

第1章 C语言程序设计概述

计算机的工作过程就是执行各种程序的过程，而程序是用计算机程序设计语言编写的。C语言作为一种长盛不衰的高级程序设计语言，被广泛应用于操作系统和应用软件的开发。本章主要介绍计算机程序设计语言的相关概念、C语言程序的基本结构、C程序的运行过程、程序设算法及其描述方法、程序设计风格等内容。

1.1 计算机程序设计语言概述

计算机系统由硬件系统和软件系统两大部分组成。硬件系统是指构成计算机的各种机械部件和电子元件组成的设备和装置，是组成计算机系统的物质基础。软件系统是控制、管理计算机各硬件设备如何工作的所有程序文件和数据文件的总称，是计算机系统的头脑和灵魂。没有软件的计算机是一台“裸机”，什么也不能干，计算机的工作过程就是执行各种程序的过程。程序是指为完成特定功能而编写的指令的集合，这组指令依据既定的逻辑控制计算机的运行。所有程序都是用计算机语言编写的。

随着计算机技术的迅速发展，计算机程序设计语言经历了机器语言、汇编语言到高级语言的发展历程。程序设计方法也伴随着计算机硬件技术的提高而不断发展，可分为三个阶段，即面向计算机的程序设计、面向过程的程序设计和面向对象的程序设计。

高级语言使用一种接近人的自然语言和数学公式的形式编写程序。使用高级语言编写的程序与计算机的硬件结构及指令系统无关，具有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好的描述各种算法，而且易于学习掌握。近几十年来，为解决不同的实际问题出现了几百种高级语言，目前较流行的有 Java、C、C++、C#、Pascal、Python、Prolog、Basic、FoxPro、Perl、Ruby 等。

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的，它由早期的编程语言 BCPL(Basic Combined Programming Language) 发展演变而来。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为人们所认识，到了 20 世纪八十年代，C 开始进入其它操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

C 语言是一种面向过程的结构化程序设计语言，它层次清晰，便于按模块化方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强，它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，它还可以直接访问内存的物理地址，进行位(bit)一级的操作，因此 C 语言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，也适合于应用软件的开发。此外，C 语言还具有语言简练、源程序短、效率高、可移植性强等特点。

随着微型计算机的日益普及，出现了许多 C 语言版本。由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准局(ANSI)为 C 语言制定了一套 ANSI 标准，成为现行的 C 语言标准。

面向过程的程序设计必须按照算法的实现过程逐条编写程序，通知计算机一步一步怎么做。20 世纪 80 年代后，面向对象的程序设计概念日益普及。所谓面向对象，是通过类和对象把程序涉及的数据结构和对它施行的操作有机地组织成模块，对数据及数据处理细节进行最大限度的封装，从而使开发出来的软件易于重用、修改、测试、维护和扩充，C++就是在 C 语言的基础上增加了面向对象的思想发展而来的。为了满足网页开发的需要，1994 年又出现了 Java 语言，Java 语言脱胎于 C++，不仅支持面向对象的程序设计思想，且具有软硬件平台无关的特点，适合于进行网络开发。2000 年，Microsoft 公司推出的 Microsoft Visual

Studio.NET 是一个具有公共语言子集的开发平台，实现了多种语言及其类库的无缝集成，C# 是专为这一平台推出的全新语言。C# 也派生于 C 和 C++，具有语法简洁、面向对象、与 Web 紧密结合、卓越的安全性能、灵活性和兼容性俱佳等特点，成为 .NET 平台一流的网络编程工具。

从面向过程的 C 到面向对象的 C++，再到 Windows 平台上可视化的 C++ 开发工具 Visual C++，直到网络平台上面向对象的 Java 和 C#，各种适应新的编程环境的程序设计语言也在层出不穷，但很多语言都是基于 C 语言发展而来的，而且开发系统程序（如操作系统和嵌入式系统）和低层应用程序（如接口程序、通信和自动控制等）仍然是非 C 莫属。C 语言在程序设计语言中可谓常青树，在近年来的世界程序设计语言排名中一直稳定或稍有上升，对于理工科学生来说，是一种最佳的编程入门学习语言。

1.2 C 语言程序结构特点及其运行

本节我们来初步了解 C 语言程序的结构特点，并通过 Visual C++6.0 集成开发环境学习 C 程序的运行方法。

1.2.1 C 语言程序的结构特点

下面通过几个简单的 C 语言程序，学习 C 语言程序的结构特点。

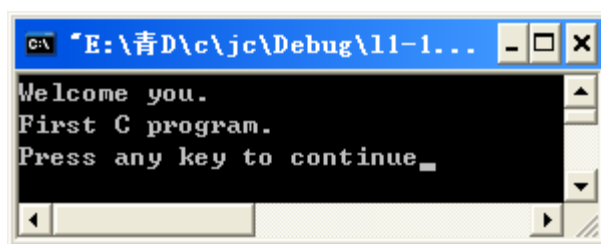
【例 1.1】在显示器上分行显示“Welcome you.”和“First C program.”。

程序如下：

```
#include<stdio.h>

void main()                /* 函数名 */
{                          /* 函数体放在一对花括号中 */
    printf("Welcome you.\nFirst C program.\n");/* 输出内容 */
}
```

运行结果为：



本例是一个最简单的 C 语言程序。

第 1 行是编译预处理命令 include。当程序中要使用 C 系统提供的标准函数或其他文件时，一定要用 include 命令包含函数对应的文件，以在编译前将对应文件嵌入到该处成为源程序的一部分。每一个程序都必须要有输出语句，因此必须包含标准输入输出头文件 stdio.h。

第 2 行是函数名 main()。一个 C 源程序由若干个函数组成，至少有一个用 main() 命名的主函数，程序从主函数开始运行，也结束于主函数。函数都要有类型说明，放在函数名前，void 表示空类型，没有返回值。

第 3 行到第 5 行为函数体。函数体是程序的执行部分，由若干语句构成，写在一对花括号“{}”内，本例的函数体只有一条输出语句。C 语言的输入与输出操作都是由系统函数完成的，printf() 为标准输出函数。printf() 函数的输出内容取决于第一对双引号“”内的字符串，一般原样输出该字符串，仅在出现以“%”开头的输出格式时有变化。“\n”是换行符。

在一对“/*”和“*/”内的内容为注释。程序编译时，碰到“/*”将不对后面的内容进

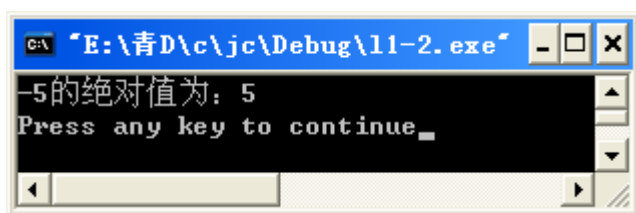
行编译直到出现“*/”为止。注释内容是对程序的解释以增加程序的可读性，不是执行代码。

【例1.2】 求-5 的绝对值。

程序如下：

```
#include<stdio.h>
#include<math.h>
void main()
{
    int x,y;                /* 定义程序中用到的变量 x 和 y */
    x=-5;                   /* 将初值-5 赋给变量 x */
    y=abs(x);               /* 求-5 的绝对值，结果赋给变量 y*/
    printf("-5 的绝对值为: %d\n",y); /* 输出变量 y 中的结果*/
}
```

运行结果为：



本例使用了系统函数 abs() 求绝对值，因此，要包含对应头文件 math.h。其它系统函数参见附录 3。%d 是输出格式，表示输出一个十进制整数，输出时其出现的位置将代入逗号后面对应变量的值。本例的%d 表示代入 y 的值 5，并以十进制整数形式显示。

【例1.3】 由键盘输入任意三个整数，输出其中的最大数。

我们用两种形式编写程序代码。

程序 1 如下：

```
#include <stdio.h>
void main( )
{
    int x,y,z,max;
    scanf("%d%d%d",&x,&y,&z);          /* 运行时，由键盘输入 3 个整数 */
    if(x>y)max=x;else max=y;          /* 比较 x 和 y，将大者赋给 max */
    if(max<z)max=z;                  /*比较 max 和 z，将大者赋给 max */
    printf("%d,%d,%d 中的最大值为: %d\n",x,y,z,max); /* 输出结果 */
}
```

运行结果为：



Scanf() 是系统提供的标准输入函数，该函数在程序运行时，为逗号后的各变量赋值，该函数要求变量名前须加取地址符“&”。用户每次运行程序，可根据实际情况随机灵活地输入不同的变量值。本例程序运行后，用“空格”隔开输入 3 个数后回车，系统自动顺次将其

赋值给变量 x、y、z。

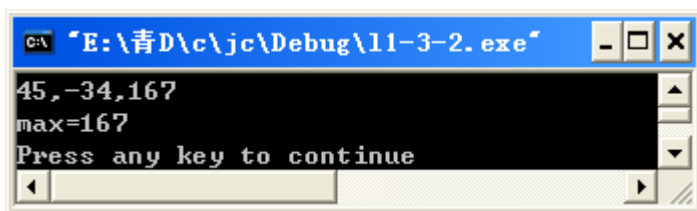
程序 2 如下：

```
#include <stdio.h>

void main()
{
    int max(int x,int y);          /* 对自定义函数 max 做声明*/
    int a,b,c,d;
    scanf("%d,%d,%d",&a,&b,&c);
    d=max(a,b);                  /* 调用函数 max 找出 a,b 中大者赋给 d*/
    d=max(d,c);                  /* 再次调用 max 找出 3 个数中大者赋给 d*/
    printf("max=%d\n",d);
}

int max(int x,int y)            /* 自定义函数 max，从两个数中找出大者*/
{
    int m;
    if(x>y) m=x;                /*比较 x 和 y，将大者赋给 m */
    else m=y;
    return(m);                  /* 返回找到的大数*/
}
```

运行结果为：



程序 2 中包括两个函数 main() 和 max()。max() 函数的功能是找到任意两个数中的大数，并将结果返回给调用它的函数。本例通过在主函数中两次调用 max() 函数，完成从 3 个数找最大数功能。把一些功能相对独立的程序段编成一个函数，避免了重复代码，降低了代码量和复杂度，可以反复调用，C 语言就是通过函数实现了程序的模块化。

由上面例题可以看出：

(1) C 语言程序是由函数构成的。一个 C 源程序不论由多少个文件组成，包含多少个函数，都有且仅有一个 main 函数，即主函数。一个 C 程序总是从 main 函数开始执行，而不论 main 函数在程序中的什么位置，程序的执行也结束于主函数，其它函数通过函数调用被执行。

(2) 每个函数包括函数名和函数体两大部分。除主函数外，其它函数按 C 语言命名规则自主命名。函数体包含在一对大括号“{”和“}”中，由若干语句构成。

(3) C 程序书写格式自由，可在一行内写多条语句，也可将一条语句写在多行上（上行末加续行符“\”）。每条语句必须以分号“;”结尾。

(4) 程序代码一般用小写字母书写，C 语言严格区分大小写。

(5) 注释部分仅增加程序的可读性，是给用户看的，不会被编译，可以出现在程序的任意地方。写在双边注释符“/*……*/”中的注释内容可以占部分行、单行、多行；写在单边注释符“//”后的注释内容仅占一行，至行末为止。

(6) 源程序中的预处理命令通常放在源程序的最前面，预处理命令之后不能加分号。

(7) 函数体内语句一般按四大功能顺次排列，即首先对变量与函数做声明，然后为变量

赋初值，再进行数据计算与处理，最后输出结果。

1.2.2 C 语言程序的运行

使用高级语言编写的程序统称为源程序，源程序不能为计算机所识别。一个 C 语言程序的执行需要经过编辑、编译、连接和运行四个步骤，才能得到程序结果，如图 1-1 所示。

(1) 编辑。通过纯文本编辑软件输入和编辑源代码，并保存成源程序文件。在 VC++6.0 环境下，源程序文件的扩展名为 .c 或 .cpp，如 ex.cpp。

(2) 编译。通过编译系统将源程序代码编译成计算机能直接识别的二进制形式，即目标程序。在编译过程中，系统自动检查源程序中的句法和语法错误，如发现错误，会报告错误类型及在程序中的位置，帮助用户修改错误。如未发现错误，则自动生成对应的目标文件。目标程序文件是与源程序同主名的 .obj 文件，如 ex.obj。

(3) 连接。将形成的目标程序、库函数或其他目标程序连接装配成一个可执行程序。大多数的编译系统都允许使用一个命令同时完成编译和连接操作。可执行程序文件是与源程序同主名的 .exe 文件，如 ex.exe。

(4) 运行。运行可执行程序，得到程序结果。

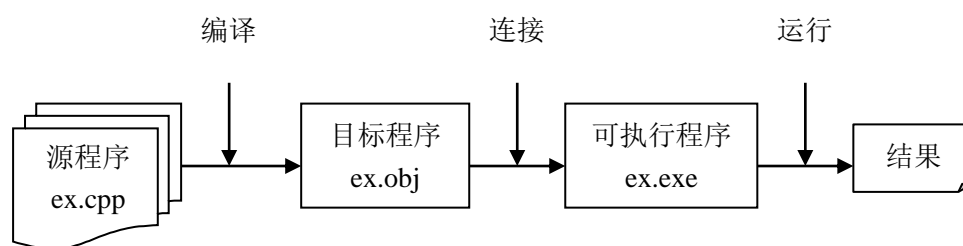


图 1-1 C 语言程序运行步骤

1.2.3 Visual C++6.0 集成开发环境的使用

集成开发环境 (Integrated Development Environment, IDE) 是集程序的编辑、编译、连接以及运行等功能于一体的程序开发软件，利用 IDE 可以大大提高程序开发的效率。目前，用于 C 程序开发的集成环境有 Turbo C、borland C、Microsoft C 和 Microsoft Visual C++ 等。Microsoft Visual C++6.0 (简称 VC++6.0) 是 Microsoft 公司 1998 年推出的基于 Windows 平台的集成开发环境，不仅可以开发 C++ 程序，也可以开发 C 程序。

本书内容均针对 VC++6.0 编写。

在 VC++6.0 中，一个 C 应用程序被称为一个项目或工程 (project)，它是由应用程序中所需要的所有文件组成的一个有机整体，一般包括源文件、头文件和资源文件等。项目能自动将其包含的文件进行分类和管理，从而大大减轻了程序员的负担。项目被置于项目工作区 (workspace) 的管理之下。一个项目工作区可以包含多个项目，甚至是不同类型的项目，这些项目之间相互独立，但共用一个项目工作区的环境设置。

1. 在 VC++6.0 中运行一个 C 程序的步骤

(1) 启动 VC++6.0。选择“开始”→“程序”→“Microsoft Visual C++6.0”命令。打开如图 1-2 所示窗口。

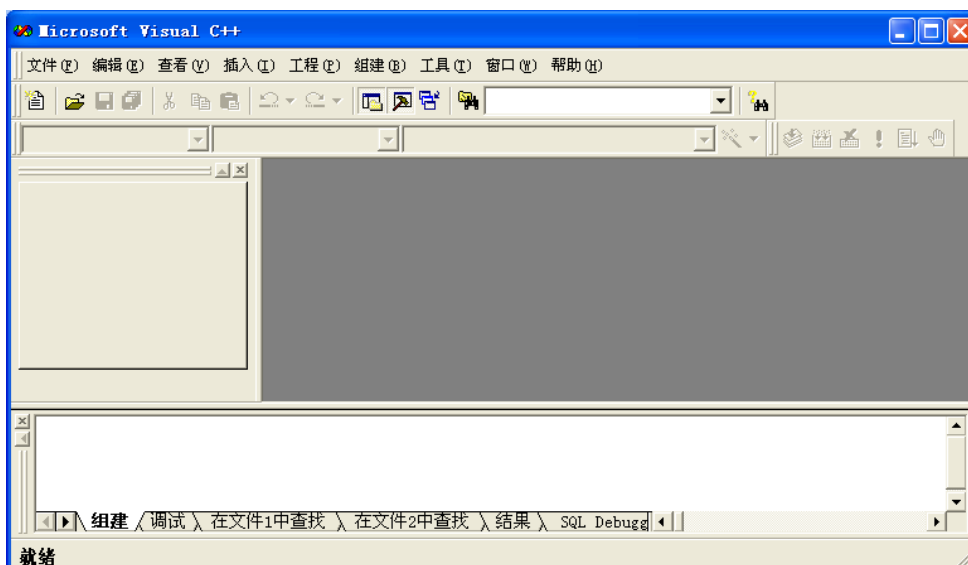


图 1-2 Visual C++6.0 主窗口

(2) 创建项目工作区。在 VC++6.0 窗口中，选择“文件”→“新建”命令，弹出如图 1-3 所示“新建”对话框，选择“工程”选项卡中的“Win32 Console Application”，指定项目名称和存放位置，然后“确定”，出现如图 1-4 所示对话框，选择“一个空工程”→完成→确定，创建项目和项目工作区，并在左侧项目工作区中显示与项目有关的信息。



图 1-3 “新建”项目对话框



图 1-4 “Win32 Console Application” 对话框

(3) 创建源文件。打开“新建”对话框中的“文件”选项卡，选择“C++ Source File”，指定文件名，如图 1-5 所示。单击“确定”，创建源程序文件。



图 1-5 “新建” 文件对话框

(4) 编辑源程序文件。在编辑窗口中输入编辑源代码，随时单击“保存”命令。

(5) 编译。选择“组件” → “编译”命令，对源程序进行编译，生成目标文件。编译结果会显示在下面的输出窗口中，如图 1-6 所示。如果编译有错误必须修改错误重新编译，

直至编译通过。

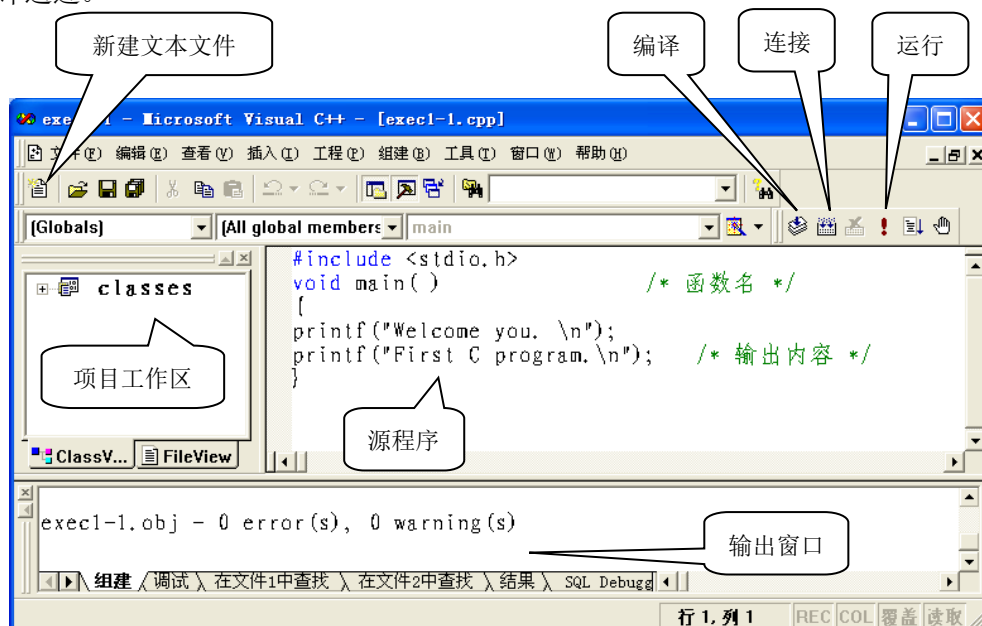


图 1-6 源文件的编辑、编译、连接和运行

(6) 连接。编译通过后，选择“组件”→“组件…”文件命令，生成可执行文件。

(7) 运行。选择“组件”→“执行…”文件命令，执行程序，得到结果。

另外，如果只是运行 C 程序，可以简化以上操作。步骤是：打开 VC++6.0→单击“新建文本文件”→输入源程序→保存文件（文件名必须加.c 或.cpp 扩展名）→编译→运行。

源程序的建立、编译、连接和运行等操作命令的选择直接单击工具栏上的工具按钮完成也更加快捷、方便，如图 1-6 所示。

新项目工作区和项目创建完成后，在项目文件夹中会有.c 或.cpp 的源文件、.dsp 项目文件、.dsw 项目工作区文件、.ncb 发布文件，还有一个 Debug 文件夹存放编译和连接生成的各种文件。

2. C 程序调试错误

C 程序调试时，可能会出现以下类型的错误。

(1) 编译错误。编译错误是编译系统在编译时发现的错误，也叫语法错误。编译错误分为错误（Error）和警告（Warning）。Error 通常是由于关键字拼写错误、不可识别的标识符、语句末尾缺少分号、不配对的圆括号或花括号等引起的错误，必须修改，否则编译无法通过。Warning 通常是因为数据类型转换、不恰当的类型说明符等引起的错误，编译系统无法确定是否真正错误，便以警告的形式告知用户。一般来说，警告并不影响后续的编译、连接和运行，但有的警告不加纠正，可能导致严重的运行错误。因此，对警告应分析原因，尽量予以消除。

有时，编译后会出现大片错误，这种情况往往是一种假象，实际可能是由一两个错误连带引起的。因此，在调试程序时，应当首先纠正最前面的或容易发现的错误，然后重新编译，逐步减少错误量，而不是修改所有错误后再重新编译。

(2) 运行错误。运行错误是在运行过程中发现的错误。比如对负数开平方根、数据发生溢出等。

(3) 逻辑错误。逻辑错误是由于使用了错误的算法、公式、数据等引起的错误。这种错误系统并不报错，必须由编程人员自行发现和纠正。

1.3 程序设计算法及其描述方法

计算机程序是指为完成特定功能而编写的指令的集合，程序设计就是用计算机语言对所要解决的问题进行完整而准确的描述过程，但是程序设计并不就是简单的编写程序代码的过程。程序可以表示为：

程序=算法+数据结构+程序设计方法+语言工具和环境

程序设计方法是编写程序的指导思想，决定了以什么样的方式组织编写程序，也决定了一个程序的成功与否；语言和环境是编写程序的工具，负责制造程序；而算法则是灵魂，是解决问题（处理数据）的方法和步骤；数据结构是算法加工的对象。

一个完整的程序设计过程大体可分为四个阶段：

（1）分析问题：充分研究与分析要解决的问题，明确问题的需求，即应用程序要实现什么功能，达到什么目的。

（2）设计算法：根据问题需求，设计具体算法，描述算法的具体实现步骤。解决同一问题可能有多种算法，应选择合适的算法。

（3）编写程序：根据选定的算法，选择适当的程序开发语言与环境，再按照算法编写程序代码。

（4）程序验证：进行程序的正确性证明、测试与调试。应设计各种情况下的验证数据，保证每一条件的成立与不成立都被测试过，以证明程序正确无误。

可见，编写程序实际上就是在实现某种算法，算法在计算机程序设计中是非常重要的。

1.3.1 算法的概念及特征

算法（Algorithm）就是解决问题的步骤和方法。能够对一定规范的输入，在有限时间内获得所要求的输出。解决一个问题的过程，就是实现一个算法的过程。一个解题步骤、工作计划、生产流程、音乐乐谱等都可称为“算法”。

如果一个算法有缺陷，或不适合于某个问题，则即使执行了这个算法也不会解决该问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。一个算法的优劣可以用空间复杂度与时间复杂度来衡量。

一个算法应该具有以下五个重要的特征。

（1）有穷性：一个算法应包括有限的操作步骤，能在执行有穷的操作步骤之后结束。

（2）确定性：算法的计算规则及相应的计算步骤必须是唯一确定的，既不能含糊其词，也不能有二义性。

（3）输入：一个算法有零个或多个输入，以刻画运算对象的初始情况。

（4）输出：一个算法有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

（5）可行性：算法中的每一个步骤都是可以在有限的时间内完成的基本操作，并能得到确定的结果。

任何简单或复杂的算法都是由基本功能操作和控制结构这两个要素组成的。计算机的基本功能操作包括以下四个方面。

（1）逻辑运算：与、或、非等。

（2）算术运算：加、减、乘、除等。

（3）数据比较：大于、小于、等于、不等于、大于等于、小于等于等。

（4）数据传送：输入、输出、赋值。

而控制结构是指各操作之间的执行顺序。按照结构化程序设计的观点，任何复杂的算

法都可以由顺序结构、选择（分支）结构和循环结构这三种基本结构来实现。这种结构化程序结构清晰、易于正确性验证、易于纠错，提高了程序的可靠性，保证了程序的质量。

1.3.2 算法的描述方法

为了将算法正确的表示出来，需要使用算法描述工具。算法有多种描述方法，可概括为两大类：文字描述（语言描述）和图形描述。下面介绍几种常见的描述方法。

（1）自然语言：是最简单的一种算法描述工具，如汉语、英语等。其优点是通俗易懂，易于掌握，但也有繁琐、容易产生歧义的缺点。

（2）流程图：使用不同的图框表示不同类型的操作，用带箭头的线表示操作的执行顺序。相对于自然语言来说流程图更简洁直观。

流程图中常用的图框和规定如图 1-7 所示。基本控制结构的流程图如图 1-8 所示。

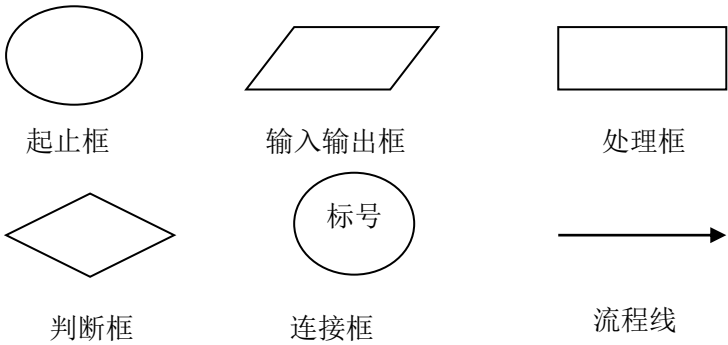


图 1-7 流程图常用图符

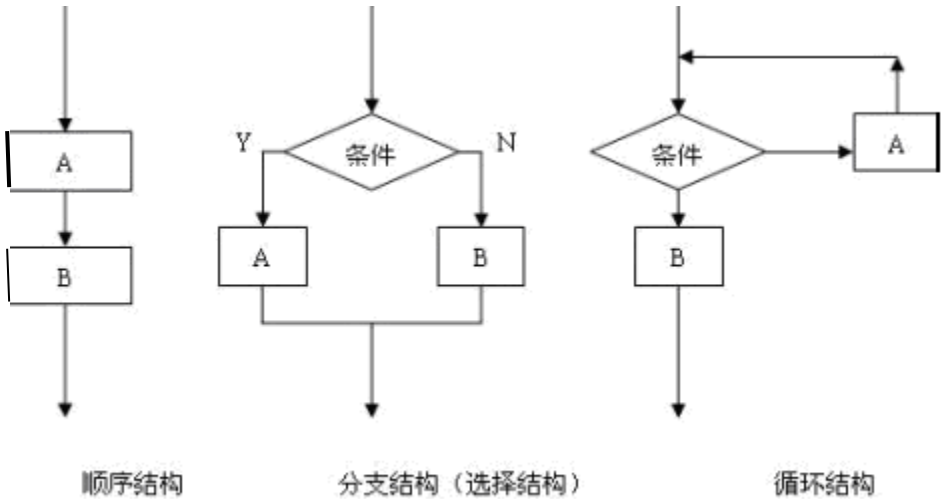


图 1-8 三种基本控制结构流程图

（3）N—S 图：描述算法的另一种常见方法，主要省掉了流程图中的流程线，使得图形更紧凑。N—S 图的基本结构描述形式如图 1-9 所示。N—S 图的优点是能直观地用图形表示算法，去掉了导致程序非结构化的流线，缺点是修改不方便。

（4）高级语言：用某种高级语言去描述算法，其好处是可以直接在计算机上运行，但受计算机语言严格的格式要求和语法限制，对用户要求较高。

(5) 伪代码：是介于自然语言与计算机语言之间的一种用文字和符号相结合的算法描述工具，形式上跟计算机语言比较接近，但没有严格的语法规则限制。通常是借助某种高级语言的控制结构，中间的操作可以用自然语言，也可以用程序设计语言描述，这样，既避免了严格的语法规则限制，又比较容易最终转换成程序代码。

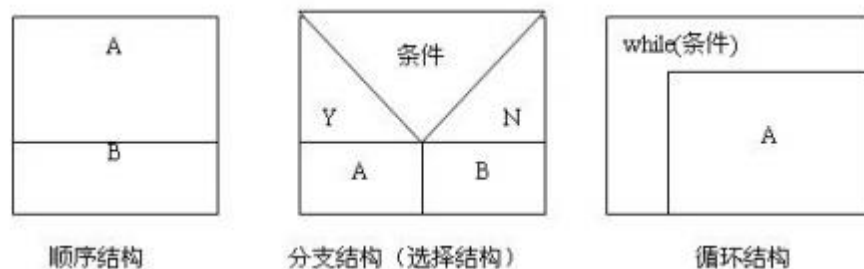


图 1-9 三种基本控制结构 N-S 图

在为具体问题设计算法时，选用哪种算法描述工具并不重要，重要的是一定要把算法描述得简洁、正确，不会产生理解上的“歧义性”。

例如：计算如下分段函数

$$y = \begin{cases} x+1 & x > 0 \\ x-100 & x = 0 \\ x \times 20 & x < 0 \end{cases}$$

①使用算法语言描述如下：

第一步：输入变量 x 的值。

第二步：判断 x 是否大于 0。若大于 0，则 y 为 $x+1$ ，然后转第五步；否则进行下一步（即第三步）。

第三步：判断 x 是否等于 0。若等于 0，则 y 为 $x-100$ ，然后转第五步；否则进行下一步（即第四步）。

第四步： y 为 $x \times 20$ （如果第二、三步条件不成立，则进入第四步时， x 肯定小于 0）。

第五步：输出变量 y 的值。

第六步：结束程序执行。

②使用流程图表示，如图 1-10 所示。

③使用 N-S 图表示，如图 1-11 所示。

④以高级语言 C 来描述的算法如下：

```
int x,y;
scanf("%d",&x);
if(x>0) y=x+1;
else if(x==0) y=x-100;
elsey=x*20;
printf("x=%d,y=%d";x,y) ;
```

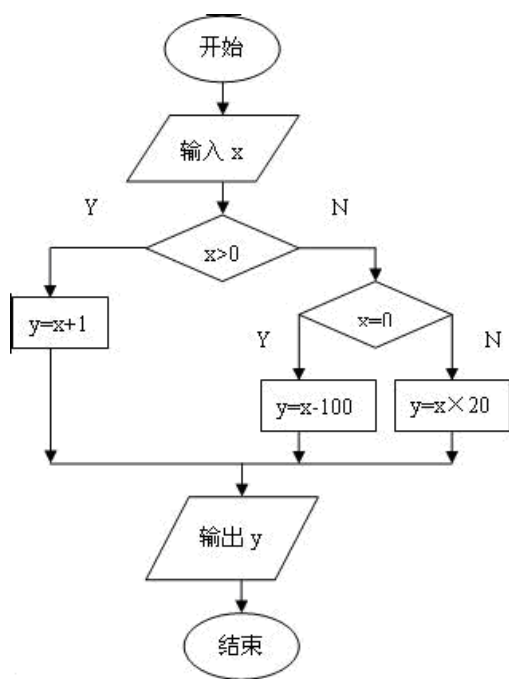


图 1-10 求分段函数算法的流程图

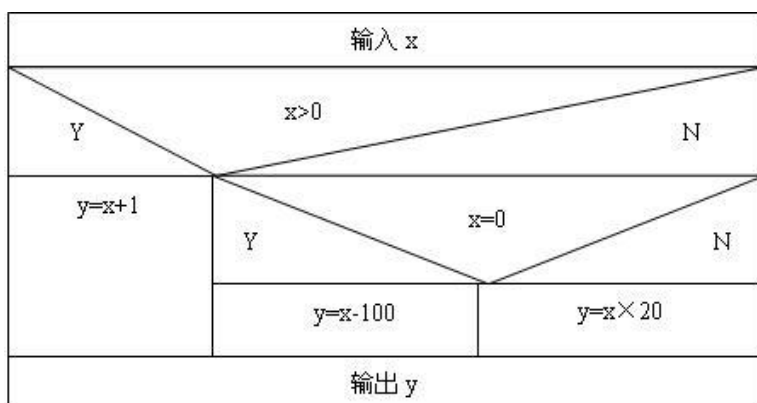


图 1-11 求分段函数算法的 N-S 图

1.3.3 程序设计风格

程序设计风格是指编制程序时所表现出来的特点、习惯和逻辑思路等。程序设计风格会深刻影响软件的质量和可维护性，良好的程序设计风格可以使程序结构清晰合理，使程序代码便于维护。培养良好的程序设计风格对于初学者来说非常重要。

程序设计的风格总体来说应强调简单和清晰，程序必须是可理解的，即做到“清晰第一，效率第二”。为提高程序的可阅读性，形成良好的程序设计风格，主要应注意和考虑以下因素：

1. 源程序文档化

(1) 标识符按意取名。具有实际意义的命名有助于对程序功能的理解。

(2) 程序应加注释。正确的注释能够帮助理解程序。注释分序言性注释和功能性注释。序言性注释通常置于每个模块的起始部分，给出程序的整体说明，主要内容有：程序标题、

程序功能说明、主要算法、接口说明、程序位置、开发简历、程序设计者、复阅者、复审日期、修改日期等。功能性注释一般嵌在源程序内部，说明程序段或语句的功能以及数据的状态。

(3) 在程序中利用空格、空行、缩进等技巧可以使程序层次清晰，结构一目了然。

2. 数据说明方法

为使数据定义更易于理解和维护，一般应注意以下几点：

(1) 数据说明顺序规范化。使数据说明的次序固定，可以使数据的属性更易于查找，从而有利于测试、纠错与维护。例如按以下顺序：常量说明、类型说明、全程量说明、局部量说明。

(2) 变量说明有序化。一个语句说明多个变量时，各变量名按字母序排列。

(3) 对复杂的数据结构加注释，说明在程序实现时的特点。

3. 语句的结构

语句结构应简单直接，不能为了追求效率而使代码复杂化。为了便于阅读和理解，不要一行多个语句。不同层次的语句采用缩进形式，使程序的逻辑结构和功能特征更加清晰。要避免复杂的判定条件，避免多重的循环嵌套。表达式中使用括号以提高运算次序的清晰度等。

4. 输入和输出

在编写输入和输出程序时，考虑以下原则：

(1) 输入操作步骤和输入格式尽量简单。

(2) 应检查输入数据的合法性、有效性，报告必要的输入状态信息及错误信息。

(3) 输入一批数据时，使用数据或文件结束标志，而不要用计数来控制。

(4) 交互式输入时，提供可用的选择和边界值。

(5) 当程序设计语言有严格的格式要求时，应保持输入格式的一致性。

(6) 输出数据表格化、图形化。

5. 效率

指处理机时间和存储空间的使用，对效率的追求要明确以下几点：

(1) 效率是一个性能要求，目标在需求分析给出。

(2) 追求效率应建立在不损害程序可读性或可靠性基础上，要先使程序正确，再提高程序效率，先使程序清晰，再提高程序效率。

(3) 提高程序效率的根本途径在于选择良好的设计方法、良好的数据结构算法，而不是靠编程时对程序语句做调整。

使用 C 语言编写程序除了必须遵守 C 的语法要求以外，还应遵守一系列的规约，这些规约不是强制性的，但是约定成俗的编程风格。

(1) 所有代码必须采用 ANSI C 标准，保持代码简单清晰，避免使用复杂语句。

(2) 程序结构清晰易懂，单个函数的程序行数建议不超过 100 行，代码行长度尽量不超过 80 个字符。每个语句行只做一件事情。

(3) 尽量使用标准库函数和公共函数。程序针对要达成的目标设计，力求简单，避免出现冗余的代码。

(4) 表达式中多使用括号可避免二义性。

(5) 要有足够多的注释充分解释源代码。每个源程序文件都有文件头说明，每个函数也都要有函数头说明。适当处加空行或空格也是一种特殊的注释，可提升可读性。

- (6) 利用缩进来显示程序的逻辑结构，缩进量一致并以“Tab”键为单位。
- (7) 循环、分支的层次不要超过 5 层，避免出现不必要的分支，尽量避免使用 goto 语句。
- (8) 所有变量在调用前必须赋值。不要进行浮点数的比较，如 $10.0*0.1==1.0$ 的比较不可靠。

习 题

一、单项选择题

1. 以下叙述中正确的是 ()。
A. 程序设计的任务就是编写程序代码并上机测试
B. 程序设计的任务就是确定所用数据结构
C. 程序设计的任务就是确定所用算法
D. 以上三种说法都不完整
2. 下列叙述中，不符合良好程序设计风格的是 ()。
A. 程序的效率第一，清晰第二
B. 程序的可读性好
C. 程序中有必要的注释
D. 输入数据前要有提示信息
3. 以下叙述正确的是 ()。
A. C 语言程序将从源程序中的第一个函数开始执行
B. 可以在程序中由用户指定任意一个函数作为主函数，程序将从此开始执行
C. C 语言规定必须用 main 作为主函数名，程序将从此开始执行，在此结束
D. main 可作为用户标识符，用以命名任意一个函数作为主函数
4. 以下叙述中正确的是 ()。
A. C 程序中的注释只能出现在程序的开始位置和语句的后面
B. C 程序书写格式严格，要求一行内只能写一个语句
C. C 程序书写格式自由，一个语句可以写在多行上
D. 用 C 语言编写的程序只能放在一个程序文件中
5. 计算机能直接执行的程序是 ()
A. 源程序
B. 目标程序
C. 编译程序
D. 可执行程序

二、填空题

1. C 源程序要先经过_____变成目标程序，再经过_____生成可执行程序后才能运行。
2. 每一个 C 程序必须有一个也只能有一个_____函数。
3. 程序的执行要从_____函数开始。
4. 主函数在程序中的位置可以_____。
5. 注释内容应写在一对_____内。在此之内的内容系统将_____编译。
6. C 程序是以_____符号作为一条语句的结束符。

三、简答题

1. 写出结构化程序设计的三种基本结构。
2. 运行一个 C 程序必须要经过哪四个步骤？
3. 分别用流程图和 N-S 图描述算法：判断一个数是否能被 3 和 5 整除。

四、程序改错

指出并修改下面程序中的错误。

```
include <stdio.h>
main {}
```

```
/* this program prints the number of weeks in a year. */  
(  
    int s  
    s:=52;  
    print(There are s weeks in a year");
```