

第 1 章

程序设计思想与方法



我亦无他，惟手熟尔。

——欧阳修《卖油翁》

项目案例

有一杯水和一杯油，现在需要我们告诉机器人将两个杯子里面的液体交换，请问你有什么好的方法吗？



问题驱动

- (1) 如何跟计算机交流？
- (2) 为什么要学 C 语言？
- (3) 如何利用 C 语言解决问题？

章节导读

程序设计 (Programming) 是指通过编写代码，使用计算机语言将解决问题的步骤和逻辑转化为计算机可以执行的指令的过程。掌握程序设计的核心思想和方法，培养解决问题的能力 and 编程思维，是软件开发设计工作的重要基石。本章将对计算机语言、C 语言的来龙去脉、程序设计的流程和步骤等进行介绍，帮助学习者理解如何跟计算机交流，为什么要学习 C 语言，如何进行程序设计等问题，同时帮助学习者建立扎实的编程基础、培养读者解决问题的思维方式。通过理论与实践相结合，学习者可以逐步成长为一名优秀的程序员。

① 这句话强调了熟能生巧的道理。这里告诉大家学习程序设计没有捷径，只有通过大量的实践和练习，才可以提高自己的程序设计的水平和分析并解决问题的能力。

1.1 如何跟计算机交流

计算机硬件系统由运算器、控制器、存储器、输入设备和输出设备 5 部分组成。其中运算器的主要功能是算术运算和逻辑运算；控制器是控制整个计算机，向计算机的其他部件发出控制信号，使它们协调一致工作的部件；存储器的主要功能是存放程序和数据；输入设备和输出设备用于输入和输出信息，在日常生活中，人们通常通过键盘、鼠标、触摸屏等输入输出设备来操作计算机。



计算机系统是由硬件系统和软件系统组成，所有操作都是通过触发软件来完成的。没有安装任何软件的计算机称为裸机，一般情况下人们不方便也不喜欢与裸机打交道，而是习惯通过相关软件来使用计算机。那么软件是什么？软件是一系列按照特定顺序组织的计算机数据和指令的集合，由计算机语言设计而成。因此，人们跟计算机进行交流，其实是通过计算机语言来完成的。

1.1.1 计算机语言

语言 (Language) 是人类最重要的交际工具，是人们进行沟通交流的各种表达符号。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。语言是传承人类文明成果最重要的工具，是人类交流思想的最主要媒介，是推动人类文明发展和社会进步的主要动力。如汉语是世界上使用人数最多的语言，是博大精深的中华文明的传承者，是讲好中国故事的有力武器。

计算机语言 (Computer Language) 指人与计算机之间进行通信交流的工具，是人与计算机之间传递信息的媒介。计算机系统的最大特征是指令通过一种语言传达给机器，也就是说如果要命令计算机做一个什么事情，那么必须是用计算机语言来告诉计算机做什么、怎么做，这样计算机才能搞明白。

当代青年都了解，学语言时应该是从词法、语法、语义和语音 4 个方面入手。计算机语言与此类似，除了语音之外，还要从词法、语法和语义 3 个方面进行学习。其中词法是语言中符号构成规则和其含义；语法是符号组成语句的规则；语义就是符号和语句在特定语境中的含义。

【例 1-1】解释“a-b;-b;”的含义。

【解】这是计算机 C 语言中的一条逗号运算语句。

其中“a”“b”“-”“，”“;”这 5 个符号有其自己特定的含义；符号根据计算机 C 语言语法规则构成一条语句，在特定语境表达唯一的意思，如“-”在“a-b”中表示减法运算，在“-b”中表示负号运算。

1.1.2 计算机语言的发展

计算机语言总的来说分为机器语言、汇编语言、高级语言 3 大类，而这 3 种语言也恰恰是计算机语言发展历史中的 3 个重要阶段，如图 1.1 所示。

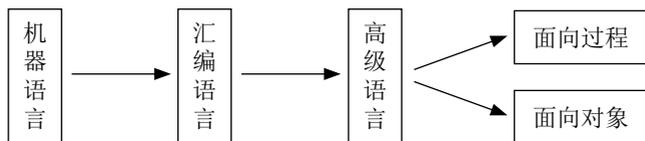


图 1.1 计算机语言发展的三个阶段

(1) 1946年2月14日，世界上第一台计算机 ENIAC 诞生。它使用最原始的穿孔卡片，用“打孔”和“不打孔”方式分别表示 1 和 0，以此与计算机进行交流。这与人类语言差别极大，我们将其称为机器语言。机器语言是第一代计算机语言，本质上是计算机能唯一识别的语言，由 0、1 序列构成的指令码组成。如：10000000 表示加法运算，10010000 表示减法运算。

(2) 机器语言的二进制码让人难以记忆和理解。如人与人交流称呼时用姓名比用身份证号码更方便一样，人类凭借智慧用符号代替二进制码，如用 ADD 代替 10000000 表示加法，从而出现了第二代计算机语言，称为汇编语言，也叫符号语言。机器语言和汇编语言可以直接对硬件进行操作，程序员编写程序时需要掌握相关硬件知识，所以我们将这两种语言称为面向机器的语言。

(3) 汇编语言尽管还是复杂，用起来也容易出错，但它有力地推动了计算机语言向更高级的语言发展，最终进入了“面向人类”的高级语言。高级语言是一种接近于人们的使用习惯和数学表达，与计算机的硬件结构及指令系统无关的程序设计语言。根据其编程思想的不同，高级语言分为两类：一是以事务处理的过程为中心，强调处理步骤，称为面向过程程序设计 (Procedural Programming Language) 语言，如 C、Fortran 等语言；另一类以事务处理的参与者为中心，强调的是参与者行为动作，称为面向对象程序设计语言 (Object Oriented Programming Language)，如 C++、Java 等语言。例如，利用计算机处理图书馆借书事务，面向过程就会按“查询→拿书→登记”这个过程进行程序设计，面向对象就会按借书事务中读者、图书、管理员等这些对象的行为动作进行程序设计。面向对象和面向过程是两种不同程序设计思想，二者各有特点，又相互交叉融合。

1.1.3 程序设计

计算机的全称是电子数字计算机，通过电压的高低来表示数据，因此它是一种以二进制数据形式在内部存储信息、以程序存储为基础、由程序自动控制的电子设备。

程序设计 (Programming) 是指给出解决特定问题的过程，是软件构造活动中的重要组成部分。程序是一条条有序指令的集合，是为实现某种功能或解决某个特定问题按一定的规则放在一块的一堆代码。

【例 1-2】求圆的面积。

【解】第一步告诉计算机圆的半径 r 和圆周率 π 的值；第二步告诉计算机求面积公式 $s = \pi \times r^2$ ；最后告诉计算机将面积 s 在屏幕上显示出来。

这个问题的解决过程往往以某种程序设计语言为工具来告诉计算机。通过 C 语言将一条条指令有序告诉计算机，帮助我们解决特定问题的过程，称为 C 语言程序设计。

1.2 为什么要学 C 语言

在人工智能时代，人工智能技术已经成为当今社会发展的重要驱动力，它的应用领域越来越广泛，涵盖了医疗、金融、教育、交通等领域。随着新一代信息技术的不断发展，各种语言也如雨后春笋般出现。这时有同学会问：我们为什么还要学习 C 语言？这里先不回答这个问题，而是让大家了解 C 语言的由来、特点和应用领域，再从中寻找出答案。



1.2.1 C 语言发展史

当计算机语言发展到第三代，就进入了“面向人类”的高级语言。高级语言是一种接近于人类使用习惯的程序设计语言。它允许用英文来写计算程序，程序中的符号和算式也与日常使用的数学式子差不多。高级语言发展于 20 世纪 50 年代中叶到 70 年代，第一个计算机语言是诞生于 1957 年的 Fortran 语言，由 IBM 公司设计。

C 语言诞生于美国的贝尔实验室，它的祖先是 BCPL 语言。1967 年，英国剑桥大学的马丁·理查兹 (Martin Richards) 对 CPL (Combined Programming Language) 语言进行了简化，于是产生了 BCPL (Basic Combined Programming Language) 语言。1970 年，美国贝尔实验室的肯尼斯·汤普森 (Kenneth Thompson) 以 BCPL 语言为基础，设计出很简单且很接近硬件的 B 语言 (取 BCPL 的首字母)，并用 B 语言写了第一个 UNIX 操作系统。在 1972 年，美国贝尔实验室的丹尼斯·里奇 (D. M. Ritchie) 在 B 语言的基础上设计出了一种新的语言，并取 BCPL 的第二字母作为这种语言的名字，它就是 C 语言。1973 年年初，C 语言的主体完成，汤普森和里奇迫不及待地开始用它完全重写了 UNIX 操作系统。

为了使 UNIX 操作系统推广，1977 年，丹尼斯·里奇发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年，美国电话电报公司 (AT&T) 贝尔实验室正式发表了 C 语言。同时，布莱恩·科尔尼干 (B.W.Kernighan) 和丹尼斯·里奇 (D. M. Ritchie) 合著了著名的《The C Programming Language》一书 (通常简称为 K&R，也有人称之为 K&R 标准)。但是，当时的 K&R 并没有定义一个完整的标准 C 语言。后来美国国家标准化协会 (American National Standards Institute, ANSI) 在此基础上制定了一个 C 语言标准，于 1983 年发布，通常称为 ANSI C。

K&R 第一版在很多语言细节上不够精确，对于 PCC 这个“参照编译器”来说，它显得不切实际；K&R 甚至没有很好地表达它所要描述的语言。1983 年夏天，ANSI 在计算机和商业设施制造商协会 (Computer and Business Equipment Manufacturers Association, CBEMA) 的领导下建立了 X3J11 委员会，目的是创建一个 C 标准。X3J11 在 1989 年年末提出了一个他们的报告 ANSI 89。国际标准化组织 (International Organization for Standards, ISO) 在 1990 年采纳了这个标准，进一步提升了 C 语言的标准化。这次标准化的内容通常被称为 ANSI/ISO C 或 C89/C90。

1994 年，ISO 修订了 C 语言的标准。1995 年，ISO 对 C90 做了一些修订，即“1995

基准增补 1(ISO/IEC/9899/AMD1:1995)”。1999 年, ISO 又对 C 语言标准进行修订, 在基本保留原来 C 语言特征的基础上, 针对需要增加了一些功能, 其中就包括 C++ 的一些功能, 并命名为 ISO/IEC9899:1999, 简称“C99”。

2001 年和 2004 年, C 语言先后进行了两次技术修正。2011 年 12 月 8 日, ISO 正式公布 C 语言的新国际标准草案——ISO/IEC 9899:2011, 即 C11。新的标准提高了对 C++ 的兼容性, 并将新的特性增加到 C 语言中。新功能包括支持多线程, 基于 ISO/IEC TR 19769:2004 规范支持 Unicode, 提供更多用于查询浮点数类型特性的宏定义和静态声明功能。2018 年 6 月, ISO 发布了 ISO/IEC9899:2018, 简称 C18(或 C17)。C18 标准没有引入新的语言特性, 只对 C11 进行了补充与修正。2022 年 9 月 3 日, ISO 于 Open Standards(计算机标准开放组织)网站上发布了新的 C 语言标准定稿, 称为 ISO/IEC 9899:2023, 简称 C23。

目前流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的, 但不同版本的 C 编译系统所实现的语言功能和语法规则又略有差别。本书内容以 ANSI C 为标准进行讲述。

1.2.2 C 语言的特征

C 语言的特征可以概括为以下几点。

1. 语言简洁、紧凑, 使用方便、灵活

C 语言只有 32 个关键字和 9 种控制语句, 程序书写形式自由, 主要用小写字母表示。这种简洁性使得 C 语言易于学习和使用。

2. 运算符和数据类型丰富, 表达力强

C 语言包含了 34 种运算符, 范围广泛, 功能强大。这使得 C 语言在表达复杂的算法和逻辑时非常灵活。同时 C 语言提供了丰富的数据类型, 包括整型、浮点型、字符型、数组类型、指针类型、结构体类型、联合体类型等。C 语言提供了丰富的运算符和数据类型, 以及强大的函数库, 使得其能够表达复杂的算法和数据结构, 可以满足不同场景下的需求。

3. 具有低级语言的特点

C 语言允许直接访问物理地址对硬件进行操作, 可以进行位(bit)操作。这使得 C 语言在嵌入式系统、操作系统等底层编程领域具有广泛的应用。

4. 生成目标代码质量高, 程序执行效率高

C 语言编译后生成的代码质量高, 运行速度快, 占用内存资源少。这使得 C 语言在高性能计算和实时控制等领域具有优势。

5. 可移植性好

C 语言是一种中级语言, 它保持了与汇编语言或机器语言的接近性, 但又克服了汇编语言过于依赖于具体机器硬件的缺点。因此, 用 C 语言编写的程序可以方便地移植到不同的计算机平台上。

6. 程序结构清晰，可模块化

C 语言提供了函数、宏定义、类型定义等机制，使得程序可以方便地划分为多个模块，提高了代码的可读性和可维护性。

综上所述，C 语言以其简洁性、灵活性、高效性和可移植性等特点，在软件开发领域具有广泛的应用和重要的地位。

1.2.3 C 语言的应用

C 语言不仅可以用于编写应用程序，还可以用于编写系统软件、驱动程序等。它的应用范围非常广泛，从嵌入式系统到高性能计算都有涉及，具体体现在下面几个应用场景。

1. 操作系统和嵌入式系统

C 语言最初是为开发操作系统而设计的，如 UNIX、Linux 和 Windows 的内核都大量使用 C 语言。C 语言在嵌入式系统中的应用也非常广泛，在家电、汽车、医疗设备中能够直接操作硬件，是开发嵌入式系统和应用程序的最佳选择。

2. 通信协议和数字信号处理

C 语言用于开发无线通信系统的软件，如协议栈、基带处理和信道编码等，能确保高效的无线数据传输和接收。在卫星通信领域，C 语言用于实现卫星通信系统的控制和信号处理软件，能确保数据传输的稳定性。C 语言对硬件的操作能力使其在对性能有严格要求的应用中表现出色，如网络程序的底层和网络服务器端底层、地图查询等。

3. 生物医学工程和金融与交易系统

在生物医学工程领域，C 语言用于开发医疗设备的软件，如心电图机、超声波仪和医学图像处理算法。在金融与交易系统中，C 语言用于开发高频交易系统、风险管理系统和金融建模工具，处理大规模金融数据时表现出色。

4. 游戏开发和图形处理

C 语言在游戏软件开发中也有广泛应用，许多游戏软件都是用 C 语言编写的。此外，C 语言还具有很强的绘图能力和数据处理能力，适用于系统软件、三维和二维图形的处理。

C 语言有着悠久的历史 and 庞大的用户群体，具有高效性、灵活性和可移植性等特点，在软件开发领域具有广泛的应用和重要的地位。学习和掌握 C 语言，对于从事软件开发、系统维护、科学研究等工作的人来说，具有重要的意义。因此，学习 C 语言不仅可以提高编程技能和解决问题的能力，还可以为未来的职业发展打下坚实的基础，这算是我们要学习 C 语言的理由吧。

1.3 如何利用 C 语言解决问题



C 语言是一种强大的编程语言，可以用来解决各种问题，范围从简单的数

值计算到复杂的系统编程。但无论是简单问题，还是复杂问题，都必须遵循程序设计的基本方法，而方法是我们处理问题的思维方式。

程序设计 (Programming) 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。一般程序设计方法的基本步骤如图 1.2 所示。

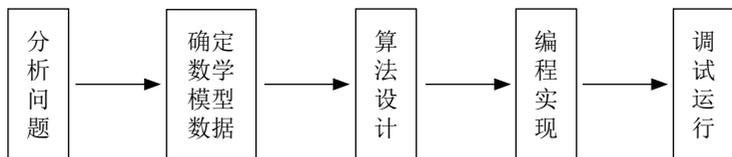


图 1.2 程序设计方法的基本步骤

1.3.1 分析问题

在着手解决问题之前，应该通过分析充分理解问题，明确原始数据、解题要求、需要输入输出的数据及形式等，具体如下。

- (1) 弄清程序要完成的功能。
- (2) 如果数据有输入，分析输入数据的格式和类型。
- (3) 对输入的数据做什么样的处理。
- (4) 处理结果使用哪个数据类型，如何保存，按什么格式输出。

比如，一杯水和一杯油的交换问题，程序功能是要完成两个杯中的液体交换，而不是杯子位置交换；输入数据应该为两容器中不同液体的质量，为简化问题，可以将其看成整数，也可以直接赋值；程序中主要对两个输入数据进行交换处理；最后将交换的结果在屏幕上输出，告知处理结果。

1.3.2 确定数据结构及模型

数据结构，指的是数据的存储结构及关系。数据模型，是对客观对象的数据特征的抽象描述。现实世界中的对象无法直接进入计算机，需要把它抽象为特定符号，放到计算机的存储器中，方便计算机进行处理。程序设计爱好者在学习过程中，要逐步树立程序设计的抽象思维，这是学好程序设计的重要基础。

比如，一杯水和一杯油的交换问题。水杯、油杯、水和油等都是无法直接放入计算机中处理的，需对其进行抽象描述，例如可以用 x 表示水杯，其值表示水的质量； y 表示油杯，其值表示油的质量；同时问题转换为 x 与 y 的值交换问题，记为 $x \leftrightarrow y$ 。

1.3.3 算法设计

所谓算法，是为解决某一特定的问题所给出的一系列确切的、有限的操作步骤，也称为解决方案 (方法)。

1. 算法设计准则

(1) 正确性。算法应当满足特定的需求，对于所有合法的输入数据都能得到正确的结果。

(2) 可读性。算法应当易于人的理解，以便于调试和维护。清晰的代码结构、合理的变量命名和模块化设计，可以使算法在后续的改进和调试中更加容易。

(3) 健壮性。当输入数据非法时，算法应能恰当处理，而不是产生错误的输出。处理错误的方法不应是中断程序执行，而是返回一个表示错误或错误性质的值。

(4) 高效性。算法的执行时间和存储量需求都应当尽可能低。时间复杂度衡量了算法运行时间随着输入规模增长的变化情况，优秀的算法应追求尽可能低的时间复杂度。

(5) 可扩展性。算法应该能够适应未来可能的变化和扩展，比如处理更大的数据量或者更复杂的任务。

(6) 模块化。将算法分解为独立的模块，由每个模块负责完成特定的功能，这样不仅提高了算法的可读性和可维护性，也有助于代码复用。

(7) 通用性。算法应该尽可能地适用于各种不同的应用场景和数据类型，而不仅仅局限于特定的问题或数据。

2. 算法的特征

算法是对计算机的解决问题方案的描述，每一步都应该是可以执行并且是确定、没有歧义的。使用者可以不通过输入设备进行输入，但是计算机一定要有一个输出的结果，这也要求算法步骤必须有限，否则计算机按照算法的步骤一直算下去，就一直不会有结果输出。因此，算法必须具备以下 5 个基本特征。

(1) 确定性：算法最基本的要求之一。它意味着算法的每一步操作都是明确的、无歧义的。在确定的算法中，对于任何给定的输入，算法都会产生相同的输出。只要输入保持不变，算法的行为应该是一致的。

(2) 有穷性：指算法必须能够在有限的步骤内结束，不应该无限循环。在实际的计算机中，算法必须能够在有限的时间内完成。

(3) 输入：算法至少应该有一个或多个输入。这些输入是算法在执行过程中所需的数据。算法的输入可以从用户处接收的，也可以是算法内部预设的。

(4) 输出：算法至少应该有一个结果输出。输出可以是一个数值、一个状态、一个图形或者其他形式的输出。

(5) 可行性：算法的每一步都应该是在实际计算机中可执行的。算法的步骤不能涉及实际无法完成的操作，如无限的循环或者超出计算机处理能力的计算等。

算法的 5 个特性共同构成了一个算法的基本框架，了解这些特性对于学习算法和编程至关重要，因为它们不仅是计算机科学的基本概念，也是评估和设计算法的标准。在分析和设计算法的过程中，这些特性是必须要考虑的，以确保算法的有效性和实用性。

3. 算法的描述

算法描述是指对设计的算法用一种方式进行详细的展现，以便与人交流。算法可以采用多种方式来描述，包括自然语言、伪代码、程序流程图等。这些描述方式在问题的描述上存在一定的差异，其中自然语言较为灵活但不够严谨，而伪代码和程序流程图则具有较高的严谨性，但可能缺乏灵活性。

(1) 自然语言描述。使用人类语言对算法进行描述，易于理解但不够精确，不易于扩展。

【例 1-3】求 5 的阶乘。

【解】最直观方法是 $1 \times 2 \times 3 \times 4 \times 5$ ，其基本步骤如下。

步骤 1：先求 1 乘以 2，得到 2 的阶乘 2。

步骤 2：将步骤 1 得到的结果 2 乘以 3，得到 3 的阶乘 6。

步骤 3：将步骤 2 得到的结果 6 乘以 4，得到 4 的阶乘 24。

步骤 4：将步骤 3 得到的结果 24 乘以 5，得到 5 的阶乘 120。

(2) 伪代码描述。一种介于自然语言和编程语言之间的描述方式，既具有自然语言的易读性，又具备编程语言的严谨性。它不依赖于具体的编程语言，而是使用类似于自然语言的语法来描述算法的结构和逻辑。

【例 1-4】求 5! 的阶乘。

【解】求 5! 的伪代码表示如下。

```
BEGIN
  n := 5
  result := 1
  FOR i FROM 1 TO n DO
    result := result * i
  END FOR
  PRINT result
END
```

(3) 程序流程图描述。程序流程图用图的形式画出程序流向，是算法的一种图形化表示方法，具有直观、清晰、更易理解的特点。程序流程图是目前描述算法的常用方法，一般分传统流程图和 N-S 流程图两类。其中传统流程图使用一组特定图形符号描述程序运行具体步骤，常用符号如图 1.3 所示。N-S 图又称盒图，它完全去掉了流程线，算法的每一步都用一个矩形框描述，把一个个矩形框按执行的次序链接起来，就是一个完整的算法描述。



图 1.3 传统流程图常用符号

在结构化程序设计中，程序的执行流程由 3 种基本结构组成：顺序结构、选择结构和循环结构。流程图对不同结构画法有所不同。

- 顺序结构：程序中的语句按先后顺序逐条执行。

【例 1-5】求 a 与 b 的和。

【解】用传统流程图和 N-S 图表示其算法，分别如图 1.4 和图 1.5 所示。

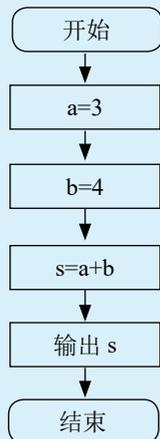


图 1.4 传统流程表示

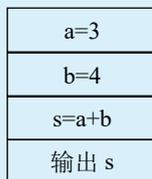


图 1.5 N-S 图表示

- 选择结构：在执行程序中的选择结构语句时，该结构将根据不同的条件执行不同分支的语句。

【例 1-6】求 a 与 b 中的最大值。

【解】用传统流程图和 N-S 图表示其算法，分别如图 1.6 和图 1.7 所示。

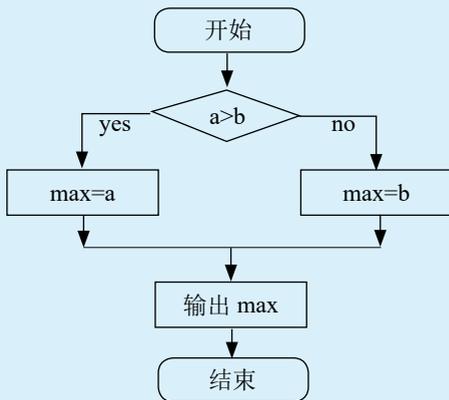


图 1.6 传统流程表示

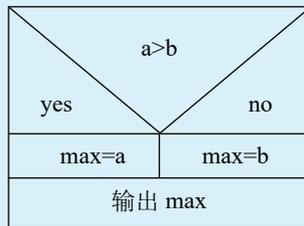


图 1.7 N-S 图表示

- 循环结构：当条件满足时，就执行循环体，否则就退出循环结构。

【例 1-7】求 5 的阶乘。

【解】用传统流程图和 N-S 图表示其算法，分别如图 1.8 和图 1.9 所示。

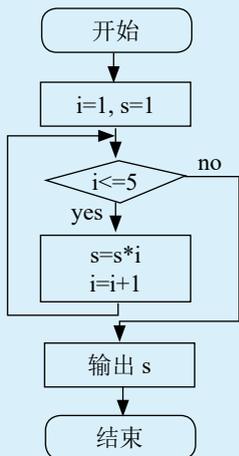


图 1.8 传统流程表示

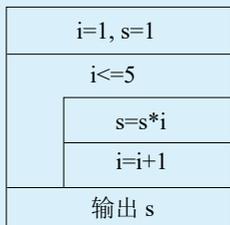


图 1.9 N-S 图表示

1.3.4 编程实现

编程实现是指将设计的方案、理念或者算法转换为可以工作的、有效的代码或程序的过程。在编程领域，将设计好的算法用代码表达出来是实现的典型例子。但这不仅仅是编程语言语法的堆砌，还包括代码的优化、错误处理、可维护性和可扩展性等多个方面的考虑。编制 C 语言程序的基本步骤一般包括编辑、编译、链接和执行等，具体步骤如图 1.10 所示。

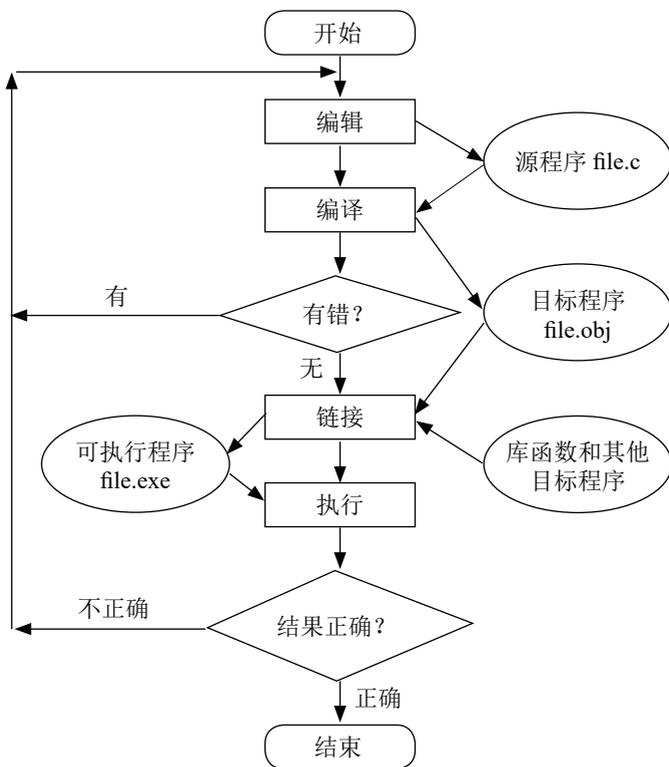


图 1.10 编制 C 语言程序的基本步骤

(1) 编辑：使用 C 语言编写源代码，这是程序开发的第一步。源代码是程序员用高级语言编写的，人类可读的文本文件。

(2) 编译：使用 C 语言编译器将源代码转换为可执行的二进制文件。编译过程包括词法分析、语法分析、语义检查、中间代码生成、代码优化和目标代码生成等阶段。如果在编译过程中发现语法错误，编译器会提供错误提示，以方便开发者进行修正。

(3) 链接：一旦源代码被编译成目标文件，还需要将这些目标文件与系统库链接起来，形成一个完整的可执行程序。这个过程称为链接，它将各个目标文件和系统库中的函数和变量合并，生成最终的可执行文件。

(4) 执行：运行编译和链接后，生成的是可执行程序文件。这个文件包含了可以被计算机处理器执行的指令，执行程序将执行这些指令，完成预定的任务。

在编制 C 语言程序过程中，会生成源程序、目标程序和可执行程序 3 种程序文件，其特征如表 1.1 所示。

表 1.1 源程序、目标程序和可执行程序的特征

	源程序	目标程序	可执行程序
内容	程序设计语言	机器语言	机器语言
是否可执行	不可以	不可以	可以
文件名后缀	.c 或 .cpp	.obj	.exe

编程实现的整个工作都可以在集成开发环境中完成，目前 C 语言常用的集成开发环境包括以下几种。

- Visual Studio。这是 Windows 平台下的标准 IDE，功能强大，支持多种编程语言。它提供了丰富的开发工具和调试器，如代码编辑器、编译器、调试器等。这个环境虽然功能强大，但不适合初学者使用，因为其安装包较大，包含许多暂时用不到的工具。
- Eclipse。这是一个开源的集成开发环境，支持多种编程语言，其中就包括 C 语言。Eclipse 具有可扩展性强的特点，可以通过插件增加额外的功能，适合需要跨平台开发的用户。
- Code::Blocks。这是一款轻量级的集成开发环境，专门针对 C 和 C++ 语言。它提供了简洁的界面和易于使用的功能，适合初学者使用或小型项目开发。
- Dev-C++。这是一款适用于 Windows 环境的 C/C++ 开发工具，功能简洁，适合初学者使用。它没有完善的可视化开发功能，调试功能较弱，但自带的编译器能满足初学者。
- Xcode。这是苹果公司为 Mac OS X 和 iOS 开发的集成环境，支持多种编程语言，其中就包括 C 语言。它提供了丰富的工具和调试器，适合 Mac 用户使用。

这些集成开发环境各有其特点和适用场景，开发人员可以根据自己的需求选择合适的 IDE 进行开发。本书将以 Dev-C++ 集成环境讲解 C 语言程序设计。

1.3.5 调试运行

程序调试运行是指对程序进行查错和排错，一般应经过以下几个步骤。

1. 静态检查 (人工检查)

在将代码放入编译器编译之前,先仔细对程序进行人工检查,查看程序的结构是否清晰合理、各函数间的调用关系是否正确,检查变量命名是否规范、有无拼写错误,以及检查语法上是否存在明显问题,比如括号是否匹配、有没有漏写分号等。

2. 利用编译程序检查语法错误

- 编译错误 (Error)。由 C 编译系统对程序进行查错时,如果出现编译错误,必须根据错误提示找出错误的位置并改正。有时代码中一个错误可能导致产生一大批编译错误,这种情况下应该从上到下逐一改正,修改一两个之后再次编译。
- 编译警告 (Warning)。警告也需要根据情况处理,若不处理可能产生运算误差等。

3. 运行程序并检查逻辑错误

排除语法错误后运行程序,输入多组不同的数据测试程序,检查运行结果是否符合要求。如果运行结果错误,这可能是由于程序中存在逻辑错误。此时通常需要对照流程图检查算法逻辑,确定程序的逻辑流程是否正确。对于怀疑出错的地方,可设置断点,查看程序当前的状态;也可一步一步地执行,监控变量值的变化,通过输出的值来判断程序是否按照预期运行,从而找到出错的程序段,缩小查错范围。



1.4 项目实战



➤ 我们有一杯水和一杯油,现在告诉机器人将两个杯子里面的液体交换,请问你有什么好的方法吗?



1.4.1 项目问题分析

项目是要完成两个杯中的液体交换,而不是杯子位置交换;输入数据简化成整数,在程序中直接赋值,分别表示两种液体质量;程序主要对两个输入数据进行交换处理;最后将交换结果在屏幕上输出。

1.4.2 数据模型的构建

水杯、油杯以及水和油都是无法直接放入计算机中处理的,我们对其进行抽象描述,构建正确的数据模型,具体如下。

- `int x;` //x 表示水杯,其值表示水量
- `int y;` //y 表示油杯,其值表示油量
- 问题描述: $x \leftrightarrow y$ 。

1.4.3 算法的设计

【算法 1】用一个空杯 z 作为缓冲,完成水和油的交换,具体算法描述如图 1.11 所示。

【算法 2】不用一个空杯做缓冲，利用水和油的密度不同来完成交换，具体算法描述如图 1.12 所示。

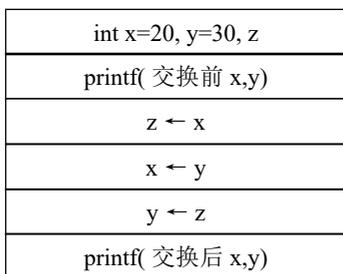


图 1.11 算法 1 的 N-S 图描述

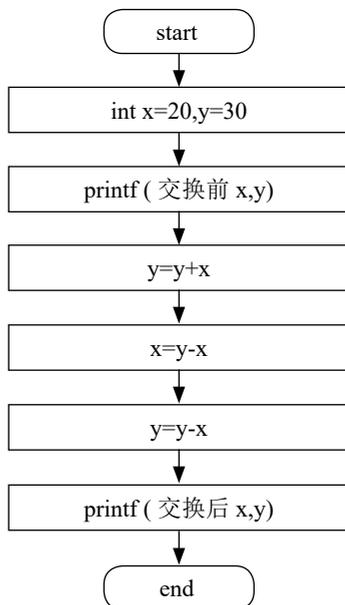


图 1.12 算法 2 的传统流程图描述

1.4.4 代码编写

C 语言源代码可以在任何文本编辑器中编辑，也可以在集成环境下直接编辑。

【算法 1】

```

/* This is the first C program */
#include <stdio.h>
int main ( )
{
    int x=20, y=30, z;
    printf(" 交换前 :x=%d, y=%d!\n",x,y) ;
    z=x;
    x=y;
    y=z;
    printf(" 交换后 :x=%d, y=%d!\n",x,y) ;
    return 0;
}
  
```

其中第 1 行为注释语句，用于帮助程序员理解代码。C 语言中的注释由“/*”开始、由“*/”结束，这种注释方式可以实现多行注释；使用“//”可进行单行注释。

第2行为预处理语句，告诉编译器到哪里找使用的系统库函数。如果要使用系统库函数，需用预处理指令包含其相应的头文件。预处理指令都是以“#”号开头。本程序中使用了printf库函数，该函数的声明放在stdio.h头文件中。预处理语句有两种形式：一种是使用尖括号“<>”包含系统头文件，如#include<stdio.h>，此时系统会先搜索编译器指定的路径，然后搜索标准系统头文件路径。另一种使用双引号(“”)包含用户自定义的头文件，如#include“myheader.h”，此时系统会先在搜索当前工作目录，然后搜索系统头文件路径。

从第3行起是主函数main。C程序由函数构成，有且仅有一个main函数，它是程序的入口和出口，即程序从main的第1个“{”开始执行，到最后一个“}”结束。“{}”中包含的语句称为函数体，函数体中每一条语句以分号结束，其中“int x=20, y=30, z;”为变量定义语句；“printf(“交换后 :x=%d, y=%d!\n”,x,y);”和“printf(“交换后 :x=%d, y=%d!\n”,x,y);”为输出语句；“z=x; x=y; y=z;”这3条语句完成数据交换；“return 0;”为函数返回语句。

C语言源文件无程序行的概念，程序中可使用空行和空格，语句都是以分号结尾。C语言所使用的符号为英文状态下符号，字母区分大小写。

【算法2】

```

/* This is the first C program */
#include <stdio.h>
int main ( )           // 无参数、有返回值的主函数
{
    int x=20, y=30;    // 定义变量 x 表示水杯, y 表示油杯, 并赋初值
    printf(" 交换前 :x=%d, y=%d!\n",x,y); // 交换前输出
    // 开始交换
    y=x+y;
    x=y-x;
    y=y-x;
    printf(" 交换后 :x=%d, y=%d!\n",x,y); // 交换后输出
    return 0; // 返回
}

```

1.4.5 程序调试运行

我们采用 Dev-C++ 5.11 集成环境，程序的编辑、调试、运行都在集成环境下完成。如果您还没有安装的话，可以从课程网站 (<https://www.xueyinonline.com/detail/246393878>) 资料中下载安装。

1. 集成环境安装运行

(1) 从课程网站下载 devcpp5.11.rar 后，解压此压缩包，双击 devcpp 5.11.exe 安装文

件, 开始安装。首先弹出 Installer Language(安装语言选择)对话框, 如图 1.13 所示, 选择 English(英文), 单击 OK 按钮, 确定安装语言。

(2) 弹出 License Agreement(许可证)窗口, 如图 1.14 所示。直接单击 I Agree 按钮, 同意安装许可。

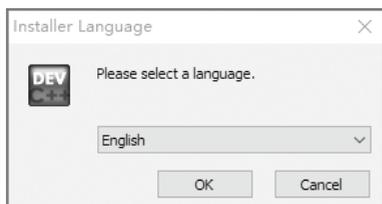


图 1.13 安装语言选择窗口

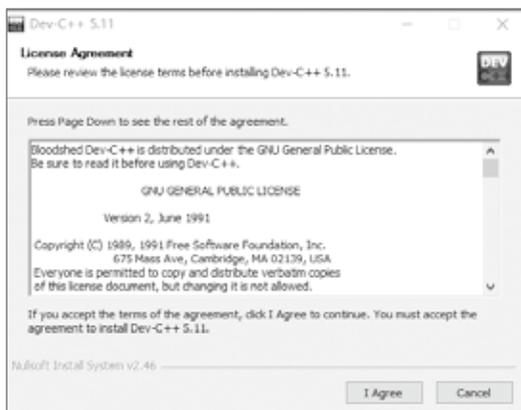


图 1.14 许可证窗口

(3) 弹出 Choose Components(组件选择)窗口, 如图 1.15 所示, 安装类型选择 Full(默认值), 然后单击 Next 按钮。

(4) 弹出 Choose Install Location(选择安装位置)窗口, 如图 1.16 所示。可以单击 Browse 按钮选择新的安装位置。单击 Install 按钮, 开始安装文件。

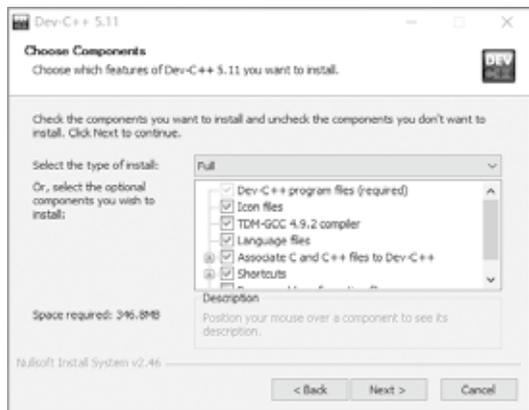


图 1.15 组件选择窗口

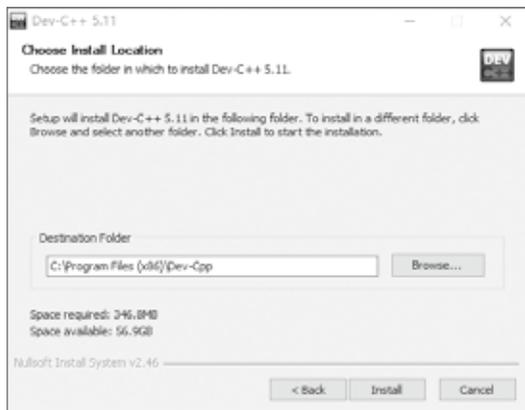


图 1.16 选择安装位置窗口

(5) 等待 Dev-C++ 安装完毕, 弹出完成安装向导窗口, 如图 1.17 所示。保留选中“Run Dev-C++ 5.11”复选框, 单击 Finish 按钮, 完成安装。

(6) 安装结束后, 在 Windows 中选择“开始”→“Bloodshed Dev-C++”→“Dev-C++”命令启动集成环境; 也可以用桌面快捷方式启动。当首次运行 Dev-C++ 时, 会弹出首次运行配置窗口, 如图 1.18 所示。选择语言为“简体中文”, 然后单击 Next 按钮, 进入下一步。



图 1.17 完成安装向导窗口



图 1.18 首次运行配置窗口

(7) 弹出主题选择窗口，如图 1.19 所示。在此可以选择编辑器的字体、颜色和图标，通常保持默认值即可。设置完毕后，单击 Next 按钮，进入下一步。

(8) 在弹出的已设置成功窗口中，直接单击 OK 按钮，完成首次安装配置。紧接着会出现 Dev-C++ 的主界面，如图 1.20 所示。



图 1.19 主题选择窗口



图 1.20 Dev-C++ 的主界面

2. 源程序编辑

(1) 在 Dev-C++ 主界面中，选择“文件”→“新建”→“源代码”命令，可以在编辑窗口直接编辑源程序，也可将编辑好的源程序复制过来，其操作与其他文本编辑器类似，如图 1.21 所示。

(2) 编辑完成后，选择“文件”→“保存”命令保存文件，此时可以选择文件保存位置、文件名和保存类型。保存位置和文件名可以根据自己喜爱设置，保存类型可以选择 C 或 C++ 源文件，如图 1.22 所示。



图 1.21 Dev-C++ 的编辑窗口

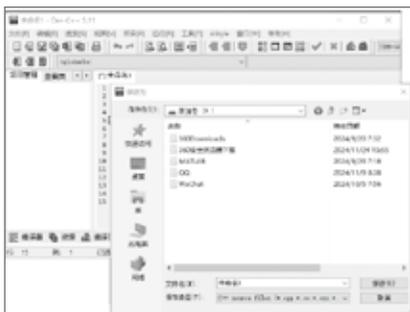


图 1.22 Dev-C++ 的保存窗口

3. 编译运行

(1) 选择“运行”→“编译”命令，或者按 F9 键，即可编译，文件如图 1.23 所示。编译文件时，主要检查语法错误，将 C 代码文件翻译成二进制汇编文件，这是因为计算机无法识别 C 文件。只能识别汇编文件。编译器完成编译工作，不需要我们手工操作，只要保证按照 C 语言的语法编辑好源程序即可。如果源程序有语法错误，编译窗口(图 1.23 下方)会显示编译错误结果，大家可以按提示进行修改。

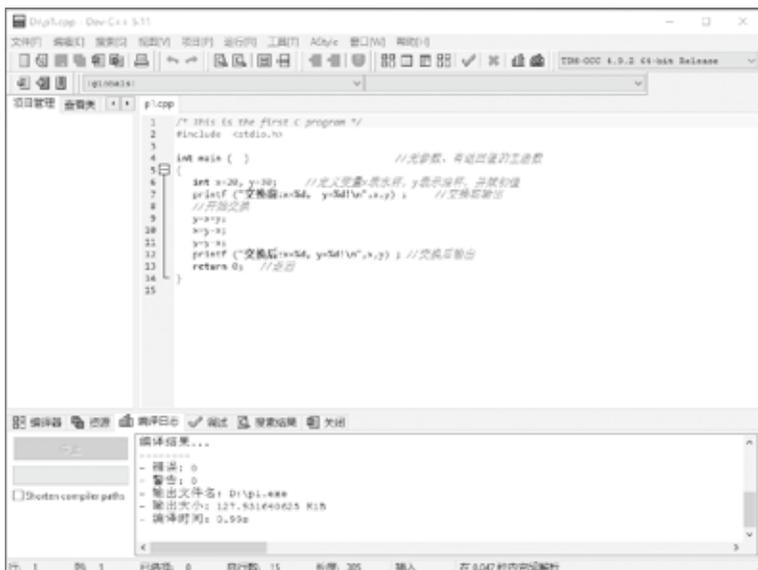


图 1.23 Dev-C++ 的编译窗口

(2) 选择“运行”→“运行”命令，或者按 F10 键，即可运行文件，结果如图 1.24 所示。运行是执行 EXE 文件(编译生成的目标文件经链接后生成 EXE 文件)。当然，也可以将编译和运行结合，选择“运行”→“编译运行”命令或者按 F11 键即可。



图 1.24 程序运行结果窗口

4. 调试运行

当程序存在逻辑上的一些错误，也就是经常说的 bug，我们肉眼一时无法识别时，可以通过调试来修改完善程序。打开一个 C 语言源文件，一定要先编译它，这样做的原因一是看代码是否能够编译通过；二是调试运行时都是需要先编译的。

首先，调试的关键是设置“断点”，具体方法是在要设置断点的那行代码开头处的数字上单击，如图 1.25 中红色一行所示。可以设置多个断点，也可以只设置一个断点。如果设置了多个断点，程序会在断点与断点之间进行调试。如果只有一个断点，程序会从设置断点处开始，随着点击一步一步执行，直到程序结束。

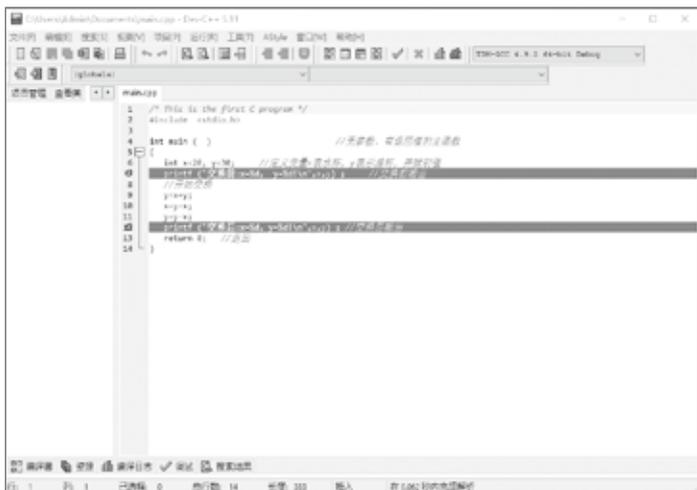


图 1.25 断点设置窗口

选择“运行”→“运行”命令或者按 F5 键，即可进入调试状态。调试开始后，可以单击“下一步”按钮，让程序运行到需要的位置。图 1.26 中蓝色一行表示当前程序运行的位置。也可以单击“添加查看”按钮，实时查看变量的值。

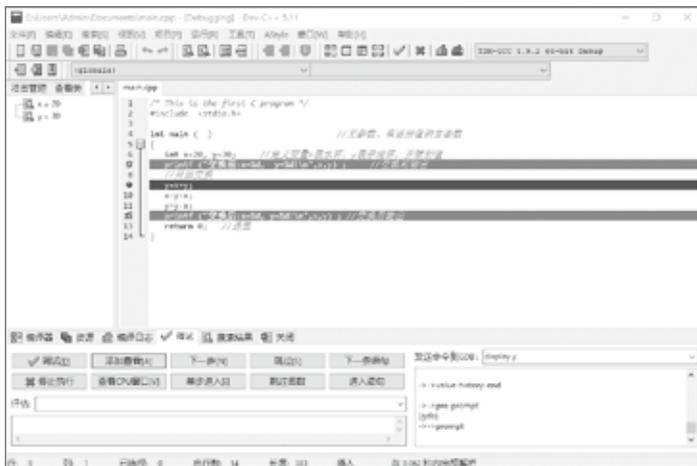


图 1.26 调试窗口

最后简单介绍一下调试窗口中常用的工具：“调试”——开始调试运行；“添加查看”——实时查看定义的变量、数组等值，它们会在左侧空白栏显示；“下一步”——让程序往下进行单步运行模式；“停止执行”——退出“调试”。



总结拓展



【本章小结】

本章主要是让读者对程序设计有一个整体的认识，帮助读者搞清楚如何跟计算机交流、为什么要学习 C 语言、如何利用 C 语言解决问题，培养读者的程序设计思维，掌握程序设计的方法，为后面的学习打下坚实的基础。其中主要介绍了以下几个方面内容。

(1) 通过理解程序设计与计算机语言等基本概念，形成程序设计理念。

(2) 了解 C 语言的来源与发展，认识学习 C 语言的重要性。

(3) 掌握 C 语言程序的基本结构。C 程序由函数构成，有且仅有一个 `main` 函数，它是程序的入口和出口。C 语言的语句都是以分号结尾。

(4) 掌握程序设计的基本步骤，了解 DEV 集成环境，能设计编写第一个 C 语言程序。

(5) 养成良好的编程风格：用锯齿形书写格式，多加注释。变量名、函数名见名知义，必须是合法的标识符，大小写敏感，先定义后使用。

本章所讨论的水与油交换问题中，两种算法各有优势，但都忽略了杯子的容量。这将涉及计算机数据存储，详情在后续章节将重点讨论。

【思政故事】

这一天，永远值得我们铭记

1964 年 10 月 16 日 15 时整，在新疆罗布泊荒漠闪过一道强烈亮光，惊天动地的巨响之后，巨大火球转为蘑菇云冲天而起——中国第一颗原子弹爆炸成功。“东方巨响”震惊世界，同时向世界庄严宣告：中国人民依靠自己的力量，打破了超级大国的核垄断。这是中国人民为保卫国家安全、维护世界和平、实现民族复兴而进行的一场伟大的科学探索，也是中国科技史上的一座里程碑。这是一个值得每个中国人骄傲和铭记的大日子。

积弱积贫，百废待兴

中国研制第一颗原子弹时，正值新中国成立不久。当时新中国处在内忧外患的严峻境地，面对着世界两大阵营的“冷战”、帝国主义对我国的各种封锁与核讹诈，以及后来苏联的背信弃义。近代中国积弱积贫，由于没有强大的国防，成了任人撕咬的肥肉。要使新中国成为谁也啃不动的硬核桃，真正站得住、不受人欺负，就要有尖端武器。毛泽东主席

在1958年6月的军委扩大会议上指出：“原子弹，没有那个东西，人家就说你不算数。那么好，我们就搞一点。搞一点原子弹、氢弹、洲际导弹，我看有10年工夫是完全可能的。”在苏联的援助下，中国于1958年建成了第一座实验性原子反应堆。这一年，成立了以万毅为部长的国防部第五部，负责领导特种部队的组建工作；成立了以聂荣臻为主任的国防科学技术委员会，负责统一领导国防科学技术研究工作；不久，国防部第五部合并到国防科委。1959年6月，苏联中止合同，随后撤走了专家。毛泽东毅然决定：自己动手，从头摸起，准备用8年时间，造出原子弹。他明确指出：“要下决心搞尖端技术。赫鲁晓夫不给我们尖端技术，极好！如果给了，这个账是很难还的。”中央把原子弹工程定名为“596工程”，要造“争气弹”。我们面前是完全空白的，没有先进的计算机，只有几台老式的手摇式计算机；研究人员更多只能依靠纸笔、计算尺等原始的工具。但随着纸张不断加厚，原子弹的理论设计也一步一步推进。

干惊天动地事，做隐姓埋名人

1960年春天，中央军委命令陈士榘将军率领中国的第一批特别工程部队进入罗布泊，开始了中国第一个核试验基地的工程建设。同时，中共中央在七、八月召开工作会议，讨论如何克服面临的严重困难，以及发展国防科学技术特别是尖端技术的问题，提出要“埋头苦干，发愤图强，自力更生，奋勇前进”，并采取了一系列重大措施：一是加强领导，组织全国各科研、生产部门协作攻关；1962年11月，成立了以周恩来任主任、罗瑞卿任办公室主任、国务院几位副总理及中央军委有关部门领导参加的专门委员会。二是遵照“缩短战线，任务排队，确保重点”的原则，对其他一些尖端武器发展项目，除保留一定的骨干力量继续攻关外，暂缓进行，以集中力量研制原子弹。三是选调技术骨干100名、大中专毕业生6000名，培养充实原子弹研制队伍。

中央专委在周恩来领导下，组织各方面的力量，及时在人力、物力、财力等方面进行调度，卓有成效地组织了全国大协作，解决了研制原子弹中遇到的100多个重大问题，安排了原子弹所需的特殊材料、部件和配套产品2万余项的研制生产，大大加快了原子弹研制的步伐。广大科技工作者在科研和生活条件十分艰苦的环境下，凭着为祖国争光的勇气，克服重重困难，发挥聪明才智，攻克了一道道难关，经过反复试验论证，于1963年3月提出了中国第一颗原子弹理论设计方案；同时，西北核武器试验场和研制基地的建成，为全面突破原子弹技术创造了条件。于是，党中央将国防科研人员陆续迁往大西北，开始进入研制原子弹的总攻阶段。

在美、英、苏三国联合遏制中国进行核试验的大背景下，中国的科学家们努力工作、发愤图强，在核武器的研究方面取得了一系列重大的突破。彭桓武、邓稼先、周光召、胡思得、周毓麟、孙清河、李德元、朱建士、秦元勋等科技理论家完成了理论的设计；王淦昌、吴世法、陈能宽、林传骝等人进行了爆炸物理试验研究；钱三强、何泽慧、王方定等人进行了中子物理试验研究；惠祝国、祝国梁等人进行了引爆控制研究；郭永怀、龙文光等人

进行了结构设计方面的研究。到 1964 年夏天，我国终于全面突破了原子弹技术难关，取得了原子弹研究方面的巨大成就。经过广大科技人员奋发图强，1964 年 6 月 6 日，经过爆轰模拟试验，我国胜利实现了预先的设想。

中国人民通过艰苦奋斗、自强不息、克服各种困难，终于研制出第一颗原子弹。新中国第一颗原子弹的成功研制，打破了帝国主义对我国的核讹诈，增强了国家的国防力量。同时原子弹的爆炸成功，代表了中国科学技术的新水平，有力地打破了超级大国的核垄断，提高了中国的国际地位。正是有了这些先辈们义无反顾的献身与付出，在 60 多年前，中国第一颗原子弹才得以横空出世。也正是在前辈们不畏艰难、勇于牺牲的精神鼓舞和激励下，一代又一代中国人实现了一次又一次的突破。60 多年后的今天，东风快递使命必达、航母舰队依次入列、神舟飞船遨游太空……只要秉持先辈们的这种精神，相信未来我们定将创造出更多中国奇迹。

致敬先辈！

【课后练习】

一、选择题

1. 一个 C 程序的执行是从()。
 - A. 本程序的 main 函数开始，到 main 函数结束
 - B. 本程序文件的第一个函数开始，到本程序文件的最后一个函数结束
 - C. 本程序文件的第一个函数开始，到本程序的 main 函数结束
 - D. 本程序的 main 函数开始，到本程序文件的最后一个函数结束
2. 以下叙述正确的是()。
 - A. 在对一个 C 程序进行编译的过程中，可发现注释中的拼写错误
 - B. 在 C 程序中，main 函数必须位于程序的最前面
 - C. C 语言本身没有输入输出语句
 - D. C 程序的每行中只能写一条语句
3. 一个 C 语言程序是由()。
 - A. 一个主程序和若干个子程序组成
 - B. 函数组成
 - C. 若干过程组成
 - D. 若干子程序组成
4. 以下叙述中错误的是()。
 - A. C 语言的可执行程序是由一系列机器指令构成的
 - B. 用 C 语言编写的源程序不能直接在计算机上运行

- C. 通过编译得到的二进制目标程序需要链接才可以运行
 D. 在没有安装 C 语言集成开发环境的机器上不能运行 C 源程序生成的 .exe 文件
5. 以下叙述不正确的是()。
- A. 一个 C 源程序必须包含一个 main 函数
 B. 一个 C 源程序可由一个或多个函数组成
 C. C 程序的基本组成单位是函数
 D. 在 C 程序中, 注释说明只能位于一条语句的后面
6. 标准 C 语言程序的源文件名的后缀为()。
- A. .c
 B. .cpp
 C. .obj
 D. .exe

二、填空题

1. 计算机程序设计语言的发展, 经历了从 ____、____ 到 ____ 的历程。
2. 计算机能唯一识别的语言是 ____。
3. C 语言程序是由 ____ 构成的。
4. 每个 C 语言程序中有且只有一个 ____ 函数, 它是程序的入口和出口。
5. 引用 C 语言标准库函数, 一般要用 ____ 预处理命令将其头文件包含进来。

三、程序分析题

1. 分析图 1.27 所示流程图, 指出该算法所完成的功能。

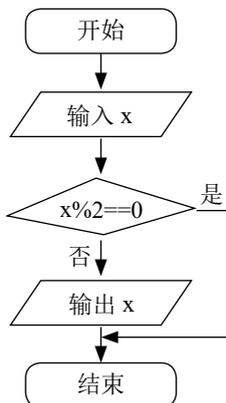


图 1.27 流程图

2. 设计从 a、b、c 三个数中找出一个最大数的算法, 用流程图表示该算法。

四、程序设计题

1. 编写一个 C 语言程序，要求输入两个整数 a 和 b ，交换 a 和 b 的数值后，在屏幕上输出交换结果。

2. 编写一个 C 语言程序，要求运行时输出下面图形。

```
*****
```

```
    Hello AI World!
```

```
*****
```

3. 同学们进入大学学习，一定要做出合理规划。大家想预测一下四年后的你是什么样的吗？我们可以用程序来计算：假设同学们进入大学是站在同一起跑线（记为 1.0），一天比一天进步或退步一点点（0.01），大学四年后的你到底有多强大？

➡ 提示：实数定义可用 `double`， x^y 可以使用 `pow(x,y)` 函数，该函数在 `math.h` 头文件中声明。