

第 1 章 C 语言程序设计概论

学习目标

- ⌘ 了解程序设计语言的基本知识。
- ⌘ 了解 C 语言程序的基本构成。
- ⌘ 了解计算机求解问题的步骤，理解算法的概念和程序设计思想和方法。
- ⌘ 了解 C 语言的发展和特点。
- ⌘ 了解 Visual C++ 2010 集成开发环境的使用方法。

“为什么要学习 C 语言？”这个问题被很多同学问起。也许我们会得到各种各样的答案，但 C 语言对锻炼我们的思维能力是毋庸置疑的。本章从自然语言出发，由浅入深为读者介绍程序设计语言、简单的 C 语言程序结构、实现问题求解的过程、C 语言的发展历史及特点，在学习过程中要求了解 C 语言程序的基本组成、计算机求解问题的步骤及算法和程序的概念，对 C 语言程序设计有一个初步认识和了解。

1.1 问题提出：程序设计概述

1. What

“什么是程序设计语言？”这个问题即将开始学习的时候，我们还是很疑惑。程序设计语言简称编程语言，是人们用计算机解决问题的方法的具体体现，是人与计算机进行交流和通信的语言。人们使用程序设计语言进行程序设计，为计算机编写规则，让计算机按自己的意愿自动处理数据，因此程序设计语言提供了一种表达数据和处理数据的功能。

C 语言是典型的过程式和结构化程序设计语言，是非常受欢迎的一种高级语言，应用广泛。C 语言既有诸如 PASCAL、FORTRAN 等高级语言的特点，又具有汇编语言中的位、地址、寄存器等概念，拥有其他许多高级语言所没有的底层操作能力，既可以应用在操作系统和应用程序层面，又可以用在需要对硬件进行操作的环境。因此，有人把 C 语言看成唯一的介于高级语言和低级语言之间的语言。

2. Why

当今社会，大数据、云计算、物联网等概念随处可见，人工智能、虚拟现实等应用



微视频

逐步普及，信息技术的应用已经并正在继续改变着人们的工作和生活，这些新技术和应用最重要的核心是程序。机器人、火箭升天、航空母舰、高科技武器、智慧农业、智能电网、工业 4.0、智能汽车和高铁等都是在程序的控制下才能工作的。

如今，最基本的工作、生活、学习、娱乐活动也离不开程序。日常生活中使用的网络电视、冰箱、微波炉、洗衣机等智能家用电器，以及网上点餐、约车、购物、水、电、煤缴费、转账、理财等互联网生活方式，微信、微博、QQ 等社交软件，没有程序都不可能实现。工作中，人们需要运行程序完成网上办公、视频会议、E-mail 收发、财务等管理功能。学习中要依赖程序完成选课、网络作业的提交、查找和下载学习资料、在线讨论、观看视频。生活中，视频点播、电视回放、网络小说、订票、订座、网络游戏等离不开程序。

我们生活在一个程序控制的年代。互联网、智能移动设备、云计算、大数据等的共同基础、共同指挥官就是程序。程序改变了人类的生活方式，推动了社会的发展。程序像空气和水一样，无处不在。

可能有人会说：我学的不是计算机专业，将来也不可能从事程序员的职业，我需要学会编程吗？殊不知，今天的编程不是一个狭义的概念，不仅包括懂不懂编程知识，也包括能不能以编程思维考虑问题等，学会编程可以更从容地应对现代社会的各种问题。人人都应该了解程序，懂程序，会编程，需要知道程序是什么、程序能改变世界依靠的是什么、程序是如何工作的，等等。

苹果公司创始人乔布斯说：我觉得每个人都应该学习一门编程语言。学习编程教你如何思考，就像学习法律一样，学法律并不一定要为了做律师，但法律教你一种思考方式，学习编程也是一样。我把计算机科学看成是基础教育，每个人都应该花一年的时间学习编程。为了号召全体民众学习编程，2016 年 2 月，美国政府专门投资 40 亿美元，推出了“全民计算机科学行动计划”，要求全体民众，特别是从幼儿园到大学的学生，都要学习编程。

学习程序设计是为了学习利用计算机科学的知识进行问题求解的方法，是训练如何设计出能解决问题的计算机系统的实践能力，是学习抽象形式较丰富的一种新思维方式。

在“互联网+”时代，信息技术已改变了人们的工作和生活方式，同时需要人们与现代信息技术相适应的思维方式，称为计算思维。编程思维是计算思维的重要内容，学习程序设计的更重要的任务就是训练并学会使用编程思维。

3. When & Where

“C 语言能用在什么时候、什么地方？”C 语言目前主要用于以下 4 方面：① 编写操作系统的不二之选，它为操作系统而生，几乎所有的操作系统底层都是由 C 或者 C++ 语言编写而成的，如 Windows、UNIX、Linux、Android 等，凡涉及系统底层的功能实现都离不开 C 和 C++ 语言；② 对程序运行效率有苛求的地方，如现在流行的云计算、物联网、智能终端的底层实现仍然是 C 语言的天下；③ 应用在网络安全领域，C 语言、脚本语言、UNIX 至今仍被认为是黑客必须苦练的三大技艺，C 语言被广泛应用于网络安全的方方面面；④ 程序设计的入门级语言，C 语言肩负着传播计算思维的责任，通过学习 C 语言程序设计，可以了解抽象、递归、复用、折中等计算思维，能在各行各业中更有效地利用计算机工具解决复杂问题。

4. How

“如何学习 C 语言？”C 语言的学习主要在于对数据表达、流程控制和算法等方面的理解及掌握。语法是次要的，更重要的是学习程序设计的基本思想和方法。C 语言学习可以按照“阅读参考书、阅读代码、编写调试实际程序、上网参与讨论、研究综合性应用”的顺序进行。初学者刚开始用 C 语言进行程序设计时可能感到无所适从，可以先模仿教材中的程序，理解并修改程序，多读多写多实践，循序渐进，直到掌握利用 C 语言解决复杂实际问题的能力。

本章将从自然语言出发，由浅入深介绍程序设计语言、简单的 C 程序结构、计算机求解问题的方法和步骤、C 语言的发展历史及特点，带领读者步入 C 程序设计的大门。

1.2 程序设计语言

1.2.1 自然语言和计算机语言

自然语言是人类在自身发展过程中形成的语言，是一种自然地随着文化而演化的语言，如汉语、英语、日语等。人类有别于一般动物，一个重要特征就在于人类创造并使用语言及作为其载体的文字。

自然语言是人类交流和思维的主要工具，是人与人之间传递信息的媒介，但是计算机目前不能识别、理解和执行人类的自然语言。人们要与计算机交流信息，使计算机按人的意图工作，必须解决人与计算机之间的“语言”问题。用于人与计算机之间通信的语言就是计算机语言（Computer Language）。

与自然语言相比，计算机语言具有如下特点：① 严格定义，有严格的语法；② 语义上无二义性；③ 比自然语言要精简；④ 是人能掌握和书写，计算机可识别和理解的。总之，计算机语言是人与计算机之间传递信息的媒介，是人用来表达需求并控制计算机执行这种需求的专门语言。根据分工的不同，计算机语言大致可以分为以下几种：

- ❖ 形式化需求规格语言，用于严格地、无二义地表达计算机用户需求的计算机语言，如 Z 语言、VDM 语言。
- ❖ 软件设计语言，用于表达软件设计策略、设计结构和算法的计算机语言，如统一建模语言 UML、软件体系结构设计语言 ACME。
- ❖ 程序设计语言，通常简称编程语言，让程序员能够准确地定义计算机所需要使用的的数据，并精确地定义在不同情况下计算机所应当采取的行动。典型的如 C 语言、Java 语言等。程序设计语言是最重要的计算机语言。
- ❖ 其他计算机语言，如用于表达网页内容的 HTML、用于计算机数据交换的 XML、用于数据库操作的 SQL 等。

计算机语言的发展也是一个不断演化的过程，其根本的推动力就是抽象机制更高的要求，以及对程序设计思想的更好的支持，具体地说，就是把机器能够理解的语言提升到也能够很好地模仿人类思考问题的形式。例如，程序设计语言的演化从最初的机器语言到汇编语言，再到各种结构化高级语言，最后到支持面向对象技术的面向对象语言。

1.2.2 程序设计语言介绍

程序设计语言（Programming Language）是最重要的计算机语言，是用于书写计算机程序的语言，通常简称编程语言。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。

一般来说，程序设计语言的基本成分不外乎 4 种。① 数据成分，用于描述程序中所涉及的数据；② 运算成分，用于描述程序中所包含的运算；③ 控制成分，用于表达程序中的控制构造；④ 传输成分，用于表达程序中数据的传输。

经过 50 多年的发展，目前已开发出上千种程序设计语言，如 FORTAN、COBOL、ALGOL、C、PASCAL、Java 等。按语言级别，程序设计语言有低级语言和高级语言之分。低级语言包括字位码、机器语言和汇编语言，其特点是与特定的机器有关，功效高，但使用复杂、烦琐、费时、易出差错。其中，字位码是计算机唯一可直接理解的语言，但由于它是一连串的字位，复杂、烦琐、冗长，几乎无人直接使用。机器语言是表示成数码形式的机器基本指令集，或者是操作码经过符号化的基本指令集。汇编语言是机器语言中地址部分符号化的结果，或进一步包括宏构造。

高级语言的表示方法要比低级语言更接近于待解问题的表示方法，其特点是在一定程度上与具体机器无关，易学、易用、易维护。当高级语言程序翻译成相应的低级语言程序时，一般来说，一个高级语言程序单位要对应多条机器指令，相应的编译程序所产生的目标程序往往功效较低。

程序设计语言一般有过程式语言和非过程式语言之分。过程式语言的主要特征是用户可以指明一系列可顺序执行的运算，以表示相应的计算过程。例如，FORTRAN、COBOL、ALGOL60、C 等都是过程式语言。非过程式语言所描述的则是事物间的某种关系，没有动态执行次序的概念，如 Prolog 语言是逻辑式程序设计语言，C++ 和 Java 则是面向对象的程序设计语言。

1.3 初识 C 语言程序

为了让读者对 C 语言有一个感性认识，先看两个 C 语言程序。

【例 1-1】 在屏幕上输出 “Hello World!”。

程序源代码如下：

```
1  #include <stdio.h>           /*编译预处理命令*/
2  int main(void)               /*定义主函数*/
3  {                             /*花括号：括起函数体*/
4      printf("Hello World!\n"); /*调用 printf()函数，输出文字至显示器，\n 为换行符*/
5      return 0;                /*函数返回值为整数 0，“;”表示一个语句的结束*/
6  }
```

运行结果如下：

```
Hello World!
```

【例 1-2】 求两个数中的最大值。

程序源代码如下：

```
1  #include<stdio.h>
2  int max(int a,int b)           /*定义 max 函数*/
3  {
4      if(a>b) return a;
5      else return b;           /*把结果返回主调函数*/
6  }
7  int main(void)                /*主函数*/
8  {
9      int x, y, z;              /*定义变量*/
10     printf("input two numbers:\n");
11     scanf("%d%d",&x,&y);        /*输入变量 x 和 y 的值*/
12     z=max(x,y);                /*调用 max()函数*/
13     printf("maxmum=%d",z);     /*输出结果至显示器*/
14     return 0;
15 }
```

运行结果如下：

```
输入：5,8
```

```
输出：maxmum=8
```

这里并不要求初学者能完全理解上述程序，希望读者先对 C 程序有个初步的印象，上述程序中的内容将会在以后各章中逐步介绍。从上可知，C 源程序的结构特点如下：

① C 程序全部由一个一个的函数构成，但必须包含且只能包含一个主函数（main()函数），其他函数可被主函数调用或相互调用。被调函数可以是 C 函数库中的函数，如上例中的 printf()函数和 scanf()函数，也可以是用户自己编的函数，如上面的 max()函数。

② 一个函数由函数头和函数体两部分组成。函数头定义了函数名、函数类型、函数形式参数名及其类型，如“int max(int a,int b)”。函数体即函数头下“{ }”内的部分，函数体定义了该函数所要实现的功能。函数体一般包括：

- ❖ 变量定义部分，如例 1-2 中 main 函数中的“int x, y, z;”。

- ❖ 执行部分，由若干语句组成，每条语句都以“;”结束。

③ 一个 C 程序中函数出现的次序可以是任意的，但总是从 main()函数开始执行。

④ C 语言程序书写格式比较自由，一行可有多个语句，一个语句也可写成多行（但不能将一个单词分开）。每条语句都必须以“;”结尾，但预处理命令、函数头和“{ }”之后不能加“;”。

⑤ “/*”与“*/”之间的内容构成 C 语言程序的注释部分，“/*”与“*/”之间的内容可以是一行，也可以是多行。注释部分不参与程序的编译和执行，只是起说明作用，增加程序的可读性。

从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时应遵循以下规则：

- ❖ 一个说明或一个语句占一行。

- ❖ “{ }”括起来的部分通常表示了程序的某一层结构。“{ }”一般与该结构语句的第一个字母对齐，并单独占一行。

- ❖ 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，以便看起来更清晰，增加程序的可读性。
- 在编程时应力求遵循这些规则，以养成良好的编程风格。

1.4 实现问题求解的过程

1.4.1 计算机求解问题步骤

我们做任何事情都要遵循一定的步骤，如做菜的过程如图 1-1 所示。同理，计算机求解问题也要遵循一定的步骤，如图 1-2 所示。



图 1-1 做菜的过程

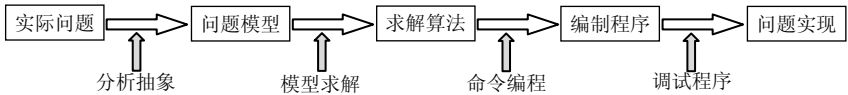


图 1-2 计算机求解问题的步骤

1.4.2 问题分析和建模

下面通过一个具体的例子来说明计算机求解问题的步骤。

【例 1-3】 问题：求 1~100 间所有偶数的和。

本问题求在一定范围内（1~100）、满足一定条件（偶数）的若干整数的和，是一个求累加和的问题。

这类问题的基本求解方法是：设置一个变量（如 `sum`），将其初值置为 0，然后在指定的范围（1~100）内寻找满足条件（偶数）的整数，将它们累加到 `sum` 中。为了处理方便，将正在查找的整数也用一个变量（如 `i`）表示。所以，累加过程的 C 语言语句为：

```
sum = sum + i;
```

表示把 `sum` 的值加上 `i` 后，重新赋值给 `sum`。这个累加过程要反复做，需用程序设计语言的循环语句来实现，过程如下：① 判别 `i` 是不是偶数，用分支控制语句来实现；② 对循环次数进行控制，通过 `i` 值的变化来控制，即 `i` 的初值设为 1，每循环一次加 1，一直加到 100 为止。

基于上述解决问题的思路，可以逐步明确解决问题的步骤，即确定解决问题的算法。

1.4.3 算法设计

1. 算法的概念

广义上说，算法是为了解决一个问题而采取的方法和步骤。狭义上说，算法是指在有限步内解决一个具体问题而规定的意义明确的解题步骤的有限集合。概括地说，算法是

指解题方案的准确而完整的描述。从程序的角度来说，也可以说算法是一个有限条指令的集合，这些指令确定了解决某一特定类型问题的运算序列。

算法应该具有以下 5 个重要特征。

- ❖ 有穷性：算法必须保证执行有穷步之后结束。
- ❖ 确定性：算法的每条指令必须具有确切的含义，读者理解和程序实现时不会产生歧义性；也就是说，对相同的输入只能得到相同的输出。
- ❖ 可行性：算法应该是能执行的，如“计算 $X/0$ ”是不允许的。
- ❖ 输入：算法有零个或多个输入。
- ❖ 输出：算法具有一个或多个输出。

2. 算法的描述方法

算法的描述方法可以归纳为以下 4 种。

- ❖ 自然语言，易写易读，但存在表达冗长和语义多义性的缺陷。
- ❖ 图形，如 N-S 图、程序流程图，具有简洁、直观、准确的特性。
- ❖ 算法语言，即计算机语言、程序设计语言、伪代码。
- ❖ 形式语言，用数学的方法，可以避免自然语言的二义性。

用各种算法描述方法所描述的同一算法，该算法的功用是一样的，允许在算法的描述和实现方法上有所不同。

本书使用程序流程图来描述算法。程序流程图又称为程序框图，它定义了一些基本的图框，并用带箭头的直线（称为流程线）把各种图框连接起来，箭头表示处理的流向。常用的图框如表 1-1 所示。

表 1-1 常用框图

形 状	名 称	含 义
	起止框	表示一个算法的开始与结束
	数据框	框中指出输入或输出的数据内容
	处理框	框中指出所进行的处理
	判断框	逻辑判断或检查
	箭头线	流程的路线
	圆圈	连接点

上述例 1-3 求 1~100 间所有偶数和的程序流程图如图 1-3 所示。

3. 三种基本的程序控制结构

图 1-3 中包含顺序结构、分支结构和循环结构三种基本的程序控制结构（如图 1-4 所示）。从上至下依次执行一系列操作的程序称为顺序结构，根据不同条件选择所要执行的操作的程序称为选择结构，可以多次重复执行某一部分处理的程序称为循环结构。

三种基本的控制结构具有如下共同特点：

- ❖ 只有一个入口，一个出口。
- ❖ 结构内的每一部分都有机会被执行到。
- ❖ 结构内不存在死循环。

由基本结构所构成的算法称为结构化算法。

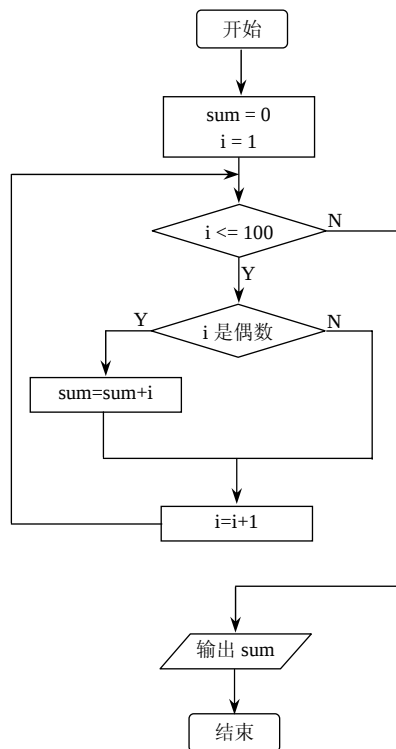


图 1-3 程序流程图示例

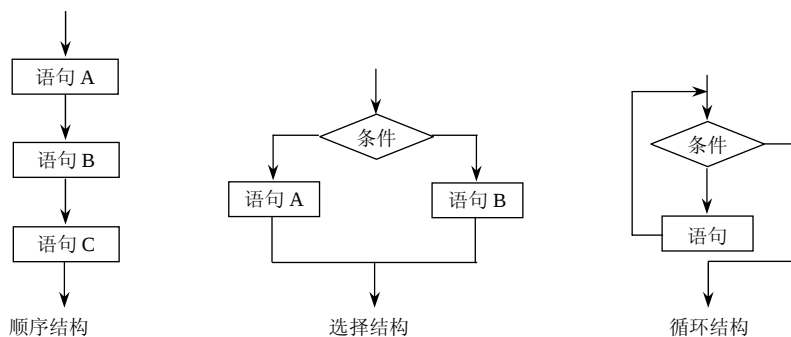


图 1-4 三种基本的程序控制结构

结构化算法的特点：不存在无规律的转向，只能自上而下顺序执行，只在各基本结构内存在分支或跳转。

非结构化算法：流程中存在无规律的跳转，不是自上而下顺序执行。

1.4.4 程序设计

计算机程序或者软件程序（通常简称程序）是指使用某种程序设计语言编写的一组指示计算机每一步动作的指令，如例 1-1 和例 1-2 就是用 C 语言编写的计算机程序。在《计算机软件保护条例》中的定义为：为了得到某种结果而可以由计算机等具有信息处理能力的装置执行的代码化指令序列，或者可被自动转换成代码化指令序列的符号化指令序

列或者符号化语句序列。

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。由于程序是软件的本体，软件的质量主要通过程序的质量来体现，在软件研究中，程序设计的工作非常重要，内容涉及有关的基本概念、工具、方法及方法学等。程序设计往往以某种程序设计语言为工具，给出这种语言下的程序。程序设计过程应当包括分析、设计、编码、测试、排错等阶段。专业的程序设计人员常被称为程序员。

按照结构性质，程序设计有结构化程序设计与非结构化程序设计之分，前者是指具有结构性的程序设计方法和过程（如 C 语言程序设计），具有由基本结构（即顺序结构、选择结构和循环结构三种）构成复杂结构的层次性，后者反之。按照程序设计的成分性质，程序设计有顺序程序设计、并发程序设计、并程序序设计、分布式程序设计之分。按照程序设计风格，程序设计有逻辑式程序设计、函数式程序设计、对象式程序设计之分。

一个程序应包括以下内容：① 对数据的描述，指定数据的类型和组织形式，即数据结构；② 对操作的描述，即操作步骤（算法）；③ 采用某种程序设计方法进行设计。如结构化程序设计方法与非结构化程序设计方法等；④ 采用某一种计算机语言来表示，程序设计语言提供了一种表达数据和处理数据的功能，要求程序员按照语言的规范编程，即“程序 = 算法+数据结构+程序设计方法+语言工具和环境”。在这 4 方面中，算法是灵魂，数据结构是加工对象，语言是工具，编程需采用合适的方法。算法是解决“做什么”和“怎么做”的问题。

上述例 1-3 求 1~100 所有偶数的和的程序如下：

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int i, sum = 0;
5      for(i = 1; i <= 100; i++)
6          if(i%2 == 0)
7              sum = sum + i;
8      printf("%d\n", sum);
9      return 0;
10 }
```

运行结果如下：

2550

其中：

- ❖ 对数据的描述：1~100 间的数、和等数据采用整型数据结构描述。
- ❖ 对操作的描述：即算法，见图 1-3。
- ❖ 采用的程序设计方法：面向过程的结构化程序设计方法。
- ❖ 采用的计算机语言：C 语言。

1.5 C 语言的发展历史及特点

1.5.1 C 语言的发展

汇编语言可以直接对硬件进行操作，如对内存地址的操作、位（bit）操作等。早期的

操作系统等系统软件主要用汇编语言编写，如 UNIX 操作系统。由于汇编语言依赖于计算机硬件，程序的可读性和可移植性比较差。为了提高可读性和可移植性，最好改用高级语言，而一般高级语言难以实现汇编语言的某些功能，人们设想能否找到一种既具有一般高级语言特性，又具有低级语言特性的语言，集它们的优点于一身，于是 C 语言就在这种情况下应运而生了，之后成为国际上广泛流行的计算机高级语言。C 语言适合作为系统描述语言，即用来写系统软件，也可用来编写应用软件。

C 语言是在 B 语言的基础上发展起来的，它的根源可以追溯到 ALGOL 60。1960 年出现的 ALGOL 60 是一种面向问题的高级语言，离硬件比较远，不宜用来编写系统程序。1963 年，英国的剑桥大学推出了 CPL (Combined Programming Language)，在 ALGOL 60 的基础上更接近硬件，但规模比较大，难以实现。1967 年，英国剑桥大学的 Martin Richards 简化了 CPL，推出了 BCPL (Basic Combined Programming Language)。1970 年，美国贝尔实验室的 Ken Thompson 以 BCPL 为基础，进一步简化，使得 BCPL 能挤压在 8 KB 内存中运行，这个很简单的而且很接近硬件的语言就是 B 语言（取 BCPL 的第一个字母）。第一个 UNIX 操作系统使用 B 语言编写，并在 DEC PDP-7 上实现。1971 年，PDP-11/20 上实现了 B 语言，并写了 UNIX 操作系统。但 B 语言过于简单，功能有限，并且和 BCPL 一样都是“无类型”的语言。

1972 年至 1973 年间，贝尔实验室的 D.M. Ritchie 在 B 语言的基础上设计了 C 语言（取 BCPL 的第二个字母）。C 语言既保持了 BCPL 和 B 语言的优点（精练、接近硬件），又克服了它们的缺点（过于简单、数据无类型等）。最初的 C 语言只是为描述和实现 UNIX 操作系统提供一种工具语言而设计的。1973 年，K. Thompson 和 D.M. Ritchie 两人合作，把 90% 以上的 UNIX 用 C 语言改写，即 UNIX 第 5 版。原来的 UNIX 操作系统是 1969 年由美国的贝尔实验室的 K. Thompson 和 D.M. Ritchie 开发成功的，用汇编语言编写。这样，UNIX 使分散的计算系统之间的大规模联网以及互联网成为可能。

后来，C 语言多次被改进，但主要在贝尔实验室内部使用，直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们普遍注意。1977 年出现了不依赖于具体机器的可移植 C 语言编译程序，使 C 移植到其他机器时所需做的工作大大简化，这也推动了 UNIX 操作系统迅速在各种机器上实现。随着 UNIX 的日益广泛使用，C 语言也迅速得到推广，1978 年以后，C 语言已先后移植到大、中、小、微型机上，如 IBM System/370、Honeywell 6000 和 Interdata 8/32，已独立于 UNIX 和 PDP 了。现在 C 语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

1983 年，美国国家标准化协会 (ANSI) X3J11 委员会根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。ANSI C 比原来的标准 C 有了很大的发展。1987 年，ANSI 又公布了新标准——87 ANSI C，目前流行的 C 编译系统都是以它为基础的。广泛流行的各种版本 C 语言编译系统虽然基本部分是相同的，但也有一些不同，在微型机上使用的有 Visual C、Borland Turbo C、Dev-C++、Quick C 和 AT&T C 等，它们的不同版本略有差异。Java、C++、C#都是以 C 语言为基础发展起来的。本书以 87 ANSI C 标准展开讲解。

1.5.2 C 语言的特点

① 简洁紧凑、灵活方便。C 语言只有 32 个关键字、9 种控制语句，程序书写自由，主要用小写字母表示，把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作，这三者是计算机最基本的工作单元。

② 运算符丰富。C 语言的运算符包含的范围很广泛，有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 语言的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符，可以实现在其他高级语言中难以实现的运算。

③ 数据结构丰富。C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等，能用来实现各种复杂的数据类型的运算，并引入了指针，使程序效率更高。

④ C 语言是结构化语言。C 语言是以函数形式提供给用户的，这些函数可方便地被调用，并具有多种循环、条件语句控制程序流向，从而使程序完全结构化。

⑤ C 语言语法限制不太严格、程序设计自由度大。一般的高级语言语法检查比较严，能够检查出几乎所有的语法错误，而 C 语言允许程序编写者有较大的自由度。

⑥ C 语言允许直接访问物理地址，可以直接对硬件进行操作，因此 C 语言既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样对位、字节和地址进行操作。

⑦ C 语言程序生成代码质量高，程序执行效率高，一般只比汇编程序生成的目标代码效率低 10%~20%。

⑧ C 语言适用范围大，可移植性好。其突出的优点就是适合多种操作系统，如 DOS、Windows、UNIX，也适用于多种机型。

当然，C 语言也有自身的不足，比如：C 语言的语法限制不太严格，对变量的类型约束不严格，影响程序的安全性，对数组下标越界不作检查等。从应用的角度，C 语言比其他高级语言较难掌握。

C 语言既有高级语言的特点，又具有汇编语言的特点，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。

1.5.3 C 语言的基本符号

任何程序设计语言如同自然语言一样，都具有自己一套对字符、单词及一些特定符号的使用规定，也有对语句、语法等方面的使用规则。C 语言所涉及的规定很多，其中主要有基本字符集、标识符、关键字、语句和标准库函数等。这些规定构成了 C 程序的最小语法单位。例如，例 1-2 中的 a、b、x、y、z 是标识符，int、if 是关键字，return a 是语句，scanf 和 printf 是标准库函数等，这些都是由 C 语言规定的基本字符组成的。

1. 基本字符集

C 程序是 C 语言基本字符构成的一个序列。C 语言的基本字符集包括：

① 数字字符：0、1、2、3、4、5、6、7、8、9

② 拉丁字母：A、B、…、Z、a、b、…、z（注意：字母的大小写是可区分的，如

abc 与 ABC 是不同的)

③ 运算符: +、-、*、/、%、=、<、>、<=、>=、!=、==、<<、>>、&、|、&&、||、^、~、(、)、[、]、->、.、!、?、:、,、;

④ 特殊符号和不可显示字符: _ (连字符或下划线)、空格、换行、制表符。

对初学者来说, 书写程序要从一开始就养成良好的习惯, 力求字符准确、工整、清晰, 尤其要注意区分一些字形上容易混淆的字符, 避免给程序的阅读、录入和调试工作带来不必要的麻烦。

2. 标识符

在程序中有许多需要命名的对象, 以便在程序的其他地方使用。如何表示在一些不同地方使用的同一个对象? 最基本的方式就是为对象命名, 通过名字在程序中建立定义与使用的关系, 建立不同使用之间的关系。为此, 每种程序语言都规定了在程序里描述名字的规则, 这些名字包括: 变量名、常数名、数组名、函数名、文件名、类型名等, 通常被统称为“标识符”。

C 语言规定, 标识符由字母、数字或下划线(_)组成, 它的第一个字符必须是字母或下划线。这里要说明的是, 为了标识符构造和阅读的方便, C 语言把下划线作为一个特殊字符使用, 它可以出现在标识符字符序列里的任何地方, 特别是可以作为标识符的第一个字符出现。C 语言还规定, 标识符中同一个字母的大写与小写被看成不同的字符, 这样, a 和 A、AB 和 Ab 是互不相同的标识符。表 1-2 是合法的和非法的两组 C 标识符。

表 1-2 标识符示例

合法的 C 标识符	不合法的 C 标识符	说 明
call_name	call...name	非字母数字或下划线组成的字符序列
test39	39test	非字母或下划线开头的字符序列
_string1	-string1	非字母或下划线开头的字符序列

在 C 程序中, 标识符的使用很多, 使用时要注意语言规则。在例 1-2 的程序中, a、b、x 等是变量名, main 和 max 是函数名, 它们都是符合 C 语言规定的标识符。ANSI C 标准规定, 标识符的长度可达 31 个字符, 但一般系统使用的标识符的有效长度不超过 8 个字符。

3. 关键字

C 语言有一些具有特定含义的关键字, 用作专用的定义符。这些特定的关键字不允许用户作为自定义的标识符使用。C 语言的关键字绝大多数是由小写字母构成的字符序列, 它们是: auto、break、case、char、const、continue、default、do、double、else、enum、extern、float、for、goto、if、int、long、registe、return、short、signed、sizeof、static、struct、switch、typedef、union、unsigned、void、volatile、while。

4. 语句

语句是组成程序的基本单位, 能完成特定操作, 语句的有机组合能实现指定的计算处理功能。所有程序设计语言都提供了满足编写程序要求的一系列语句, 都有确定的形式和功能。C 语言中的语句有以下几类。

❖ 选择语句: if, switch。

- ❖ 循环语句: for, while, do-while。
- ❖ 转移语句: break, continue, return, goto。
- ❖ 表达式语句。
- ❖ 复合语句。
- ❖ 空语句。

这些语句的形式和使用见后续相关章节。

5. 标准库函数

标准库函数不是 C 语言本身的组成部分, 是由 C 编译系统提供的一些非常有用的功能函数, 如 C 语言没有输入/输出语句, 也没有直接处理字符串的语句, 而一般的 C 编译系统都提供了完成这些功能的函数, 称为标准库函数。常用的有数学函数、字符函数和字符串函数、输入/输出函数等几大类。

在 C 语言处理系统中, 标准库函数存放在不同的头文件 (也称标题文件) 中, 如输入/输出一个字符的函数 `getchar()` 和 `putchar()`、有格式的输入/输出函数 `printf()` 和 `scanf()` 等就存放在标准输入/输出头文件 `stdio.h` 中, 求绝对值函数和三角函数等数学函数存放在标准输入/输出头文件 `math.h` 中。这些头文件中存放了关于这些函数的说明、类型和宏定义, 而对应的子程序则存放在运行库 (*.lib) 中。使用时只要把头文件包含在用户程序中, 就可以直接调用相应的库函数了, 即在程序开始部分用如下形式:

<code>#include <头文件名></code>	或	<code>#include "头文件名"</code>
------------------------------------	---	------------------------------

标准库函数是语言处理系统中一种重要的软件资源, 在程序设计中充分利用这些函数, 常常会收到事半功倍的效果。所以, 读者在学习 C 语言本身的同时, 应逐步了解和掌握标准库中各种常用函数的功能和用法, 避免自行重复编制这些函数。

需要说明的是, 不同 C 编译系统提供的标准库函数在数量、种类、名称及使用上都有些差异。但就一般系统而言, 常用的标准函数基本上是相同的。附录 E 中列出了一些常用的标准库函数。

1.6 C 语言程序的开发环境

1.6.1 C 语言的一般上机步骤

我们对 C 语言源程序结构有了总体的认识, 那么如何在机器上运行 C 语言源程序呢? 任何高级语言源程序都要“翻译”成机器语言, 才能在机器上运行。“翻译”的方式有两种: 一种是解释方式, 即对源程序解释一句执行一句; 另一种是编译方式, 即通过编译系统先把源程序“翻译”成目标程序 (用机器代码组成的程序), 再经过连接装配后生成可执行文件, 最后执行可执行文件而得到结果。

C 语言是一种编译型的程序设计语言, 采用编译的方式将源程序翻译成目标程序 (机器代码)。运行一个 C 程序, 从输入源程序开始, 要经过编辑源程序文件 (*.c 或 *.cpp)、编译生成目标文件 (*.obj)、连接生成可执行文件 (*.exe) 和执行 4 步, 如图 1-5 所示。

在编译、连接时, C 语言编译系统往往会提供出错信息, 包括出错位置 (行号)、出错提示信息等。编程者可以根据这些信息, 找出相应错误所在并修改。有时系统提示了一

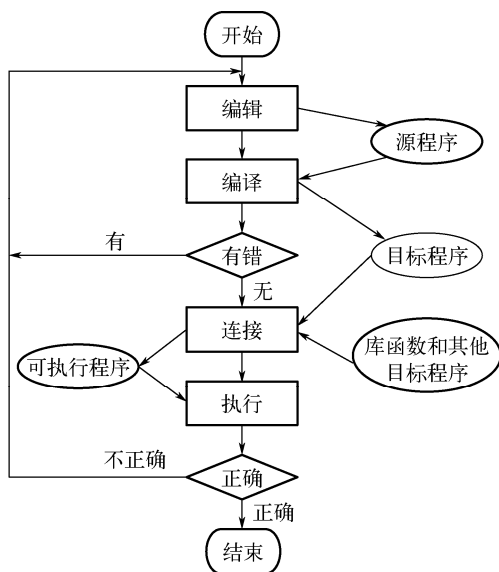


图 1-5 C 语言的运行步骤

大串错误信息，并不表示真的有这么多个错误，往往是因为程序中的一两个错误引起的，所以当纠正了几个错误后，应该重新编译连接一次，然后根据最新的出错信息继续纠正，这是程序调试的一个好方法。

有些程序通过了编译连接，并能够在计算机上运行，但得到的结果与预期的结果不一样，这类错误被称为逻辑错误。这类在程序执行过程中的错误往往难以改正。错误的原因一部分是程序书写错误带来的，如应该使用变量 x 的地方写成了变量 y ，虽然没有语法错误，但意思完全错了；另一部分可能是程序的算法不正确，解题思路不对，得到的结果和预期的结果不一样，如预期求两个整数的和，在程序中却写为两个整数的差，得到的结果肯定与预期的不一样。还有一些程序计算结果有时正确，有时不正确，如求一个整数除以 2 的商，如果将这个商定义为 `int` 型变量，那么在该整数为偶数时正确，奇数时就会错误，这些现象往往是编程时对各种情况考虑不周所致。

解决运行错误的首要步骤就是错误定位，即找到出错的位置和错误的原因，才能予以纠正。通常，我们需要先设法确定错误的大致位置，然后通过 C 语言提供的调试工具找出真正的错误。但需要注意，本书中大部分的程序在调试时，调试工具都能直接找到程序的错误，也有部分比较复杂的程序，当程序执行出错时，调试工具发现的错误未必一定是程序中的真正错误。在将来的实际软件开发中，这样的情况将会出现得更多，所以有经验的程序员往往都认为，寻找程序的错误不能只依靠计算机，也需要我们自己掌握好的方法，在实际的程序调试中积累丰富的经验。

1.6.2 Visual C++ 2010 集成开发环境

Visual C++2010 Express（学习版）是微软发布的免费版本，用户可以免费注册使用。Visual C++2010 是一个 Windows 环境下支持 C、C++和 C++/CLI 的 IDE（集成开发环境），支持 C++ 98 标准，是适合初学者学习 C 或 C++的开发工具。本书将在实验教程详细介绍 Visual C++2010 Express 的使用，在此只简单介绍如何使用 Visual C++2010 编辑调试 C 语

言程序。Visual C++2010 的主界面如图 1-6 所示。



图 1-6 Visual C++ 2010 的主界面

同其他集成 IDE 编辑器类似，在 Visual C++ 2010 Express 中运行 C 程序也需要经过编辑、保存、编译、运行等步骤。

1. 新建项目

在实际工作中，程序通常由多个文件构成。为了方便对文件的管理，Visual C++ 2010 把文件组织在一个项目中。在部分编译环境中，项目也被称为工程。相较以往版本，Visual C++ 2010 不再支持创建单独的源代码文件以完成编辑、编译、运行，单独创建的源代码文件必须附加到项目中方可进行编辑调试。项目创建的具体步骤如下：

<1> 在起始页面选择“新建项目”，或者通过菜单依次选择“文件”→“新建”→“项目”，也可以直接单击工具栏的“新建项目”按钮。

<2> 在出现的界面中选择左边目录中的“Visual C++”的“Win32”，在右边子窗口中选择“Win32 控制台应用程序”，然后在下面的“名称”栏中输入项目名（注意不是源代码文件名称），选择好保存位置，再单击“确定”按钮，进入“Win32 应用程序向导”。单击“下一步”按钮，进入“应用程序设置”界面（如图 1-7 所示），即可完成项目的创建。

<3> 勾选“附加选项”中的“空项目”，单击“完成”按钮，即可完成项目的创建，进入如图 1-8 所示的界面，窗口左边出现一个“解决方案资源管理器”。

2. 编辑源代码文件

Visual C++ 2010 支持新建文件，也支持把已经编辑好的源文件附加到项目中。

附加文件：如果已经编辑好程序源代码，则选择“添加”→“现有项”，以附加源代码文件。

新建文件：在图 1-8 所示界面的“解决方案资源管理器”下右击“源文件”，选择“添加”→“新建项”，在出现的“新添加项”界面中选择“C++文件”，然后输入源代码名称、选择保存位置，单击“添加”按钮，即可进入源代码编辑界面。



图 1-7 应用程序设置界面

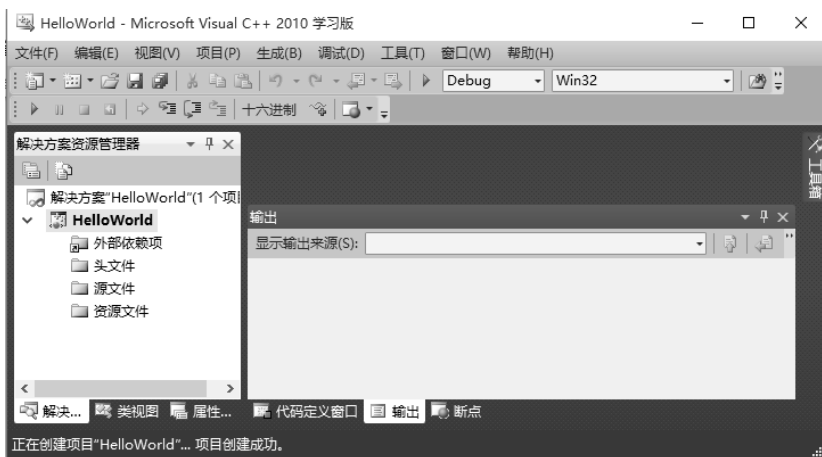


图 1-8 项目创建成功后的界面

注意：如果编写 C 语言代码，建议在名称后加后缀名“.c”；如果编写 C++ 代码，建议在名称后面加“.cpp”；扩展名也可以不加，如果不加，默认文件的扩展名是“.cpp”。

保存文件：在程序编辑窗口输入程序代码，单击工具栏中的“保存”按钮或依次选择菜单“文件”→“保存”，即完成了程序文件的保存。

3. 程序编译

程序代码的输入完成后，接着需要检查程序是否有错误。操作步骤为：依次选择菜单“生成”→“编译”（或按快捷键 Ctrl+F7），编译器将检查程序中是否有语法错误，如果有，将给出错误提示，错误的提示中包含行号，读者可以根据行号分析代码，改正错误。

4. 可执行文件生成（连接）

编译成功之后，选择菜单“生成”→“生成***”（***代表设定的项目名称）运行”，Visual C++ 2010 将在输出窗口显示项目生成可执行文件情况，如果无法生成可执行文件，则可能是程序中有逻辑错误，需要在编辑窗口对源代码进行修改。程序的显示结果就在窗口中。

5. 程序运行

可执行文件生成后，即可运行程序，以查看运行结果。同时 Visual C++ 2010 也支持不经过编译、连接步骤，而直接运行，此时如有错误，会直接在输出窗口中显示。

<1> 选择“调试”→“启动调试”，或单击工具栏中的“启动调试”按钮，即可查看到运行结果。**注意：**如果在程序结束前也没有暂停语句，就会出现运行窗口一闪而过的情况。解决方法如下，在程序源代码中增加头文件包含语句：

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

在程序结束前增加程序暂停语句：

```
system("pause");
```

<2> 按快捷键 Ctrl + F5，运行程序。

6. 程序调试

编译时可以检查出源代码中的语法错误，但是检查不出源代码中的逻辑错误。有时候程序虽然能够编译并运行，运行结果却不是预想的结果，那是因为源代码中存在有逻辑错误。

Visual C++ 2010 设置有“调试”菜单，包括“逐语句 F11”、“逐过程 F10”、“切换断点 F9”、“调试 F5”，使用这些命令可以完成程序的调试。

除 Visual C++ 2010 外，C 语言程序设计编辑和调试还可以使用 Dev-C++、CodeBlocks 等，这些调试环境将在实验指导中作详细介绍。本书后续章节的所有程序都以 Visual C++ 2010 作为编辑调试环境。

1.7 科学计算器和学生成绩管理系统

本书以“一基两能”为出发点，培养学生运用 C 语言解决实际问题的能力，“一基”是指 C 语言的基本语法知识和规则，“两能”是指算法设计能力和程序实现能力。因此，本书主要以科学计算器和学生成绩管理系统两个实际项目来讲解 C 语言的相关知识和应用，程序的实现在后续章节中逐步实现。

1. 科学计算器的功能及结构

简易计算器能实现以下功能：

- ❖ 数制转换。
- ❖ 加、减、乘、除、求幂、求模、求平方根运算。
- ❖ 比较数的大小。
- ❖ 求和、阶乘、求素数运算。
- ❖ 正弦、余弦函数计算。
- ❖ 指数、对数计算。
- ❖ 简单的四则混合运算（支持括号、优先级）。

简单计算器由 6 个模块组成，如图 1-9 所示。

- ❖ 主函数 void main(): 调用各主要模块，实现简单计算器的整体功能。
- ❖ 主界面函数 void ZhuJieMian(): 使用有关图形函数模拟出可视化的计算器外观界面，并提供用户输入窗口。

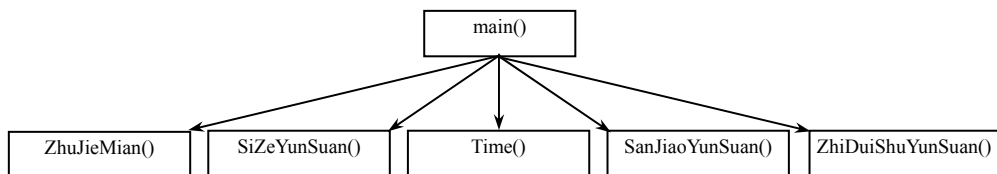


图 1-9 简单计算器模块

- ❖ 时钟函数 `int Time()`：在模拟的可视化计算器中显示当时的时间。
- ❖ 四则混合运算函数 `void SiZeHunHeYunSuan()`：实现简单的加、减、乘、除混合运算，并支持优先级和带括号的运算。
- ❖ 三角运算函数 `void SanJiaoYunSuan()`：实现基本的三角函数运算 $\sin(x)$ 和 $\cos(x)$ 。
- ❖ 指对数函数 `void ZhiDuiShuYunSuan()`：实现基本的指数和对数运算。

2. 学生成绩管理系统的功能及结构

学生成绩管理系统的功能如图 1-10 所示，主要由 7 个主要功能模块组成。

- ❖ 信息输入：输入学生成绩信息（包括学生学号、姓名、各门课程的成绩等）。
- ❖ 查询：输入学号或名字查询某学生各门课程的成绩及平均成绩。
- ❖ 信息插入：输入新的学生信息并插入到原来的列表中进行排序。
- ❖ 信息修改：输入学号，显示该学生的成绩信息，并在此基础上进行修改。
- ❖ 信息删除：输入学号，删除该学生的成绩信息。
- ❖ 排序：对学生数据按学号或者平均分进行排序，并显示排序结果。
- ❖ 信息输出：显示全部学生的成绩信息。

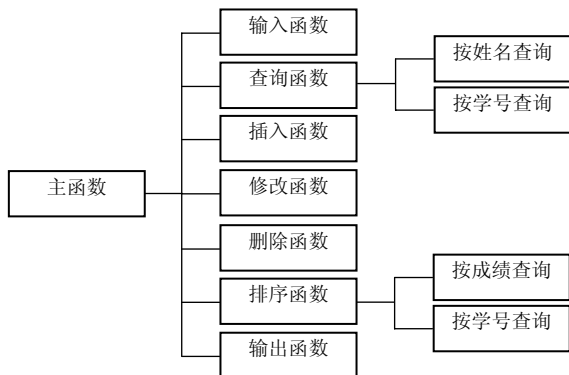


图 1-10 系统功能模块

学生成绩管理系统运行界面如图 1-11 和图 1-12 所示。

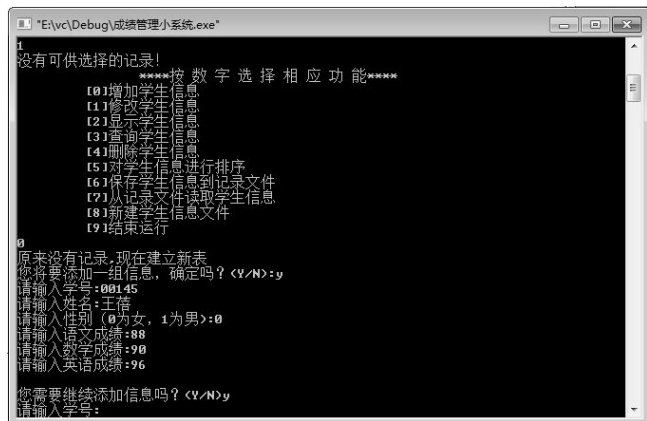


图 1-11 系统运行界面



图 1-12 系统源程序及运行界面

本章小结

本章介绍了程序设计语言的基础知识，首先从自然语言引入程序设计语言，并对程序设计语言作了介绍。然后以两个简单的 C 语言程序介绍了 C 程序的组成结构，以生活中的实际问题为例介绍了计算机求解问题的步骤，还讲解了 C 语言的发展历史和特点。在学习过程中，读者应了解 C 语言程序的基本组成、算法的概念、结构化程序设计的方法。最后从学习并理解 C 语言的语法和语义出发，讲述了 Visual C++ 2010 集成开发环境下 C 源程序的上机运行步骤和调试方法，使读者对 C 语言程序的上机使用有初步的认识和了解。

习 题 1

一、填空题

1. 当一个 C 语言程序只有一个函数时，这个函数的名称是_____。
2. 一个函数由_____和_____两部分组成。
3. C 语言程序的基本单位或者模块是_____。
4. 编写一个 C 程序，上机运行，要经过哪几个步骤_____。
5. 在 Visual C++ 2010 环境中，通过文字编辑建立的源程序文件的扩展名是_____；编译后生成目标程序文件，扩展名是_____；连接后生成可执行程序文件，扩展名是_____；运行得到结果。
6. C 语言程序的语句结束符是_____。

二、思考题

1. 查找资料，了解有哪些常见的程序设计语言，它们各自的特点和主要应用领域是什么？
2. 什么是算法？算法的特性是什么？
3. 请写出 C 语言的主要特点。

4. 一个已编好的 C 程序源文件 (program.c)，在计算机上运行，应该经历哪些步骤？
5. 编写一程序，显示下列文字信息：

```
This is My First Program!  
I am a student!
```

6. 用流程图描述在三个数中输出最大数的算法。