_。
- (10) 已知字母 `a` 的 ASCII 十进制代码为 97, 则执行下列语句后的输出结果为\_\_\_\_\_。

```
char a = 'a';
a--;
printf("%d, %c\n", a + '2' - '0', a + '3' - '0');
```

## 3.2.4 综合应用

### 1. 程序填空

功能: 模拟超市结账时将多出的几分进行四舍五入。以下程序运行后实际应付 313.86 元, 而抹去分后实际支付 313.90 元。

```
1 #include<stdio.h>
2 int main()
3 {
4 float pay1 = 56.75f, pay2 = 72.91f, pay3 = 88.90f, pay4 = 26.87f, pay5 = 68.51f;
5 float total; //表示商品实际应付总金额
6 float money; //表示抹分后实付总金额
7 /***** FILL *****/
8 total = _____;
9 printf("实际应付 %.2f 元\n", total);
10 /***** FILL *****/
11 money = _____;
12 printf("抹分之后支付 %.2f 元\n", money);
13 return 0;
14 }
```

请在程序的下画线处填入正确的内容, 并把下画线删除, 使程序得出正确的结果。

**注意:** 不要增行或删行, 也不得更改程序结构。

#### 【实验提示】

(1) C 语言编译器默认实型常量为双精度 `double` 类型, 若希望实型常量以单精度 `float` 类型参与运算, 则在实型常量后添加“`f`”或“`F`”, 如程序第 4 行所示。

(2) 第 9、12 行 printf 函数中的“%.2f”表示 float 型数据输出,小数位数占 2 位,第 3 位四舍五入,整数部分按实际宽度输出。

(3) 实际应付金额多出的几分实现四舍五入的计算公式:  $(total * 10 + 0.5) / 10$ , 程序中需要将分子强制转换为 int 类型,但最后计算实际支付金额时,还需要避免整型数除以整型数。

(4) 思考: 若要将几角几分全部四舍五入抹去取整结账,该如何修改程序?

## 2. 输入并运行程序

功能: 阅读、理解、运行程序,写出程序运行结果。

```
1 #include <stdio.h>
2 int main()
3 {
4 float a;
5 int b,c;
6 char d,e;
7 a = 5.5;
8 b = (int)a;
9 c = b + 'a';
10 d = c;
11 e = '\\';
12 printf("a = %f\\nb = %d\\nc = %d\\nd = %c\\ne = %c\\n", a, b, c, d, e);
13 }
```

### 【实验提示】

(1) 本程序主要学习数据类型的 3 种转换方式: 自动转换、赋值转换和强制类型转换。

(2) 第 7 行,赋值转换,实型常量 5.5 系统默认为 double 类型,系统自动把 5.5 转换为单精度 float 类型,赋值给变量 a,但编译时此行会提示警告信息。

(3) 第 8 行,强制类型转换,把 a 的值 5.5f 强制转换成 int 类型,即舍弃小数部分,将整数 5 赋给变量 b,注意: 变量 a 的值不变,仍为 5.5f。

(4) 第 9 行,先将字符常量 'a' 自动转换为 ASCII 码 97 参与运算,得到整型数 102,赋给变量 c。

(5) 第 10 行,赋值转换,将整数 102 转换成 ASCII 码对应的字符 'e',赋给字符变量 d。

(6) 第 11 行, '\\ ' 为转义字符,表示字符 '\\ '。

## 3. 程序改错

功能: 设圆半径  $r=1.5$ , 圆柱高  $h=3$ , 求圆球表面积、圆球体积、圆柱体积。用 scanf 函数输入数据,输出计算结果,输出时要求有文字说明,取小数点后 2 位数字。修改注释行下面一程序的错误,不要增行或删行,也不得更改程序结构。

```
1 #include<stdio.h>
2 #define PI 3.14159
3 int main()
4 {
```

```
5 float r, h, sq, vq, vz;
6 printf("请输入圆半径 r, 圆柱高 h:");
7 / ***** ERROR ***** /
8 scanf("%f, %f", &r, &h);
9 sq = 4 * PI * r * r;
10 / ***** ERROR ***** /
11 vq = 3/4 * PI * r * r * r;
12 vz = PI * r * r * h;
13 printf("圆球表面积为: sq = %6.2f\n", sq);
14 printf("圆球体积为: vq = %6.2f\n", vq);
15 printf("圆柱体积为: vz = %6.2f\n", vz);
16 return 0;
17 }
```

#### 【实验提示】

- (1) 第 2 行, 程序中反复用到常量 3.141 59, 因此定义符号常量 PI。
- (2) 第 8 行, 根据 scanf 函数的语法规则, 正确使用 scanf 函数。
- (3) 第 11 行, 要注意避免整型数除以整型数。
- (4) 第 13~15 行, “%6.2f”表示 float 型数据输出的格式, 小数位 2 位, 总的宽度 6 位, 实际宽度不足 6 位, 左边补空格; 若实际宽度超过 6 位, 按实际宽度输出。

#### 4. 程序设计

功能: 从键盘输入一个 4 位正整数, 将 4 位正整数的个位、十位、百位、千位分离出来, 求各位数之和, 并将各位数逆序组成一个新的 4 位数, 如输入 1234, 各位数之和为 10, 逆序组成新的 4 位数为 4321。请将下列程序补充完整。

```
1 #include <stdio.h>
2 int main()
3 {
4 int a, b; //存放输入的数及逆序得到的新数
5 int ge, shi, bai, qian; //存放分离的每一位数
6 int sum; //存放各位数之和
7 / ***** BEGIN ***** /
8
9
10 / ***** END ***** /
11 return 0;
12 }
```

#### 【实验提示】

- (1) 分析题目, 将题目中的信息进行抽象, 需要定义 7 个整型变量, 例如用 a、b 分别表示输入的数和逆序得到的新数, ge、shi、bai、qian 分别表示各位数, sum 表示各位数之和。

- (2) 在分离每一位时, 要用到除法运算中整型除以整型商为整型及求余运算, 例如:

千位:  $3478/1000=3$

百位:  $3478/100\%10=4$

十位:  $3478/10\%10=7$

个位:  $3478\%10=8$

(3) 逆序得到的新数: 个位 $\times 1000$  + 十位 $\times 100$  + 百位 $\times 10$  + 千位。

(4) 程序最后输出各位数之和及逆序得到的新数。

### 3.3 实践拓展

经过前面“单词”“短语”“句子”“段落”的学习、练习与应用,应该可以独立地写一篇精彩的“文章”了。为了让文章更好看、更实用,在这里通过两个拓展练习学习如何设置好看的运行窗口和打印有趣的图案。

#### 3.3.1 设置运行窗口字体及背景颜色

C语言运行窗口默认设置为黑底白字,其实计算机的输出可以是彩色的,右击运行窗口标题栏,打开“属性”对话框可以设置窗口字体及背景颜色。另外,还可以编程调用 Windows 系统命令实现窗口字体及背景颜色的设置。输入并运行如下程序:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 /* 十六进制数表示颜色
4 0-黑 1-蓝 2-绿 3-浅绿 4-红 5-紫 6-黄 7-白 8-灰
5 9-淡蓝 A-淡绿 B-淡浅绿 C-淡红 D-淡紫 E-淡黄 F-亮白 */
6 int main()
7 {
8 system("title 劝学"); //设置运行窗口标题
9 system("mode con cols = 80 lines = 40"); //设置窗口大小,宽度为 80px,高度为 40px
10 system("color F4"); //设置运行窗口背景色为 F-亮白,文字颜色为 4-红
11 printf("\t «劝学»\n\n");
12 printf("\t 颜真卿\n\n");
13 printf("\t 三更灯火五更鸡,\n\n");
14 printf("\t 正是男儿读书时。\n\n");
15 printf("\t 黑发不知勤学早,\n\n");
16 printf("\t 白首方悔读书迟。\n\n");
17 printf("\n");
18 return 0;
19 }
```

#### 【实验提示】

(1) 利用 system 函数调用 Windows 系统命令,来设置 C 语言运行窗口的字体及背景颜色,它包含在头文件 #include <stdlib.h> 中。

(2) 第 3~5 行,多行注释,说明用十六进制数表示的各种颜色。

(3) 第 8 行,利用 system 函数调用 Windows 系统外部命令 title,设置运行窗口标题为“劝学”。

(4) 第 9 行,利用 system 函数调用 Windows 系统外部命令 mode con,设置窗口宽度为 80px,高度为 40px。

(5) 第 10 行,利用 `system` 函数调用 Windows 系统外部命令 `color`,设置颜色。`color` 命令后面有两个十六进制数,第一个十六进制数表示背景颜色,如 F 表示背景颜色为亮白色,第二个十六进制数表示文字颜色,如 4 表示红色。如果 `color` 后面只有一个十六进制数,则该数表示文字颜色,背景默认为黑色。

(6) 第 11~17 行,输出唐朝诗人颜真卿的著名诗篇《劝学》,程序中转义字符 `'\t'` 相当于 Tab 键,将当前位置移到一个 Tab 键位置。

### 3.3.2 打印绿底白字小飞机图案

读者可以利用前面所学的知识,尝试以编程的方式让计算机打印如图 3-6 所示的绿底白字小飞机图案。

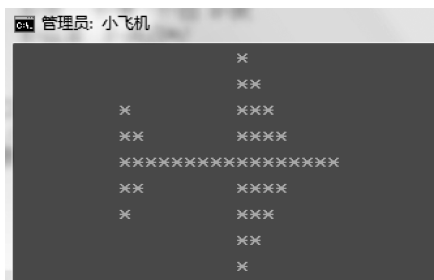


图 3-6 绿底白字小飞机图案

#### 【实验提示】

- (1) 模仿 3.3.1 节的程序,利用 `system` 函数设置运行窗口的标题、背景颜色、文字颜色。
- (2) 分 9 行输出模拟飞机形状,合理使用转义字符 `'\t'`, `'\n'`。
- (3) 修改程序实现一面小红旗、五角星等图案的输出显示。