



高职高专电子信息类“十二五”规划教材

项目式C语言教程

主 编 陈和洲

副主编 王小平 陈晶瑾 张冬梅



西安电子科技大学出版社
<http://www.xduph.com>

高职高专电子信息类“十二五”规划教材

内容简介

项目式 C 语言教程

主 编 陈和洲

副主编 王小平 陈晶瑾 张冬梅

参 编 舒廷杰 刘海舒

(1) 在每个项目开始的时候引入新的知识点，当解决完所有问题后，再对整个项目进行代码。

(2) 项目问题提出后，围绕该问题，从解决该问题的角度出发，进行项目设计。

(3) 项目选择具有代表性，项目 1 为显示广告语，学生可以任意输入广告语，并显示在屏幕上。

出各种图案，项目 2 为显示广告语，学生可以任意输入广告语，并显示在屏幕上。

(4) 项目内容的深入，各种知识的学习不再枯燥。

(5) 项目之间的衔接，二维数组。

(6) 注重项目

本书具体包括项目 1 通过字的方法，何为字用法——显示字优秀的开源 C/C++ 编译程序，最后介绍

项目 2 利用求值运算符，并讲解了如何从大量数据求和问题，此外还详细讲解了格式化输出 printf 函数的使用。

西安电子科技大学出版社

林蝶收联 “五十二” 类息自毛申空高朋高

内 容 简 介

本书采用项目方式组织内容,从应用出发,利用实际问题引出语法,从解决问题的角度出发来讲解知识点。本书共 10 个项目,内容包括显示广告语、完成数据计算、菜单设计、大量数据求和、成绩的计算、多门功课成绩的计算、用结构体处理学生成绩、编写一个日历程序、为函数设置多个返回值、大数求平均值问题。各个项目都配有习题,便于读者练习 C 语言编程方法。

本书可以作为高职院校相关专业的教材,也可以作为自学 C 语言的参考用书。

本书配有电子教案,需要者可在出版社网站下载。

陈和洲 编 主

陈冬羽 黎晶莉 王小平 编 主 编

陈和洲 黎晶莉 王小平 编 参 考

图书在版编目(CIP)数据

项目式 C 语言教程/陈和洲主编. —西安:西安电子科技大学出版社,2014.8

高职高专电子信息类“十二五”规划教材

ISBN 978-7-5606-3430-2

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

中国版本图书馆 CIP 数据核字(2014)第 162935 号

策 划 王 飞 刘玉芳

责任编辑 张 玮

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2014 年 8 月第 1 版 2014 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 15

字 数 353 千字

印 数 1~3000 册

定 价 24.00 元

ISBN 978 - 7 - 5606 - 3430 - 2/TP

XDUP 3722001-1

西安电子科技大学出版社

如有印装问题可调换

前 言

C 语言是一种计算机程序设计语言。它既有高级语言的特点，又有汇编语言的特点；它既可以作为操作系统设计语言，编写系统程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。因此，它的应用范围广泛。本书采用项目方式，从应用出发介绍 C 语言，通过解决实际问题引出语法。

与其他的项目式教材相比，本书的特点如下：

(1) 在每个项目开始的时候引入问题，然后分析问题并进行分解，在逐步解决问题的同时引入新的知识点，当解决完所有问题后，在项目的最后会给出所讨论问题的完整程序代码。

(2) 项目问题提出后，围绕该问题，从解决问题的角度出发来讲解知识点。

(3) 项目选择具有代表性。项目 1 为显示广告语，学生可以发挥想象力在屏幕上打出各种图案；项目 5~7 选择期末成绩处理这个跟学生密切相关的问题，让学生理解起来更加容易。

(4) 项目内部结构紧凑，以实际问题的解决方法为引线，逐步推进，随着问题探讨的深入，各种知识点和技巧性的东西随之出现，让学生在不知不觉中学会了语法，语法学习不再枯燥。

(5) 项目之间连贯性强。项目 5、6、7 围绕同一个问题进行，逐步推进，贯穿了数组、二维数组、结构体、字符串等相关内容。

(6) 注重项目间知识点的连续性。

本书具体包括以下内容：

项目 1 通过书写广告语这样一个简单的工作，首先讲解了在屏幕上显示字符和汉字的方法、何为计算机编程、C 语言程序的结构、main 函数的概念、printf 函数的基本用法——显示字符等，由此扩展，实现了在屏幕上显示复杂的图案。其次介绍了一款优秀的开源 C/C++ 集成开发环境 (IDE) Code::Blocks，详细说明了如何使用 Code::Blocks 编译程序。最后介绍了 C 语言的发展及其优点。

项目 2 利用求两个数平均值的问题引入了 C 语言基本数据类型、算术运算符和赋值运算符，并讲解了如何从键盘上输入一个数字，如何利用复合的赋值运算符求解大量数据求和问题，此外还详细介绍了格式化输出 printf 函数的使用。

项目 3 是设计一个菜单，通过在键盘上输入不同的命令而使程序执行不同的任务。该项目是在项目 2 的基础上进行设计的，然后引入分支结构、if 语句，讲解了关系运算符、逻辑运算符、运算符的优先级和结合性，并且利用 switch 语句实现了菜单的设计。

项目 4 对项目 2 进行了扩展，通过求 10 个数的平均值，引入了大量数据求和的方法：for 语句、while 语句、do-while 语句，综合讲解了结构化程序设计的相关概念，总结了结构化程序设计的三种基本结构：顺序、分支、循环。

项目 5 以计算一门功课的成绩为例，引入了数组概念，讲解了如何利用 for 循环结构处理数组相关问题，介绍了数组元素在内存中的存储方式，以及排序算法中的冒泡算法和选择算法。

项目 6 通过处理多门功课的成绩引入了二维数组，并对双重循环进行了介绍。

项目 7 在成绩中增加了姓名一栏，引入了字符串和字符数组，并且因为同一个人的数据不再同质而引入了结构体数据类型，并对数据结构做了概念上的介绍，讲解了 typedef 的使用方法。

项目 8 利用一个日历程序的编写讲解了函数的概念、库函数的使用、函数的编写、函数之间的参数传递，并说明了“自顶向下、逐步求精”的设计思想如何在程序实践中得到应用。

项目 9 通过一个例子提出函数返回多个数值的问题，讲解了指针的概念、指针的使用、指针与数组的联系和区别等。

项目 10 介绍了大数据求和问题，引入了位运算，介绍了基本运算规则与、或、非和异或，同时介绍了这些规则经常使用的方法和场合。

本书由重庆航天职业技术学院陈和洲担任主编，重庆城市管理职业学院王小平及重庆航天职业技术学院陈晶瑾、张冬梅担任副主编，四川外国语大学教育技术中心舒廷杰及重庆工业职业技术学院刘海舒也参与了本教程的编写工作。其中，陈和洲对全书的编写思想和大纲进行了总体策划，负责全书的统稿，并编写了项目 1、项目 8、项目 10；陈晶瑾编写了项目 2、项目 3；王小平编写了项目 4、项目 9；张冬梅编写了项目 5；刘海舒编写了项目 6；舒廷杰编写了项目 7。

由于编者水平有限，书中难免存在不足之处，恳请广大读者批评指正。

编者

2014 年 1 月

目 录

项目1 显示广告语	1	项目3 菜单设计	43
1.1 项目要求	1	3.1 项目要求	43
1.2 项目描述	1	3.2 项目描述	43
1.3 Code::Blocks 的安装和使用	6	3.2.1 数据输入部分	45
1.3.1 Code::Blocks 的安装	7	3.2.2 数据处理部分	46
1.3.2 Code::Blocks 的汉化	7	3.3 if 语句	48
1.3.3 配置 g++ 编译器及调试器	8	3.3.1 关系运算符	50
1.3.4 编写程序	9	3.3.2 布尔型数据	51
1.4 C 语言的发展史及其特点	14	3.3.3 逻辑运算符	51
1.5 总结	16	3.3.4 简化求值	53
1.6 习题	16	3.3.5 运算符的优先级和结合性	54
项目2 完成数据计算	18	3.4 程序实现	55
2.1 项目要求	18	3.5 switch 语句	56
2.2 项目描述	18	3.6 选择结构嵌套问题	58
2.3 计算机中的数据类型	19	3.7 条件运算符	61
2.3.1 变量和常量	19	3.8 总结	62
2.3.2 整型数据	21	3.9 习题	63
2.3.3 浮点型数据	23	项目4 大量数据求和	68
2.3.4 字符型数据	24	4.1 项目要求	68
2.4 算术运算符和赋值运算符	26	4.2 项目描述	68
2.4.1 算术运算符和算术表达式	26	4.3 完成大量数据求和	69
2.4.2 赋值运算符和赋值表达式	27	4.3.1 复合赋值运算符	69
2.4.3 优先级和结合性	27	4.3.2 自增和自减运算符	70
2.5 任意两个数相加	29	4.3.3 重复多次操作	70
2.5.1 从键盘上输入变量的值	29	4.3.4 循环和循环变量	71
2.5.2 格式化输入(scanf)	32	4.3.5 循环中的初始值	72
2.6 求平均值	33	4.3.6 define 的用法	75
2.6.1 数值类型转换	34	4.4 继续讨论循环问题	75
2.6.2 格式化输出	35	4.4.1 for 语句	76
2.7 总结	38	4.4.2 while 语句	77
2.8 习题	39	4.4.3 do-while 语句	78

4.4.4 goto 语句.....	79	6.4.4 求出每门课程的平均分.....	124
4.4.5 循环嵌套.....	80	6.4.5 完整的程序.....	124
4.5 break 和 continue 语句.....	81	6.5 总结.....	127
4.6 结构化程序设计.....	83	6.6 习题.....	128
4.6.1 结构化程序设计的概念.....	83	项目 7 用结构体处理学生成绩	131
4.6.2 结构化程序设计的基本结构.....	84	7.1 项目要求.....	131
4.6.3 结构化程序设计的原则.....	84	7.2 项目描述.....	131
4.6.4 结构化程序设计的特点、优点 和缺点.....	85	7.3 字符.....	132
4.7 总结.....	85	7.4 字符串和字符数组.....	133
4.8 习题.....	86	7.4.1 字符数组的输入和输出.....	134
项目 5 成绩的计算	93	7.4.2 字符串处理函数.....	136
5.1 项目要求.....	93	7.5 结构体.....	138
5.2 项目描述.....	93	7.5.1 定义新的结构体类型.....	138
5.3 什么是数组.....	93	7.5.2 定义结构体类型变量的方法.....	139
5.3.1 数组声明.....	94	7.5.3 结构体变量的初始化.....	141
5.3.2 引用数组元素.....	94	7.5.4 使用结构体变量.....	141
5.3.3 初始化数组元素.....	96	7.5.5 结构体数组.....	141
5.4 数组元素在内存中的形式.....	98	7.6 解决项目问题.....	142
5.4.1 地址的概念.....	98	7.7 typedef 自定义数据类型.....	147
5.4.2 运算符 sizeof.....	99	7.8 总结.....	148
5.4.3 数组在内存中的存储.....	100	7.9 习题.....	148
5.4.4 引用超过数组范围的元素.....	101	项目 8 编写一个日历程序	154
5.5 排序算法.....	101	8.1 项目要求.....	154
5.5.1 冒泡排序.....	102	8.2 项目描述.....	154
5.5.2 选择排序.....	106	8.3 函数.....	154
5.6 总结.....	108	8.3.1 库函数.....	155
5.7 习题.....	109	8.3.2 函数声明.....	155
项目 6 多门功课成绩的计算	114	8.3.3 自己编写函数.....	156
6.1 项目要求.....	114	8.3.4 函数的嵌套调用.....	160
6.2 项目描述.....	114	8.3.5 函数的递归调用.....	162
6.3 二维数组.....	115	8.3.6 数组作为函数的参数.....	166
6.3.1 二维数组的概念.....	115	8.4 写一个日历程序.....	168
6.3.2 二维数组的初始化.....	116	8.4.1 实现 main 函数.....	168
6.3.3 二维数组内存存放方式.....	117	8.4.2 实现 GiveInstructions.....	169
6.4 问题求解.....	117	8.4.3 实现 GetYearFromUser.....	169
6.4.1 数据输入输出.....	118	8.4.4 实现 PrintCalendar.....	169
6.4.2 计算每个人的平均分.....	122	8.4.5 实现 PrintMonth.....	170
6.4.3 找出平均分最高的 3 人.....	123	8.4.6 完成最后的工作.....	174
		8.5 变量的作用域和生存周期.....	179

8.5.1 变量的作用域.....	179	9.6 总结.....	202
8.5.2 变量的存储方式和生存期.....	180	9.7 习题.....	203
8.5.3 作用域和生存期的小结.....	181	项目 10 大数求平均值问题	206
8.6 总结.....	182	10.1 项目要求.....	206
8.7 习题.....	183	10.2 项目描述.....	206
项目 9 为函数设置多个返回值	187	10.3 位运算符.....	207
9.1 项目要求.....	187	10.3.1 “按位与”运算符(&).....	208
9.2 问题的提出.....	187	10.3.2 “按位或”运算符().....	208
9.3 指针.....	189	10.3.3 “异或”运算符(^).....	208
9.3.1 地址和指针.....	189	10.3.4 “取反”运算符(~).....	209
9.3.2 声明指针变量.....	190	10.3.5 左移运算符(<<).....	209
9.3.3 指针的基本操作.....	190	10.3.6 右移运算符(>>).....	209
9.4 让函数返回多个值.....	192	10.3.7 复合赋值运算符.....	209
9.5 指针和数组.....	195	10.4 位运算的实际应用.....	210
9.5.1 指针运算.....	196	10.5 总结.....	211
9.5.2 指针的自增和自减.....	197	10.6 习题.....	212
9.5.3 指针和数组名.....	198	附录 A 二进制数及其他	215
9.5.4 指针作为函数参数.....	199	附录 B ASCII 码表	224
9.5.5 动态分配.....	199	附录 C 运算符的优先级和结合性	226
9.5.6 void *类型.....	200	附录 D 常用库函数	228
9.5.7 动态数组.....	201	附录 E C 语言关键字	232
9.5.8 释放内存.....	201		

要让计算机工作，首先需要告诉计算机干什么，即执行什么指令。指令是计算机能识别的集合。我们把计算机需要执行的程序写下来，然后输入到计算机中，计算机按照程序指定的工作。这些程序也称为“软件”。在计算机中，使用的是二进制数（二进制数与计算机的关系，见附录 A）。随着计算机的发展，纯粹地使用二进制数编程的要求，人们用一些英文符号代替二进制指令来进行编程，这就是汇编语言。完成从汇编语言到机器语言转换的程序称为编译程序。

机器语言和汇编语言都是面向硬件的具体操作的，对机器过分依赖，要求使用者必须对硬件结构及其工作原理都十分熟悉，这对于计算机的推广应用是不利的。随着计算机应用用途被逐渐地扩大，有越来越多的人参与到计算机的使用和编程中来，也有越来越多的程序需要编写，汇编语言已经满足不了要求。这促使人们去寻求一些与人类自然语言相近且能为计算机所接受的语意确定、规则明确、自然易懂和通用易学的计算机语言。这种与自然语言相近并为计算机所接受和执行的计算机语言称为高级语言。

此后，高级语言大量出现，如方便初学者学习编程的 BASIC 语言，用于操作系统编写的 C 语言，主要应用于情报检索、商业数据处理等管理领域的 COBOL 语言，军用编程语言 Ada 语言，以及 PASCAL 语言、Java 语言、C++ 语言、C# 语言等。

C 语言是一种计算机程序设计语言，它既有高级语言的特点，又有汇编语言的特点。C

项目1 显示广告语



1.1 项目要求

- (1) 初步认识计算机编程。
- (2) 了解 C 语言的结构。
- (3) 掌握 printf 语句输出文字的方法。
- (4) 掌握 Code::Blocks 的使用方法。
- (5) 了解 C 语言的发展史及其特点。

1.2 项目描述

广告是企业在进行宣传时所采用的一种必不可少的方式，如摩托罗拉手机广告“Hello MOTO”、李宁运动服饰广告“一切皆有可能”、农夫山泉广告“农夫山泉有点甜”等。不管是平面广告还是电视广告，除了文字之外还包括图片、声音等。本项目通过计算机编程显示最简单的文字，但不显示视频、图片、声音等。

要让计算机工作，首先需要告诉计算机干什么，即执行什么指令。程序就是一组指令的集合。我们把计算机需要执行的程序写下来，然后输入到计算机中，计算机将会自动执行程序指定的工作。这些程序也称为“软件”。在计算机中，使用的是二进制数(关于二进制数与计算机的关系，见附录 A)。随着计算机的发展，纯粹地使用二进制数编程无法满足要求，人们用一些英文符号代替二进制指令来进行编程，这就是汇编语言。完成从汇编语言到机器语言转换的程序则称为编译程序。

机器语言和汇编语言都是面向硬件的具体操作的，对机器过分依赖，要求使用者必须对硬件结构及其工作原理都十分熟悉，这对于计算机的推广应用是不利的。随着计算机的用途被逐渐地扩大，有越来越多的人参与到计算机的使用和编程中来，也有越来越复杂的程序需要编写，汇编语言已经满足不了要求。这促使人们去寻求一些与人类自然语言相接近且能为计算机所接受的语意确定、规则明确、自然直观和通用易学的计算机语言。这种与自然语言相近并为计算机所接受和执行的计算机语言称为高级语言。

此后，高级语言大量出现，如方便初学者学习编程的 BASIC 语言，用于操作系统编写的 C 语言，主要应用于情报检索、商业数据处理等管理领域的 COBOL 语言，军用编程语言 Ada 语言，以及 PASCAL 语言、Java 语言、C++ 语言、C# 语言等。

C 语言是一种计算机程序设计语言，它既有高级语言的特点，又有汇编语言的特点。C

语言由美国贝尔研究所的 D.M.Ritchie 于 1972 年推出, 1978 年之后被移植到大、中、小及微型计算机上。它既可以作为操作系统设计语言, 编写系统应用程序, 也可以作为应用程序设计语言, 编写不依赖计算机硬件的应用程序。它的应用范围广泛, 具备很强的数据处理能力, 不仅仅是在软件开发上, 而且各类科研都需要用到 C 语言, 所以我们选择 C 语言来学习计算机编程。

接下来我们就开始用 C 语言完成本项目的要求——写广告词。首先以摩托罗拉手机的广告“Hello MOTO”为例进行说明。

例 1-1 摩托罗拉手机广告“Hello MOTO”文字显示。

```
/*=====
*程序名称: HelloMOTO.c
*功能: 显示摩托罗拉手机广告“Hello MOTO”
*=====*/
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello MOTO!\n");

    return 0;
}
```

运行这个程序会在显示屏上显示如下文字:

Hello MOTO!

这里用一个带圆角的方框代表程序运行时计算机显示器上会出现的文字, 在本书中都会以这种方式来显示程序运行的结果。

说明:

(1) 位于“/*”和“*/”中间的所有内容都是注释。这部分不是 C 语言的语句, 是对程序所做的说明, 是写给编程人员看的, 计算机编译的时候会自动跳过本段。注释是程序中必不可少的部分, 增加了代码的可读性。一个好的程序必须加上合适的注释。

除了用/*和*/来标注块注释外, 还可以用“//”来对一行代码进行注释。注释可以写在任何位置。

(2) #include <stdio.h> 用于说明程序中包含了一个库文件。#include 是一条预处理指令, 就是在进行程序编译之前预先处理的指令。其作用是把 stdio.h 文件的内容调出来放在此位置, 这样在程序中就可以使用 stdio.h 文件的内容了。

所谓的库就是一种工具集合, 这些工具由其他程序员编写, 用于执行特定的功能。本程序中使用的库是由 ANSI C 提供的标准输入输出库(stdio.h)。如果在程序中还需要其他的库, 则对每一个库都必须使用一行#include。

在编写程序的时候，使用库提供的工具可以省去自己编写这些工具的麻烦。要使用一个库，必须提供足够的信息让编译器知道库里有哪些工具可以使用。对于库的说明写在了一个头文件(.h)中。如 `stdio.h` 是一个头文件，它定义了标准的输入输出库的内容。

`#include` 有两种形式，分别为

```
#include "stdio.h"
```

以上两种 `#include` 指令形式的区别是：使用尖括号表示在编译系统的包含文件目录中去查找被包含文件(包含目录是由用户在设置环境时设置的)，这称为标准形式。使用双引号则表示首先在当前的源文件目录中查找被包含文件，若未找到才按照标准形式到编译系统的包含文件目录中查找。如果使用系统提供的库文件，则用标准形式比较好；若包含用户自己的库文件，则可以把该库文件放在用户源文件目录中，然后用双引号的形式引用。

(3) `int main()`

```
{
    printf("Hello MOTO!\n");

    return 0;
}
```

是主函数。C 语言程序的功能是由函数来实现的，程序总是从 `main` 函数开始执行，在 `main` 函数中结束，而且不论 `main` 函数在整个程序中的位置如何。

函数可以理解为实现某个功能的独立程序段。一个函数主要由以下几部分组成：

函数类型 函数名(参数类型 参数名)

```
{
    函数体;
}
```

`int` 就是函数的类型，是整型类型(整型类型在项目 2 中讲解)；`main` 就是函数的名称；`{ }` 表示了这个 `main` 函数的开始和结束的位置；函数体是在 `{ }` 中间的语句，实现了程序的功能。

(4) “`return 0;`”语句用于返回数据。我们运行的所有程序都是基于操作系统(如 Windows 系统或者 Linux 系统)的，程序运行完成后要向操作系统报告运行的状态。“`return 0;`”是向操作系统返回一个 0 值，表示程序正常运行结束。

(5) “`printf("Hello MOTO!\n");`”语句用于调用函数 `printf` 并输出一段文字。要在显示屏上显示文字，需要涉及 CPU、内存、显卡、显示屏以及操作系统等的工作原理，操作非常复杂，C 语言开发者编写了一个输出函数 `printf`，让使用 C 语言的编程人员可以不用了解上述工作原理就可以进行编程。在程序设计中，通过函数名来使用该函数的行为称为调用，在 `HelloMOTO.c` 中的语句

```
printf("Hello MOTO!\n");
```

就是对函数 `printf` 的调用。

在调用一个函数的时候，我们需要提供一些额外的信息。如 printf 可以在屏幕上显示信息，但是显示什么信息呢？这些额外的信息由括号内的参数说明。参数就是一个函数调用程序提供给函数的信息。此处的参数是由双引号内的一串字符组成的。

当程序执行时，printf 会逐个地显示“H”、“e”、“l”等，直到所有的字符都出现在屏幕上。在字符串的结尾部分，是一个特殊字符\n，称为换行字符。当 printf 输出最后的“!”时，光标停留在文本末尾的“!”后面。当函数 printf 输出换行字符(\n)时，光标会移动到下一行的开始位置，就像我们按下了 Enter 键一样。在编写程序时，必须在合适的位置使用“\n”，否则所有的输出都将不分行地显示在一起。

对于例 1-1 的程序，除了“printf("Hello MOTO!\n");”是实现了程序的功能外，其他的全部是结构性的语句。这就是说，如果另外编写一个程序，只需要把“printf("Hello MOTO!\n");”一句换成实现相应功能的语句就可以了，程序的其他部分不需要改变。接下来我们编写李宁运动服饰的广告词。

例 1-2 李宁运动服饰广告词“一切皆有可能”文字显示。

```

/*=====
*程序名称: Lining.c
*功能: 显示李宁运动服饰广告词“一切皆有可能”
*=====*/
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("一切皆有可能!\n");
    return 0;
}

```

运行这个程序会在显示屏上显示如下文字：



有了这个基础，显示农夫山泉的广告词就非常容易了。

例 1-3 农夫山泉广告词“农夫山泉有点甜”文字显示。

```

/*=====
*程序名称: NonFu.c
*功能: 显示农夫山泉广告“农夫山泉有点甜”
*=====*/
#include <stdio.h>
#include <stdlib.h>

```

```
(5) int main()
```

```
{
```

```
    printf("农夫山泉有点甜!\n");
```

```
    return 0;
```

```
}
```

1.3 运行这个程序会在显示屏上显示如下文字:

```
农夫山泉有点甜!
```

利用 printf 函数可以显示很多内容。可以用如下的程序显示《诗经》中的名句：“关关雉鸣，在河之洲。窈窕淑女，君子好逑。”

例 1-4 《诗经》文字显示。

```
/*=====
```

```
*程序名称: songs.c
```

```
*功能: 按下面的格式显示
```

```
    诗经
```

```
    关关雉鸣,
```

```
    在河之洲。
```

```
    窈窕淑女,
```

```
    君子好逑。
```

```
*/=====
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    printf("诗经\n");
```

```
    printf("\n");
```

```
    printf("关关雉鸣, \n");
```

```
    printf("在河之洲。 \n");
```

```
    printf("窈窕淑女, \n");
```

```
    printf("君子好逑。 \n");
```

```
    return 0;
```

```
}
```

程序运行后会在显示器上显示如下结果:

诗经

关关雎鸠，
在河之洲。
窈窕淑女，
君子好逑。

根据上面的例题，编写程序显示图 1-1 所示的图案。

通过上述例子我们学会了使用 C 语言编写一个在显示屏上显示文字的程序，但是这样的程序还不是计算机的可执行程序。C 语言程序文件保存的文件后缀名为“.c”，称为源程序文件，如 HelloMOTO.c。这个文件需要经过预处理、编译、链接三个步骤才能变成可执行程序。完成这三个步骤的工具软件分别称为预处理器、编译器、链接器。



图 1-1

预处理器对源程序进行转换，以生成符合编译器要求的等价的 C 语言源程序。由于这个等价的 C 语言源程序不会被保存，所以对于编程人员来说是不可见的。

编译器用于将预处理器转换好的 C 语言源程序文件生成目标文件，也就是“obj 文件”。其文件名与 C 语言程序文件相同，不同的编译器其后缀名也不同。同时，编译器还可以查找程序中的错误，如语法错误、类型错误、声明错误等。

链接器将库文件以及由编译器生成的若干个目标文件整合成一个可执行程序。在 Windows 系统中，可执行程序的后缀名为“.exe”。

在众多的 C 语言编译器中，GNU/GCC 是一款优秀的开源编译器。Code::Blocks 使用 GNU/GCC 编译器，是一款专门的 C/C++ 集成开发环境。集成开发环境(IDE)中包含了完成预处理、编译、链接等步骤的各种工具软件，我们只需要点击一个按钮就可以实现多个步骤，直接生成最终的可执行程序。另外，集成开发环境还在“项目管理”、“源代码管理”、“代码分析”等功能上提供了大量的工具，所以我们在编写程序时，往往只使用某一款集成开发环境。本书中的所有程序都在 Code::Blocks 中编译通过。

1.3 Code::Blocks 的安装和使用

编程工具种类繁多，其中图形化的编程工具相比其他工具可以更为方便地进行程序的编写和调试。具有代表性的跨平台 IDE 中就集成了图形化的编程工具。Code::Blocks 除了具备 IDE 典型的集编辑、编译、调试于一体的功能之外，还具有以下特点：

- (1) 开源。开源不仅仅意味着免费，它还意味着有更好的学习途径。
- (2) 跨平台。Code::Blocks 可以在 Windows、Linux、Mac OS 等多个平台上运行。
- (3) 跨编译器。Code::Blocks 支持 GCC/g++、Visual C++、Borland C++、Intel C++ 等 20 多款编译器。
- (4) 采用插件式框架。Code::Blocks 采用开放体系，具有良好的功能扩展能力。

(5) 由 C++ 写成。Code::Blocks 程序本身就是用 C++ 语言编写的，无需安装额外的、往往是庞大的运行环境。

(6) 内嵌可视化 GUI 设计。Code::Blocks 的图形界面采用 wxWidgets 设计，同时也支持使用 wxWidgets 进行可视化图形界面设计。

(7) 支持多国语言。通过附加不同的语言包，可以实现菜单的多国语言化。安装中文语言包，界面菜单是方便中国人使用的汉字。

1.3.1 Code::Blocks 的安装

Code::Blocks 最新版本是 13.12，可以在官方网站 <http://www.codeblocks.org/downloads/26> 下载。本书使用 10.05 版本，因为汉化文件基于 10.05 版本。如果是 Windows 系统，则下载 codeblocks-10.05mingw-setup.exe，此文件自带了编译器，对于初学者来说比较方便。

1. 在 Windows 系统中安装

双击下载好的文件 codeblocks-10.05mingw-setup.exe 即可以开始安装。安装时要选择全部安装(Full: All plugins, all tools, just everything)，防止有些库未安装的情况出现。选择路径时，点击 Browse 选择要安装的目录，默认路径为 C:\Program Files\codeblocks。最后点击 Finish，即可完成安装过程。

2. 在 Linux 系统中安装

在 Linux 系统中安装 Code::Blocks，应该首先安装基本的编译环境。在 Fedora14 中可以打开终端，执行命令：`sudo yum install gcc gcc-c++ *aclocal -y`，即可将所需要的编译工具安装完毕。

然后安装 Code::Blocks。以 Fedora14 为例，打开终端，输入命令：`sudo yum install codeblocks* xterm -y`，即可自动安装。

1.3.2 Code::Blocks 的汉化

1. Windows 系统中的汉化

(1) 第一次启动 Code::Blocks 时，会出现图 1-2 所示的对话框。选择编译器，选择“GNU/GCC Compiler”，点击“OK”即可。

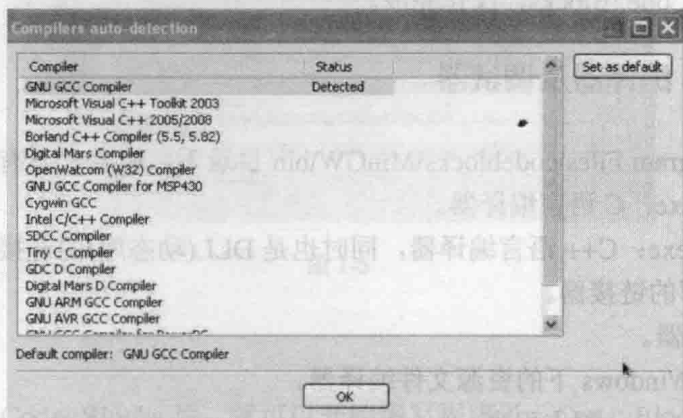


图 1-2

(2) 启动程序，进入 Code::Blocks 主界面，这时会弹出一个名为“Tip of the Day”的小对话框，如图1-3所示。这是每日提示，把“Show tips at startup”前的勾去掉，就可以使该提示不再显示。

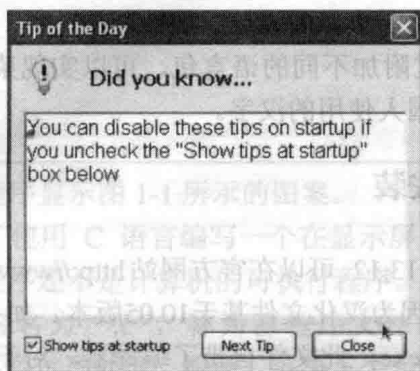


图 1-3

(3) 安装完毕后 Code::Blocks 是英文显示的，我们需要进行汉化。汉化包在 http://www.d2school.com/codeblocks/download/d2school_codeblocks_chinese_locale.7z 下载，解压后会产生一个 zh_CN 文件夹。先在 C:\Program Files\codeblocks\share\codeblocks\ 目录中新建名为 locale 的文件夹，并将获得的 zh_CN 文件夹复制到 locale 目录中，然后打开软件，选择菜单“settings→Environment→View”，勾选“Internationalization(will take place after restart)”，选择选项“Chinese(Simplified)”。重新启动 Code::Blocks，即可完成汉化。

2. Linux 系统中的汉化

- (1) 终端中执行命令：`sudo mkdir /usr/share/codeblocks/locale`，新建 locale 文件夹。
- (2) 把 zh_CN 文件夹复制到刚才建立的目录下。假设 zh_CN 文件夹放在~/目录下，则执行命令：`sudo cp -R ~/zh_CN/ /usr/share/codeblocks/locale`。
- (3) 打开软件，选择菜单“settings→Environment→View”，勾选“Internationalization(will take place after restart)”，然后选择“Chinese(Simplified)”。
- (4) 执行命令：`sudo chmod -R 777 /usr/share/codeblocks/locale/`，把 zh_CN 文件夹权限改为 777。
- (5) 重新启动 Code::Blocks，汉化完成。

1.3.3 配置 g++ 编译器及调试器

首先到 C:\Program Files\codeblocks\MinGW\bin 目录下，检查一下有没有以下文件：

mingw32-gcc.exe: C 语言编译器。

mingw32-g++.exe: C++ 语言编译器，同时也是 DLL(动态库)的链接器。

ar.exe: 静态库的链接器。

gdb.exe: 调试器。

windres.exe: Windows 下的资源文件编译器。

mingw32-make.exe: 制作程序。

确认没有问题后,则点击 Code::Blocks 主菜单“设置”(汉化前的“Settings”),选中“编译器和调试器设置...”,在出现的对话框中,在左侧选中“全局编译器设置”,然后对照图 1-4,检查右侧的“可执行工具链”下的配置是否正确无误。

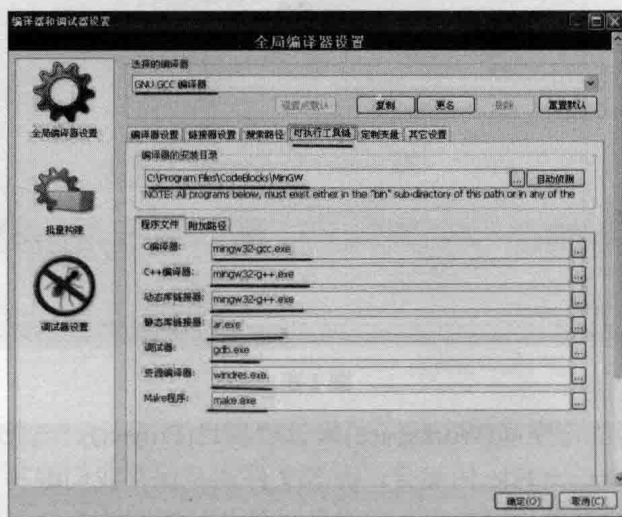


图 1-4

一般来说,程序正常安装好后这些配置都是没有问题的。但是如果上述配置不正确,Code::Blocks 将无法进行工作。

若是在 Linux 系统中,则需打开菜单“设置→编译器和调试器设置→全局编译器设置”,检查其中的配置是否与图 1-5 中的一样。

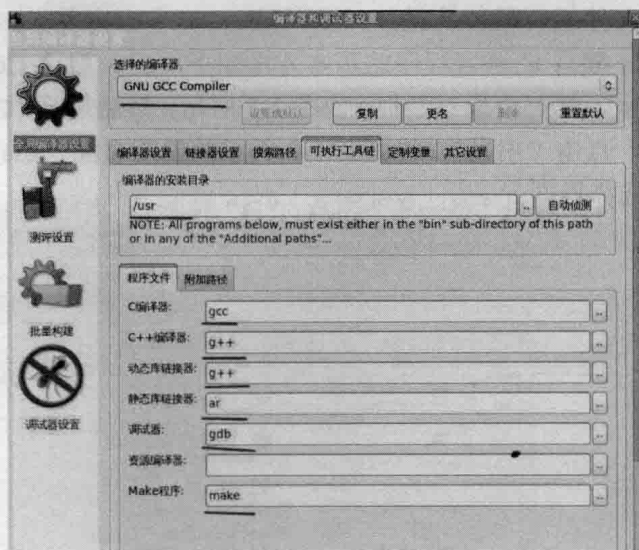


图 1-5

1.3.4 编写程序

安装和配置好 Code::Blocks 后,就可以开始编写程序了。Code::Blocks 的界面如图 1-6 所示。