

全国计算机等级考试培训教材

(二级·上册)

主编：潘悦飞

黄河出版社

全国计算机等级考试培训教材

(二级·上册)

主 编 潘悦飞

副主编 曹灵芝 林晓娟

黄河出版社

责任编辑 张清训 葛春亮 封面设计 蒋士雷 张宪峰

图书在版编目(CIP)数据

全国计算机等级考试培训教材(二级)/潘悦飞主编.

济南:黄河出版社,2007.4

ISBN 978-7-80152-800-1

I. 全… II. 潘… III. 电子计算机—水平考试—教材

IV. TP3

中国版本图书馆 CIP 数据核字 (2007) 第 038347 号

书名 全国计算机等级考试培训教材(二级·上册)

主编 潘悦飞

出版 黄河出版社

发行 黄河出版社发行部

(济南市英雄山路 21 号 250002)

印刷 济南和平商务有限公司

规格 850 毫米×1168 毫米 16 开本

11.5 印张 280 千字

版次 2006 年 4 月第 1 版

印次 2006 年 4 月第 1 次印刷

印数 1-2000 册

书号 ISBN 978-7-80152-800-1/G·159

定价 60.00 元(全三册)

目 录

第一章 C 语言概述

第一节 计算机系统的基本组成	1
1.1 基本组成概述	1
1.2 计算机系统的硬件	2
1.3 计算机系统的软件	3
第二节 C 语言程序的构成和格式	4
2.1 在屏幕上输出“my first program!”	4
2.2 在屏幕上输出三排“*”号	5
2.3 在屏幕上输出数据值 100 和 100.1	6
2.4 求矩形周长	7
2.5 编写函数 fun, 其功能是求两个整数的和	8

第二章 数据类型、输入输出操作和常用运算符

第一节 标识符、常量、变量	9
1.1 标识符	9
1.2 常量	10
1.3 变量	10
第二节 C 语言中的基本数据类型	11
2.1 进制转换	11
2.2 整数类型	12
2.3 实数类型	15
2.4 字符类型	16
第三节 数据输出	18
3.1 printf 函数的调用格式	18
3.2 printf 函数中常用的格式符	18
3.3 调用 printf 函数时的注意事项	20
3.4 用 putchar 输出字符类型数据	21
第四节 数据输入	21
4.1 scanf 函数的调用格式	21
4.2 scanf 函数中常用的格式说明	22
4.3 调用 scanf 函数时的注意事项	24
4.4 用 getchar 函数输入字符类型数据	24
第五节 C 语言中常用的运算符	25
5.1 算术运算符	25

5.2	强制类型转换运算符	26
5.3	赋值运算符	27
5.4	自增自减运算符	29
5.5	逗号运算符	30
5.6	关系运算符	30
5.7	逻辑运算符	32
5.8	复合语句和空语句	34

第三章 函数

第一节	函数的分类	37
1.1	主函数和一般函数	37
1.2	库函数和用户自定义函数	37
1.3	主调函数和被调函数	38
第二节	用户自定义函数的定义和返回值	39
2.1	用户自定义函数的定义	39
2.2	函数的返回值	41
2.3	函数的调用和函数说明	42
第三节	实在参数和形式参数之间的数据传递	44

第四章 选择结构

第一节	程序设计的基本知识	48
1.1	程序设计的基本步骤	48
1.2	算法	48
1.3	流程图	49
第二节	结构化程序设计	50
2.1	顺序结构	50
2.2	选择结构	50
2.3	循环结构	51
第三节	由 if 语句构成的选择结构	53
3.1	不带 else 的简单 if 语句	53
3.2	带 else 的简单 if 语句	56
3.3	嵌套的 if 语句	57
第四节	条件表达式	60
第五节	switch 语句	61
5.1	switch 语句	61
5.2	break 语句	62

第五章 循环结构

第一节 while 语句	66
1.1 while 循环流程图	66
1.2 编写程序, 求 $1+2+3+\cdots+100$ 的值	66
1.3 while 循环的一般形式	68
1.4 while 循环的执行过程	68
1.5 编程, 数字字符转换成对应的整型数据	68
第二节 for 语句和用 for 语句构成的循环结构	69
2.1 用 for 循环编写程序, 求 $1+2+3+\cdots+100$ 的值	69
2.2 for 循环的一般形式	69
2.3 for 循环语句流程图及其执行过程	69
2.4 格式说明	70
2.5 编程	70
第三节 do-while 语句	71
3.1 直到型循环的流程图	71
3.2 用 do-while 语句编写程序, 求 $1+2+3+\cdots+100$ 的值	71
3.3 do-while 循环一般格式	72
3.4 编程	72
第四节 循环结构的嵌套	73
第五节 break 和 continue 在循环体中的应用	75
5.1 break 语句	75
5.2 continue 语句	76

第六章 指针

第一节 变量的地址和指针	81
1.1 变量与单元	81
1.2 指针变量	81
第二节 指针变量的定义和赋值	82
2.1 定义格式	82
2.2 指针变量的类型和基类型	83
2.3 指针变量的赋值	83
2.4 给指针变量赋空值	84
第三节 指针变量的操作	84
3.1 通过指针变量引用存贮单元	84
3.2 移动指针	85
3.3 指针比较	86

第四节 函数之间地址值的传递·····	86
4.1 数据传递·····	86
4.2 函数间返回数据·····	88
第七章 数组	
第一节 一维数组的定义和引用·····	90
1.1 变量、数组简介·····	90
1.2 一维数组的定义·····	91
1.3 一维数组的初始化·····	91
1.4 一维数组元素的赋值·····	92
1.5 一维数组元素的引用·····	92
1.6 一维数组元素的遍历·····	92
第二节 一维数组与指针·····	93
2.1 三个运算符·····	93
2.2 一维数组名代表数组首地址·····	94
2.3 用数组名表示一维数组·····	94
2.4 用指针变量表示一维数组·····	95
2.5 函数之间对一维数组和数组元素的引用·····	96
第三节 一维数组应用举例·····	97
第四节 二维数组的定义和引用·····	98
4.1 二维数组的定义·····	98
4.2 二维数组的初始化·····	99
4.3 二维数组元素的赋值·····	101
4.4 二维数组元素的引用·····	101
4.5 二维数组元素的遍历·····	102
第五节 二维数组与指针·····	102
5.1 二维数组名为二重指针常量·····	102
5.2 把二维数组看成两个一维数组·····	103
5.3 用指针数组表示二维数组·····	104
5.4 用行指针来表示二维数组·····	105
第六节 二维数组举例·····	106
第八章 字符串	
第一节 字符串常量·····	110
1.1 字符串常量·····	110
1.2 字符串常量的返回值·····	111
第二节 用一维字符数组表示字符串·····	111

2.1	用一维字符数组表示字符串的要求	111
2.2	一维字符数组初始化为字符串	112
2.3	一维字符数组赋值为字符串	113
第三节	用指针表示字符串	113
3.1	指针初始化为字符串	113
3.2	指针赋值为字符串	114
3.3	两种表示字符串方式的区别	114
第四节	字符串的输入与输出	115
4.1	用 printf 输出字符串	115
4.2	用 scanf 输入字符串	115
4.3	用 puts 输出字符串	116
4.4	用 gets 输入字符串	116
第五节	用于字符串处理的函数	116
5.1	字符串复制函数	116
5.2	字符串连接函数	117
5.3	求字符串长度函数	117
5.4	字符串比较函数	118
第六节	字符串数组	118
6.1	用二维字符数组表示字符串数组	118
6.2	用字符型指针数组表示字符串数组	119
第九章 结构体、共用体和用户自定义类型		
第一节	用 typedef 说明一种新类型	123
第二节	结构体类型和结构体数组	123
2.1	对结构体类型的现实需求	123
2.2	结构体的类型说明	124
2.3	定义结构体类型的变量、数组和指针	124
2.4	结构体变量的初始化	127
2.5	结构体数组的初始化	127
2.6	引用结构体变量中的数据	128
2.7	结构体数组举例	129
第三节	存储分配	130
3.1	静态存储分配	130
3.2	动态存储分配	131
第四节	链表	133
4.1	一种特殊的结构体	133

4.2	静态链表	133
4.3	带头结点动态链表的创建	134
4.4	带头结点动态链表的遍历	136
4.5	动态链表的插入	136
4.6	动态链表的删除	136
第五节	共用体	137
5.1	举例说明结构体和共用体变量的不同	137
5.2	共用体的类型说明	137
5.3	共用体变量的定义、初始化	138
5.4	共用体变量的引用	138
5.5	利用共用体的特点分别取出成员数据	139
第十章 文件		
第一节	文件概述	143
1.1	文件的概念	143
1.2	文件的分类	143
1.3	文件的读写操作	143
第二节	文件指针	144
2.1	文件指针概述	144
2.2	文件指针的定义	144
第三节	打开文件	144
3.1	fopen 函数	144
3.2	文件的打开方式	145
3.3	C 语言中的三个标准文件	145
第四节	判断文件是否结束	145
第五节	关闭文件	146
第六节	文件的读写操作	146
6.1	fscanf 函数和 fprintf 函数	146
6.2	putc(fputc)函数和 getc(fgetc)函数	147
6.3	fputs 函数和 fgets 函数	147
6.4	fwrite 函数和 fread 函数	147
第七节	文件定位函数	148
7.1	fseek 函数	148
7.2	ftell 函数	149
7.3	rewind 函数	149

第十一章 位运算

第一节 位运算符	151
第二节 位运算符的运算功能	152
2.1 位运算符的运算功能	152
2.2 位数不同的运算数之间的运算规则	153

第十二章 对函数的进一步讨论

第一节 传给 main 函数的参数	154
第二节 函数指针	156
第三节 函数的递归调用	158

第十三章 用户标识符的作用域和存储类

第一节 变量的分类及其作用域和生存期	161
1.1 变量的分类	161
1.2 作用域和生存期	162
第二节 局部变量	163
2.1 auto 变量	163
2.2 register 变量	163
2.3 静态类局部变量	163
第三节 全局变量	165
3.1 全局变量的作用域和生存期	165
3.2 扩展全局变量作用域	166
3.3 静态全局变量	167
第四节 函数的存储分类	168
4.1 用 extern 说明函数	168
4.2 用 static 说明函数	168

第十四章 编译预处理

第一节 宏替换	169
1.1 不带参数的宏定义	169
1.2 带参数的宏定义	170
1.3 终止宏定义	171
第二节 文件包含	172
2.1 文件包含的概念	172
2.2 文件包含命令的一般格式	172
2.3 文件包含命令的作用	172
2.4 文件包含说明	172

第一章 C 语言概述

第一节 计算机系统的基本组成

1.1 基本组成概述

当今，计算机已广泛应用于社会生活的各个领域，成为大众化的现代工具。简单地讲，计算机是具有内部存储能力、由程序自动控制的机械电子设备。一个完整的计算机系统应包括硬件系统和软件系统两大部分。

计算机硬件系统是指组成一台计算机的各种物理装置，它们是由各种实在的器件所组成。这些器件看得见、摸得着。如：键盘、鼠标、显示器等。直观地看，计算机硬件是一大堆设备，它是计算机进行工作的物质基础。

计算机软件系统是计算机系统的重要组成部分，是指在硬件设备上运行的各种程序、数据以及有关的资料。

相对于计算机硬件而言，软件是计算机的无形部分，但它的作用是很大的。这好比是人们为了看电视，就必须要有电视机，这是硬件条件；但仅有电视机还看不成电视，还必须要电视节目，这是软件条件。由此可知，如果只有好的硬件，而没有好的软件，计算机是不可能发挥出它的优越性的。

日常生活中，经常听见有人这样说，“这个人是搞编程的”、“那个人是做软件开发的”。那么“软件 = 程序”吗？软件 = 程序 + 文档。所谓程序其实就是能完成一定功能的内部存储的指令的集合。编写一个程序就好比写一篇文章。文章由语句组成，程序也是由语句组成。文章的语句由字符组成，程序中的语句由指令组成。写一篇文章，我们可以用汉语，也可用英语、日语等；同样，编写一个程序就是选择一种高级语言（如 C 语言或 Basic 语言）来编写指令。文档，其实就是与程序相关的各种资料，如各种数据、帮助文件、使用说明书等。

一般微型计算机系统的简单组成如图：

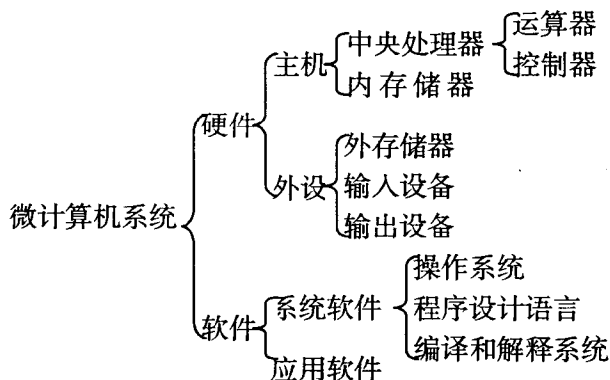


图 1.1

1.2 计算机系统的硬件

一般微型计算机的硬件系统由以下几部分组成：

1.2.1 中央处理器

中央处理器(CPU)在微型计算机中称为微处理器，它是计算机系统的核心，主要包括运算器和控制器两个部件。计算机发生的所有动作都是受微处理器控制的。

运算器主要完成各种算术运算(如加、减、乘、除)和逻辑运算(逻辑加、逻辑乘和逻辑非运算)。

控制器是计算机的指挥控制中心。我们说程序是指令的集合，无数条指令运行时，必然有一个运行先后顺序，这个顺序就是由控制器来控制的。

1.2.2 存储器

存储器好比是一座大型宾馆，由很多房间组成，每个房间，我们称为存储单元。宾馆的房间是用来住人的；存储器的存储单元是用来存放数据和程序的。房间的大小用面积来衡量，有单人间、双人间、四人间等；存储单元的容量用字节数来衡量，最小的存储单元占一个字节。一个字节可以存放 8 位的二进制信息（二进制位：0 和 1）。每个房间都有门牌号，门牌号是该房间的地址；同样，存储单元的每一个字节都有一个编号，这个编号称为地址。微处理器就是按地址来存取存储器中的数据。

1) 存储器分类

计算机的存储器分为内存（储器）和外存（储器）。内存储器实际上就是主机箱内的内存条；常用的外存储器有磁盘（硬盘和软盘）、光盘、磁带。

我们在数学上学过了整数和小数，现在分别看一下两类数据在存储器中的存储形式。

整数（如 4）：用 int 来描述，表明占 2 个字节的空间

500	00000100	低字节
501	00000000	高字节

小数（如 1.25）：用 float 来描述，表明占 4 个字节的空间

1000		低字节
1001		中低字节
1002		中高字节
1003		高字节

2) 内存和外存的关系

打个形象的比喻，内存好比是我们的家，外存好比是客人住宿的宾馆。家的空间很小，一二百平方米足够了，但宾馆的房间面积远远大于我们的家，因此内存的容量小，外存的容量大。换个角度讲，当我们回家时，只需用钥匙打开房门，很简单，但当我们住宾馆时，需拿着身份证去服务台办理入住手续、交押金，然后才能领到钥匙，由服务人员领到房间，走

的时候也很麻烦,因此我们说内存的数据存取速度比外存的要快得多。

总之,内存速度快、容量小,而外存速度慢、容量大。因此,我们总是把大量的数据放在外存。当需要处理数据时,把数据从外存调入内存进行加速,使之接近 CPU 的速度,再让 CPU 进行处理;当 CPU 处理完时,首先把数据送到内存,降低数据的速度,再送到外存进行保存。

无论是内存还是外存都是用来存储数据和程序的。数据在内存中的存储方式为常量、变量、数组,它们分别代表一定的存储单元,我们可以用这些单元的名称或地址来表示这些单元;数据在外存中的存放形式为文件,我们可以通过文件的路径和文件名称来操作文件对应的数据。

1.2.3 输入设备

输入设备是外界向计算机传送信息的装置。在微型计算机系统中,最常用的输入设备有键盘和鼠标器。

1.2.4 输出设备

输出设备的作用是将计算机中的数据信息传送到外部媒介,并转化成某种为人们所需要的表示形式。在微机系统中,最常用的输出设备是显示器和打印机。

磁盘驱动器既可以从磁盘读出数据,也可以往磁盘写数据,因此,它既是输入设备,也是输出设备。

1.3 计算机系统的软件

微型机的软件系统可以分为系统软件和应用软件两大类。

1.3.1 系统软件

系统软件是指管理、监控和维护计算机资源(包括硬件和软件)的软件。常见的系统软件有操作系统、各种语言处理程序以及各种工具软件等。在这里,我们重点介绍一下操作系统和程序设计语言。

1) 操作系统

操作系统是最底层的系统软件,它是对硬件系统功能的首次扩充,也是其他系统软件和应用软件能够在计算机上运行的基础。操作系统主要起服务、支持的作用。

2) 程序设计语言

程序设计语言是软件系统的重要组成部分,它一般分为低级语言和高级语言。

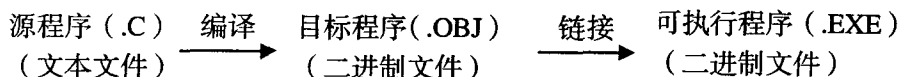
低级语言分为机器语言和汇编语言。自然界中只有两种状态最容易识别,比如说:灯要么开,要么关;门要么开,要么关,如果半开半闭,就不容易识别了。因此为了表示这两种状态,我们用数字 0 和 1 表示。而计算机恰好只能够识别 0 和 1 代码。由 0、1 代码编写的语言,我们称为机器语言(又称二进制语言)。机器语言是针对计算机开发的语言,可以被计算机硬件直接识别。但这种语言,直观性比较差,我们人类不容易识别。为了便于理解与记忆,人们在机器语言的基础上加少量能帮助记忆的符号,这就产生了汇编语言。汇编语言和机器语言一般是一一对应的。由于汇编语言采用了助忆符,因此,它比机器语言直观,容易理解和记忆,用汇编语言编写的程序也比机器语言编写的程序易读、易检查、易修改。但

是,计算机不能直接识别用汇编语言编写的程序,必须由汇编程序将汇编语言源程序翻译成机器语言程序后,计算机才能识别并执行。汇编语言只在机器语言的基础上加少量的助忆符,只有专业人员才可读懂,普通大众仍不容易读懂。因此,在汇编语言的基础上产生了高级语言。

高级语言分为面向过程的语言(C、BASIC、PASCAL)和面向对象的语言(VC、VB、DEPHI)。面向过程的语言和面向对象的语言不存在谁比谁高级之说,它们只是不同时期的产物。面向过程的语言的操作类似于在DOS环境下操作,界面很不友好,输入一条命令,执行一条命令。面向对象的语言的操作类似于在WINDOWS环境下操作,采用图形用户界面,直观,易操作。

通常,我们都是采用高级语言编写程序。用高级语言编写的程序,我们称为源程序。但计算机在执行的过程中并不识别高级语言编写的源程序,只识别二进制语言编写的程序。这就需要进行语言转换。

以C语言为例,语言转换过程:



C语言源程序经过C语言编译程序编译之后生成一个后缀为.OBJ的二进制文件目标程序,然后由链接程序把.OBJ文件与C语言提供的标准函数→库函数连接起来生成后缀为.EXE的可执行文件。.EXE的可执行文件就可以被计算机识别,运行。

1.3.2 应用软件

应用软件是指除了系统软件以外的所有软件,它是为了解决某个应用领域中的具体问题而开发的软件。比如说:Word——文字处理软件,是用来处理文字的;Excel——电子表格,是用来处理数据的;Explore——因特网浏览器,可以上网查资料等。

第二节 C语言程序的构成和格式

本节将通过几个简单的程序例子,介绍C程序的一些基本构成和格式,使读者对C语言程序有一个初步的了解。

2.1 在屏幕上输出“my first program!”

```
main()
{
    printf("my first program!");
}
```

运行结果,在屏幕上输出了一行字符: my first program!

说明:

1) 编写一个程序就好比写一篇文章。写一篇文章需要遵循一定的语法知识。编程的本质很简单,就是要遵循一定的“规则”。一篇稍微长的文章是分很多段落的,每个段落代表一个相对完整的含义。我们前面讲程序是指令的集合,这个集合的排列不是无序的,它被分成很多代表独立功能的模块,每个模块我们称之为“函数”,它类似一篇文章中的段落。在本例中只有一条指令 `printf('my first program!');`,该指令用花括号集合起来放在函数 `main` 中。

2) 上面讲的例子很简单,它只有一个函数组成,函数名称为 `main`,我们把它称为主函数。一个 C 程序必须有一个且只有一个主函数。C 语言规定,无论主函数放在什么位置,程序首先执行主函数,即使在多个函数组成的程序中,主函数是程序执行的开端,并且最终在主函数中结束。

3) 上述 `main` 函数有两部分组成:`main()` 为函数的首部,其余部分为函数体。在函数首部中,`main` 为函数的名称,是 C 语言的关键字(后面再介绍什么是关键字),只能采用小写字母。`main` 虽然没有参数,但后面的括号不能省略掉。

4) 函数体是函数的主要组成部分,其起始标志为“{”,结束标志为“}”。程序的指令基本都存放在函数体中,这些指令体现在多条语句中。一篇文章中的段落也是由语句组成的,如果用中文编写,语句的结束用空心句号“。”表示,如果用英文编写,语句的结束用实心句号“.”表示。用 C 语言编写的文章用分号“;”表示语句的结束。上述程序中只有一条语句。

5) 程序是为人服务的,因此必然要输出结果给人阅读。在 C 语言中,在显示器屏幕上输出信息用 `printf` 函数来实现。就像在数学中学过的正弦函数 $\sin(30^\circ)$,其中 `sin` 为函数名称,小括号内的为参数。同理,`printf` 为函数名称,为小写字母,后面的括号不能漏掉,括号内的文本 `my first program!` 在双引号内,双引号内的内容将在屏幕上原样输出。例如,我们用如下函数 `printf("***);`,则在屏幕上会输出三个“*”号。

2.2 在屏幕上输出三排“*”号

```
main()
{
    printf("***\n");
    printf("***\n");
    printf("***\n");
}
```

运行结果: ***

-

程序的功能:在屏幕上输出三行“*”号,每行有 3 个“*”。

说明:

1) 该程序中只有一个函数 `main()`，而且 `main` 函数由三条语句构成，每条语句都是由一个 `printf` 函数构成。

2) ‘`\n`’ 表示换行符，它的作用是将光标移到下一行的开始。“_” 表示光标。

再如: `main()`

```
{
    printf("****");
    printf("****\n");
    printf("****\n");
}
```

运行结果: *****

同学们在学习的过程中应通过将这两个程序的运行结果相对比来掌握 ‘`\n`’ 的用法。

2.3 在屏幕上输出数据值 100 和 100.1

```
main()
{
    printf("%d  ",100);
    printf("%f  ",100.1);
}
```

运行结果: 100 100.100000

程序功能: 分别将数据 100 和 101.1 以整数和小数的形式输出到屏幕上。

说明: 在数学上有整数和小数之分，在 C 语言中小数被称为实数。由于数值分类型，我们必须规定好各类数据的输出格式。在 C 语言中，整数用 `%d` 表示；实数用 `%f` 表示，默认输出六位小数。

通过上述三个例子，我们能简单了解 `printf` 函数的格式:

1) `printf()` 函数有两个参数，第一个参数是双引号括起来的字符串，(如“`%d` ”)，第二个参数是输出项表 (如 100)，这两个参数之间应用逗号隔开。

2) 第一个参数的内容可由三部分组成:

普通字符: 起解释、说明的作用，应原样输出。常见普通字符: `a=`、`sum=`、逗号、空格等，例 1.1 中的 “my first program!”，例 1.2 中的 “`****`” 都是普通字符，原样输出。

转义字符: 如 ‘`\n`’，字母 `n` 加了反斜线后改变了原字母 `n` 的含义，整体代表“回车换行”。其余的转义字符，我们将在后面讲。

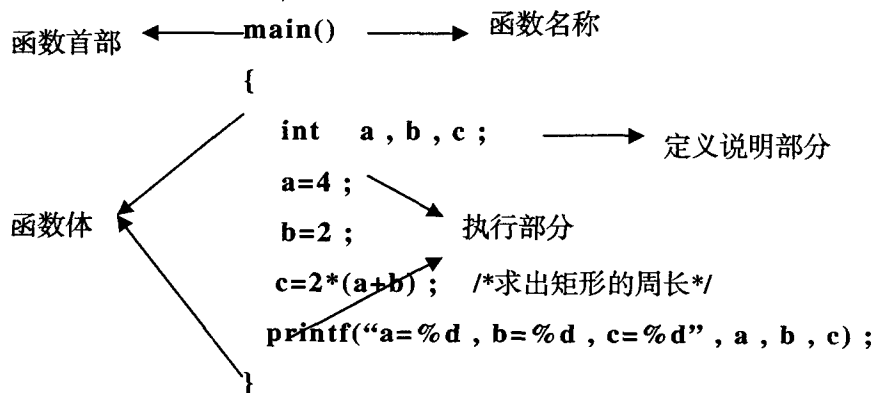
格式说明: 由 “`%`” 和字母构成，规定输出项的输出格式，如 “`%d`” 和 “`%f`” 分别如前所述。

3) 第二个参数输出项表: 该项可以省掉。如果有多个输出项，则由逗号隔开，如

printf("%d,%f", 100, 101.2); 中有两个输出项 100 和 101.2, 它们之间用逗号隔开。该语句的功能: 将 100 和 101.2 分别以整数形式和小数的形式输出到屏幕上。

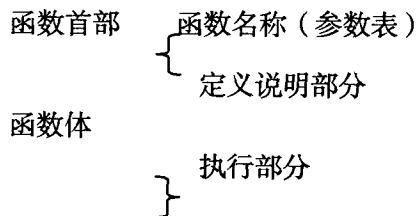
再如: **printf("a=%d\n b=%d",10, 20);** 屏幕上显示: a=10
b=20

2.4 求矩形周长



说明:

1) 通过该例, 掌握一般函数的构成格式



2) 函数首部由函数名称构成, 函数名称可以无参数, 但小括号一定不能省略。函数首部的最后没有分号, 因为函数首部不是语句。函数体由定义说明部分和执行部分构成, 定义说明部分一定要放在执行部分前面。定义说明部分和执行部分放在一对花括号内。

3) 程序中的 a、b、c 分别表示矩形的长、宽和周长。矩形的长、宽和周长的值是可以发生改变的, 因此 a、b、c 是变量。变量是指在程序执行过程中, 其值可以改变的量。变量应先定义后使用, 即定义说明部分应放在执行部分前面。**int a, b, c;** 为定义说明部分, 该语句的功能是定义变量 a、b、c 为整数类型, 此处的 int 翻译成中文为“基本整型”, 是整数的一种。小数在 C 语言中用 float 来表示。定义变量是 C 语言为变量向内存申请存储单元, 类似我们打电话预定房间, 但在赋值前, 单元是空的, 相当于客人还未入住。**a=4; b=2; c=2*(a+b);** 就是把所赋的数据放到变量名称所代表的存储单元中 (相当于预定房间后, 实际入住)。

4) 在汉语中, 一句话未说完, 用逗号分隔。同样, C 语言中的一条语句如果未结束用逗