

## 第 1 章

# 程序设计与 C 语言

工欲善其事，必先利其器。

——《论语》

语言是思维的物质外壳。

——《辩证唯物主义常识》

### ◎ 章节导读

随着以计算机、物联网、云计算、人工智能、5G 通信、智能制造、工业互联网等技术为代表的第四次工业革命的蓬勃发展，计算机技术已经成为各行各业的基础生产力。在日常工作、学习过程中，都离不开计算机的辅助。了解程序设计的基本思想，理解计算思维的概念，并借助一种程序设计语言实现工程问题的求解是理工科学生解决工程问题的一种基本技能。

C 语言是目前广泛使用的程序设计语言之一，它既具备高级语言的特性，又具有直接操纵计算机硬件的能力，并以其丰富灵活的控制和数据结构、简洁而高效的语句表达、清晰的程序结构和良好的可移植性拥有大量的使用者。经验丰富的程序员可以用 C 语言编写出快速高效的代码，数以百万计的 C 程序代码充分证明了它的生命力。C 语言被广泛的应用于系统软件、应用软件、嵌入式开发、游戏开发等众多领域。掌握和使用 C 语言进行程序设计，可极大的提高学生利用计算机技术解决工程问题的能力。本章将帮助大家了解程序设计的概念，认识 C 程序设计语言并学会使用具体的编译软件实现单程序的开发。

### 章节目标

1. 理解程序、程序设计、算法的概念。
2. 理解 C 语言的特点、应用领域。
3. 理解 C 语言程序的基本组成和基本概念。
4. 能够借助 Dev-C++ 工具软件编写简单的 C 程序。

## 1.1 程序与程序设计语言

### 1.1.1 程序

“程序”的概念在计算机诞生之前就广而用之，它是指完成某项工作而以某种语言编制出来的动作序列或工作流程，是一种有效的管理方式，如“工作程序”、“二十大会议流程”等。良好的“程序”能优化工作流程，可以高效协调的推进工作或任务向预设的方向发展。细节决定成败，任何单位和个人，任何事情，“程序”都非常重要，在现实生活中，对于我们将要开展的相关工作，先设计好规范有效的“程序”，再按“程序”有计划有步骤地开展任务，势必事半功倍的完成任务。

计算机程序是为了让计算机完成某项特定的任务，按照计算机程序设计语言所要求的规范编制出来的一组工作指令，它表述了程序员要求计算机执行的具体操作流程。对计算机而言，程序就是一组指令，计算机按照指令的流程执行并求得结果。对编程人员而讲，计算机程序就是根据要解决的问题，按照具体的程序设计语言格式规范书写出来的指令集合。

根据问题的求解规模，程序可能简单也可能复杂。简单的程序只有几条指令，复杂的程序可能达到千万条指令的级别。对于复杂的程序，需要团队合作完成，对于程序员而讲，团队意识和协作精神是必备的职业素养。

程序设计是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具，给出这种语言下的代码实现。程序不等同于编程，编程只是程序设计中的一小部分。程序设计过程包括分析、设计、编码、测试、排错等不同阶段。受制于问题规模、客户需求、技术及程序员风格等多方面的因素制约，程序设计是一种在各种约束条件和相互矛盾的需求之间的一种平衡，因此任何程序没有完美，只有更优。

### 1.1.2 程序设计语言

自1946年第一台计算机ENIAC面世，到目前为止，计算机程序设计语言的发展经历了机器语言、汇编语言、高级语言三个阶段，每一个阶段的改变都极大的提升了程序设计的效率。

#### 1. 机器语言

机器语言是采用计算机指令格式并以二进制编码表达各种操作的语言。因为计算机只能识别0和1，因此机器语言就是0/1组成的代码。实际上，在计算机内部，无论是一部电影还是一首歌曲或是一张图片，最终保存的都是0/1代码，因为CPU只能执行0/1代码。

用机器语言编写的计算机能够直接理解和执行的程序，执行速度快，运算效率高，占用存储空间小。但是机器语言程序难读、难记、编程难度大、调试修改麻烦，而且不同型号的计算机具有不同的机器指令系统。

## 2. 汇编语言

汇编语言是一种符号语言，它用助记符来表达指令功能，如：

```
MOV AX BX
```

```
ADD CX DX
```

汇编语言和机器语言都是面向机器的语言，都是为特定的计算机系统而设计的。但汇编语言程序比机器语言程序易读、易写，并保持了机器语言执行速度快、占用存储空间小的优点。然而由于汇编语言的语句功能简单，程序的编写较复杂，且汇编程序不能被计算机直接识别和执行，需要由一种起翻译作用的程序（称为汇编程序），将其翻译成机器语言程序（称为目标程序），计算机才能执行，因此汇编语言程序的可移植性差、代码编写复杂。机器语言和汇编语言都被称为低级语言。

汇编语言开发的软件，执行效率非常高，但是开发效率非常低，所以在常见的应用程序开发中不会使用汇编语言，只有在对效率和实时性要求极高的关键模块才会考虑汇编语言，例如操作系统内核、驱动、仪器仪表、工业控制等。

## 3. 高级语言

人类最擅长的是制造工具并使用工具，在长期从事计算机编程的过程中，人们逐渐意识到，设计一种接近于自然语言或数学语言，同时编写出来的程序并不为机器型号所限制的语言是促进计算机普及应用的根本。经过历代计算机科学家的努力，1956年，第一个完全脱离机器硬件的高级语言—FORTRAN问世了，70多年来，共有几百种高级语言出现，有重要意义的有几十种，影响较大、使用较普遍的有FORTRAN、ALGOL、COBOL、BASIC、LISP、NOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、VC、VB、Delphi、JAVA、Python等。

高级语言（High-Level Language）是一种相对于汇编语言和机器语言而言更加抽象化的编程语言，它使用类似于英语的语法和结构，使得程序员可以更容易地理解和编写程序。高级语言通常比汇编语言和机器语言更加易于阅读、编写和维护，适合用来编写大型复杂的应用程序。高级语言的编译器或解释器可以将高级语言代码转化为机器语言或汇编语言，然后由计算机执行。常见的高级语言包括C语言、Java、Python、Ruby、PHP等。不同的高级语言有不同的特点和适用场合。例如，C语言是一种面向系统编程的语言，适合编写操作系统、驱动程序、PC端软件、Linux应用开发、游戏开发、单片机和嵌入式系统等软件；Java是一种面向对象编程的语言，适合编写网络应用、企业级软件等，近年来又持续发力于大数据领域；Python是一种脚本语言，适合编写科学计算、数据分析、人工智能、Web应用等，近年来势头强劲，业务增长迅猛；Objective-C是一种专门用于苹果产品的软件开发工具。

高级语言的主要优点是代码可读性高、编写效率高、维护成本低。相对于汇编语言和

机器语言而言，高级语言的抽象化程度更高，使得程序员可以更加关注问题的本质而不是底层的实现细节。每一种高级语言都会提供大量的库和框架，这些工具可以帮助程序员快速开发部署应用程序。缺点是执行效率相对较低，不适合对程序执行效率要求特别高的应用场景。

高级语言的下一个发展目标是智能化应用，也就是说：只需要告诉某种人工智能编译软件，数据的输入记录是什么，要完成的数据加工以及输出形式，系统就能自动生成算法，自动进行数据处理，并将结果呈现给用户，它是一种非过程化的程序语言。

## 1.2 结构化程序设计与算法

### 1.2.1 结构化程序设计的概念

在实际应用中，随着问题规模和复杂性的增加程序会越来越臃肿，如果不按一定的规则或流程去编写程序，会造成程序的可读性差、后期维护成本大、软件质量无法保障。因此需要一整套规范来指导程序设计，于是 E. W. Dijkstra 在 1965 年提出结构化程序设计的概念。其基本原则是“自顶向下、逐步细化、模块化设计、结构化编码”。

(1) 自顶向下：程序设计时，应先考虑整体，再考虑细节，先考虑全局目标，再考虑局部目标。

(2) 逐步细化：对复杂问题，应设计一些子目标进行过渡，逐步细化。

(3) 模块化设计：一个复杂的问题往往是由若干个简单的问题构成的。模块化就是把程序要解决的总目标分解为规模小的分目标，再进一步分解为具体的小目标，把每个小目标称为一个模块。

(4) 限制使用 GOTO 语句：程序中大量使用 GOTO 语句会导致程序结构混乱，结构化程序一般要尽量避免使用 GOTO 语句进行跳转。

### 1.2.2 结构化程序的基本结构与特点

1996 年，Boehm 与 Jacopini 证明了程序设计语言仅仅使用顺序结构、选择结构和循环结构，就足以表达各种其他结构的程序设计方法。它们的共同特征是：严格地只有一个入口和一个出口。采用结构化程序设计方法编写程序，可使程序结构易读、易理解、易维护。基本上，用下面三种基本结构就可以实现结构化程序设计。

#### 1. 顺序结构

顺序结构是指按照程序语句的先后顺序，从始至终，一条语句一条语句地按先后次序执行，它是最简单也是最常用的基本结构。顺序结构如图 1-1 所示。

#### 2. 选择结构

选择结构又称分支结构，它包括简单选择和多分支选择。这种结构可以根据给定条

件，判断执行哪一个分支中的语句，每一个分支都有机会被执行到，但对于一次具体的执行，只能执行其中一个分支语句，不可能存在同时执行两个分支的语句的情况。选择结构如图 1-2 所示。

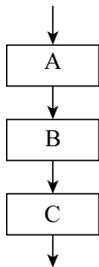


图 1-1 顺序结构

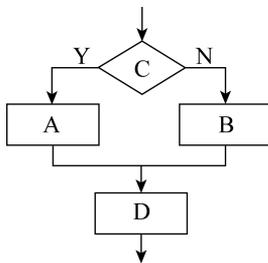


图 1-2 选择结构

### 3. 循环结构

循环结构又称重复结构，它根据给定的条件判断是否重复执行某一段相同的运算（循环体）。利用循环结构可以大大简化程序的语句，循环结构分为当型和直到型两种，分别如图 1-3 和图 1-4 所示。

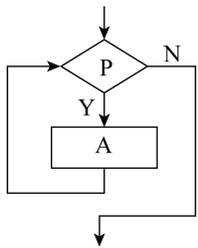


图 1-3 当型结构

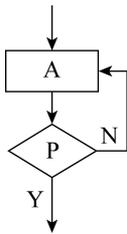


图 1-4 直到型循环

遵循结构化程序设计的原则，按结构化程序设计的方法设计出来的程序易于理解，用户使用起来方便，后期维护成本低；结构化程序设计极大地提高了软件开发的效率，降低了软件开发成本。

## 1.2.3 算法的概念及其特性

计算思维是运用计算机科学的基础概念去求解问题、设计系统和理解人类的行为。它包括了涵盖计算机科学之广度的一系列思维活动。用计算思维实现问题求解，需要经过以下几个步骤：

(1) 对问题进行抽象与映射，将客观世界的实际问题映射成计算空间的计算求解问题，建立解决问题的数学模型。当有多个数学模型可用时，需要对模型进行分析、归纳、假设等优化，选择最优模型。

(2) 将建立的数学模型转换成计算机所理解的算法和语言，也就是将数学模型映射或

分解成计算机所理解和执行的计算步骤。

(3) 编程验证，就是将所设计的算法翻译成计算机能理解的指令，即用某一种计算机语言描述算法（这就是计算程序）。然后通过上机实践，完成问题求解。

在这个过程中，编程人员始终以问题的抽象、问题的映射、问题求解算法设计等为主线展开讨论，编写程序只不过是使用一种计算机语言去实施问题求解，而对数学模型进行转换所得到的算法才是计算机问题求解的灵魂。所谓算法（Algorithm），就是一组明确的、有序的、可以执行的步骤集合。算法的概念要求步骤集是有序的，这样就要求算法中的各个步骤必须拥有定义完好的、顺序执行的结构。

算法应具有 5 个特性：有穷性、确定性、有零个或多个输入、有一个或多个输出，可行性。

有穷性：是指一个算法必须保证执行有限步骤之后结束，且有时间限制；

确定性：是指算法的每一步骤必须有确切的定义，算法步骤必须没有二义性，不能产生理解偏差；

输入/输出：有零个或多个输入是指描述运算对象的初始情况，其中零个输入是指算法本身给出了初始条件；有一个或多个输出是指算法必须要有结果，没有输出的算法是毫无意义的；

可行性：算法的每一步都必须可行的，每一步都通过执行有限次数完成。原则上能够有效运行，得到确定的结果。

算法主要分为两类：数值运算算法和非数值运算算法。数值运算算法主要用于解决数值求解问题，例如求方程的根、求函数的定积分、求最大公约数等。非数值运算算法主要用于解决需要用逻辑推理才能解决的问题，常见的是用于事务管理领域，例如图书检索、人事管理、行车调度管理、人机围棋大战、智能检索与呈现、定制服务等。

#### 1.2.4 算法的表示方法

算法的表示方法主要有：自然语言、流程图、伪代码、PAD 图等。本书所有的算法均使用自然语言或流程图表示。

##### 1. 用自然语言描述算法

自然语言是指人们在日常生活中的语言，它可以是汉语、英语或其它语言。使用自然语言对算法进行描述，人们容易理解算法的思路和步骤。此种表述方式简单，易于交流和沟通，但是由于自然语言描述容易存在歧义和不确定性，造成不同的人阅读算法有不同的理解。因此，除了一些简单的算法外，一般不建议使用自然语言描述算法。

##### ► 案例 1-1 用自然语言描述求解 5 的阶乘

Step1: 令  $p=1$ ;

Step2: 令  $i=1$ ;

- Step3: 使  $p$  和  $i$  相乘, 将结果放在  $p$  中;
- Step4: 使  $i$  的值加 1;
- Step5: 若  $i$  的值不大于 5, 返回重新执行 Step3、Step4、Step5; 若  $i$  的值大于 5, 执行 Step6;
- Step6: 在输出计算结果  $p$ 。

## 2. 用流程图描述算法

流程图是算法的图形化表示, 也称为程序流程图。它采用不同的图元形状来描述程序的走向。这种方法灵活、自由、形象、直观, 可表示任何算法。常用的流程图制作软件有微软公司出品的 Vision, 基于流程图的可视化编程工具 Raptor, 编程爱好者也可使用 Word 或 WPS 自带的图形工具进行流程图制作。

流程图采用美国国家标准化协会 (American National Standards Institute, ANSI) 规定的一组几何图形来描述算法, 在图形上使用简洁的文字和符号表示不同的操作, 用流程线表示算法步骤的执行方向。常用流程图符号如图 1-5 所示。图中各符号的含义如下。

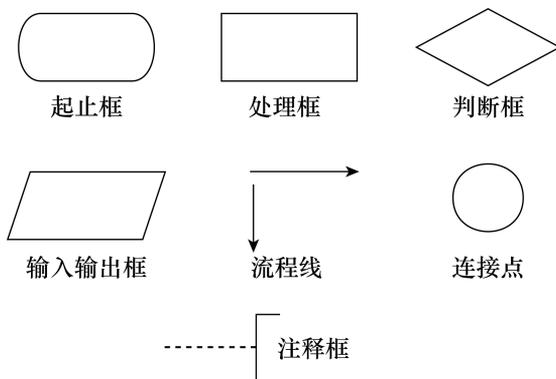


图 1-5 常用流程图符号示例

- (1) 起止框: 表示算法的开始和结束。
- (2) 处理框: 表示某种处理操作, 比如给变量赋值、两个数据的相加操作等。
- (3) 判断框: 对一个给定的条件进行判断, 依据判定结果选择下一步的操作。判断框有一个入口, 两个出口。两个出口分别对应条件成立和不成立时的执行方向。
- (4) 输入输出框: 表示输入输出操作。
- (5) 流程线: 用于连接各图框, 表示算法的执行顺序。
- (6) 连接点: 将画在不同地方的流程线连接起来。
- (7) 注释框: 用于对流程图中某些框的操作进行说明。

### ► 案例 1-2 用流程图描述求解 5 的阶乘

求解  $5!$  本质上是两个数的乘法问题。因此, 需要设计一个变量  $p$  保存初值 1, 另一个变量  $i$  既可作为操作数递增变量, 同时也可作为循环控制变量使用。变量  $i$  与  $p$  每做一

次乘法运算后，变量  $i$  便以步长为 1 进行递增，继续循环计算  $i$  与  $p$  的乘法，直至  $i$  增长超过 5 后运算结束。算法所体现的思维方式就是通过将从 1~5 这 5 个数的连乘运算分解为两个数的循环乘法运算，实现了将复杂问题分解为多个具有相同运算结构的简单问题。具体的流程图见图 1-6。

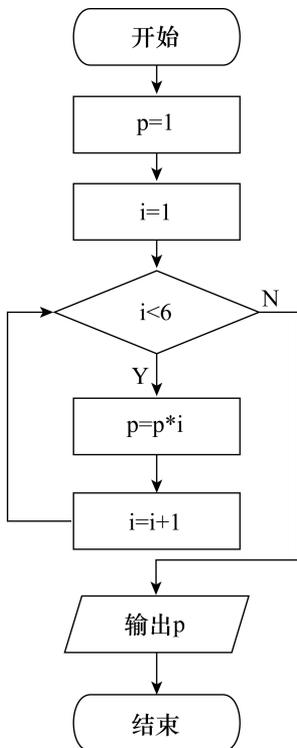


图 1-6 用流程图表示求解  $5!$  算法

## 1.3 C 语言的发展历史及其特点

### 1.3.1 C 语言的发展历史

作为非常流行的高级程序设计语言，C 语言的发展颇具戏剧性，经历了 A-B-C 三个阶段，是诸多科学家的集体智慧，是 UNIX 操作系统的副产品。

C 语言的原型是 ALGOL 60，业界称之为 A 语言。1963 年，剑桥大学将 ALGOL 60 发展成为 CPL 语言，即组合编程语言。1967 年，剑桥大学的马丁·理查兹 (Martin Richards) 对 CPL 进行了简单，开发出了 BCPL (Basic Combined Programming Language)，即基本组合编程语言。

1969 年，贝尔实验室的肯尼思·汤普森 (Ken Thompson) 使用 BCPL 语言在 PDP-7

计算机上实现了 UNIX 的最初版本，UNIX 是一套多用户、多任务、多层次的操作系统。同时对 BCPL 语言进行了修改，为它取了一个非常有趣的名字“B”语言，意思是将 CPL 语言煮干，提炼出它的精华。B 语言是一种无类型、可直接对机器字操作的语言。

1972 年到 1973 年间，贝尔实验室的丹尼思·里奇（Dennis Ritchie）在 B 语言的基础上增加了类型（Data type）和结构（Structure）的概念，设计出了 C 语言。C 语言一经问世，便因强大的功能和优越性能而迅速普及。

1973 年，Ken. Thompson 和 D. M. Ritch 合作把 UNIX 操作系统的 90% 以上的部分用 C 语言改写，发表了 UNIX 版本 5，至此，UNIX 操作系统的修改、移植都相当顺利，UNIX 和 C 的完美结合，为 UNIX 日后的应用普及打下了坚实的基础。后来二者又反复多次对 C 语言作了改进，并将其移植到 IBM、VAX 等多种计算机上。1978 年，布莱恩·克尼汉（Brian Kernighan）和丹尼思·里奇（Dennis Ritchie）以 UNIX 版本 7 中的 C 语言编译程序为基础，出版了《The C Programming Language》一书，从而使 C 语言成为目前广泛流行的程序设计语言。

此后。又有多种语言在 C 语言的基础上产生，如 C++、Java、C# 和 Python 等，如 Python、PHP 官方版本是使用 C 语言实现的，不少语法类似 C 语言的语法，虽然平时做项目的时候看不到 C 语言的影子，但是如果深入想学习 PHP 和 Python，那就要有 C 语言基础了。因此，学好 C 语言，再学习和掌握其他高级程序设计语言会事半功倍。

1983 年美国国家标准学会（American National Standards Institute, ANSI）对 C 语言进行了标准化，推出了第一个 C 语言标准草案，通常称之为 83 ANSI C（简称 C83）。1989 年，ANSI 发布了完整的 C 语言标准 89 ANSI C（简称 C89）。1990 年，国际标准化组织（International Organization for Standardization, ISO）采纳了 89 ANSI C，并以国际标准发布，即 ISO C 标准（ISO 9899-1990），也即是今天人们常说的 ANSI C。1999 年，ISO 对 C 标准做了全面修订，形成了 C 标准 ISO/IEC9899: 1999，简称 C99。目前，C 语言最新标准是 2018 年发布的 ISO/IEC 9899: 2018，简称 C18。C18 规范了 C 语言的语法、语义和实现，并对 C 程序的行为、编译、解释和执行等方面进行了全方面的讲解，为 C 语言程序员和开发者提供了一个标准化的工作平台。

目前各厂家提供的 C 编译器有 Microsoft Visual C++、Turbo C、Borland C 等，都未能实现 C99 以后标准所建议的全部功能，因此本书采用 ANSIC 标准来叙述。

## ► 思政点滴

为了适应国家安全和航空航天空间软件产业规范化需要，国防科学技术工业委员会在 2004 年 4 月发布了 GJB5369-2005，即航天型号软件 C 语言案例子集。该子集规定了 C 语言软件的编程准则，准则分为推荐和强制两种类型，推荐类为参照执行标准，强制类为必须执行标准，标准适用于航天型号 C 语言软件。2013 年 7 月，中国人民解放军总装备部、航天科工集团、船舶重工集团、航天科技集团、航空工业集团共同起草了 GJB8114-2013，即 C/C++ 语言编程安全子集，提出软件编程标准，以提高国家军用软件的安全性，并作为静态规则检查的依据。新标准规定了 C/C++ 语法严于语法规定的安全子集，用于编写军用

安全关键软件的指导意见，分为软件编程中强制要求遵循的编码规则或推荐参照编码规则。近几年，国家更是接连出台了相关军用软件开发的技术规范，此类规范的提出，对于提升我国军用软件的质量，保障国家安全提供了坚定的技术保障。

### 1.3.2 C语言的特点

早期的C语言主要用于UNIX操作系统的开发，C标准确立后，C语言被广泛应用于各个领域，成为至今为止最优秀的程序设计语言之一，其主要的优缺点有以下几个方面。

#### 1. C语言的主要优点有：

(1) 语言简练、紧凑，使用方便、灵活。标准C语言一共有32个关键字、9种控制语句，程序编写形式自由，支持大小写敏感。C语言的这些特点使得编程者的个性易于发挥，能够写出更好的程序。

(2) 运算符丰富。C语言共有34种运算符，括号、赋值、强制类型转换等都以运算符的形式出现，从而使得C语言的表现能力和处理能力极强，也使C语言比其他语言更易实现算法。

(3) 数据结构丰富。C语言的数据类型有基本数据类型和构造数据类型，能够方便实现各种复杂的数据结构。

(4) C语言是一种结构化语言。用函数作为程序的基本单位，具有易于实现、层次清晰，便于按模块化方式组织程序的控制流语句（if~else语句、switch语句、while语句、for语句等）。

(5) C语言语法灵活、限制不会十分严格。例如，整型数据、字符型数据和逻辑型据可以通用，对数组下标不作越界检查等，让程序员拥有更大的发挥空间。但另一方面语法检查也容易引起语法错误。

(6) C语言可以直接访问内存的物理地址，进行位（bit）一级的操作。由于C语言现了对硬件的编程操作，因此C语言集合了高级语言和低级语言的功能，既可用于系统的开发，也适合于应用软件的开发。因此，C语言被广泛地移植到了各类单片机上，在嵌入式程序开发中拥有一席之地。

(7) C语言还具有效率高、可移植性强等特点，仅需要极少的修改就能用于不同型号的计算机和操作系统。

#### 2. C语言的主要缺点有

(1) 程序更容易隐藏错误，与汇编语言相似，很多错误要运行时才能检测出来。

(2) 其灵活性带来的副作用是C程序可能会难以被理解，所以编写C程序的时候最好遵循现代程序设计规范，养成写好注释和程序文档的好习惯，以便于后期较为方便容易的维护程序。

(3) 用C语言编写大型程序会难以修改，它缺乏现代编程语言提供的“类”“包”之类的机制。

总之 C 语言既有高级语言的特点，又具有汇编语言的特点；既是一个成功的系统设计语言，又是一个较好的应用程序设计语言；既能用来编写不依赖计算机硬件的应用程序，又能用来编写各种系统程序。是一种受欢迎、应用广泛的程序设计语言。

### 1.3.3 C 语言的应用领域

C 语言的特点决定了它的应用领域比较广泛，通常 C 语言常被应用在以下领域。

#### 1. 操作系统

C 语言可以开发操作系统，例如个人桌面领域的 Windows 操作系统内核、服务器领域的 Linux 操作系统内核、FreeBSD、苹果公司研发的 MacOS 等。

#### 2. 应用软件

C 语言可以开发应用软件。在企业数据管理中，需要可靠的软件来处理有价值的数据库。由于 C 语言具有高速、稳定等特性，企业数据管理中使用的数据库（如 Oracle、MySQL、MS SQL Server 和 SQLite 等）都是使用 C 语言开发的。此外，金山办公软件 WPS、微软的 Office 办公软件、功能强大的数学软件 MATLAB 等都是使用 C 语言开发的。

#### 3. 嵌入式开发

使用 C 语言可以进行嵌入式底层开发。当今生活的各个方面都在趋向智能化，智慧农业、智慧交通、智慧城市、智慧家庭等已不再是设想。例如，智能手环、工业智能机器人、汽车智慧驾驶系统等都是智能化的体现。这些智能系统的组成部分，如底层的微处理器控制的传感器、蓝牙、Wi-Fi 网络传输模块，上层的半导体芯片的驱动库、嵌入式实时操作系统 FreeRTOS 和 UCOS 等，大部分都是用 C 语言开发的。

#### 4. 游戏开发

因为 C 语言具有强大的图像处理能力、可移植性、高效等特点，所以在一些大型的游戏开发中，游戏环境渲染、图像处理等都使用 C 语言来实现。除此之外，成熟的跨平台游戏库 OpenGL、SDL 等也是用 C 语言开发的。

## 1.4 C 语言程序的结构及 2 个实例

C 语言是结构化程序设计语言，它由三种基本结构，即顺序结构、选择结构、循环结构组成。C 语言程序的基本组成单位是函数，函数与函数之间相互独立，共同配合完成一项任务。C 语言规定源程序有且必须只有一个主函数（main() 函数）。用 C 语言编写的程序是源程序，需经过编译和链接才能被计算机识别和执行。下面通过两个具体的案例来说明 C 程序的基本组成。

## 1.4.1 实训任务 1 输出指定的图案

### ► 案例 1-3 问题描述

程序中根据业务流程的需要打印各种各样的图案，如有以下图案：

```
*****
```

```
欢迎进入 C 语言的精彩世界！
```

```
为实现科技强国、助力民族复兴而努力学习编程！
```

```
*****
```

#### 【问题分析与实现】

任务是由两行星花和两行标语组成，对于两行星花，可以写一个函数，由其实现打印出来指定个数的"\*"；对于两行标语，可通过系统输出函数 printf() 实现。

```
#include<stdio.h>
#define ST1  "欢迎进入 C 语言的精彩世界!\n"
#define ST2  "为实现科技强国、助力民族复兴而努力学习编程!\n"
#define N 60
void printstar()
{
    int i;
    for(i=0;i<=N;i++) //循环打印 N 个星花,使用符号常量便于后期修改星号个数
        printf("*");
    printf("\n");      //换行
}
int main()
{
    printstar();      /* 打印第一行星号 */

    printf(ST1);     /* 欢迎进入 C 语言的精彩世界! */
    Printf(ST2);     /* 为实现科技强国、助力民族复兴而努力学习编程! */
    Printstar();     /* 打印第二行星号 */
    return 0;
}
```

#### 程序解析：

##### 1. 函数结构

C 语言程序是由一个或多个函数所组成的，函数的结构为函数首部+函数体，函数体

以 “}” 标识开始，以 “{” 标识结束。例如：

```
函数类型 函数名(参数类型参数名) //函数的首部
{ //函数体,包括两部分,声明部分及可执行部分。
    声明部分
    运行部分
}
```

C 程序有且仅有一个 main 函数，其在源程序的位置任意。主函数以外的其他函数可以是系统提供的库函数，也可以是用户自定义的函数。如本程序中使用了用户自定义函数 printstar() 及输出库函数 printf()。其中库函数 printf() 的功能是把要输出的内容输送到显示器上；用户自定义函数 printstar() 负责输出指定的星花。

C 程序的运行是从主函数开始的。当程序运行之初，系统首先查找到 main 函数，从 main 函数开始的大括号进入函数体，并根据程序的语句按序依次运行，直到遇到 main 函数最后的大括号结束。主函数是整个程序的控制部分，当主函数运行结束，整个 C 程序的运行也结束了。主函数类似于整个程序的管理者，程序由它开始至它结束。

## 2. 程序语句

C 语言程序的每一个函数是由若干条语句组成的，语句必须以分号结束。一条语句可以写在多行上，一行也可以写多条语句。但为了保障程序的可读性，建议编程时一行写一条语句。

## 3. 预编译命令

在 C 语言中，以 # 开头的命令是预处理命令，不是一条语句，所以其后面不能加分号。本示例代码段中有两个预处理命令，即 #define N 60，用于定义符号常量 N；#include <stdio.h>，用以包含头文件，里面有将要使用到的输入输出函数的原型定义等。

预编译命令是整个编译过程开始之前进行的工作。如本例中 “#include” 是预编译命令，用于控制 C 语言编译器在进行编译、链接操作过程中的行为。其含义是，在编译时将一个包含文件的内容添加到当前程序中。包含文件由 #include 命令后面的内容所指定，它是一个独立的磁盘文件，该文件包含了可被程序或编译器使用的信息，最常用的是扩展名为 “.h” 的头文件。C 语言提供了大量的库函数，编程时可根据具体的需求，将相应的文件包含到当前源文件中来。

## 4. 变量定义

就像每个人都有自己的名字和身份证信息一样，在 C 程序中为了区别每个变量、函数、构造数据类型，都须为它们取不同的名字。C 语言规定，使用变量之前必须首先定义变量。变量定义就是将变量的名称以及变量要存储的信息类型告知编译器。如当前程序中的子函数 printstar() 中定义了一个变量 i，其类型是整型，初值是 0，在程序中的作用是作为循环变量控制输出星花的个数。

变量或符号常量是编程时根据问题的需要所定义和使用的用户自定义标识符，在定义时需要遵循标识符的命名规则。C 语法规则规定，标识符只能由字母、数字、下划线组成，且

第一个字母必须是字母或者下划线，同时标识符区分大小写字母，如 If 和 if 是两个不相同的标识符。

另外标识符不能与程序中具有特殊意义的关键字相同，不能和用户所编写的函数名、C 语言系统库函数相同，在程序中标识符尽量不要重复，尽量能让用户看到名字之后就可以知道它的含义。

## 5. C 程序的结构

C 程序是结构化程序，一般由三个基本结构组成，即顺序结构、选择结构及循环结构。本例中的 printstar() 使用了循环结构，主函数是典型的顺序结构。任何程序，无论复杂或简单，均由三种基本结构组成。编程时需要分析问题，确立程序所选用的结构，然后再进行编码。

## 6. 程序注释

C 程序中的注释就如文言文中的注释一样，使用注释能增强程序的可读性，便于程序的阅读者理解程序的功能。但无注释并不影响程序的运行，因为程序在编译时自动忽略掉注释。程序员要养成良好的使用注释进行编程的职业习惯。常用的程序注释方式有以下两种。

(1) 单行注释：以“//”作为引导符，只能在一行内进行注释。

(2) 多行注释：所有的注释内容都写在“/\*”和“\*/”之间，其中可以换行。

## 1.4.2 实训任务 2 输出三个者中的最大值

### ► 案例 1-4 问题描述

高中生小明对计算机编程比较感兴趣，报名参加了学校组织的软件开发兴趣班。在编程课上指导教师要求小明从键盘上输入三个整数，比较其大小后，将最大的整数输出。

#### 【问题分析与实现】

生活中两个数进行比较大小比较容易达到，现在要求三个数进行比较，可以先将两个数中的最大数找出来，然后与第三个数进行比较从而得出最大值。因此解决本问题的步骤如下：

Step1：从键盘上输入三个整数。

Step2：找出两个数当中的最大值。

Step3：将两个数当中的最大值与第三个数进行比较，得出三个数中的最大值。

Step4：输出结果。

```
#include<stdio.h>
int main()
{
    int a,b,c,max;           //定义三个变量,用来存储输入的数据
    scanf("%d%d%d",&a,&b,&c); //格式化输入数据
```

```

max=a>b? a:b;      //条件运算符,将两个数中的较大数赋值给 max
max=max>c? max:c;
printf("a=%d,b=%d,c=%d,三个数中的最大值为%d\n",a,b,c,max);
return 0;
}

```

## 程序解析:

### 1. 格式化输入输出函数

C语言没有输入输出语句，其输入输出是通过格式化控制库函数 `printf()` 及 `scanf()` 实现的。两者都是库函数，在 `stdio.h` 头文件中，使用时需要将其用 `include` 命令包含进来。两个函数有固定的格式控制参数，具体可参阅第二章第七小节。

本例中，`scanf()` 通过 `%d` 控制输入的三个数是整数，并将其赋值给变量 `a`，`b`，`c`。数据输入时需以空格、回车或 `Tab` 作为数据的间隔。`printf()` 格式控制串有普通字符和格式控制字符，对于普通字符原样输出，格式控制字符以相应的输出项按指定的格式替换。

### 2. 赋值表达式

C语言中的数值计算要通过赋值表达式来实现。在C语言中，`=`称为赋值符号，将等号右边的运算结果赋值给左边的变量，如案例2中的 `max=a>b? a: b;` 是将条件运算的结果赋值给变量 `max`。C语言有丰富的运算符和表达式，本书将在第二章进行详细的阐述。

### 3. 条件运算符

在语句 `max=a>b? a: b;` 中，`?:` 是C语言中的条件运算，它是C语言唯一的一个三目运算符，表达式 `max=a>b? a: b` 的含义是，如果 `a` 比 `b` 大，则将 `a` 的值赋值给 `max`，否则将 `b` 的值赋值给 `max`。

从以上两个案例中可以看出，一个简单的C语言程序的基本组成包含：注释、预处理命令、主函数、其他函数。稍为复杂的源程序是由多个函数组成，每一个函数可完成相对独立的功能，函数与函数之间是调用关系，每一个函数都由函数的首部和函数体组成。任何一个源程序，都有且仅有一个 `main()`，程序从其执行到其结束。

简单的C程序编写起来不难，读者一开始要养成良好的程序编码习惯，比如在编写C程序时，应注意编程风格，采用逐层缩进的形式进行编码；保持一行仅写一条语句良好的风格；进行适当的注释；采用统一且规范的命名规则等。

## 1.5 C程序的编译机制

C语言是一种编译型的高级语言，程序员在编程时仅需关注程序逻辑本身，而源程序编译成可执行的机器语言由相应的编译程序执行。大多数编译程序都提供了集成的开发环境，在这样的一个环境中，程序员可以把写完的源代码经过相应的菜单操作生成可执行的

计算机程序。对于编程者本身而言，无需过多的关注源程序到目标程序的过程。但了解 C 程序的编译机制，可以帮助读者更好的进行程序设计和调试，从而能编写出更高质量的程序。C 语言程序从源代码到可执行代码需要一般要经过编辑、预处理、编译、链接、运行 5 个步骤。

### 1. 编辑 (Edit)

程序的编辑过程就是代码书写的过程。为了实现计算机执行编程者期望任务，利用编辑软件，将源程序逐一输入到计算机中，形成一个源程序文件的过程。这期间必须严格遵守 C 语言的语法规则，要特别注意编辑程序是否添加了额外的格式字符或特殊字符，如全角符号、中文标点符号等。理论上可以使用各种各样的文本编辑器进行代码书写，如记事本、Vim、Word、WPS 等文本编辑器，但为了更好地提高代码书写的调试的效率，建议使用集成开发环境，市场上的集成开发环境很多，个别集成开发环境对硬件环境有所要求。建议读者使用 Dev-C++、Code::Blocks、Microsoft Visual Studio 等。

### 2. 预处理 (Pretreatment)

预处理主要处理代码中以“#”开头的预处理语句。预处理完成后会生成 .i 文件。预处理操作具体包括以下几项。展开所有宏（如#define），将宏替换为对应定义的值；处理所有条件编译指令，例如#ifdef、#ifndef、#endif；处理文件包含语句（如#include），将包含的文件直接插入语句所在的位置；删除所有注释，添加行号和文件标识，以便在调试和编译出错时快速定位到错误所在行。

### 3. 编译 (Compile)

将预处理形成的 .i 文件作为编译程序的输入，进行编译。编译程序会自动分析、检查源程序文件的语法错误，并按两类错误类型（Warning、Error）报告出错行和原因。用户根据报告信息修改源程序，再编译，直到程序正确后，输出中间目标程序文件 .obj。编译过程相对复杂，主要进行词法分析、语法分析、语义分析、优化处理等。

### 4. 链接 (Link)

使用链接程序，将上一步形成的中间目标文件与所指定的库文件和其他中间目标文件进行链接。这期间可能出现缺少库文件或库文件不在指定路径等链接错误，同样链接程序会报告出错误信息。用户根据错误报告信息再修改源程序，再编译，再链接，直到程序正确无误后输出可执行文件。

### 5. 运行 (Run)

上述情况完成并确认无误后，就可以运行可执行文件，得到运行结果。当然也可能由于解决问题的算法而使源程序编写具有逻辑错误，得到错误的运行结果；或者由于语义上的错误，例如用 0 做除数，出现运行时错误（Division by zero）。这就需要检查算法，重新从编辑源程序开始，直到运行结果正确。如何保证结果的正确性？需要设计出测试计划和用例，进行全面、细致而艰苦的测试工作。

## 6. C 语言集成开发环境之 Dev-C++

使用高级语言编程时，通常需要使用一个集成开发环境 IDE（Integrated Developing

Environment，简称 IDE）来进行编辑、编译、运行和调试工作。在学习 C 语言的过程中，选择一个适合自己的编辑器非常重要。C 语言的集成开发环境有许多种，最常用的包括：Code::Blocks、Eclipse、CLion、Visual Studio Code、NetBeans、Codelite、Dev-C++等。

Dev-C++是一个 Windows 环境下的适合于初学者使用的轻量级 C/C++集成开发环境（IDE）。它是一款自由软件，遵守 GPL 许可协议分发源代码，集合了 MinGW 中的 GCC 编译器、GDB 调试器和 AStyle 格式整理器等众多自由软件。

Dev-C++使用 MinGW/GCC 编译器，遵循 C/C++标准。开发环境包括多页面窗口、工程编辑器以及调试器等，在工程编辑器中集合了编辑器、编译器、连接程序和执行程序，提供高亮度语法显示，帮助开发者尽量减少编辑错误，拥有完善的调试功能，可满足初学者与编程高手的不同需求，特别适合于非商业级普通开发者使用，是学习 C 语言和 C++的首选开发工具。在后面的章节中，如没有特别的说明，所有的程序都是基于 Dev-C++环境下的调试结果。

Dev-C++开发环境下载官网：<https://sourceforge.net/projects/orwelldevcpp/>。下载之后可以直接安装，在安装过程中按照对话框的操作提示执行即可。在 Windows 中安装完毕之后，在桌面上会出现如图 1-7 所示的图标。双击 Windows 桌面上的 Dev-C++图标，可以启动 Dev-C++。启动之后的编辑界面显示如下：

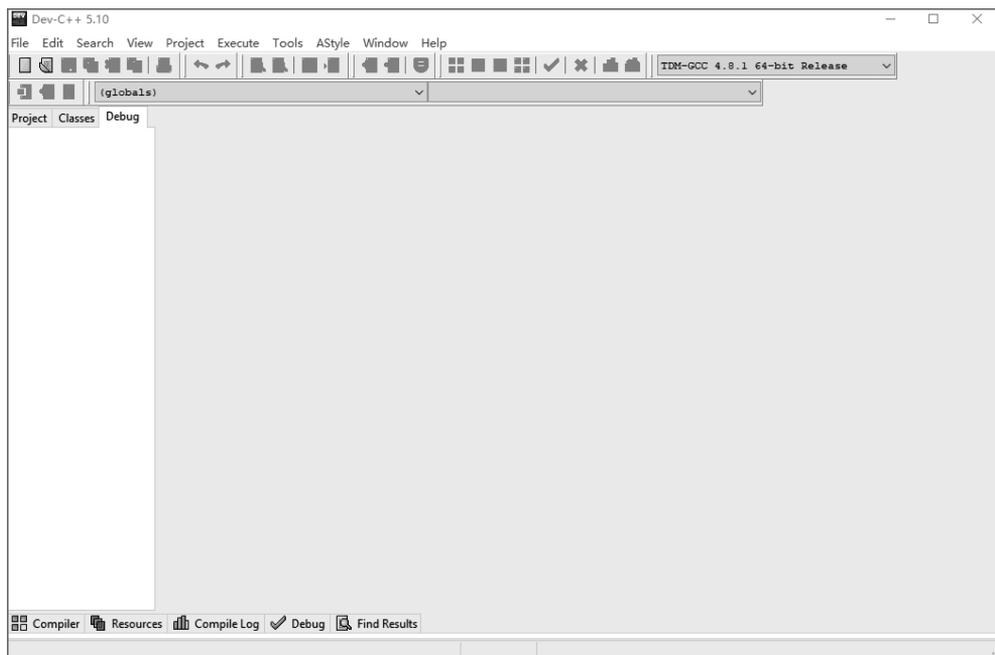


图 1-7 Dev-C++启动界面

Dev-C++初始界面为英文界面，读者可单击菜单栏中“Tools”的“Environment Options”命令项，在 General 选项卡中，在 Languages 下拉列表中选择“简体中文/Chinese”。参见图 1-8 操作：

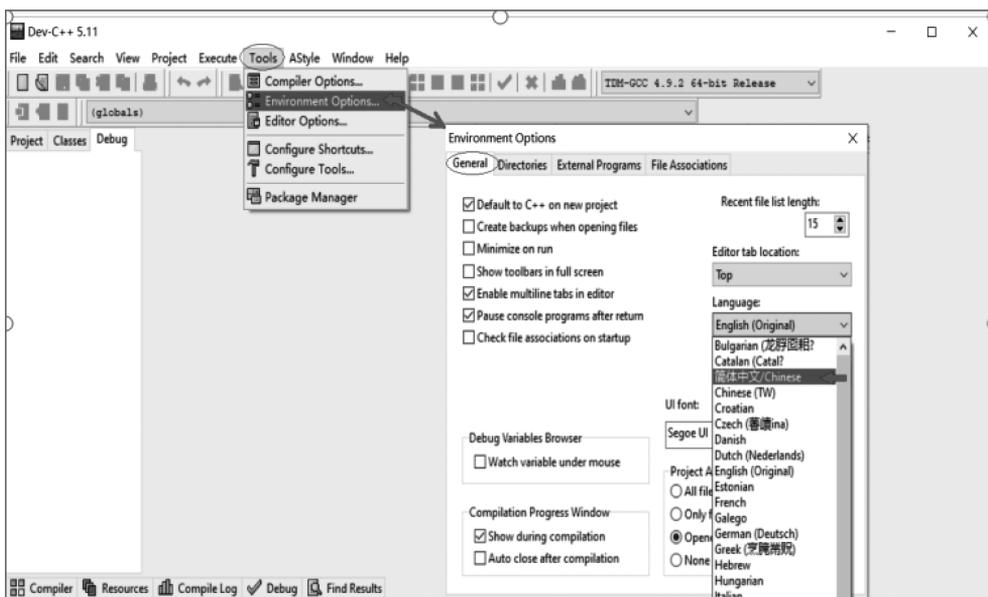


图 1-8 Dev-C++编辑环境设定

(1) 新建源文件：打开 Dev-C++，在上方菜单栏中选择“文件>新建>源代码”，或者通过按下 Ctrl+N 组合键，新建一个源文件，在编辑窗口中输入源代码并做初步检查。

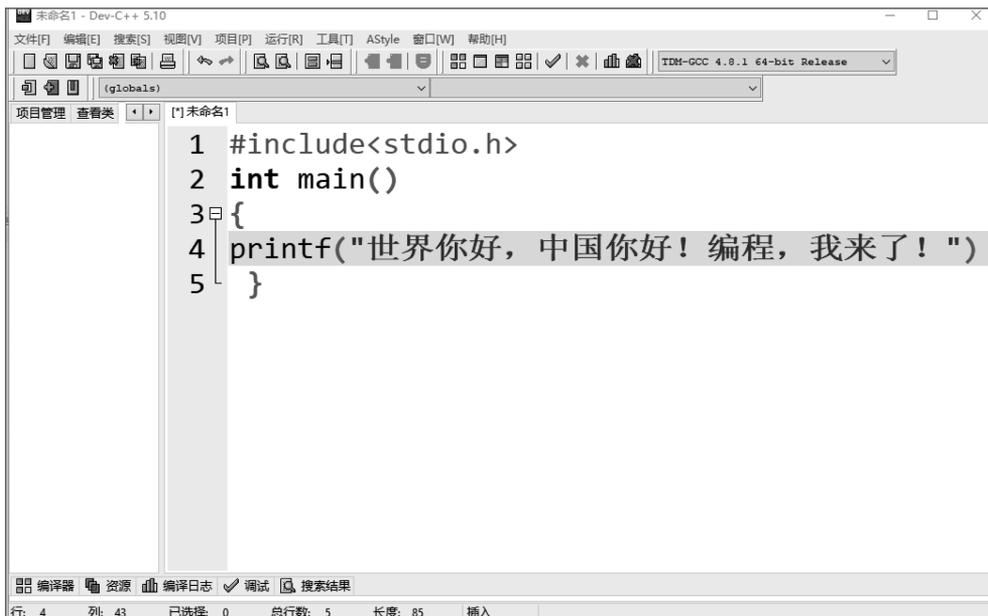


图 1-9 Dev-C++编辑源程序

(2) 生成可执行程序：使用菜单“运行>编译”或者按下 F9 键，则将 Myfirst.cpp 源文件的编译成 Myfirst.exe 文件。如果代码没有错误，会在下方的“编译日志”窗口中看到编译成功的提示，如图 1-10 所示。

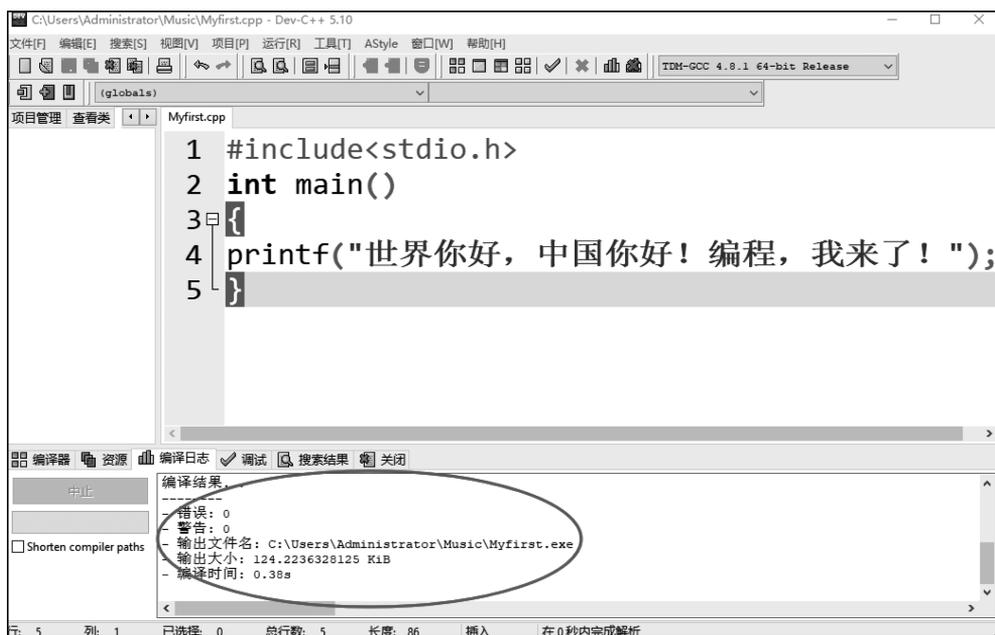


图 1-10 Dev C++生成可执行文件

(3) 运行程序：编译完成后，打开源文件所在的目录，会看到目录下新增了一个名为 Myfirst.exe 的可执行文件。双击 Myfirst.exe 运行或者在 Dev 界面上点击运行快捷键，即可出现如图 1-11 所示的运行窗口。

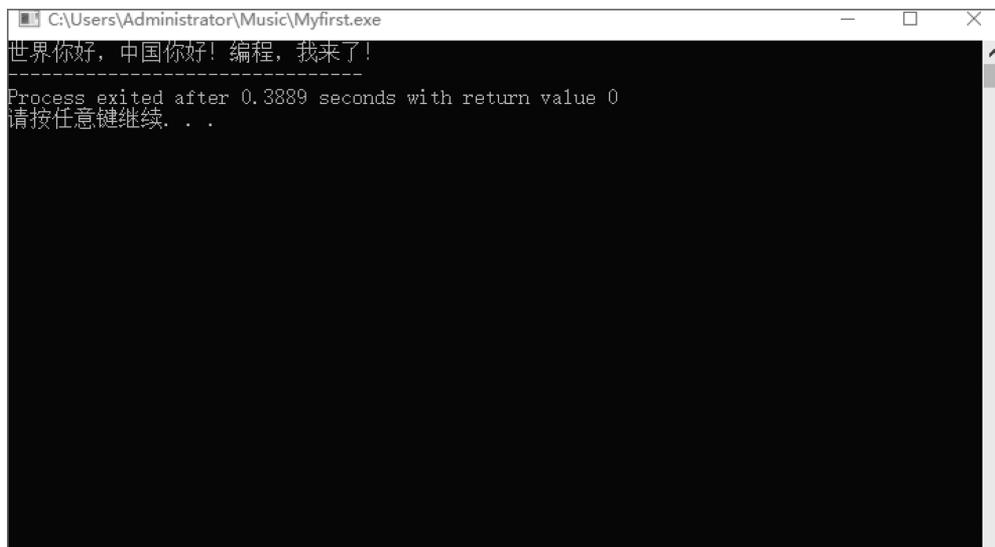


图 1-11 Myfirst.exe 执行界面

读者也可以将编译运行合二为一，方法是使用菜单“运行>编译运行”，或者直接按下 F11 键，如果代码没有错误，就会出现图 1-11 的运行界面。

## ► 思政点滴

《论语·卫灵公》：子贡问为仁。子曰：“工欲善其事，必先利其器。居是邦也，事其大夫之贤者，友其士之仁者。”常言说得好，“磨刀不误砍柴工”。工匠在做工前要先打磨好工具，操作起来就能得心应手，想要学好程序设计，首先要熟练掌握将要使用到的工具。

## 1.6 关于学习编程的建议

为了能让读者更快捷、更轻松学会程序设计，学会用C语言解决学习和工作中遇到的具体问题，特提供以下几点建议供读者参考：

(1) 编程是为了解决问题，解决问题的前提是对问题的分析，“设计是一个发现问题，而不是发现解决方案的过程”。一个优秀的程序员，要具备计算思维的能力，学习从计算机思维的角度去分析问题，设计方案。好的方案设计出来了，剩下的就是代码的实现，那是简单而又简单的事情。

(2) 编程语言是语言又不是语言。C语言、Python语言等各类编程语言仅是实现代码的工具，而不是解决问题的方法。学习编程语言，不是像学习自然语言一样去死记语法，而是掌握计算机程序的设计方法，方法懂了，学习也就事半功倍了。如若读者一味的背语法、背代码，将永远学不会程序设计。就像古代人习武一样，要掌握武术的“道”，而不是死记硬背每一招每一式。就像“George Carrette”说的一样，“首先学习计算机科学和那些理论，然后摸索出你自己的编程风格，然后把那些都忘掉，乱来就成。”

(3) “书读百遍、其义自见”。古人都懂得的道理，现代人往往把它抛到了脑后。特别是现在这个技术宰制的世界，一切似乎都被碎片化了，短的形式几乎笼罩了生活的每一个细节。如果编程仅仅是几个小视频能帮你解决的事情，你还怎能期许学科知识的完整！离开整全的视野，你的技术又将在何处安顿呢？不管信息处理的方式如何演进，书还得一页一页的读，从第一行到最后一行。

(4) “实践是检验真理的一切标准”。如果光读书，不练习，不写代码，将永远学不会程序设计。多读一些典型的好程序，掌握程序设计思想的好方法，然后去实践，是提升编程能力的最佳途径。在编程时要注意掌握良好的编程风格，注意编写简单、正确、易懂的语句，而不要刻意追求编程技巧，写出一些难以看懂的语句。

(5) 调试、还是调试。任何问题，解法都可能不是唯一的，尝试多几种方法去解决问题，编程的思路就开阔了。一个问题的解决方案，在逻辑上看是没有问题的，但可能存在着隐藏的Bug，程序员通过调试，可以发现错误，排除错误，又能进一步理解程序的行为，这也是一次再学习的过程。

(6) 参加一些学科竞赛。鼓励读者，特别是在校的大学生读者，多参加一些编程比

赛、创新比赛、学科竞赛。不是为了拿奖，过程重于结果，在准备比赛的过程中让自己动起来，锻炼自己的实践能力。

无论是你将来从事何种职业，无论你将来的工作中用不用得上计算机编程，学习编程思想，掌握计算机思维都是大有裨益的，期待读者能从本书的阅读与学习中收获点什么。

## 习 题 1

### 一、选择题

- 下面选项中表示主函数的是 ( )。
  - main()
  - int
  - printf()
  - return
- C 语言属于 ( )。
  - 汇编语言
  - 高级语言
  - 机器语言
  - 以上均不属于
- (多选) 关于 main() 函数，下列说法中正确的是 ( )。
  - 主函数必须有返回值
  - 一个 C 语言程序只能包含一个 main()
  - 主函数位置任意
  - 主函数是 C 语言程序的入口
- (多选) 下列开发工具中，可以用于开发 C 语言程序的工具有 ( )。
  - Visual Studio 2019
  - Turbo C
  - Dev-C++
  - Qt
- 下列选项中不属于 C 语言优点的是 ( )。
  - 语法灵活
  - 可移植性强
  - 面向对象
  - 运算符丰富、结构清晰
- 下面合法的用户标识符是 ( )。
  - a \* b
  - student
  - 3ab
  - a-b
- C 源程序的基本单位是 ( )。
  - 函数
  - 文件
  - 语句
  - 库函数
- 以下叙述不正确的是 ( )。
  - 一个 C 源程序可由一个或多个函数组成
  - 一个 C 源程序必须包含一个 main 函数
  - C 程序的基本组成单位是函数
  - 在 C 程序中，注释说明只能位于一条语句的后面
- 一个 C 语言程序是由 ( )。
  - 若干函数组成
  - 若干子程序组成
  - 若干过程组成
  - 一个主程序和若干个子程序组成
- C 语言规定：在一个源程序中，main 函数的位置 ( )。
  - 必须在最开始
  - 必须在系统调用的库函数的后面
  - 可以任意
  - 必须在最后

## 二、填空题

1. C 语言程序的可执行文件的后缀名为\_\_\_\_\_。
2. 在 C 语言程序中，如果使用 printf()，则应在文件中包含\_\_\_\_\_。
3. \_\_\_\_\_是 C 语句的结束标志。
4. C 语言中的注释符有\_\_\_\_\_和\_\_\_\_\_两种，其中\_\_\_\_\_可用于多行注释。
5. C 语言程序中用以格式化输入的库函数为\_\_\_\_\_。
7. C 语言规定，用户自定义标识符由字母、数字、下划线组成，并且第一个字符必须是\_\_\_\_\_。
8. 结构化程序设计的三种基本结构是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
9. 结构化程序设计的基本原则是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
10. 用以定义整型变量的关键字是\_\_\_\_\_。

## 三、判断题

1. C 语言是解释型语言。 ( )
2. C 语言的源程序在运行之前必须进行编译。 ( )
3. C 语言不能用于游戏开发。 ( )
4. C 语言的注释不影响程序的运行。 ( )
5. C 语言中的 main 是程序的入口。 ( )
6. C 语言因出现时间太久，目前已不太适合系统软件开发。 ( )
7. 由于汇编语言是最靠近硬件的语言，其可移植性较好。 ( )
8. 注释在预编译时会被编译到程序中，在执行中会被解释执行。 ( )
9. 计算机语言可分为四大类：机器语言、汇编语言、脚本语言和高级语言。 ( )
10. 程序是由算法+数据结构组成。 ( )

## 四、问答题

1. 简述计算机解决问题的基本步骤。
2. 什么是 C 语言？为什么要学习 C 语言？
3. 谈一谈你对计算思维与 C 程序设计的理解。
4. 查找资料，总结目前排名前 10 位的编程语言各自的特点和主要应用领域。
5. 查找资料，总结一下自由软件和开源软件的区别。

## 五、编程题

1. 模仿案例 1-3，编程输出社会主义核心价值观的内容。
2. 模仿案例 1-4，输入 4 个整数，输出其最大值。
3. 从键盘上输入圆的半径，编程计算这个圆的面积。
4. 已知华氏温度与摄氏温度之间的转换公式为： $^{\circ}\text{C} = (5/9) \times (^{\circ}\text{F} - 32)$ 。编写一个程序，在屏幕上分别显示华氏温度  $0^{\circ}\text{F}$ ， $30^{\circ}\text{F}$ ， $60^{\circ}\text{F}$ ， $100^{\circ}\text{F}$  与摄氏温度的对照表。