

第 1 章 初识 C 语言

【内容概述】

C 语言是目前十分优秀的程序设计语言之一,它集高级语言和低级语言的功能于一体,既可用于系统软件的开发,也可用于应用软件的开发,同时它还具有高效、可移植性好等特点。本章主要介绍 C 语言的结构特点、程序组成、书写规则、上机运行过程和调试应用程序的方法。

【学习目标】

通过本章的学习,理解 C 语言程序的构成、C 语言的词法规定和书写规范,掌握 C 语言程序的上机步骤和 C 语言程序的运行环境。

1.1 程序设计语言

我们目前使用的计算机应用系统,如网上购物系统、办公系统、排版系统等,都是由计算机程序设计语言编写而成的。计算机程序设计语言通常简称为编程语言,是一组用来定义计算机程序的语法规则。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据,并精确地定义在不同情况下应当采取的行动。



程序设计语言

1. 程序设计语言的构成

语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。在程序设计语言中,这些记号串就是程序。程序设计语言有三方面的因素,即语法、语义和语用。语法表示程序的结构或形式,即表示构成语言的各个记号之间的组合规律,但不涉及这些记号的特定含义,也不涉及使用者;语义表示程序的含义,即表示按照各种方法所表示的各个记号的特定含义,但不涉及使用者;语用表示程序与使用者的关系。

2. 程序设计语言的发展

(1) 机器语言。最初程序员使用的程序设计语言是一种用二进制代码“0”和“1”形式表示的、能被计算机直接识别和执行的语言,称为机器语言。它是一种低级语言,用机器语言编写的程序不便于记忆、阅读和书写。通常不用机器语言直接编写程序。

(2) 汇编语言。在机器语言的基础上设计出了汇编语言,它可以将机器语言用便于人们记忆和阅读的助记符表示,如 ADD、SUB、MOV 等。计算机运行汇编语言程序时,首先

将用助记符写成的源程序转换成机器语言程序才能运行。汇编语言适用于编写直接控制机器操作的底层程序,它与机器密切相关。汇编语言和机器语言都是面向机器的程序设计语言,称为低级语言。

(3) 高级语言。随着计算机应用的发展,出现了高级程序设计语言,即高级语言。它是一种与硬件结构及指令系统无关,并且表达方式比较接近自然和数学表达式的计算机程序设计语言。

C 语言是一种具有很高灵活性的高级程序设计语言。1972—1973 年,贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言,后来 C 语言又做了多次改进。早期的 C 语言主要用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐被人们认识,到了 20 世纪 80 年代,C 语言开始进入其他操作系统,并很快在各类大、中、小和微型计算机中得到了广泛应用,成为当代最优秀的程序设计语言之一。

3. C 语言的特点

(1) C 语言简洁、紧凑,使用方便、灵活,只有 32 个关键字、9 种控制语句,程序主要用小写字母表示。

(2) 运算符丰富,共有 34 种。C 语言把括号、赋值、逗号等都作为运算符处理,从而使 C 语言的运算类型极为丰富,可以实现其他高级语言难以实现的运算。

(3) 数据结构类型丰富。

(4) 具有结构化的控制语句。

(5) 语法限制不太严格,程序设计自由度大。

(6) C 语言允许直接访问物理地址,能进行位(bit)操作,能实现汇编语言的大部分功能。

(7) 生成目标代码的质量高,程序执行效率高。

(8) 与汇编语言相比,用 C 语言写的程序可移植性好。

但是,C 语言对程序员要求也高,程序员用 C 语言写程序会感到限制少、灵活性大、功能强,但较其他高级语言在学习上要困难一些。

1.2 简单的 C 语言程序介绍

1.2.1 简单的 C 语言程序实例

用 C 语言语句编写的程序称为 C 语言程序或 C 语言源程序。下面先介绍两个简单的 C 语言程序,从中分析 C 语言程序的特性。

【例 1.1】 用 C 语言编写一个程序,输出“你好,我的朋友!”。

程序代码:

```
/* ex1_1.c: 输出欢迎词 */
#include <stdio.h>

int main()                                /* 定义主函数 */
```

```

{
    printf("你好,我的朋友!\n");    /* 输出“你好,我的朋友!” */
}

```

程序运行结果如图 1.1 所示。

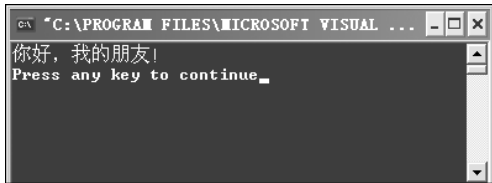


图 1.1 例 1.1 的程序运行结果



简单的 C 语言程序介绍

程序说明如下。

(1) 程序中的 `main()` 代表一个函数,其中 `main` 是函数名,`int` 表示该函数的返回值类型。`main()` 是一个 C 语言程序中的主函数,程序从主函数开始执行。一个 C 语言程序有且只有一个主函数。一个 C 语言的程序可以包含多个文件,每个文件又可以包含多个函数。函数之间是相互平行、相互独立的。执行程序时,系统先从主函数开始运行,其他函数只能被主函数调用或被主函数调用的函数所调用。

(2) 函数体用 `{}` 括起来。`main()` 函数中的所有操作语句都在这一对 `{}` 中间。即 `main()` 函数中的所有操作都在 `main()` 函数的函数体中。

(3) `#include <stdio.h>` 是一条编译预处理命令,声明该程序要使用 `stdio.h` 文件中的内容。`stdio.h` 文件中包含了输入函数 `scanf()` 和输出函数 `printf()` 的定义。编译时系统将头文件 `stdio.h` 中的内容嵌入程序中该命令位置。C 语言中编译预处理命令都以“`#`”开头。

C 语言提供了三类编译预处理命令:宏定义命令、文件包含命令和条件编译命令。例 1.1 中出现的 `#include <stdio.h>` 是文件包含命令,其中尖括号内是被包含的文件名。

(4) `printf()` 函数是一个由系统定义的标准函数,可在程序中直接调用。`printf()` 函数的功能是将要输出的内容输出到显示器上,双引号中的内容要原样输出。“`\n`”是换行符,即在输出完“你好,我的朋友!”后回车换行。

(5) 每条语句用“`;`”号结束。

(6) `/* ... */` 括起来的部分是一段注释。“`/*`”是注释的开始符号,“`*/`”是注释的结束符号,它们必须成对使用。

【例 1.2】 输入两个正整数,计算并输出两数的和。

程序代码:

```

/* ex1_2.c: 求两个正整数的和 */
#include <stdio.h>
int main()                                //主函数
{
    int a,b,num;                          //定义三个整型变量
    printf("请输入两个正整数!\n");
    scanf("%d",&a);                      //输入数据给变量 a
    scanf("%d",&b);                      //输入数据给变量 b
    num= a+b;                             //变量 a 和变量 b 的值相加,然后将结果赋给变量 num
}

```

```
printf("相加结果是%d\n",num); //输出变量 num 的值  
}
```

程序运行结果如图 1.2 所示。

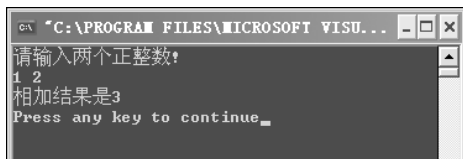


图 1.2 例 1.2 的程序运行结果

程序说明如下。

(1) “int a,b,num;”是变量声明。声明了三个整型变量 a、b、num。C 语言的变量必须先声明后使用。

(2) 程序中的 scanf 是输入函数的名字。程序中 scanf()函数的作用是输入 a、b 的值,&a 和 &b 中的 & 的含义是取地址。此 scanf()函数的作用是将两个数据分别输入变量 a 和 b 的地址所标识的单元中,也就是输入给变量 a 和 b。

(3) “num=a+b;”是将 a、b 两变量的值相加,然后将结果赋值给整型变量 num。

(4) “printf(“相加结果是%d \n”,num);”是调用库函数 printf()输出 num 的结果。“%d”为格式控制,表示 num 的值以十进制整数的形式输出。

(5) “//”之后的内容是注释语句,作用与“/* */”相同。两种注释语句的区别是:“//”后面的注释语句只能在一行中,不能跨行;若注释语句太长,需要占多行,则每行注释之前均需使用“//”符号。“/* */”中的注释语句内容可跨行,无须每行均加“/* */”。

1.2.2 C 语言程序的构成和书写规则

1. C 语言程序的构成

(1) C 语言程序是由函数构成的,函数是 C 语言程序的基本单位。一个源程序至少包含一个 main()函数,即主函数,但可以包含若干个其他函数。被调用的函数可以是系统提供的库函数,也可以是用户根据需要自己编写的函数。

(2) main()函数是每个程序执行的起始点,一个 C 语言程序不管有多少个文件,有且只能有一个 main()函数。一个 C 语言程序总是从 main()函数开始执行,而不管 main()函数在程序中处于什么位置。可以将 main()函数放在整个程序的最前面,也可以放在整个程序的最后面,或者放在其他函数之间。

(3) 源程序可以有预处理命令(include 是其中一种),预处理命令通常放在源文件或源程序的最前面。

(4) 每个语句都必须以分号结尾,但预处理命令、函数头和大括号“}”之后不加分号。

(5) 标识符和关键字之间至少加一个空格以示间隔,空格的数目不限。

(6) 源程序中需要解释和说明的部分可用“/* */”或“//”加以注释。注释是给程

序阅读者看的,机器在编译和执行程序时将忽略注释内容。

2. C 语言程序的书写规则

(1) 在 C 语言中,虽然一行可写多个语句,一个语句也可占多行,但是为了便于阅读,建议一行只写一个语句。

(2) 应该采用缩进格式书写程序,以便增强层次感、可读性和清晰性。低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写。

(3) 用{}括起来的部分通常表示程序的某一层结构。

(4) 为便于程序的阅读和理解,在程序代码中应加上必要的注释。

1.3 C 语言的字符集和词汇

1.3.1 C 语言的字符集

程序是由命令、变量、表达式等构成的语句集合,而命令、变量等是由字符组成的,字符是组成语言的最基本的元素。任何一种语言都有其特定意义的字符集,C 语言字符集由字母、数字、空白符、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

1. 字母

小写字母为 a~z,大写字母为 A~Z,均为 26 个。

2. 数字

数字为 0~9,共 10 个。

3. 空白符

空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用;在其他地方出现时,空白符只起间隔作用,编译程序对它们忽略不计。因此,在程序中是否使用空白符对程序的编译不产生影响,但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

4. 标点和特殊字符

标点和特殊字符既包括+、-、*、/等运算符,又包括_、&、#、!等特殊字符,还包括逗号、圆点、大括号等常用标点符号和括号。



C 语言的字符
集和词汇

1.3.2 C 语言的词汇

在字符集的基础上,C 语言允许使用相关的词汇,以实现程序中的不同职能。在 C 语言

中常使用的词汇包括标识符、关键字、运算符、分隔符、常量。

1. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外,其余都由用户自定义。C 语言规定,标识符只能是字母(A~Z、a~z)、数字(0~9)、下画线(_)组成的字符串,并且其第一个字符必须是字母或下画线。

以下是合法标识符。

a,b,aa1,b2,a_b,a_1,HE,HE_1

以下是非法标识符。

- 1a: 以数字开头。
- H\$E: 出现非法字符\$。
- -1a: 以减号开头。
- a-b: 出现非法字符-(减号)。

提示: 在使用标识符时还必须注意以下三点。

(1) 标准 C 语言不限制标识符的长度,但它受各种版本的 C 语言编译系统限制,同时也受到具体机器的限制。例如,在某版本 C 语言中规定标识符前八位有效,当两个标识符前八位相同时,则被认为是同一个标识符。

(2) 在标识符中,大小写是有区别的。例如,aa 和 AA 是两个不同的标识符。

(3) 标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号。因此,命名应尽量有相应的意义,以便阅读理解,应做到“见名知意”。

2. 关键字

关键字是由 C 语言规定的具有特定意义的字符串,通常也称为保留字。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下几类。

(1) 类型说明符。用于定义、说明变量、函数或其他数据结构的类型,如前面例题中用到的 int、double 等。

(2) 语句定义符。用于表示一个语句的功能,如以后要经常用到的 if-else 就是条件语句的语句定义符。

(3) 预处理命令字。用于表示一个预处理命令,如前面示例中用到的 include。

3. 运算符

C 语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式,表示各种运算功能。运算符由一个或多个字符组成。

4. 分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中,分隔各个变量。空格多用于语句各单词之间,用作间隔符。在关键字、标识符之间必须要有一个以上的空格符作间隔,否则将会出现语法错误。例如,把“int a;”写成“inta;”,C 语言编译器会把 inta 当成一个标识符处理,其结果必然出错。

5. 常量

C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种,在后面章节中将专门介绍。

1.4 C 语言的运行环境

1.4.1 C 语言程序的实现过程

本章所列举的两个实例是已经编写完成的、符合 C 语言语法要求的程序,叫作源程序。一个 C 语言源程序从编写到最终实现结果,需要经过编辑、编译、链接和运行四个过程,如图 1.3 所示。

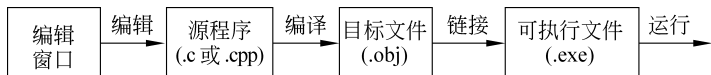


图 1.3 C 语言程序的实现过程



C 语言的运行环境

1. 编辑

对于一种计算机编程语言来说,编辑是在一定的编程工具环境下进行程序的输入和修改的过程。在编程工具提供的环境下,经过用某种计算机程序设计语言编写的程序,保存后生成源程序文件。C 语言源程序也可以使用计算机所提供的各种编辑器进行编辑,比如作为通用编辑工具的记事本、专业编辑工具 Turbo C 和 Visual C++ 等。C 语言源程序在 Visual C++ 环境下默认的文件扩展名为“.cpp”,在 Turbo C 2.0 环境下默认的文件扩展名为“.c”。本书使用的实例都是在 VC++ 环境下编辑和实现的。

2. 编译

编辑好的源程序不能直接被计算机所理解,源程序必须经过编译,生成计算机能够识别的机器代码。通过编译器将 C 语言源程序转换成二进制机器代码的过程称为编译,这些二进制机器代码称为目标程序,其扩展名为“.obj”。

编译阶段要进行词法分析和语法分析,又称源程序分析。这一阶段主要是分析程序的语法结构,检查 C 语言源程序的语法错误。如果分析过程中发现有不符合要求的语法,就会及时报告给用户,将错误类型显示在屏幕上。

3. 链接

编译后生成的目标代码还不能直接在计算机上运行,其主要原因是编译器对每个源程序文件分别进行编译。如果一个程序有多个源程序文件,编译后这些源程序文件还分布在不同的地方,因此,需要把它们链接在一起,生成可以在计算机上运行的可执行文件。在源程序中,输入/输出等标准函数不是用户自己编写的,而是直接调用系统中的库函数,因此,必须把目标程序与库函数进行链接。

链接工作一般由编译系统中的链接程序来完成,链接程序将由编译器生成的目标代码文件和库中的某些文件链接在一起,生成一个可执行文件。可执行文件的默认扩展名为“.exe”。

4. 运行

一个 C 语言源程序经过编译和链接后生成了可执行文件,可以在 Windows 环境下直接双击该文件运行程序,也可以在 Visual C++ 的集成开发环境下运行。

程序运行后,将在屏幕上显示运行结果或提示用户输入数据的信息,用户可以根据运行结果来判断程序是否有算法错误。在生成可执行文件之前,一定要保证编译和链接不出现错误和警告,这样才能正常运行。因为程序中有些警告虽然不影响生成可执行文件,但有可能导致错误结果。

1.4.2 熟悉 Visual Studio 2019 编程工具

Visual Studio 2019 是目前被广泛使用的可视化 C++ 编程工具,同时也是良好的 C 语言编程工具。在 Visual Studio 2019 编程环境下,需要首先建立工程,才能建立、编辑和执行程序,存储的 C 语言源代码文件的扩展名是.cpp。如果在创建文件前没有创建相关工程,系统在编译时会提示是否要创建活动工程。本小节将主要介绍利用编程工具编辑和执行 C 语言程序的基本方法和步骤。

1. C 语言程序的建立

在 Visual Studio 2019 编程环境中,要想建立和执行 C 语言程序文件,首先启动编程工具,建立一个工程,之后才能建立 C 语言文件,具体步骤如下。

(1) 启动 Visual Studio 2019 编程工具,选择“开始”→“所有程序”→Microsoft Visual Studio 2019 命令,可启动 Microsoft Visual Studio 2019 集成开发环境,如图 1.4 所示。

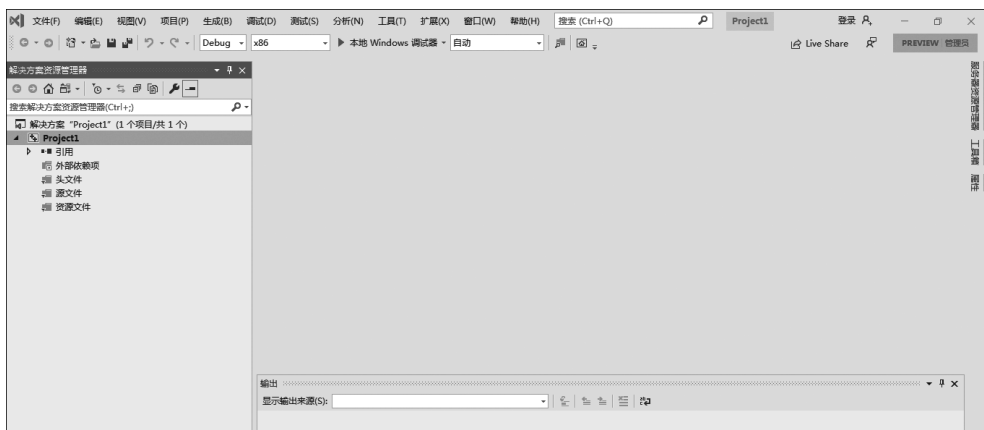


图 1.4 Microsoft Visual Studio 2019 集成开发环境

(2) 建立工程。建立工程是建立 C 语言程序的起始步骤。现在以在“C:\c_study”文件夹下建立 ex1_1 工程为例,介绍建立工程的方法。

在 Visual Studio 2019 集成开发环境下选择“文件”→“新建项目”命令,打开“新建”对话框,选择“空项目”,单击“下一步”按钮,在“配置新项目”对话框中将项目设置为 ex1_1,指定新建工程的路径为“C:\c_study\”,新建工程后的效果如图 1.5 所示。

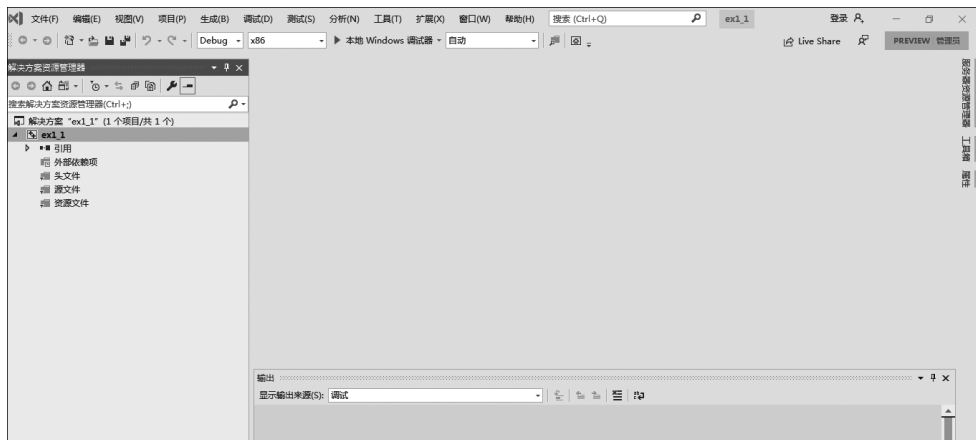


图 1.5 新建工程后的效果

(3) 建立 C 语言程序。新建完工程之后,就可以在此工程下建立 C 语言程序,具体步骤如下。

① 选择“添加”→“新建项”命令,会弹出“添加新项-ex1_1”对话框,如图 1.6 所示。在该对话框中选择“C++ 文件(.cpp)”,然后在“名称”文本框中输入 test.cpp。

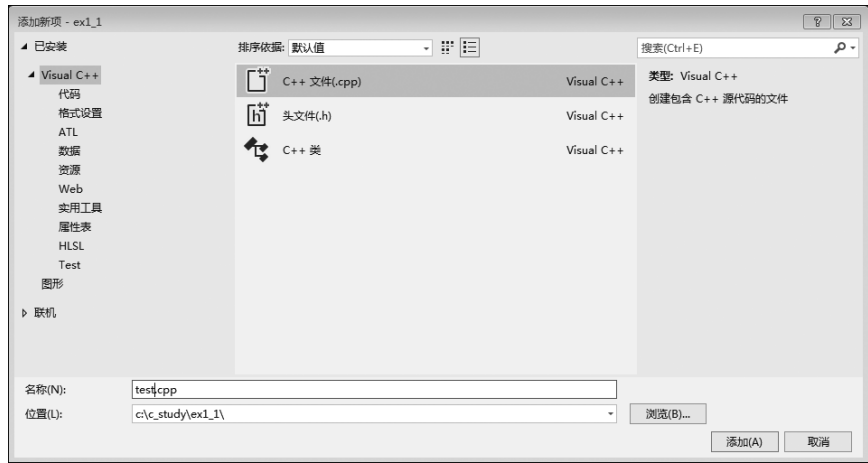


图 1.6 “添加新项-ex1_1”对话框

② 单击“添加”按钮,会显示程序编辑界面,输入 C 语言程序代码,如图 1.7 所示。

③ 选择“文件”→“保存”命令,将文件保存。

2. C 语言程序的运行

编辑好程序之后,接下来要编译和执行程序。在编译之前,应该检查并避免程序代码的错误(当然,在编译时系统也会检查出程序中的错误)。值得注意的是,用 Visual Studio

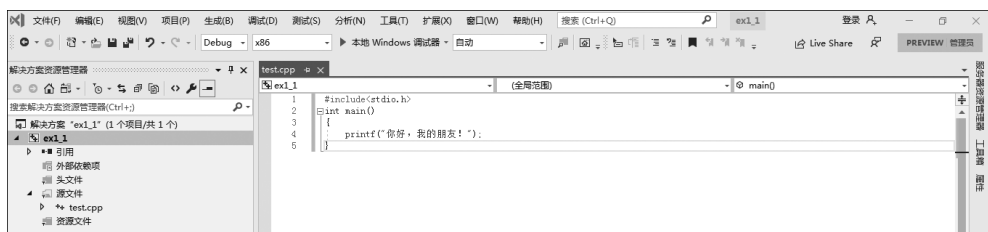


图 1.7 编辑代码

2019 编写 C 语言程序, 当使用输出语句时, “#include <stdio.h>”命令是不能缺少的, 这一点与 Turbo C 环境不同。

选择“调试”→“开始调试”命令(也可以直接按 Ctrl+F5 组合键), 调试并运行程序, 如图 1.8 所示。

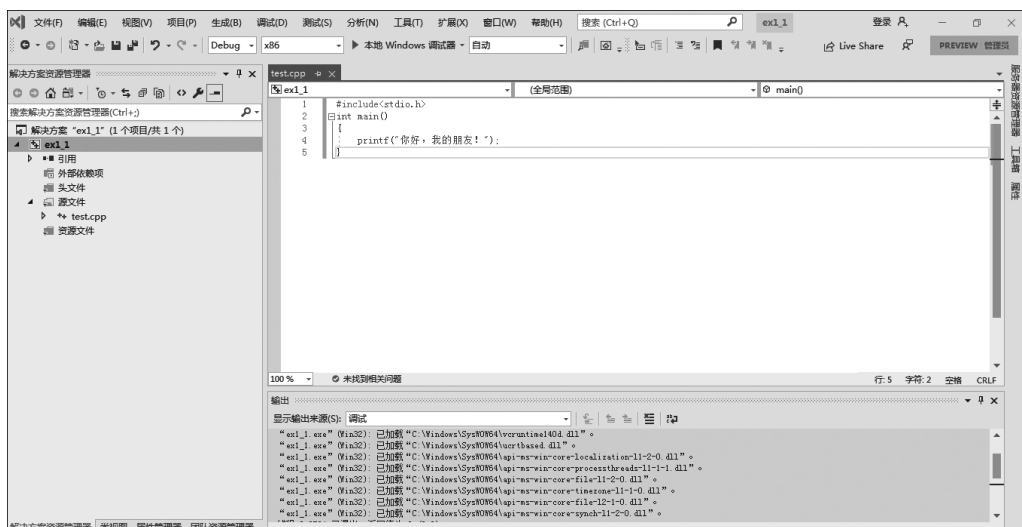


图 1.8 调试并运行程序

程序的运行结果如图 1.9 所示。

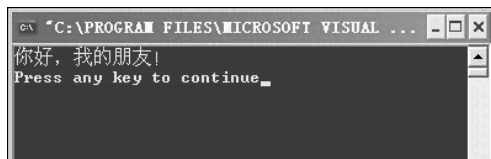


图 1.9 程序的运行结果

3. 程序错误的处理

在编写程序的时候, 程序可能会出现一些错误, 这些错误分为语法错误和逻辑错误。对于语法错误, 在对程序编译的时候, 系统会给出错误的描述、错误的位置和错误的个数, 如图 1.10 所示。