



高职高专计算机项目 / 任务驱动模式教材

C语言程序设计 项目化教程

周雅静 钱冬云 邢小英 徐济惠 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高职高专计算机项目/任务驱动模式教材

C语言程序设计 项目化教程

周雅静 钱冬云 邢小英 徐济惠 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本教材采用工学结合、任务驱动的模式编写。编写过程中,以完成“学生成绩管理系统”为主线,设立循序渐进的8个项目。通过项目分析,又将其分成若干个具体的任务,每个任务包含着C语言的若干个知识点和技能点。本书旨在提高学生对知识灵活应用的能力、实用操作能力及解决问题、处理问题的实践能力。

本教材还着重强调“任务”的目标性和教学情境的创建,使学生带着真实的任务在探索中学习。本教材以注重培养学生的实践能力为前提,理论知识传授遵循“实用为主、必需和够用为度”的准则,基本知识广而不深、点到为止,基本技能贯穿教学的始终,具体采用“技能需求、问题引导、任务驱动”的方式。

本教材既可以作为高职学生的教学用书,还可作为计算机爱好者的自学参考书和计算机培训班的教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

C语言程序设计项目化教程/周雅静等编著. —北京:电子工业出版社,2014.8

高职高专计算机项目/任务驱动模式教材

ISBN 978-7-121-23488-0

I. ①C… II. ①周… III. ①C语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第124467号

策划编辑:束传政

责任编辑:束传政

特约编辑:彭 瑛 赵海军

印 刷:北京季蜂印刷有限公司

装 订:北京季蜂印刷有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×1092 1/16 印张:18.5 字数:473.6千字

版 次:2014年8月第1版

印 次:2014年8月第1次印刷

印 数:2500册 定价:39.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至zlts@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线:(010)88258888。

前言

Preface

本教材采用工学结合、任务驱动的模式编写。编写过程中，以完成“学生成绩管理系统”为主线（选择这样一个系统的原因是学生比较熟悉有关学习成绩的业务和功能），设立循序渐进的8个项目，项目的设置做到先易后难，先单一后全面的原则。

我们在编写教材时，根据C语言的特点，将其分成两个部分：基础篇和提高篇。基础篇中有5个项目：项目1——学生成绩的总分与平均分的计算、项目2——输入学生成绩转化为等级、项目3——学生成绩的分组汇总、项目4——学生成绩的排序、项目5——根据条件进行学生成绩汇总，本篇涉及C语言知识点有数据类型、输入/输出函数、顺序结构、选择语句、循环语句、数组、函数等。提高篇中有3个项目：项目6——用指针优化学生成绩排名、项目7——学生成绩单制作、项目8——学生成绩文件管理，本篇所包含C语言的知识点有指针、结构体、文件等。

根据项目的要求，将项目分解成若干具体任务，每个任务有C语言的若干知识点，在每个任务的相关知识学习完毕后，都有实例，使学生进一步加深对知识的理解，最后还配有实践训练（基础篇中的实践训练分为初级训练和深入训练两个部分），力求做到“工作过程”和“教学过程”有机结合，切实提高学生编程能力。

本教材注重培养学生的实践能力，理论知识传授遵循“实用为主、必需和够用为度”的准则，同时以工作任务为中心组织课程内容，让学生在完成具体项目的过程中具备完成相应工作任务的能力，并构建相关理论知识，突出对学生职业能力的训练。本书所有代码可以登录华信教育资源网(www.hxedu.com.cn)免费下载。

本书由宁波城市职业技术学院的周雅静、邢小英、徐济惠及浙江工贸职业技术学院的钱冬云老师等组织编写，由周雅静负责全书的统稿。在本书的编写过程中，编者参考了大量有关C语言的书籍和资料，在此对这些参考文献的作者表示感谢。

由于作者水平有限，书中难免有疏漏和不足之处，恳请广大读者批评指正，并提出宝贵意见。

编者

2014年5月

目录

Contents

项目1 学生成绩的总分与平均分的计算

任务1-1 学生成绩的输入/输出	2
任务提出及实现	2
1. 任务提出	2
2. 具体实现	2
相关知识	2
1. C语言的结构	2
2. C语言运行环境及运行方法	3
3. 数据类型	4
4. 格式输出函数——printf()函数	5
5. 格式输入函数——scanf()函数	7
知识扩展	9
1. 转义字符	9
2. 格式化输出	9
3. 单个字符的输入/输出	11
4. 字符串常量	12
举一反三	13
实践训练	15
☆ 初级训练	15
☆ 深入训练	16
任务1-2 总分及平均分的计算	18
任务提出及实现	18
1. 任务提出	18
2. 具体实现	18
相关知识	19
1. 算术运算符和算术表达式	19
2. 赋值运算和赋值表达式	20
3. 自增、自减及逗号运算符	21
举一反三	23
实践训练	27
☆ 初级训练	27
☆ 深入训练	28

综合训练一	30
-------	----

项目2 输入学生成绩转化为等级

任务2-1 输入学生成绩, 判断其合法性	33
任务要求与实现	33
1. 任务要求	33
2. 具体实现	33
相关知识	34
1. 关系运算符与关系表达式	34
2. 逻辑运算符与逻辑表达式	35
3. 条件语句	36
4. 条件运算符与条件表达式	39
实践训练	40
☆ 初级训练	40
☆ 深入训练	42
任务2-2 输入学生成绩转化为等级	43
任务提出及实现	43
1. 任务提出	43
2. 具体实现	43
相关知识	45
1. if语句的嵌套	45
2. switch语句	47
举一反三	50
实践训练	55
☆ 初级训练	55
☆ 深入训练	58
综合练习二	60

项目3 学生成绩的分组汇总

任务3-1 求一个小组学生成绩的总分及平均分	64
任务提出与实现	64
1. 任务提出	64
2. 具体实现	64



相关知识	65	任务4-2 多个学生一门课成绩的排序	109
1. while循环	65	任务提出及实现	109
2. do...while语句	69	1. 任务提出	109
3. for语句	71	2. 具体实现(为了程序运行方便,假设只有10位学生)	109
知识扩展	74	相关知识	110
1. for语句的多种表示方法	74	1. 求最高分	110
2. break语句	75	2. 排序	111
举一反三	76	举一反三	113
实践训练	81	实践训练	116
☆ 初步训练	82	☆ 初级训练	117
☆ 深入训练	84	☆ 深入训练	120
任务3-2 求每个小组学生成绩的总分及平均分	86	任务4-3 学生姓名的输入/输出	122
任务提出与实现	86	任务提出与实现	122
1. 任务提出	86	1. 任务提出	122
2. 具体实现	86	2. 具体实现(为了程序运行简单,假设只有5位同学)	122
相关知识	87	相关知识	123
1. 常见的循环嵌套	87	1. 一维字符数组	123
2. 循环嵌套的运用	88	2. 二维字符数组	126
3. 循环嵌套的实例	90	知识扩展	127
举一反三	91	1. 字符串比较函数strcmp()	127
实践训练	93	2. 复制字符串函数strcpy()	127
☆ 初级训练	93	举一反三	128
☆ 深入训练	95	实践训练	131
综合练习三	96	☆ 初级训练	131
项目4 学生成绩的排序	99	☆ 深入训练	132
任务4-1 多位学生一门课成绩的输入/输出	100	任务4-4 多个学生多门课成绩的排序	134
任务提出及实现	100	任务提出及实现	134
1. 任务提出	100	1. 任务提出	134
2. 具体实现	100	2. 具体实现(为了在程序运行时方便,所以假设只有5位学生)	134
相关知识	100	相关知识	136
1. 一维数组的定义	100	1. 二维数组的定义	136
2. 一维数组的引用	101	2. 二维数组的引用	136
3. 一维数组的初始化	103	3. 二维数组的初始化	136
举一反三	103	4. 二维数组的应用	137
实践训练	106	举一反三	140
☆ 初步训练	106		
☆ 深入训练	108		



实践训练	143	1. 数组名作为函数参数	181
☆ 初步训练	143	2. 数组名作为函数参数的应用	183
☆ 深入训练	144	举一反三	185
综合练习四	145	实践训练	187
项目5 根据条件进行学生成绩		☆ 初级训练	187
汇总	147	☆ 深入训练	189
任务5-1 统计小组一门课程的总分及平均分	149	综合训练五	190
任务提出及实现	149	项目6 用指针优化学生成绩排名	194
1. 任务提出	149	任务6-1 了解指针	194
2. 具体实现	149	任务提出与实现	194
相关知识	150	1. 任务提出	194
1. 无参函数	150	2. 具体实现	195
2. 空函数	154	相关知识	195
3. 有参函数	154	1. 地址和指针的概念	195
举一反三	157	2. 指向变量的指针变量	196
实践训练	159	3. 指针变量作为函数参数	199
☆ 初级训练	159	举一反三	200
☆ 深入训练	164	实践训练	202
任务5-2 统计小组若干门课程的总分		任务6-2 用指针优化全班同学一门课成绩	
及平均分	166	的输入/输出	206
任务提出与实现	166	任务提出及实现	206
1. 任务提出	166	1. 任务提出	206
2. 具体实现	166	2. 具体实现 (以10位学	
相关知识	167	生为例)	206
1. 嵌套函数	167	相关知识	207
2. 嵌套函数应用	168	1. 指向数组元素的指针	207
知识扩展	169	2. 一维数组元素的指针访	
1. 递归函数	169	问方式	208
2. 递归函数的应用	169	举一反三	211
举一反三	171	实践训练	212
实践训练	173	任务6-3 用指针优化某班同学三门课成绩	
☆ 初级训练	173	的输入/输出	213
☆ 深入训练	177	任务提出及实现	213
任务5-3 输出排序后小组三门课成绩单	178	1. 任务提出	213
任务提出与实现	178	2. 具体实现	213
1. 任务提出	178	相关知识	214
2. 具体实现 (假设本小组只有		1. 二维数组的指针访问方式	214
5位同学)	178	2. 二维数组的指针访问应用	215
相关知识	181	举一反三	216



实践训练	217	项目8 学生成绩文件管理	261
任务6-4 用指针实现输出最高分的记录...	219	任务8-1 将学生成绩顺序读写到文件中...	261
任务提出及实现	219	任务提出及实现	261
1. 任务提出	219	1. 任务提出	261
2. 具体实现	219	2. 具体实现 (为了程序运行方便, 假设只有10位同学)	262
相关知识	221	相关知识	263
知识扩展	224	1. 文件的打开/关闭	263
1. 字符串的表示形式	225	2. 文件的读写	265
2. 用指针变量来实现对字符串的访问	226	举一反三	274
举一反三	227	实践训练	275
实践训练	229	任务8-2 将学生成绩随机读写到文件中...	276
综合训练六	232	任务提出与实现	276
项目7 学生成绩单制作	234	1. 任务提出	276
任务7-1 用结构体数组进行学生信息的输入/输出	235	2. 具体实现 (为了程序运行方便, 假设只有10位学生)	276
任务提出及实现	235	相关知识	277
1. 任务提出	235	1. 指针指向文件开头的函数 (rewind()函数)	278
2. 具体实现 (为了程序运行方便, 假设有5位同学)	235	2. fseek()函数	278
相关知识	236	举一反三	280
1. 结构体类型	236	实践训练	281
2. 结构体变量的引用	237	综合训练八	281
3. 结构体变量的初始化	238	附录A 运算符表	283
4. 结构体数组	239	附录B 常用标准库函数	284
5. 指向结构体类型数据的指针	241	1. 数学标准库函数 (函数原型: math.h)	284
举一反三	243	2. 输入/输出库函数 (函数原型: stdio.h)	284
实践训练	246	3. 字符函数与字符串函数 (函数原型: string.h)	285
任务7-2 排序后的学生成绩单	249	4. 其他函数 (函数原型: stdlib.h)	286
任务提出及实现	249	附录C ASCII字符编码表	287
1. 任务提出	249	参考文献	288
2. 具体实现 (为了程序运行方便, 假设只有5位同学)	250		
相关知识	251		
举一反三	253		
实践训练	256		
综合训练七	259		

学生成绩的总分与平均分的计算



技能目标

- 学会对数据进行正确的输入/输出并进行简单的汇总。



知识目标

- 理解C语言的数据类型。
- 掌握整型常量、整型变量、实型常量、实型变量、字符常量和字符变量等的概念。
- 掌握输入/输出语句。
- 了解数据运算规则。

项目要求

输入三个学生的成绩，求他们的总分及平均分。

掌握程序的运行要求

输入三个学生的成绩：79 89 90。

三个学生的总分为：258.0。

三个学生的平均分为：86.0。

说 明：分数可以任意输入。

项目分析

要完成学生成绩的平均分与总分的计算，首先，必须要学会输入成绩及输出结果；第二，必须学会对输入的成绩进行总分及平均分的计算。所以，该项目分解成两个任务：任务 1-1 是学生成绩的输入/输出；任务 1-2 是总分及平均分的计算。



任务1-1 学生成绩的输入/输出



任务提出及实现

1. 任务提出

某班级进行了一次考试，现需将该班几个学生成绩输入电脑，并按要求输出。

2. 具体实现

【例 1-1】(假设只输入三个学生的成绩)

```

#include "stdio.h" // 文件预处理
main() // 函数名
{ // 函数体开始
    int x,y,z; // 定义三个变量 x,y,z
    printf("请输入三个学生的成绩");
    scanf("%d%d%d",&x,&y,&z); // 输入三个学生的成绩
    printf("输出三个学生的成绩");
    printf("x=%d,y=%d,z=%d\n",x,y,z); // 输出三个变量 x,y,z 的值
} // 函数体结束

```

例 1-1 程序运行结果如图 1-1 所示。



图1-1 例1-1程序运行结果

从上面这段程序可分析出：

- ① 要了解 C 语言的结构。
- ② 要了解 C 语言的运行环境。
- ③ 要懂得如何定义变量及输入/输出语句的表示方法。



相关知识

1. C语言的结构

C 语言由一个或多个文件组成，而一个文件可由一个或多个函数组成。C 语言必须有一个函数名为 main 的函数，且只能有一个 main 函数。程序运行时从 main 函数开始，最后回到 main 函数。

从例 1-1 中，可以看出：C 语言函数由语句构成，语句结束符用“;”表示，但 main()、#include 不是语句，后面不需加“;”。语句由关键字、标识符、运算符和表达式构成。其中“{”



和“}”分别表示函数执行的起点与终点或程序块的起点与终点。

“//”后面的语句为注释语句，也可以写在“/*”和“*/”内。

C语言中书写格式自由，一行内可以写几个语句；但区分大小写字母（这个很重要）。用C语言写成的主函数结构如图1-2所示。

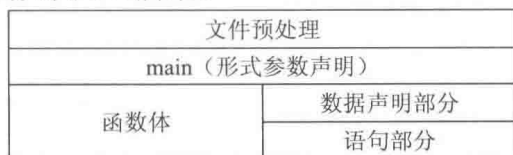


图1-2 C语言的主函数结构图

2. C语言运行环境及运行方法

C语言必须经过编辑、编译、连接的过程才能生成一个可执行程序。运行C语言的环境很多，这里主要介绍创天中文VC++环境。

打开VC++，界面如图1-3所示。

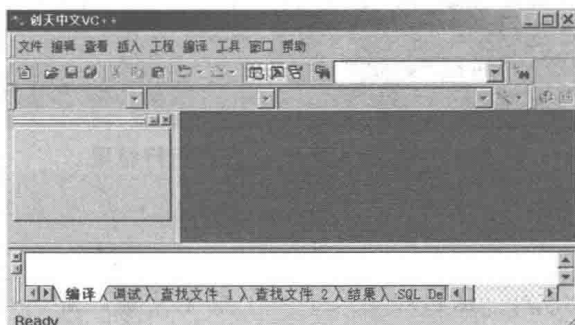




图1-3 VC++界面

单击【文件】→【新建】命令，打开对话框如图1-4所示。选择【文件】选项卡中的C++Source File选项，然后更改文件所保存的目录（若需要更改），并输入文件名，然后单击【确定】。返回VC++界面，如图1-5所示。现在就可以输入程序。当程序输入完毕后，单击按钮，或按F7键进行编译和连接。如果没有错误，则单击按钮，或按快捷键Ctrl+F5，进行运行，即可得出程序运行的结果，如图1-6所示。

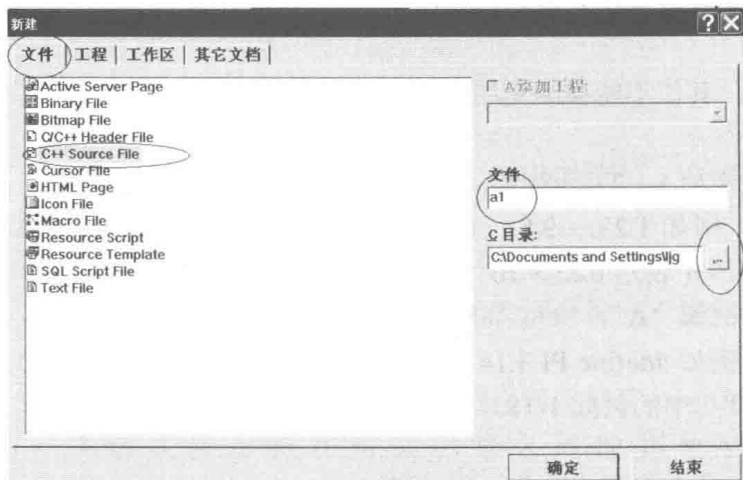


图1-4 【新建】对话框

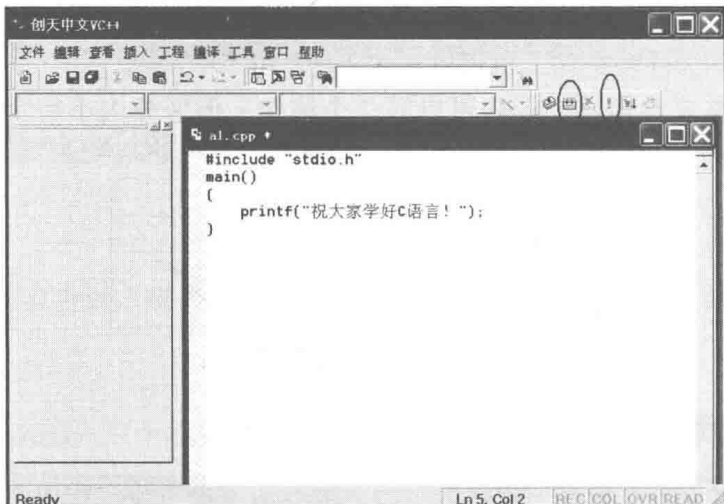


图1-5 创天中文VC++界面

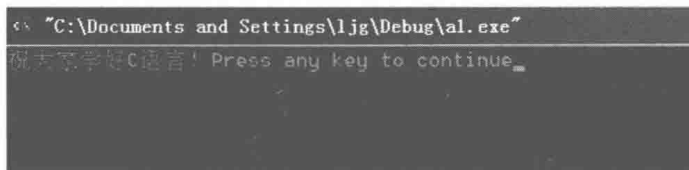


图1-6 创天中文VC++程序运行结果

3. 数据类型

由例 1-1 可知，“int x,y,z;”意思为定义了三个整型变量。那么在 C 语言中还有哪些常用的数据类型？

C 语言的数据类型可分为：

- ① 基本类型，又分为整型、实型、字符型和枚举型四种。其中枚举型暂不介绍。
- ② 构造类型，又分为数组类型、结构类型和共用类型三种。
- ③ 指针类型。
- ④ 空类型。

C 语言的数据，有常量和变量之分。

(1) 常量

在程序运行中，其值不能被改变的量称为常量。

常量可分为：

- ① 整型常量，例如 1，23，-8。
- ② 实型常量，例如 1.23，-9.8，.123，23.，0.0，2.23e-4（表示 2.23×10^{-4} ），0.23e3（表示 0.23×10^3 ），0.23E-3（表示 0.23×10^{-3} ），1.23E3（表示 1.23×10^3 ）。
- ③ 字符常量，例如 ‘A’，‘+’，‘8’。
- ④ 符号常量，例如 #define PI 3.14，则 PI 是一个符号常量，其值为 3.14，不能在程序中被改变（例如任务 1-2 中的例如 1-18）。

(2) 变量

在程序运行过程中，其值可以被改变的量称为变量。一个变量由两个要素组成，即变量



名和变量值。

每一个变量都必须有一个名字，即变量名。变量名的命名规则为：由字母或下画线开头，后面跟字母、数字和下画线。其有效长度，随系统而异，但至少前8个字符有效。如果超长，则超长部分被舍弃。

注意

C语言的关键字不能用作变量名。

说明：

(1) C语言的变量名区分大小写。即同一字母的大小写，被认为是两个不同的变量。例 Total、total、toTal 是不同的变量名。

(2) 给变量名命名时，最好遵循“见名知意”这一基本原则。例如 name/xm(姓名)、sex/xb(性别)、age/nl(年龄)、salary/gz(工资)等。

(3) 变量定义与初始化

在C语言中要求对所有用到的变量，必须先定义后使用。

变量定义的一般格式为：

数据类型 变量名 1[, 变量名 2……];

例如：

```
int x,y,z;           // 定义了整型变量 x,y,z
float a,b,c;        // 定义了实型变量 a,b,c
char c1,c2;         // 定义了字符型变量 ch1,ch2
```

变量初始化的一般格式为：

数据类型 变量名 [= 初值, 变量名 2= 初值 2……];

例如：

```
int x=1,y=2,z=3;
float a=1.1,b=1.2,c=-0.1;
char ch1='A',ch2='*';
```

4. 格式输出函数——printf()函数

printf() 函数的作用是向显示器输出若干个任意类型的数据。

printf() 函数的一般格式为：

printf("格式字符串"[, 输出项表]);

(1) 常用的格式字符串

常用的格式字符串有以下几种格式：

① 格式指示符。%d 表示带符号十进制整数，%f 表示带符号十进制小数（默认6位小数），%c 表示一个字符。

② 转义字符。例 1-1 中的 printf() 函数中的“\n”就是转义字符，输出时产生一个“换行”。

③ 普通字符。除格式指示符和转义字符之外的其他字符。例 1-1 中的 printf("x=%d,y=%d,z=%d\n",x,y,z); 的“x=”“y=”“z=”，格式字符串中的普通字符原样输出。



在汉字系统环境下，允许使用汉字。

(2) 输出项表

输出项表是可选的。在输出项中，列出要输出的任意类型的数据。如果要输出的数据不止一个，则相邻项用逗号隔开。例如：

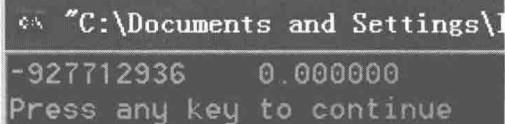
```
printf("我是一个学生\n");           // 没有输出项
printf("%d",1+2);                   // 输出 1+2 的值
printf("a=%d b=%d\n",1,1+3);       // 输出 1 的值和 1+3 的值
```

必须强调，格式指示符一定要同输出项的数据类型一致，否则会出现错误。例如，`printf("%d %f\n",1.212,5);`是错误的。因为"`%d`"是整型数格式，但1.212却是实数，同理"`%f`"是实数格式，但5却是整型数。下面用一个例子来说明。

【例 1-2】格式指示符与输出项数据类型不一致引起的错误。

```
#include "stdio.h"                 // 文件预处理，因为要用到 printf() 函数
main()                             // 主函数
{                                   // 函数体开始
    printf("%d%f\n",1.234,6);
}
```

例 1-2 程序结果如图 1-7 所示。



```
C:\Documents and Settings\I
-927712936 0.000000
Press any key to continue
```

图 1-7 例1-2程序运行结果

【例 1-3】格式化输出。

```
#include "stdio.h"
main()
{
    int x=1,y=2,z=3;           /* 定义 x,y,z 三个整型变量，并将它们的初值赋为 1,2,3*/
    float a=1.1,b=2.3;
    char c1='A',c2='B';       /* 定义 c1,c2 二个字符型变量，并将它们的初值赋为 'A' 和 'B'*/
    printf("输出 x,y,z 的值\n");           // 原样“输出 x,y,z 的值”后换行
    printf("x=%d,y=%d,z=%d\n",x,y,z);     // 输出 "x=1,y=2,z=3" 后换行
    printf("输出 a,b 的值\n");
    printf("a=%f,b=%f\n",a,b);
    printf("输出 c1,c2 的值\n");
    printf("c1=%c,c2=%c\n",c1,c2);       // 输出 c1='A', c2='B' 后换行
}
```

例 1-3 程序运行结果如图 1-8 所示。



```

c:\ "C:\Documents and ... - □ ×
输出x,y,z的值
x=1,y=2,z=3
输出a,b的值
a=1.100000,b=2.300000
输出c1,c2的值
c1=A,c2=B

```

图1-8 例1-3程序运行结果

5. 格式输入函数——scanf()函数

(1) scanf() 函数的功能

从键盘上输入若干个任意类型的数据。

(2) scanf() 函数的一般格式

`scanf("格式字符串", 输入项首地址表)`

①格式字符串。格式字符串包含3种类型:格式字符、空白字符(空格、Tab键和回车键)和非空白字符(又称普通字符)。

格式指示符与printf()函数的格式指示符相似:%d表示十进制带符号的整型数,%f表示十进制带符号的实型数,%c表示一个字符。

空白字符作为相邻2个输入数据的默认分隔符。

非空白字符在输入数据时,必须原样输入。

②输入项首地址表。由若干个输入项首地址组成,相邻输入项地址用逗号隔开。变量首地址的表示方法为:

& 变量名

其中“&”是地址运算符。例如,在例1-1中scanf("%d%d%d",&x,&y,&z)中的“&x”是指变量x在内存中的首地址。其功能是从键盘上输入3个整型数,分别存入&x,&y,&z起始的存储单元中,即输入三个整数分别赋给x,y,z。

在介绍了数据类型和输入/输出函数后,现在是否理解了例1-1?

“scanf("%d,%d",&a,&b);”意思是在键盘上输入两个数a,b,若a=1,b=2,则程序运行时在键盘中输入“1,2”。

注意

1与2之间用逗号隔开,因为scanf()函数中的两个%d之间是用逗号隔开的,如图1-9所示。

“scanf("%d%d",&a,&b);”意思是在键盘上输入两个数a,b,若a=1,b=2,则程序运行时在键盘中输入“1—2”,如图1-10所示;或者“1↔2”如图1-11所示。

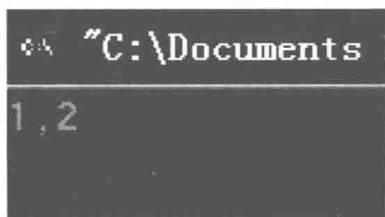


图1-9 用逗号分隔

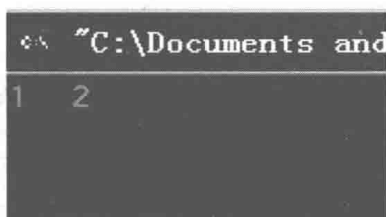


图1-10 用空格分隔

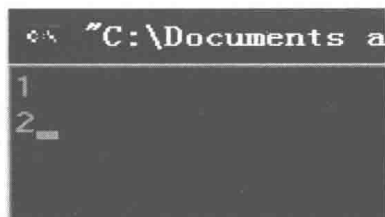


图1-11 用回车分隔

注意

1与2之间可用空格分开，也可用回车键分开。

【例 1-4】有两个人 A 和 B，数学考试成绩分别为 87 和 76，请输入 A 和 B 两人的代号及成绩，输出成绩。

分析：本题考查输入/输出语句，注意整数输入/输出的格式用 %d，字符变量的输入/输出格式用 %c，同时注意输入语句中的两个数据之间的隔开符号，在程序运行输入数据时也要用同样的隔开符号隔开。

```
#include "stdio.h"
main()
{
    char c1,c2;
    int x,y;
    printf("请输入A的成绩及代号:");
    scanf("%d:%c",&x,&c1);
    printf("请输入B的成绩及代号:");
    scanf("%d:%c",&y,&c2);
    printf("输出A的代号及成绩:");
    printf("%c:%d\n",c1,x);
    printf("输出B的代号及成绩:");
    printf("%c:%d\n",c2,y);
}
```

例 1-4 程序运行结果如图 1-12 所示。

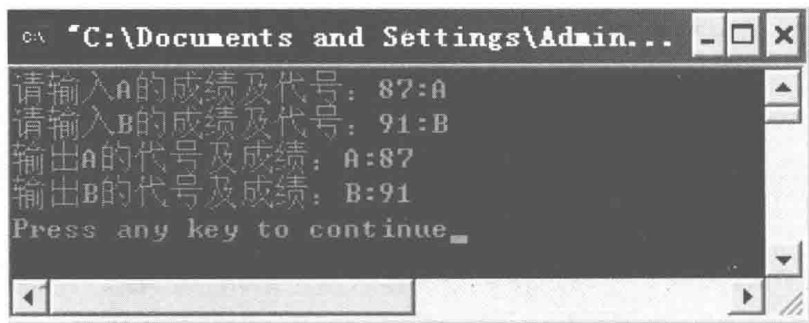


图1-12 例如1-4程序运行结果



知识扩展

1. 转义字符

前面，已经提到“\n”的转义字符，即是换行。常用的转义字符还有“\t”，即横向跳出下一个输出区。

【例 1-5】用转义字符控制输出效果。

```
#include "stdio.h"
main()
{
    printf("%d\t%d\t%d\n", 1, 2, 3);
}
```

例 1-5 程序运行结果如图 1-13 所示。

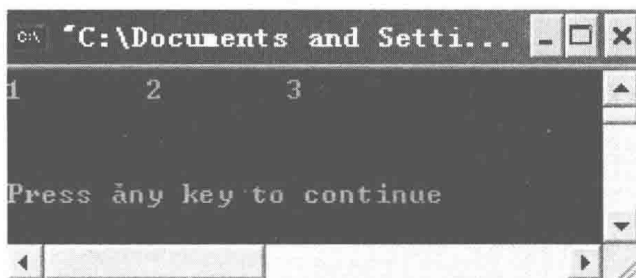


图1-13 例1-5程序运行结果

2. 格式化输出

前面，已经学过输出语句的一般格式为

```
printf(" 格式字符串 "[, 输出项表]);
```

格式字符串的内涵非常丰富，除了前面提到的转义字符和普通字符外，还有格式指示符。常用的格式字符串表示法如下：

“%5d”表示输出十进制带符号的整数，其输出域的宽度为 5 位，若数超长，则按实际宽度输出；若不足，则输出数的左端补空格。“%ld”只是表示输出的是长整型。

“%-5d”表示输出十进制带符号的整数，其输出域的宽度为 5 位，若数超长，则按实际宽度输出；若不足，则输出数的右端补空格。

“%.1f”表示输出十进制带符号的单、双精度实数，其输出域的宽度为 6 位，其中小数一位，若数超长，则按实际宽度输出；若不足，则输出数的左端补空格。

“%-7.2f”表示输出十进制带符号的单、双精度实数，其输出域的宽度为 7 位，其中小数 2 位，若数超长，则按实际宽度输出；若不足，则输出数的右端补空格。

【例 1-6】类型转换字符 d 的使用。

```
#include <stdio.h>
main()
{
```