

职业教育“十四五”新形态教材建设项目

重庆市高等教育学会高等教育科学研究课题项目教材（项目编号 CQGJ19B196）

C 语言程序设计

项目式教程

（含实训与试题）

主 编 ◎ 陈燕熙 田 楠 黄晓艳

副主编 ◎ 何婷婷 王 维

陈树林 汤 东



新形态一体化教材



活页式



习题及答案解析



课件

西南交通大学出版社

· 成 都 ·

图书在版编目 (C I P) 数据

C 语言程序设计项目式教程 : 含实训与试题 / 陈燕熙, 田楠, 黄晓艳主编. —成都: 西南交通大学出版社, 2021.8

ISBN 978-7-5643-8162-2

. C... . 陈... 田... 黄... . C 语言 - 程序设计 - 高等职业教育 - 教材 . TP312.8

中国版本图书馆 CIP 数据核字 (2021) 第 150678 号

C Yuyan Chengxu Sheji Xiangmushi Jiaocheng
(Han Shixun yu Shiti)

C 语言程序设计项目式教程
(含实训与试题)

主 编 陈燕熙 田 楠 黄晓艳

责任编辑 / 李华宇
封面设计 / 曹天擎

西南交通大学出版社出版发行
(四川省成都市金牛区二环路北一段 111 号西南交通大学创新大厦 21 楼 610031)
发行部电话: 028-87600564 028-87600533
网址: <http://www.xnjdcbs.com>
印刷: 四川玖艺呈现印刷有限公司

成品尺寸 185 mm × 260 mm
印张 16.5 字数 381 千
版次 2021 年 8 月第 1 版 印次 2021 年 8 月第 1 次

书号 ISBN 978-7-5643-8162-2
定价 48.00 元

课件咨询电话: 028-81435775
图书如有印装质量问题 本社负责退换
版权所有 盗版必究 举报电话: 028-87600562

前言

PREFACE

“C 语言程序设计”课程作为理工科计算机专业和非计算机专业学生的专业基础课，是覆盖面较大、应用广泛的一门课程，也是许多相关专业课程的前导课程，还是训练学生计算思维的重要课程。这门课程将要学习的 C 语言是非计算机专业学生计算机应用能力水平二级考试的主要语种之一。

本课程的教学目标是学习使用一种典型的程序设计语言——C 语言，建立程序设计的概念，初步掌握程序设计方法和技巧，养成良好的程序设计习惯，在项目式教学中实践软件项目开发的过程，从而具备应用程序设计解决相关专业领域内实际问题的基本能力。“C 语言程序设计”是一门实践性很强的课程，既要掌握较系统的概念和理论，又要结合实际动手编程，还要上机调试运行。通过本课程的学习，学生不但能掌握算法、基本概念，编制相应的程序，并初步掌握软件开发的基本技巧，同时也能为后继课程的学习打下坚实的基础。因此，本课程在培养具有创新能力的应用型人才培养方面占有不可或缺的地位。

本书内容

本书共包含 10 个项目，分别是：

项目 1：程序设计宏观认识，主要知识点包括 C 程序的基本结构、C 程序开发的步骤以及 VC++6.0 开发工具的使用等。

项目 2：程序设计基础知识，主要知识点包括 C 语言的数据类型、常量、变量、表达式、运算符等。

项目 3：顺序结构程序设计，主要知识点包括顺序结构的基本知识、格式化输入函数、格式化输出函数、字符输入函数、字符输出函数等。

项目 4：选择结构程序设计，主要知识点包括 if 选择语句和 switch 语句等。

项目 5：循环结构程序设计，主要知识点包括 while 循环语句、do...while 循环语句、for 循环语句、循环结构控制语句 break 和 continue 等。

项目 6：数据的批量处理，主要知识点包括一维数组、二维数组、字符数组以及字符串处理函数等。

项目 7：模块化程序设计，主要知识点包括函数的基本使用、带参宏定义、函数的递归调用、作用域及存储类型等。

项目 8：利用指针访问数据，主要知识点包括地址与指针、用指针引用一维数组和二维数组、用字符指针引用字符串、返回地址的函数等。

项目 9：用户自定义数据类型，主要知识点包括结构体类型的基本使用、结构体数组与指针变量、结构体与函数、单链表、共用体和枚举类型等。

项目 10：文件操作，主要知识点包括文件的基本知识、文件的打开与关闭、文件的顺序读写、文件的随机读写。

本书特色

本书通过整合教学内容，重构教学体系，使其具有基础性、实用性和系统性。每个项目的教学内容分为三大模块：项目基础知识讲解、项目考核、项目实训。各项目和实训内容的选取注重实用性、应用性和趣味性。

本书按照理论与实训一体化标准，以项目为导向，以任务为驱动，以经典、实用项目为载体组织教学内容，通过“做”理解 C 语言程序设计规则，掌握语法和编程技术，实现了“教、学、做”合一。

本书读者

本书可作为各类职业院校和培训机构“C 语言程序设计”课程的教材，也可作为程序设计爱好者自学的参考教材。

教学资源

本书配套有视频教学资料、课件、课后习题及其答案、项目实训等资源。

编者
2021 年 5 月

目录

CONTENT

项目 1	程序设计宏观认识	1
任务 1.1	程序宏观框架结构	1
任务 1.2	程序的构成	3
任务 1.3	程序开发过程及运行环境	5
项目考核	22
项目实训	24
项目 2	程序设计基础知识	25
任务 2.1	C 程序基础知识概述	25
任务 2.2	数据类型	27
任务 2.3	常 量	29
任务 2.4	变 量	33
任务 2.5	运算符和表达式	37
任务 2.6	数据类型转换	52
任务 2.7	算法与结构化程序设计	54
项目考核	60
项目实训	62
项目 3	顺序结构程序设计	64
任务 3.1	顺序结构的概述	64
任务 3.2	格式化输出 printf 函数	66
任务 3.3	格式化输入 scanf 函数	72
任务 3.4	字符输入输出函数	77
项目考核	80
项目实训	84
项目 4	选择结构程序设计	86
任务 4.1	if 选择语句	86
任务 4.2	多分支 switch 语句	99

项目考核	103
项目实训	105
项目 5 循环结构程序设计	106
任务 5.1 循环的概述	106
任务 5.2 while 循环语句	108
任务 5.3 do-while 循环语句	111
任务 5.4 for 循环语句	114
任务 5.5 循环结构控制语句	119
任务 5.6 循环的嵌套	122
项目考核	128
项目实训	131
项目 6 数据的处理批量	132
任务 6.1 一维数组	132
任务 6.2 二维数组	137
任务 6.3 字符数组定义及初始化	141
任务 6.4 字符数组的输入与输出	144
任务 6.5 字符串处理函数	146
项目考核	151
项目实训	154
项目 7 函数	155
任务 7.1 函数的概述	155
任务 7.2 函数的基本使用	158
任务 7.3 带参宏定义	167
任务 7.4 函数递归调用	169
任务 7.5 作用域及存储类型	173
项目考核	175
项目实训	178
项目 8 利用指针访问数据	179
任务 8.1 地址和指针	179
任务 8.2 用指针引用一维数组	184
任务 8.3 用指针引用二维数组	188
任务 8.4 用字符指针引用字符串	192
任务 8.5 返回地址的函数	194

项目考核	197
项目实训	201
项目 9 用户自定义数据类型	202
任务 9.1 结构体类型的基本使用	202
任务 9.2 结构体数组和结构体指针变量	211
任务 9.3 结构体与函数	216
任务 9.4 单链表的建立	219
任务 9.5 共用体类型	222
项目考核	226
项目实训	229
项目 10 文件操作	230
任务 10.1 文件基本知识	230
任务 10.2 文件的打开与关闭	232
任务 10.3 文件的顺序读写	234
任务 10.4 文件的随机读写	239
项目考核	242
项目实训	244
附 录	245
附录 A 关键字	245
附录 B C 语言运算符优先级和结合性	246
附录 C ASCII 码	248
附录 D C 语言常用的库函数	249
参考文献	256



项目 1 程序设计宏观认识

项目导读

计算机语言是计算机和人交流的工具，其必须遵守一定的规则，具有一定的书写格式。C 语言就是一种计算机语言，诞生于 1972 年，是由美国电话电报公司（AT&T）贝尔实验室的 Dennis M. Ritchie 设计。C 语言诞生至今已 40 多年，其发展经历了 ALGOL（1960 年）—CPL（1963 年）—B（1970 年）—C（1972 年）的演变过程。本项目首先介绍了 C 语言的程序宏观框架结构和程序的构成，然后通过几个简单的 C 语言程序实例，让读者了解 C 语言程序宏观框架结构和程序的构成，最后介绍了 C 程序的开发步骤及开发环境，其目的是使读者能够通过编写一个简单的 C 语言程序练习使用开发环境。

学习目标

- 熟悉 C 语言程序宏观框架结构；
- 熟悉 C 语言程序结构及格式特点；
- 熟悉 C 程序的开发步骤；
- 掌握使用 Visual C++ 6.0 编译器开发 C 程序的方法。

任务 1.1 程序宏观框架结构

在现实生活中，人与人之间的交流用的是语言，如汉语、英语、法语等。那么，人与计算机之间交流用的是什么呢？人和计算机交流可以采用对话形式的一问一答方式，也可采用任务方式，即把需要交流的内容集中在一起，一次性完成。前者的专业术语叫解释方式，后者叫编译方式。用 C 语言进行人和计算机交流采用的是编译方式。那么 C 语言程序的宏观框架结构又是怎样的呢？下面就通过几个简单的实例来认识一下 C 语言程序。

任务 1.1.1 基本的程序框架结构

【例 1-1】请在计算机或一些智能终端设备上打印显示“Hello, World!”的字样信息。

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

```
/*编译预处理指令*/
```

```

void main ()          /*主函数的函数头*/
{                    /*函数体的开始标记*/
    Printf ("Hello, World!\n" );    /*输出要显示的字符串*/
}                    /*函数的结束标记*/

```

本程序看似很简单，但却包含 C 语言程序的两个最基本的程序框架结构——编译预处理和函数组。

(1) 编译预处理部分，本例中的程序段如下：

```
#include <stdio.h>          /*编译预处理指令*/
```

有关编译预处理的几点说明：

编译预处理的第二种书写格式：#include “stdio.h”。

程序编译之前的准备工作，以#开头。

include 称为文件包含命令，后面尖括号中的内容称为头文件。

stdio.h 是 C 语言的系统文件，stdio 是“ standard input & output(标准输入输出)”的缩写，“.h”是文件的扩展名。

(2) 函数组部分，本例中的程序段如下：

```

void main ()          /*主函数的函数头*/
{                    /*函数体的开始标记*/
    printf ("Hello, World!\n");    /*输出要显示的字符串*/
}                    /*函数的结束标记*/

```

有关函数组的几点说明：

每个 C 语言程序都必须有一个 main 函数。

一个 C 程序总是从 main 函数开始执行，到 main 函数结束，与 main 函数所处的位置无关。

包括主函数、库函数和用户自定义函数。

函数是构成 C 语言程序的基本单位。

函数组中有且仅有一个主函数 main ()。

程序的执行以主函数为核心展开。

程序语句，用“；”作为语句结束符。

除了主体框架中的“编译预处理”和“函数组”以外，在程序代码中还有注释部分，用以增强程序的可读性，例 1-1 中 /**/ 即为程序语句增加的注释说明。

任务 1.1.2 程序注释

注释部分有以下两种格式：

注释格式：

```

//          /*单行注释的起始符号*/
/* */      /*可作为单行或多行注释*/

```

【例 1-2】从键盘输入矩形的长和宽，计算并输出其周长。

分析：

根据题意需要定义三个整型变量：int a, b, c;

周长的计算： $c=2*(a+b)$;

输入/输出函数：scanf 和 printf。

```
/* example1-2 The second C Program*/
#include <stdio.h>           //包含标准输入输出头文件
void main()                 //主函数
{   int a, b, c;           //定义整型变量，长 a、宽 b、周长 c
    printf("请输入矩形的长和宽:"); //调用输出函数，显示提示语
    scanf("%d%d", &a, &b); //数据输入，从键盘输入 a、b 值
    c=2*(a+b);             //数据计算，将计算的周长赋值给 c
    printf("该矩形周长为: %d.\n", c); //调用输出函数输出结果
}
```

例 1-2 的程序要比例 1-1 复杂一些，但是程序中的主要框架还是由编译预处理和函数数组两个部分组成。程序的开头增加了一条说明语句：/*example1-2 The second C Program*/；编译预处理部分没有变化，只有一条包含标准输入输出头文件 stdio.h 的命令，为主函数中要用到 printf()函数和 scanf()函数时做准备；函数组部分只有一个主函数 main()，主函数通过数据准备、数据输入、数据计算、结果输出等语句完成本题目的要求。初学者都可以参照例 1-2 的程序代码进行程序的编写。

课堂练习

模仿编程，输入正方形的边长，计算其面积。

任务 1.2 程序的构成

一篇格式规范的文章由字、词、句子以及段落逐级构成，C 语言程序也是如此。从微观的角度来看，C 语言程序是由标识符、程序语句、函数等表示形式构成，最终形成完整的 C 语言程序代码。

任务 1.2.1 标识符

在日常生活中，我们每个人都有不同且唯一的身份证号，C 语言与此类似，需要对所处理的数据进行唯一标识。在例 1-1 和例 1-2 的程序代码中，由 void、main、int、printf、scanf、a、b、c 等一系列符号构成了程序中的语句的函数，这些符号统称为标识符。在 C 语言中，有许多东西都需要命名，如变量名、函数名、数组名等，在命名的时候都必须遵守一定的规则，按此规则命名的符号称为标识符。

C 语言的标识符可以分为以下三类：关键字、预定义标识符和用户自定义标识符。

1. 关键字

C 语言规定了一批标识符，它们在程序中都代表着固定的含义，不能另作他用。例如，用来说明变量类型的标识符 `int`、`float`、`double` 以及 `if` 语句中的 `if`、`else` 等都已有的专门的用途，它们不能再用作变量名或函数名。注意，所有的关键字都必须小写。C 语言中的关键字请参考附录 A。

总体而言，C 语言中的关键字大致可以分为以下几类：

- (1) 数据类型说明符，如整型 `int`、字符型 `char` 等。
- (2) 语句定义符，如选择结构中的 `if`、`else`，循环结构中的 `for`、`while`、`do` 等。
- (3) 存储类型关键字，如声明自动变量的 `auto`、声明静态变量的 `static` 等。
- (4) 其他关键字，如声明常量的 `const` 等。

2. 预定义标识符

这些标识符在 C 语言中也都有特定的含义，如 C 语言提供的库函数的名字（如 `include`、`printf`、`scanf`）和预编译处理命令（如 `define`）等。

3. 用户自定义标识符

由用户根据需要定义的标识符称为用户标识符。一般用来给变量、函数、数组或文件等命名。注意，用户标识符不允许与关键字同名；建议不使用预定义标识符作为用户标识符，因为会造成二义性。

用户自定义标识符的命名规则：标识符只能由字母、下划线、数字组成，且第一个字符必须是字母或下划线，不能是数字。如 `str`、`_str1`、`str_2` 都是合法的标识符，但 `2str`、`2_str`、`&123`、`%lss0`、`M.Jack`、`-L2` 都是错误的标识符。

在用户自定义标识符使用时还应注意以下几点：

(1) C 语言对大小写字符敏感，标识符要严格区分英文字母大小写，如 `score` 和 `Score` 是两个不同的标识符。

(2) 标识符不能是 C 语言中的关键字。

(3) 应尽量使标识符具备相应的意义，使其可以“见名知义”，从而提高程序的可读性，例如学生名字变量可以命名为 `StuName`。

(4) 理论上，C 语言并不限制标识符的长度，但实际标识符的长度受不同 C 语言编译系统和机器系统的限制。标识符的长度各个系统不同，最好不要超过 8 个字符。例如，某些编译程序中规定标识符前 8 位有效，即当两个标识符前 8 位相同时，被认为是同一个标识符。

任务 1.2.2 程序语句

程序语句是构成函数的基础，根据语句在程序中所起到的作用可分为说明语句和可执行语句，通过这些语句实现对数据的描述和操作。通常每行只写一条语句，每条语句以“；”结束，如果没有写“；”，在程序编译时会出现“missing‘;’”的错误提示。

任务 1.2.3 函 数

函数是构成 C 语言程序的基本单位，而程序语句则是函数的重要组成。函数可分为主函数（如 main）、库函数（如 printf）和自定义函数。自定义函数的结构形式如下：

```

函数返回值类型  函数名 (形式参数)           //函数首部
{ 说明语句;           //函数体
  执行语句;
}

```

函数的第一行称为函数首部，大括号“{}”括起来的部分称为函数体，函数体由若干语句组成，函数体使用一对大括号“{}”作为定界符。

在使用函数时还应注意以下几点：

- (1) C 程序中可以由若干个函数和一个主函数组成。
- (2) 每一个可执行的 C 程序必须有且只能有一个 main 函数。
- (3) C 程序的执行总是从 main 函数开始，执行完 main 函数返回系统。
- (4) 定义语句应出现在执行语句之前，每条语句后面的分号不可少。多条语句可以写在同一行上，并且一条语句也可以写在多行上。

课堂练习

- (1) 在 C 语言中，用户标识符的命名规则是什么？
- (2) 以下选项中不合法的用户标识符是（ ）。
_123 printf A\$ Dim Main void
- (3) 识别例 1-2 程序中的标识符。

任务 1.3 程序开发过程及运行环境

用 C 语言编写的程序称为源程序，不能直接被计算机识别和执行，需要一种担任翻译工作的程序，即编译程序。通过程序编译把 C 语言程序代码转换为计算机能够处理的二进制目标代码。

任务 1.3.1 C 程序的开发过程

学习 C 语言就是学习 C 语言编程的过程。C 程序的开发从确定任务到得到结果一般要经历以下几个步骤。

1. 编辑程序

用编辑软件将 C 源程序输入计算机，经修改无误后，保存为一个文件。C 源程序文件的后缀为“.c”或“.cpp”。可用于编写 C 源程序的编辑软件有很多，DOS 环境下，可以使用 Turbo C；Windows 环境下，可以使用 WIN TC（Turbo C 的 Windows 版本），也可

以使用 C-Free，也可以使用 Visual C++。

2. 编译程序

在 WIN TC 或 C-Free 或 Visual C++ 下，将后缀为 “.c” 或 “.cpp” 的源程序编辑保存之后，通过快捷键或者选择菜单的方式进行编译，编译的过程是把 C 源程序代码转换为计算机可识别的代码。如果在编译过程中发现源程序有语法错误，则系统会显示出错误信息，告诉用户源程序有错误，然后用户可以重新修改源程序再进行编译，如此反复直至编译通过为止。编译通过后生成目标程序，在 Visual C++ 6.0 中，目标文件的后缀为 .obj，如 file.obj。

3. 连接程序

经过编译得到的目标文件还不能供计算机直接执行。前面已经介绍过，一个程序可能包含若干个源程序文件，而编译是以源程序文件为对象的，一次编译只能得到一个与源程序文件相对应的目标文件，所以必须把所有的目标文件连接装配起来，再与函数库相连接，生成一个可供计算机直接执行的文件，称为可执行文件（在 Visual C++ 6.0 中，可执行文件的后缀为 .exe，如 file.exe）。

4. 运行程序

只要输入可执行文件的文件名即可运行程序。在 C-Free 或 WIN TC 或 Visual C++ 下通过快捷键或选择菜单的方式即可运行程序。

任务 1.3.2 程序开发环境

如前所述，当按照 C 语言的要求编写程序后，计算机是不能直接运行它的，必须借助相应的程序或软件对编写的程序进行编译、连接，将其变成可执行程序，才能运行它。“工欲善其事，必先利其器”，我们到底需要什么工具进行 C 语言编程呢？C 语言开发环境有很多，如 Linux 操作系统下的 GCC，Windows 操作系统下常用的有 Turbo C 2.0、Turbo C++ 3.0、Dev-C++、C-Free 和 Visual C++ 6.0 等。下面将介绍一款适应性很强的 C 语言开发工具——Visual C++ 6.0。

Visual C++ 6.0 是一个功能强大的可视化软件开发工具，它集合了程序的代码编辑、程序编译、连接、运行和调试等功能。Visual C++ 6.0 也是全国计算机二级考试中指定的编译器。本书中所有程序的编写都是基于 Visual C++ 6.0 开发环境的。下面结合一个实例介绍其用法。

1. 启动 Visual C++ 6.0

安装完成 Visual C++ 6.0 后，执行“开始”菜单中的 Microsoft Visual C++ 6.0 命令，或者双击桌面上的 Visual C++ 6.0 图标，可启动 Visual C++ 6.0 开发环境，得到如图 1-1 所示的操作界面。

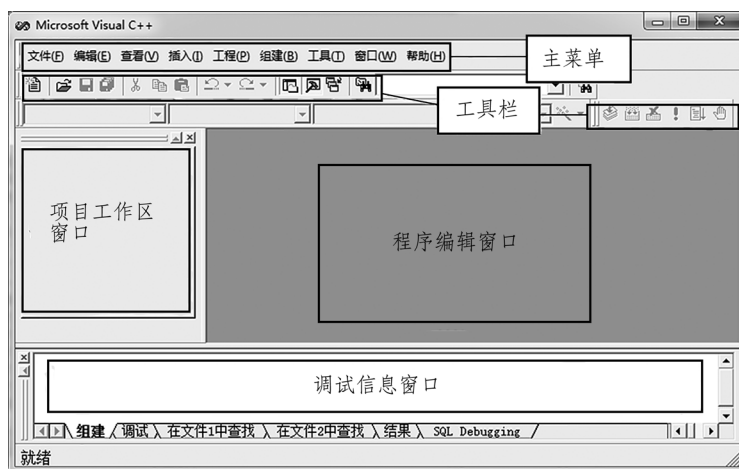


图 1-1 操作界面


如图 1-1 所示，主窗口的顶部为主菜单栏，主菜单栏下面是工具栏，工具栏中包含常用的工具按钮，如编辑程序相关的按钮和调试程序相关的按钮等。主窗口的左侧是项目工作区窗口，用来显示所设定的工作区信息；主窗口的右侧是程序编辑窗口，用来输入和编辑源程序（这时不能输入程序，需要新建文件后方可进行输入）；主窗口的下部是调试信息窗口，用来将编译、连接等结果信息进行显示。

2. 新建一个 C 源程序

(1) 在 Visual C++ 6.0 主窗口的主菜单栏中选择“文件”选项，然后在其下拉菜单中单击“新建”按钮，如图 1-2 所示。



图 1-2 “新建”按钮

(2) 此时会弹出“新建”对话框，如图 1-3 所示。选择此对话框左上角的“文件”选项卡，选择其中的“C++ Source File”选项；在右半部分的“文件名(N)”文本框中输入源程序文件的名字，如图中输入为“1-1.c”；“位置(C)”文本框中是源文件的保存地址，可以通过单击右边的  按钮修改源文件存储路径。

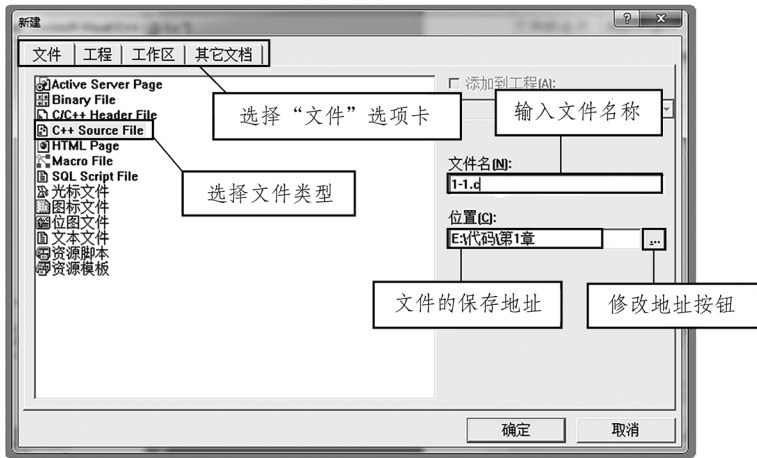



图 1-3 “新建”对话框

(3) 单击图 1-3 中的“确定”按钮后，回到 Visual C++ 6.0 主窗口，此时光标在程序编辑窗口闪烁，表示程序编辑窗口已激活，可以编辑源程序了。输入程序代码，如图 1-4 所示。



图 1-4 输入程序代码

在图最下面一行的矩形框中显示了“行 6，列 49”，表示光标的当前位置是第 6 行第 49 列。当光标位置改变时，显示的数字也随之改变。

(4) 在主菜单栏中选择“文件”选项，在其下拉菜单中选择“保存”菜单项，保存文件。也可以单击工具栏中的“保存”工具按钮 ，或者直接按【Ctrl+S】快捷键来保存文件。

3. 编译程序


(1) 在 Visual C++ 6.0 环境中，要编译源程序可单击主菜单栏中的“组建”按钮，在其下拉菜单中选择“编译[1-1.c]”菜单项，如图 1-5 所示。也可以单击工具栏中的“编译”工具按钮 ，或者直接按【Ctrl+F7】快捷键来编译程序文件。



图 1-5 编译

(2) 在选择编译命令后，屏幕上出现一个对话框，如图 1-6 所示，该对话框的意思是询问是否同意建立一个默认的项目工作区。单击“是(Y)”按钮，表示同意由系统建立默认的项目工作区，然后开始编译程序。

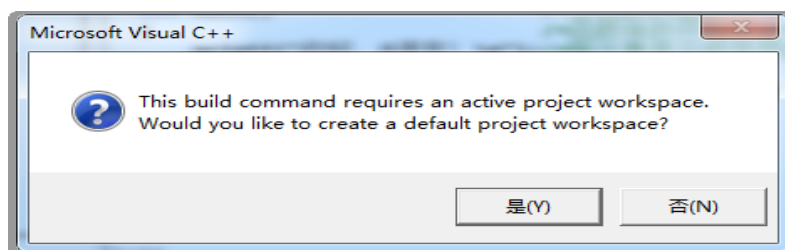


图 1-6 是否同意建立一个默认的项目工作区

(3) 编译时，编译系统会详尽检查源程序中是否有语法错误，然后在主窗口下部的调试信息窗口输出编译信息，如图 1-7 所示。程序编译结果为“0 error(s), 0 warning(s)”，且产生了一个名为“1-1.obj”的目标文件。

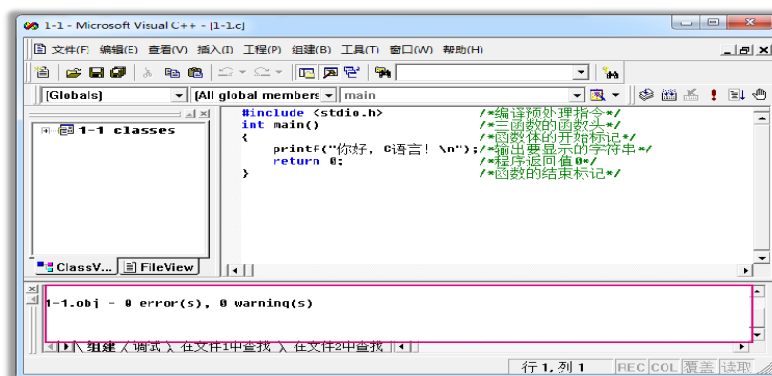


图 1-7 编译信息

(4) 如果程序中存在语法错误，则会在调试信息窗口中输出错误提示信息，如图 1-8 所示。编译结果为“1 error(s), 0 warning(s)”，说明有一个错误，双击错误提示语就会有箭头指向对应的有错误的代码行。修改错误再编译，直到没有错误为止。



图 1-8 错误提示信息

4. 连接程序

(1) 在得到目标程序“1-1.obj”后，还需要进行连接方可生成可执行文件。此时选择“组建” “组建[1-1.exe]”选项，如图 1-9 所示。也可以单击工具栏中的“组建”工具按钮，或者直接按【F7】快捷键来组建程序文件。



图 1-9 组建程序

(2) 组建程序后得到可执行文件。如图 1-10 所示，在主窗口下部的调试信息窗口输出连接信息“1-1.exe - 0 error(s), 0 warning(s)”，意思是生成可执行文件“1-1.exe”，0 个错误和 0 个警告。

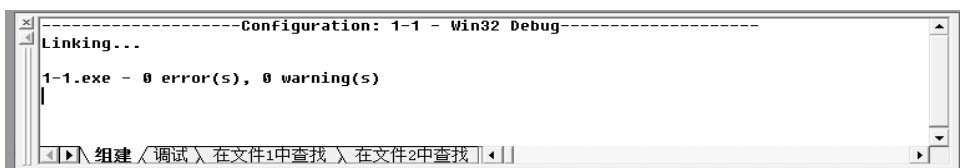


图 1-10 连接信息

5. 运行程序


(1) 在得到可执行文件后, 就可以直接运行“1-1.exe”了。为此, 可选择“**组建**”“**执行[1-1.exe]**”菜单, 如图 1-11 所示。也可以单击工具栏中的“**运行**”工具按钮 , 或者直接按【Ctrl+F5】快捷键来执行程序。



图 1-11 执行

(2) 程序运行后, 屏幕切换到输出结果窗口, 显示运行结果, 如图 1-12 所示。

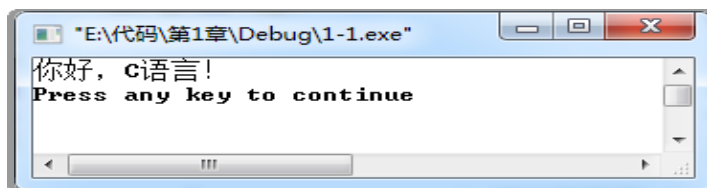


图 1-12 运行结果

任务 1.3.3 C 语言程序设计实验过程问题

问题 1: VC++环境中建立 C 源程序文件时忘了加后缀.C, 如图 1-13 所示。

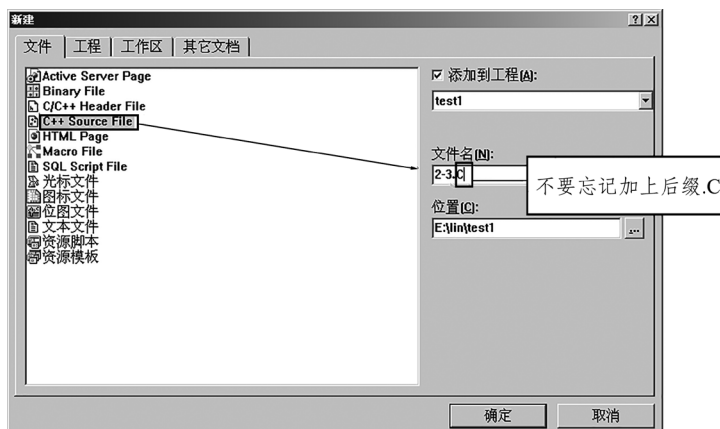


图 1-13 忘记加后缀.C

解释：VC++是 C++语言的集成开发环境，但也可以用来开发 C 程序，但在建立 C 程序源程序文件时一定要加上.c 后缀。

问题 2：同一个工程中出现了多个有 main 开头的 C 程序，如图 1-14 中的 2-1.c 与 2-2.c。

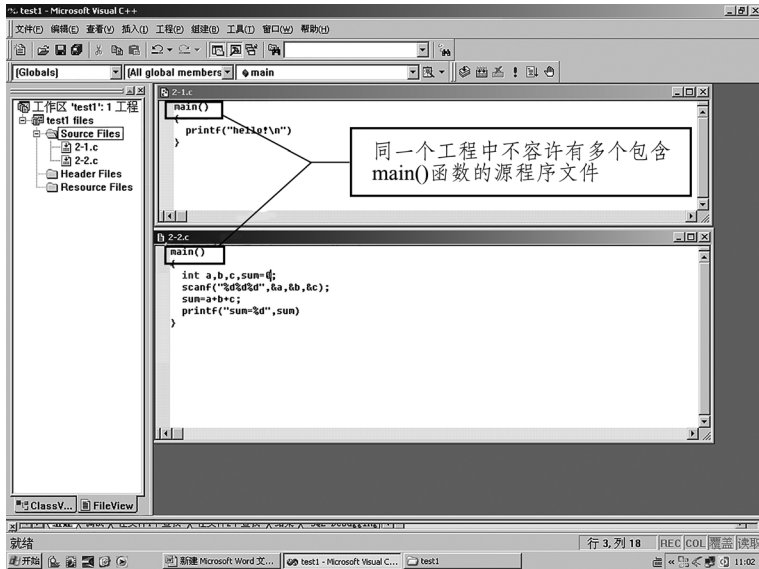


图 1-14 有多个 main 开头的 C 程序

解释：工程中可以有多个 C 程序文件，但不容许有多个包含 main()函数的源文件。鉴于现在 C 程序比较简单，一般工程中只有一个 C 源程序文件，如图 1-15 所示。

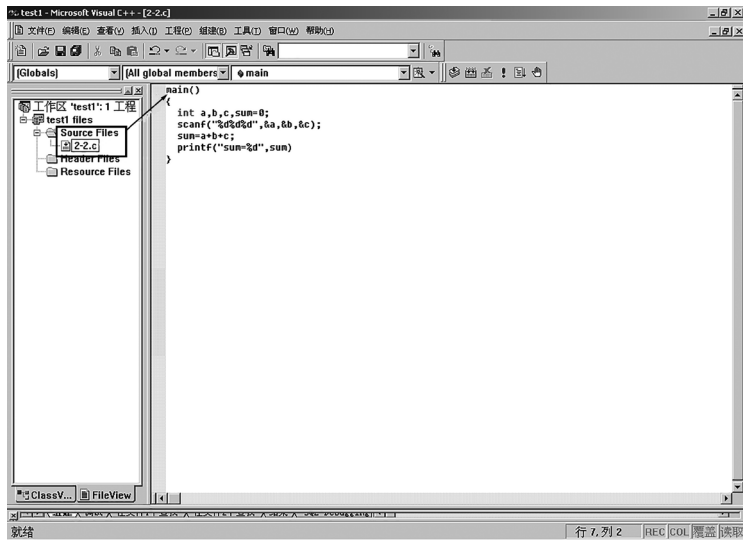
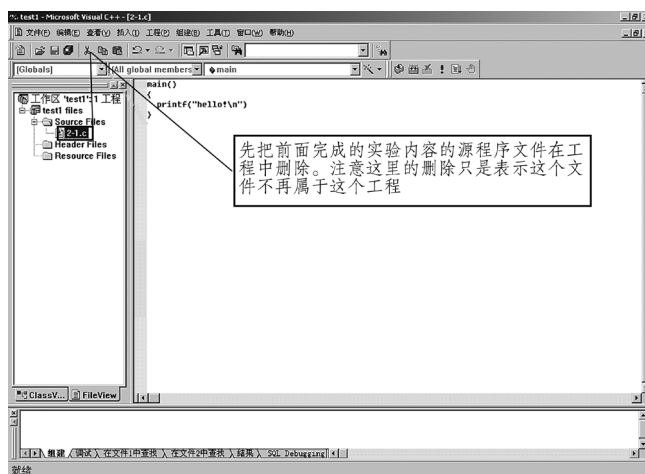


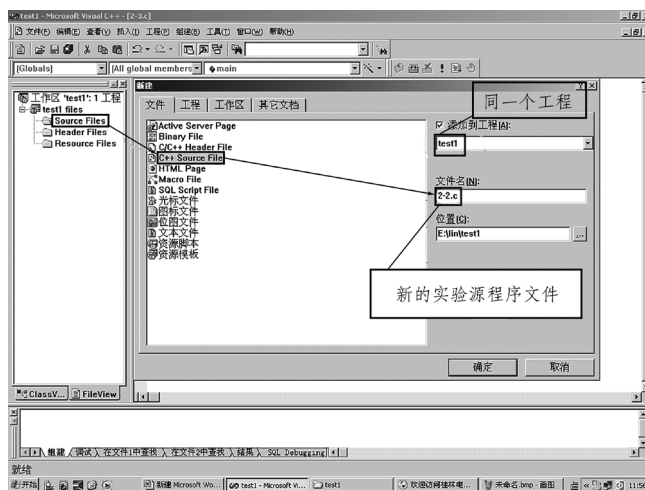
图 1-15 只有一个 C 源程序文件

问题 3：在同一个工程下做完一个实验后，没有重新开始做另一个实验。很多同学是在原有工程下直接建立 C 程序文件，但以前的 C 程序文件又没有在工程中进行删除，这就出现了前面的问题 2。

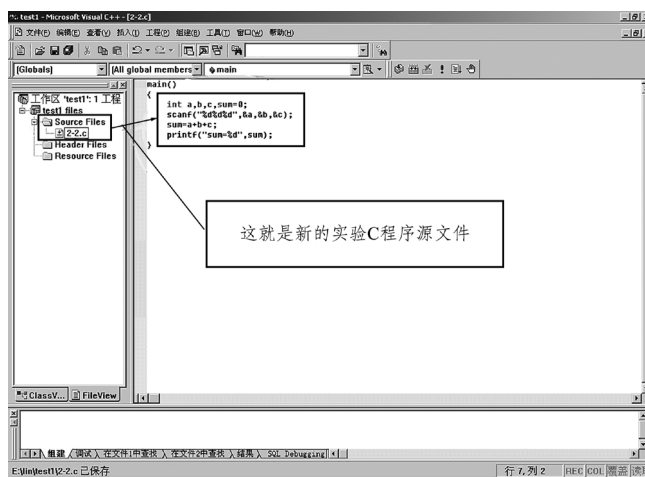
正确做法如图 1-16 所示。



(a) 第一步



(b) 第二步



(c) 新的实验 C 程序源文件

图 1-16 正确做法

问题 4：编译出现错误时不知道怎样找错误。先解释编译结果，如图 1-17 所示。

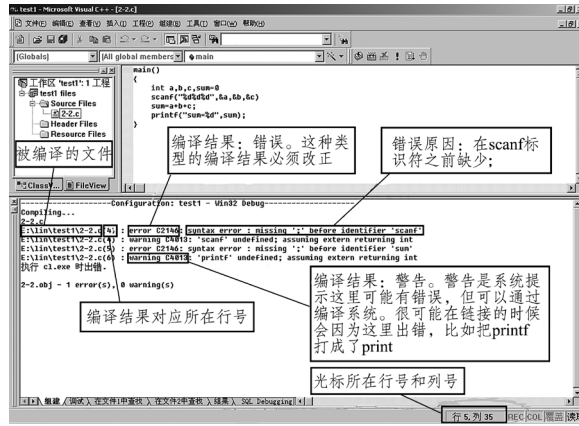


图 1-17 解释编译结果

注意：如果看不懂错误的原因请参考附录 E，也可以查询英文单词的含义。在改正一处错误之后最好重新编译一次。

问题 5：不知道在哪里输入数据和查看结果，如图 1-18 和图 1-19 所示。

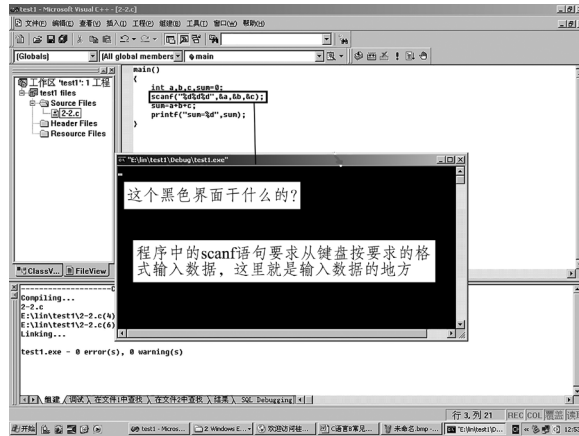


图 1-18 输入数据

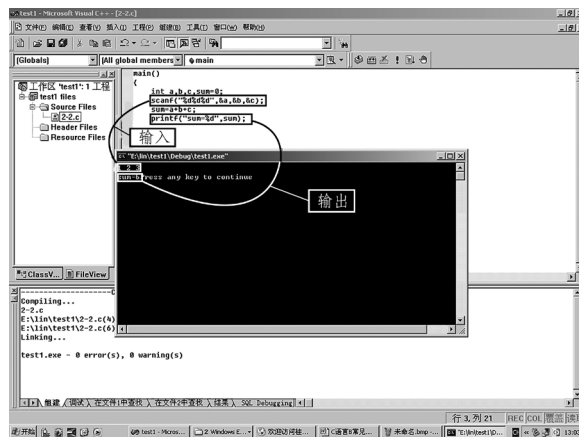


图 1-19 输出结果

任务 1.3.4 其他开发环境

1. VS 2013 开发过程

由于 08、10、12、13 版本在操作上基本没什么变化，所以此处以 13 版为例，建立一个 C++ 工程，其他版本一样。

(1) 打开 VS 2013 文件 新建 项目，如图 1-20 所示。



图 1-20 新建项目

(2) 左侧选择“Visual C++”，中间选择“Win32 控制台应用程序”，下方可以修改名称及存储位置，修改完毕后点击“确定”（这里存储在桌面，名字使用“例子”），如图 1-21 所示。



图 1-21 名称与存储位置

(3) 点击“下一步”，如图 1-22 所示。



图 1-22 应用程序向导

(4) 勾选附加选项的“空项目”，然后点击“完成”，如图 1-23 所示。



图 1-23 空项目

(5) 在右侧右击源文件 添加 项目，如图 1-24 所示。

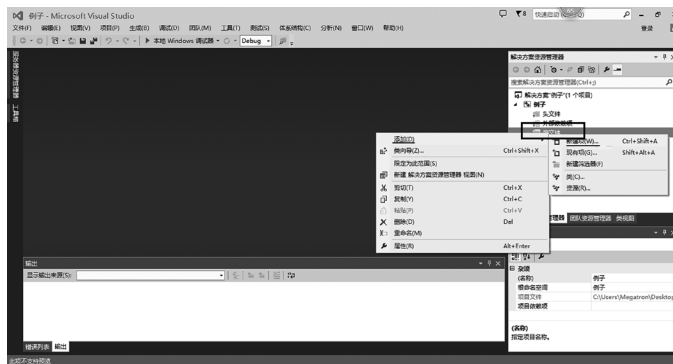


图 1-24 添加

(6) 选择 C++ 文件，这一步很关键，把下面的源 .cpp 改成想命名的名字 .c。切记一定要改成 .c（这里没有修改名字，仅修改了后缀，即“源.c”），修改完毕后点击“添加”，如图 1-25 所示。



图 1-25 修改后缀

(7) 完成后界面如图 1-26 所示，请务必等待左下角显示“就绪”二字以后才可以进行下一步操作。

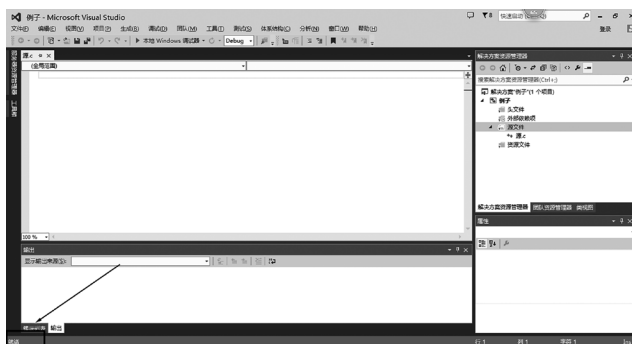


图 1-26 就绪

(8) 编写 C 程序，编写完毕后按 Ctrl+F5 键以运行编写的程序，此时界面如图 1-27 所示，点击“是”。

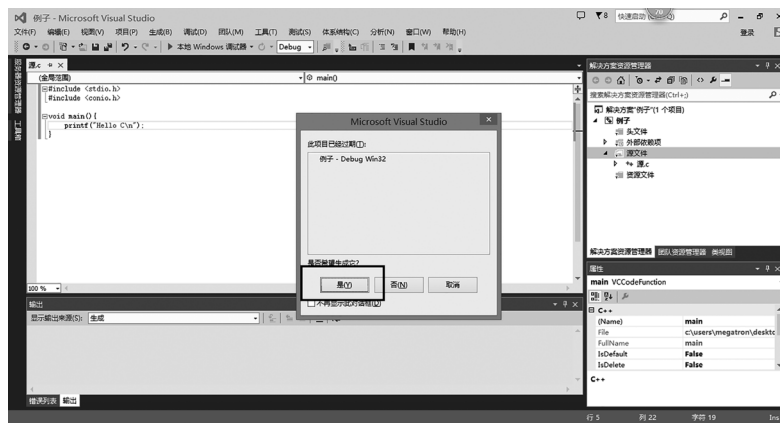


图 1-27 运行程序

(9) 然后会经过一段等待时间，完毕以后会出现如图 1-28 所示的界面。

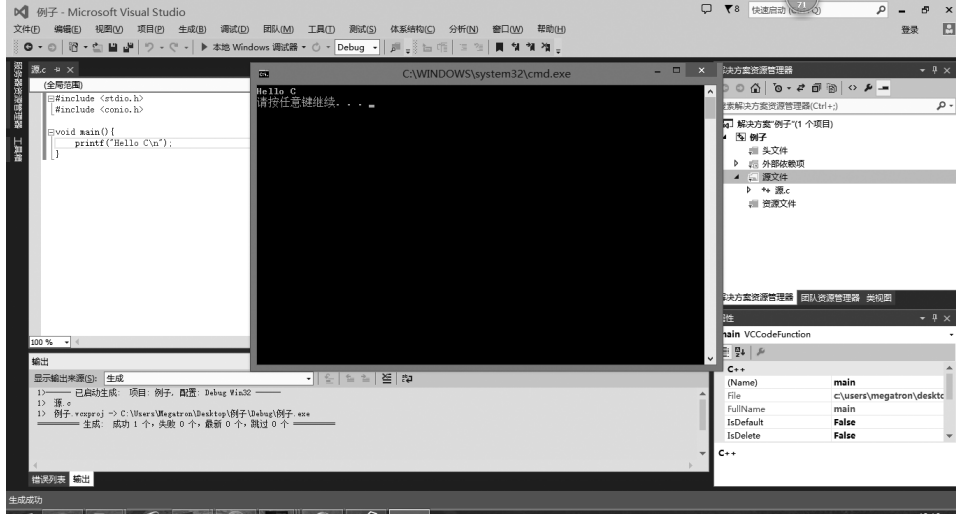


图 1-28 完成界面

2. Turbo C 开发环境

C 语言有许多集成开发环境，可以把程序的编辑、编译、连接和运行等操作全部集中在一个界面上。Turbo C 是其中较常用的一种。

1) Turbo C 的启动

可以通过在 Windows 系统中双击主程序文件 TC.EXE 来启动 Turbo C。主程序文件 TC.EXE 在安装 Turbo C 的文件夹中。启动成功后，屏幕上将显示 Turbo C 的主菜单窗口，如图 1-29 所示。



图 1-29 主菜单窗口

在 Turbo C 主菜单窗口中，用 F10 键和光标移动键可以从主菜单中选择所需的功

能。一般为了防止与其他用户混淆，用户应该建立一个专用的工作目录，来存放自己的文件。

当需设定用户文件的存放目录时，应选“File”选项，即将亮块移到“File”后按下回车键，即会弹出“文件”下拉菜单，如图 1-30 所示。

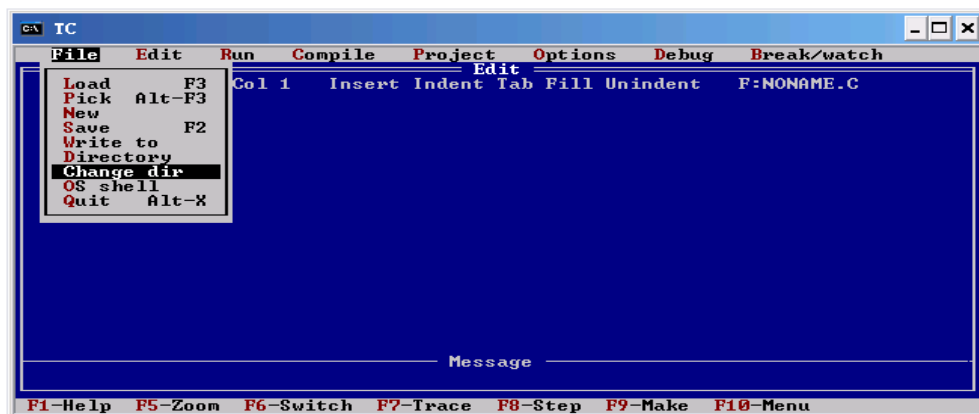


图 1-30 文件下拉菜单

将光标移至“Change dir”选项并按回车键，则弹出一个子目录提示框。用户可在该提示框中输入用户文件的存放目录，输入完后按回车键即可，如输入 D:\user (D 盘中已创建好 user 文件夹)，如图 1-31 所示。

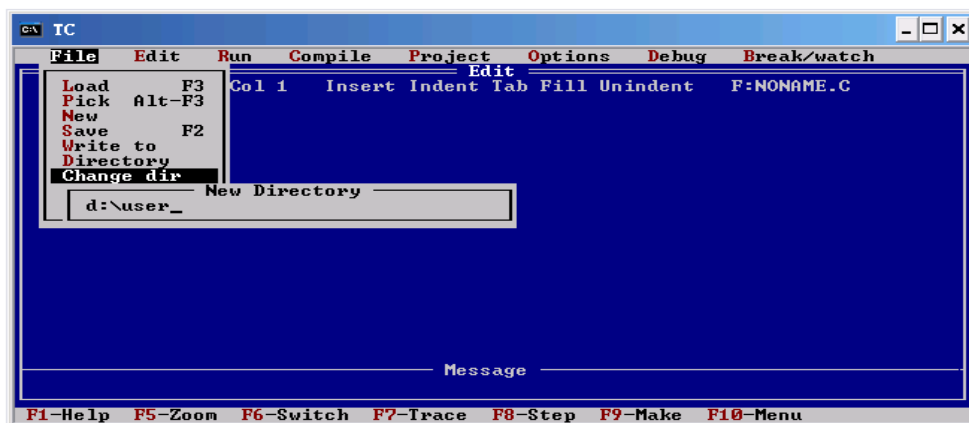


图 1-31 用户文件存放目录

设定好用户文件的存放目录后，用户编写的源文件将会保存在这里，最后，还需要设定编译、连接后的可执行文件的存放目录。用刚才的方法，选择“Options”选项下拉菜单中的“Directories”选项并按回车键，将弹出子菜单，从中选择“Output directory”选项并输入 D:\user。这里设定的就是编译、连接后的可执行文件所存放的目录，如图 1-32 所示。

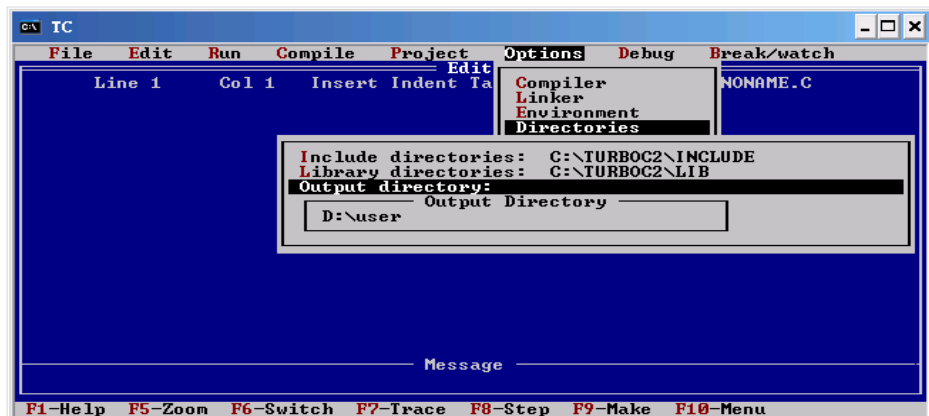


图 1-32 可执行文件存放目录

2) 编辑源程序

在“File”选项的下拉菜单中，选择“New”选项，系统将会给出一个空白编辑窗口，在此编辑源程序。编辑过程中，可用到以下常用的编辑键：

(1) 用光标移动键分别向上、下、左、右移动。用“Home”键到达行首，“End”键到达行末。

(2) 用“Delete”键删除光标所在的字符，用“Backspace”键删除光标左侧的字符。

(3) 用“Insert”键控制工作状态是否为插入状态。按下“Insert”键可看到屏幕编辑窗口上有“Insert”时为插入状态，此时可在屏幕当前光标处插入输入的字符。在插入状态下，再按一下“Insert”键可取消插入状态，状态行上的“Insert”标识消失，此时输入的字符将覆盖光标处的字符。

3) 编译和连接

选择“Compile”选项，按回车键后出现下拉菜单，如图 1-33 所示。

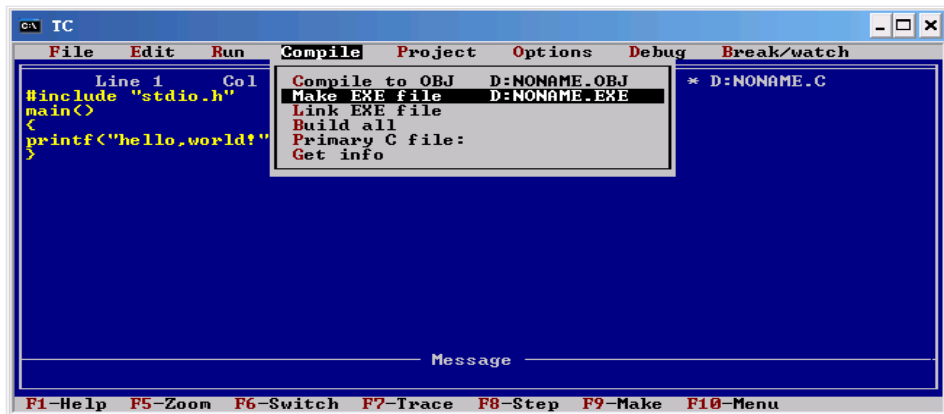


图 1-33 Compile 下拉菜单

选择“Make EXE file”选项对当前编辑窗口的源程序文件进行编译并生成目标文件（扩展名为.OBJ），连接后生成可执行文件（扩展名为.EXE）。若发现语法错误需进行修改。

4) 运 行

按“Esc”键回到主菜单，选择“Run”选项出现下拉菜单，在菜单中选择“Run”命令项，即可运行已生成的可执行文件。若发现结果不对就要重新修改源程序。程序运行后，可在“Run”选项的下拉菜单中选择“User Screen”选项查看结果，如图 1-34 所示。查看后按任意键返回编辑窗口。

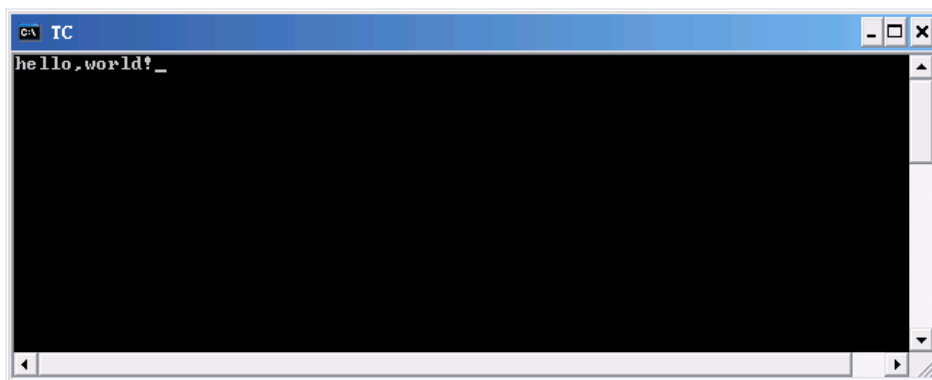


图 1-34 运行结果

5) Turbo C 的退出

在“File”选项的下拉菜单中，选择“Quit”选项，即可退出 Turbo C 集成开发环境。

项目考核

一、选择题

- 下列选项中，关于 main()函数叙述不正确的是（ ）。
 - 一个完整的 C 语言程序中，有且只能有一个 main()函数
 - main()函数是程序的主入口，程序总是从 main()函数开始执行
 - 如果没有声明 main()函数，程序将从第 1 行开始执行
 - main()函数可以有返回值，也可以没有返回值
- 下列关于注释的描述中，错误的是（ ）。
 - C 语言中的注释有两种类型：单行注释和多行注释
 - 所有的注释都可以嵌套使用
 - 单行注释用“//”符号表示，“//”后面为被注释的内容
 - 多行注释以符号“/*”开头，以符号“*/”结尾
- 在 C 语言中，程序的入口函数是（ ）。
 - main
 - printf
 - 自己定义的函数
 - #include<stdio.h>
- C 语言程序的正确运行顺序是（ ）。
 - 编译、编码、连接、运行
 - 编辑、编译、连接、运行
 - 连接、编译、编码、运行
 - 连接、编码、编译、运行
- C 语言规定：在一个源程序中，main 函数的位置（ ）。
 - 必须在最后
 - 必须在系统调用的库函数的后面
 - 必须在最开始
 - 可以任意
- 关于 C 程序的叙述，正确的是（ ）。
 - 一条语句可以写在多行，也可以多条语句写在一行
 - 从//开始的程序都是注释内容
 - /* */只能写在一行，其中的内容为注释部分
 - 一行只能写一条语句
- 构成 C 语言程序的基本单位是（ ）。
 - 字符
 - 语句
 - 函数
 - 文件
- 以下叙述正确的是（ ）。
 - 函数的函数体如果只有一条语句，则“{}”可以省略
 - C 语言中函数必须用预处理#include 将其包含进来
 - 函数体可以放在程序的任意位置
 - 函数的函数体必须用“{}”括起来
- C 语言从（ ）开始执行。
 - 函数
 - 定义变量的语句

项目实训

1. 编程要求:设计并在屏幕上打印自己程序设计的 Logo。例如：

```
*****
*
*   NWSUAF 电子信息 1 班   *****程序   *
*
*
*
*
*           BY JUANQIN WANG           *
*
*                   2020-8-11         *
*****
```

2. 为自己制作一张名片，让老师和同学们记住你的名字。名片中要求包括班级、姓名、性别、电话、专业以及兴趣爱好等内容。

