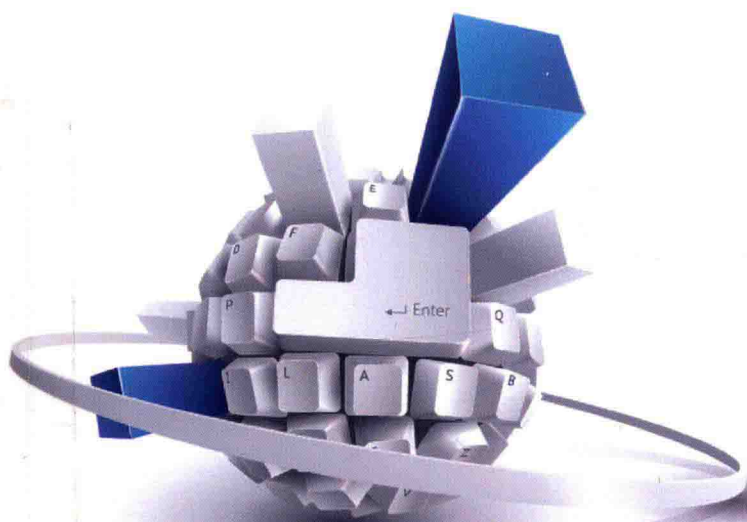


计算机类专业人才培养内涵建设项目系列教材

# C语言程序设计 与实训教程

上册

主编 蔡红



WUHAN UNIVERSITY PRESS

武汉大学出版社

计算机类专业人才培养内涵建设项目系列教材

# C 语言程序设计与实训教程 (上册)

主 编 蔡 红  
副主编 胡 悦 严 岚  
参 编 王伟卿



WUHAN UNIVERSITY PRESS

武汉大学出版社

## 图书在版编目(CIP)数据

C 语言程序设计与实训教程.上册/蔡红主编. —武汉:武汉大学出版社,2016.8  
计算机类专业人才培养内涵建设项目系列教材

ISBN 978-7-307-18568-5

I. C… II. 蔡… III. C 语言—程序设计—高等学校—教材 IV. TP312.8

中国版本图书馆 CIP 数据核字(2016)第 203419 号

责任编辑:张欣 曲生伟

责任校对:刘小娟

装帧设计:张希玉

---

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:whu\_publish@163.com 网址:www.stmpress.cn)

印刷:虎彩印艺股份有限公司

开本:787×1092 1/16 印张:12 字数:307千字

版次:2016年8月第1版 2016年8月第1次印刷

ISBN 978-7-307-18568-5 定价:30.00元

---

版权所有,不得翻印;凡购买我社的图书,如有质量问题,请与当地图书销售部门联系调换。

# 计算机类专业人才培养内涵建设项目系列教材 编写委员会

主任	刘新宇				
副主任	黄群	张健			
委员	蔡红	支永昌	田春	吴建平	
	孔令	朱景德	杨柳		

# 前 言

C 语言具有功能丰富,表达能力强,使用灵活、方便,目标程序效率高,可移植性好等特点,其广泛应用于事务处理、科学计算、工业控制及数据库等领域。

根据高职、高专的课程设置体系,C 语言被推荐为计算机类专业学生的必修课程。然而,与其他程序设计语言相比,C 语言学习难度比较大,尤其对于中高职贯通计算机类专业中职阶段学生来说,学习 C 语言是一件难度非常大的事情。因此,提供一本适合中高职教学的教材是关键因素。为此,编者根据中高职贯通计算机类专业学生的具体情况,依据十几年 C 语言的教学经验,编写了本书。

本书具有以下几个方面的特色:

1. 门槛较低,不要求具备程序设计语言的基础知识。

本书从程序设计的最基础知识讲起,例题由浅入深设计,将经典算法的来龙去脉讲解清楚,读者可以不具备程序设计语言的入门知识。

2. 以场景实例为先导,尽可能清晰易懂,并带有趣味性。

充分考虑中高职院校学生的学习特点、学习基础和接受能力,以场景实例为先导,突出职业技能培养。结合大量例题、课堂实践题、有趣的能力拓展题,激发学生的学习兴趣。

3. 深入浅出,通俗易懂,启发思维,教学互动。

结构清晰、衔接自然。实训题结合前导所讲的知识点,深入浅出。总体根据知识、能力、结构要求,按提出问题、分析问题、解决问题的步骤完成程序设计。

4. 强化实践,重视应用,例题丰富。

重视实践教学环节,结合实际应用,每项任务训练中都列举了大量的例题。通过例题,学生不仅仅可以掌握 C 语言的语法、语句,更重要的是可以具备利用编程解决实际应用问题的能力。

5. 强调良好的编程习惯。

给出合理的编程风格,指出采用统一的编程风格的重要性。

本书由蔡红担任主编,胡悦和严岚担任副主编,王伟卿担任参编。其中,任务 1 和任务 3 由胡悦编写,任务 2 和任务 4 由严岚编写,任务 5 由蔡红编写,附录 1 和附录 2 由蔡红和胡悦编写。王伟卿作为企业工程师为整本书的设计及内容安排提供了很多有益的意见和建议。全书由蔡红统稿、审核。



本书提供《C语言库函数速查手册》和《C/C++语言参考》以方便读者自学,可登录网址 <http://www.stmpress.cn/html/2016/jsdg-0815/33.html> 自行下载。

由于编者水平有限,书中疏漏及错误之处在所难免,敬请读者提出意见,以便作进一步的完善。

编 者

2016年6月

# 目 录

<b>任务 1 C 语言简单程序设计</b> .....	1
1.1 C 语言程序案例一:输出图形 .....	2
1.2 C 语言程序案例二:爸爸妈妈谁大? .....	6
1.3 C 语言程序案例三:可爱的运算器 .....	11
1.4 程序与程序设计 .....	13
1.5 算法的概念 .....	14
1.6 能力拓展 .....	16
<b>任务 2 顺序程序设计能力训练</b> .....	18
2.1 常量应用任务训练 .....	19
2.2 变量应用任务训练 .....	31
2.3 表达式/运算符应用任务训练 .....	37
2.4 能力拓展 .....	49
<b>任务 3 选择程序设计能力训练</b> .....	53
3.1 单分支算法设计任务训练 .....	54
3.2 双分支算法设计任务训练 .....	60
3.3 多分支算法设计任务训练 .....	66
3.4 能力拓展 .....	80
<b>任务 4 循环程序设计能力训练</b> .....	82
4.1 for 语句循环设计任务训练 .....	83
4.2 while 语句循环设计任务训练 .....	93
4.3 do... while 语句循环设计任务训练 .....	103
4.4 循环嵌套与循环中断设计任务训练 .....	116
4.5 能力拓展 .....	135
<b>任务 5 排序(数组)程序设计能力训练</b> .....	140
5.1 一维数组应用任务训练 .....	141
5.2 二维数组应用任务训练 .....	151



5.3 字符数组应用任务训练 .....	158
5.4 能力拓展 .....	168
附录 1 使用 Dev-C++ 开发程序的步骤 .....	178
附录 2 ASCII 码字符表 .....	182
参考文献 .....	184

# 任务 1

## C 语言简单程序设计



### ◆ 学习目标

在本任务中,我们将学习如下内容:

- 用 C 语言编写简单的程序;
- 基本输入输出语句的使用;
- 基本数据类型的使用;
- 算术运算符的使用;
- 流程图的作用和绘制。

### ◆ 重点、难点

- 使用格式化输入函数 scanf, 格式化输出函数 printf;
- 使用逻辑表达式判断变量关系;
- 使用流程图分析、设计算法。

### ◆ 提 纲

- 1.1 C 语言程序案例一: 输出图形
- 1.2 C 语言程序案例二: 爸爸妈妈谁大?
- 1.3 C 语言程序案例三: 可爱的运算器
- 1.4 程序与程序设计
- 1.5 算法的概念
- 1.6 能力拓展

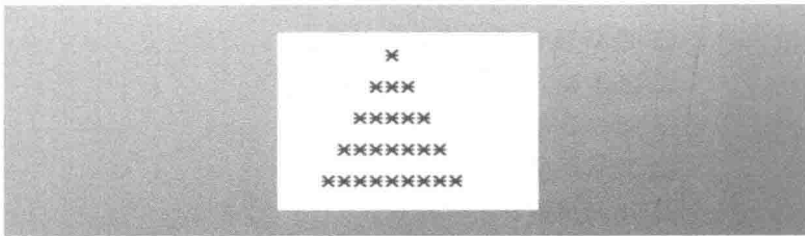




什么是 C 语言? C 语言是贝尔实验室的 Ken Thompson、Dennis Ritchie 等人开发的 UNIX 操作系统的“副产品”。C 语言是一种底层语言,不仅提供了对机器级概念(例如,字节和地址)的访问,还提供了计算机内置指令紧密协调的操作,使得程序可以快速执行。C 语言是一种小型语言,与其他许多编程语言相比,提供了一套更有限的特性集合,为此,C 语言在很大程度上依赖一个标准函数的“库”。C 语言是一种包容性语言,提供了比其他许多编程语言更广阔的自由度。此外,C 语言不像其他语言那样强制进行详细的错误检查。下面我们先运行几个小程序,就此揭开 C 语言的神秘面纱。

## 1.1 C 语言程序案例一:输出图形

请使用 C 语言在屏幕上输出一个等腰三角形。



**【问题分析】** 一个 C 语言程序是由一个或多个具有独立功能的模块组成的,这些模块称为函数。在这些函数中,必须有函数名为 main 的函数,该函数称为主函数。输出等腰三角形图形,可以使用 printf 函数实现。printf 函数是系统提供的标准输出函数,可在程序中直接调用。



### 涉及知识点

#### 1.1.1 main 函数

C 语言程序是由一个或者多个函数组成的,其中必须有一个函数是 main 函数。main 之后有一对圆括号,表示 main 是一个被称为函数的程序模块。所有的 C 语言程序都从 main 函数开始执行,函数在执行前可以通过圆括号接收一些信息,如果不需要接收信息,则圆括号中可为 void 或空。圆括号后面紧跟一对花括号,左花括号“{”表示 main 函数的函数体的开始;对应的右花括号“}”表示函数体的结束。一对花括号及位于其内的程序段被称为一个模块。模块是一种重要的 C 语言程序单元。



main 函数一般形式为:

```
int main(void)           关键字 int 表示 main 函数执行后将返回一个整数
void main(void)         main 后的关键字 void 表示 main 函数无接收信息
void main()             main 前的关键字 void 表示 main 函数无返回值
```

### 1.1.2 格式化输出 printf 函数

printf 函数的作用是显示格式串的内容,在格式串的指定位置插入需要的值。调用 printf 函数时需要提供格式串,格式串后的参数是将在显示时插入该串中的值。

printf 函数一般形式为:

```
printf(格式串,表达式 1,表达式 2,...);
```

显示的值可以是常量、变量或者更复杂的表达式,常量与变量的介绍见本任务后面的内容。printf 函数中的表达式的个数没有限制,当然,表达式也可以没有。

例如,显示一句话“Hello World!”,可写为:

```
printf("Hello World");
```

格式串必须使用双引号进行着重说明,其中包含格式字符串和普通字符串两种。格式字符串是以 % 开头的字符串,在 % 后面跟有各种格式字符,以说明输出数据的类型、形式、长度、小数位数等。普通字符串在输出时原样打印,在显示中起提示作用。例如:

```
printf("%c,%c\n",a,b);
```

语句中格式控制字符串为“%c,%c\n”,其中格式字符串有两个“%c”,普通字符串有“,”和控制符“\n”。

在格式字符串的“%”后面跟有各种格式字符,以说明输出数据的类型、形式、长度、小数位数等。如“%d”表示按十进制整型输出,“%ld”表示按十进制长整型输出,“%c”表示按字符型输出等。具体的格式字符所对应的数据类型见表 1-1。

表 1-1

printf 函数中的格式字符表

格式字符	格式字符意义
d	以十进制形式输出带符号整数(正数不输出符号)
o	以八进制形式输出无符号整数(不输出前缀 0)
x	以十六进制形式输出无符号整数(不输出前缀 0x)
u	以十进制形式输出无符号整数
f	以小数形式输出单、双精度实数
e	以指数形式输出单、双精度实数
c	输出单个字符
s	输出字符串



## 程序

## printf 函数应用

【例 1-1】 无符号数据的输出。

unsigned.c:

```

1 #include <stdio.h>
2 main()
3 { //定义整型变量a,b,并赋值
4   int a=65535,b=-3;
5   //分别用%d,%o,%x,%u四种格式输出a的值
6   printf("a=%d,%o,%x,%u\n",a,a,a,a);
7   //分别用%d,%o,%x,%u四种格式输出b的值
8   printf("b=%d,%o,%x,%u\n",b,b,b,b);
9 }
```

将标准输入输出头文件包含到程序中，  
用于调用printf函数

程序的运行结果：

```
a=65535,177777,ffff,65535
```

```
b=-3,37777777775,ffffffd,4294967293
```

## printf 用法小贴士

对于输出数据的宽度设定,可在“%”和格式字符之间加上一个整数。例如:显示一个整数 123,printf(“%5d\n%d”,123,123),则先两个空格后跟一个“123”,换行之后从第一个字符开始显示“123”。

【例 1-2】 字符串的输出。

string.c:

```

1 #include <stdio.h>
2 main()
3 { //"-表示显示的字符串靠左对齐(适用于所有输出格式),
4   //".3"则表示只显示该字符串的左端3位字符
5   printf("%3s\n%7s\n%-5.3s\n","CHINA","CHINA","CHINA");
6 }
```

程序的运行结果：

```
CHINA
  CHINA
CHI
```

**【例 1-3】** 输出实数时指定小数位数。

dicimalnum.c:

```
1 #include <stdio.h>
2 main()
3 {
4     float f=123.456;
5     //“10.2”表示f显示的宽度为10位，其中显示小数部分取2位
6     //“-”表示显示的数字靠左对齐
7     printf("%f\n%10f\n%10.2f\n%.2f\n%-10.2f",f, f, f, f, f);
8 }
```

程序的运行结果:

```
123.456001
123.456001
      123.46
123.46
123.46
```

**printf 用法小贴士**

printf 函数可以使显示的数据格式标准,但是在使用过程中必须注意以下几点:

- (1) 格式字符串中需要使用小写字母,例如“%d”,而不能是“%D”。
- (2) 在格式字符串中可以使用各种“转义字符”,如“\n”“\t”等。
- (3) 在格式字符与“%”之间只能添加整数或精度格式符,否则该格式字符将成为普通字符而输出。

(4) 若想显示“%”,则需连续使用两个“%”表示,如:

```
printf("%f\\%%",0.33333),显示为“0.33333%”。
```

**返回案例一**

**【问题分析】** 三角形等图案可以使用 printf 输出多个字符串,缩进可用空格或前面所讲的指定宽度来实现。

heart.c:

```
1 #include <stdio.h>
2
3 main()
4 {
5     printf("      *      \n");
6     printf("     ***     \n");
7     printf("    *****  \n");
8     printf("   *         \n");
9     printf("  *         \n");
10 }
```

## 1.2 C 语言程序案例二:爸爸妈妈谁大?

请输入爸爸、妈妈的年龄,然后输出谁的年龄大。

请输入爸爸和妈妈的年龄: 28 26

爸爸的年龄大

**【问题分析】** 爸爸、妈妈的年龄需要通过键盘存储到程序中,从键盘输入数据可以使用 scanf 函数。scanf 函数是系统提供的标准输入函数,可在程序中直接调用。本例输入的数据需要存储在整型变量中,输入数据的大小比较需要使用关系运算符、判等运算符。



### 涉及知识点

#### 1.2.1 变量

变量是指在程序运行过程中其值可以发生变化的量。其通常用来保存程序运行过程中的输入数据、计算中的中间结果和最终结果。变量有如下特征:

- (1) 变量就是程序数据的“容器”。变量中存放的数据称为变量值。
- (2) 每个变量都有一个名字。该名字指定了程序中使用的一个存储单元,用以存放变量值,这个名字称为变量名。
- (3) 每个变量应属于一个特定的类型,被分配内存空间的大小决定了“容器”的大小。

例如,定义一个整型变量  $a$ ,可写为: `int a`。

为变量起名要遵循 C 语言关于标识符的规定。标识符只能由字母、数字和下画线三种字符组成,且第一个字符必须为字母或下画线,下面是正确或错误标识符命名。

正确形式

`count`

`test23`

`high_balance`

错误形式

`3count`(以数字开头)

`hi! There`(含有非英文字母和空格字符)

`int`(这是 C 语言的关键字)

下面是标识符命名的一些原则和约定:

(1) 大写字母和小写字母被认为是两个不同的字符。注意 `total` 和 `Total` 是两个不同的标识符。习惯上,变量名用小写字母表示,符号常量用大写字母表示,如 `PI`、`MAX`、`MIN` 等。



(2) 当为变量选取名时,最好使用“见名知意”的标识符,即变量的名字应该能够描述出它所存储的值的意义,如表示一位学生有关信息名的变量可用 name、sex、age、score 等,以增加可读性。

(3) 标识符不能和 C 语言的保留字相同,保留字是编辑程序本身所使用的标识符。为了避免误解,标识符不能和 C 语言库函数同名,否则将失去系统规定的原意。例如 sin、cos 等。

### 1.2.2 整型变量

整型变量的基本类型符为 int。基本类型可以通过加入一些限定词来改变数据类型的意义,以便更加准确地适应各种不同的情况。这些加上限定的数据类型被称为派生数据类型。

根据数值的范围可以将整型变量分为基本整型、短整型、长整型和无符号型,具体如下所述。

(1) 基本整型:类型说明符为 int,在内存中占 2 个字节。

(2) 短整型:类型说明符为 short int 或 short,所占字节和取值范围均与基本整型相同。

(3) 长整型:类型说明符为 long int 或 long,在内存中占 4 个字节。

(4) 无符号型:类型说明符为 unsigned。无符号型又可与上述三种类型匹配而构成如下类型。

① 无符号基本整型:类型说明符为 unsigned int 或 unsigned。

② 无符号短整型:类型说明符为 unsigned short。

③ 无符号长整型:类型说明符为 unsigned long。

在整型数据类型中,当数值不会出现负数的时候,亦即只能处理非负值,就可以声明为无符号整型变量。一个无符号整型变量中可以存放的正数范围比一般整型变量中正数范围扩大一倍。如果在程序中定义 a 和 b 两个变量:

```
int a;           /* 声明 a 为基本整型 */
unsigned int b; /* 声明 b 为无符号基本整型 */
```

则变量 a 的取值范围为  $-32768 \sim +32767$ 。而变量 b 的取值范围为  $0 \sim 65535$ 。

表 1-2 列出了整型数据类型符及其含义,并列出数的范围,用户可根据数的范围来选用。

表 1-2 整型数据类型符

数据类型	数据类型符	占用字节数	数值范围
基本整型	int	2	$-32768 \sim +32767$
短整型	short	2	$-32768 \sim +32767$
长整型	long	4	$-2147483648 \sim +2147483647$
无符号基本整型	unsigned int	2	$0 \sim 65535$
无符号短整型	unsigned short	2	$0 \sim 65535$
无符号长整型	unsigned long	4	$0 \sim 4294967295$



### 1.2.3 变量的定义

C语言规定,任何变量都必须“先定义,后使用”,定义的内容包括:变量的类型、名字及初值。

变量的定义格式为:

类型说明符 变量名表;

其中,类型说明符包括 int、float、double、char 等,用来指定变量的数据类型;变量名表如果有多个变量,则彼此间要用逗号分隔开,分号是语句结束符。

下面是一些变量的声明语句的实例:

```
int a;                /* 定义一个类型为整型的变量 a,占 2 字节空间 */
long d,e,f;          /* 定义 d,e,f 三个变量同属于长整型,每个变量占 4 个字
                    节 */
int i=0,y=10,k=10;   /* 声明变量的同时为其设定初值,特别注意在设定初值的
                    时候,必须一个变量一个变量地分别设定,即使是那些变量的
                    初值相同 */
```

### 1.2.4 格式化输入 scanf 函数

printf 函数使用给定的格式输出信息,scanf 函数则是使用给定的格式输入信息。像 printf 函数的格式串一样,scanf 函数的格式串也包含普通字符和转换说明两部分。scanf 函数转换说明的用法和 printf 函数转换说明的用法本质上是一样的。

scanf 函数一般形式为:

scanf("格式控制字符串",地址列表);

在许多情况下,scanf 函数的格式只包含转换说明,例如:

```
int i,j;
float a,b;
scanf("%d%d%f%f",&i,&j,&a,&b);
```

#### scanf 函数用法小贴士

(1)scanf 函数的格式控制字符串中没有转义字符。如 scanf("%d\n",&a);是错误的。

(2)scanf 函数的格式控制字符串中的普通字符不是用来显示的,而是输入时要求照普通字符输入的。

(3)scanf 函数的格式控制字符串中转换说明的数量应与输入变量的数量一致。

(4)scanf 函数中参数的第二部分一定是地址列表,不能是表达式,每个参数前要加取址运算符“&”。



## 程序

## scanf 函数应用

【例 1-4】 输入整型值,并显示其值。

```
scanfint.c:
1 #include <stdio.h>
2 main()
3 {
4     int a,b,c;
5     printf("input a,b,c\n");
6     //整型数用%d, 输入数据是要与控制格式相符
7     scanf("%d%d%d",&a,&b,&c);
8     printf("a=%d,b=%d,c=%d",a,b,c);
9 }
```

将标准输入输出头文件包含到程序中,用于调用scanf函数和printf函数

程序的运行结果:

```
input a,b,c
15 38 97

a=15,b=38,c=97
```

## 1.2.5 逻辑表达式

可执行的 C 语言可以是执行操作(例如,数据的输入或输出,对数据进行运算),也可以是做出判断(例如,判断某位同学的成绩是否及格)。if 语句能够判断一个称为条件的语句是真话还是假话。若条件满足,即条件为真(True),则执行 if 语句体中的语句;若条件不满足,即条件为假(False),则不执行 if 语句体中的语句。

if 语句中的条件通常是由关系运算符、判等运算符和逻辑运算符构成的表达式。

## 1. 关系运算符

C 语言中的关系运算符(表 1-3)用于比较运算,与数学中的  $<$ 、 $>$ 、 $\leq$  和  $\geq$  运算符相对应,不同的是在 C 语言中,关系表达式结果是 1(真)或 0(假)。例如,表达式  $4 < 5$  为真,结果为 1;而表达式  $4 > 5$  为假,结果为 0。

表 1-3

关系运算符

符号	含义	示例
$<$	小于	$x < y$
$>$	大于	$x > y$
$\leq$	小于等于	$x \leq y$
$\geq$	大于等于	$x \geq y$