



项目一 C 语言概述

项目描述

小明曾认为计算机是无所不能的,能够自动地进行所有的工作,什么难题到计算机面前都能迎刃而解。其实计算机的每一个操作都是根据人们事先指定的指令进行的。程序就是一组计算机能识别和执行的指令,每一条指令使计算机执行特定的操作。

能力目标

1. 了解 C 语言程序的结构。
2. 掌握 C 语言程序的组成部分。
3. 掌握输入/输出函数的基本格式和功能。
4. 学会实现 C 语言程序的运行、方法和步骤。

职业素养

1. 养成良好的专业态度和正确的专业思想。
2. 培养分析能力和解决问题的能力。
3. 培养良好的编程习惯。

一、C 语言的产生与发展

C 语言是在 BASIC 语言、FORTRAN 语言和 PASCAL 语言之后出现的一种通用计算机程序设计语言,它适用编写各种系统软件,也适用于编写各种应用软件。

C 语言是 1972 年由美国 Dennis Ritchie 设计发明的,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL (Basic Combind Programming Language)发展演变而来。在 1970 年,AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进并取名为 B 的语言,最后导致了 C 语言的问世。

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,美国国家标准研究所(ANSI)为 C 语言制定了一套 ANSI 标准,并成为现行的 C 语言标准。

二、C 语言的特点

C 语言发展如此迅速,而且成为最受欢迎的语言之一,主要是因为它具有强大的功能。许多著名的系统软件,如 PC-DOS、DBASE IV 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序,就更能显示 C 语言的优势了。归纳起来 C 语言具有以下特点:

1. C 是中级语言

它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,而这三者是计算机最基本的工作单元。

2. C 是结构式语言

结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外,彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言是以函数形式提供给用户的,这些函数可以方便地调用,可多种循环、条件语句控制程序流向,从而使程序完全结构化。

3. C 语言功能齐全

C 语言具有各种各样的数据类型,并引入了指针概念,使得程序效率更高。另外 C 语言也具有强大的图形功能,支持多种显示器和驱动器,而且计算功能、逻辑判断功能也比较强大,可以实现决策目的。

4. C 语言适用范围大

C 语言还有一个突出的优点就是适合于多种操作系统,如 DOS、UNIX,同时也适用于多种机型。

三、Turbo C 的产生与发展

我们平常用的大部分都是 Turbo C。Turbo C 是美国 Borland 公司的产品,Borland 公司

是一家专门从事软件开发、研制的公司。该公司相继推出了一套 Turbo 系列软件,如 Turbo BASIC、Turbo Pascal、Turbo Prolog,这些软件都很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品,其中使用了全新的集成开发环境,即使用了一系列下拉式菜单,将文本编辑、程序编译、连接以及程序运行一体化,大大方便了程序的开发。1988 年,Borland 公司又推出 Turbo C 1.5 版本,增加了图形库和文本窗口函数库等,而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能,并可以在 Tiny 模式下直接生成“.com”(数据、代码、堆栈处在同一 64k 中)文件,还可对数学协处理器(支持 8087/80287/80387 等)进行仿真。我们目前经常使用的集成环境就是 Turbo C 2.0。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C++,它继承发展了 Turbo C 2.0 的集成开发环境,并包含了面向对象的基本思想和设计方法。习惯上我们叫它 Turbo C 3.0。

1991 年为了适用 Microsoft 公司的 Windows 3.0 版本,Borland 公司又将 Turbo C++作了更新,即 Turbo C 的新一代产品 Borland C++。

1. C 语言的历史

C 语言是在 20 世纪 70 年代初问世的。1978 年由美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 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。但是在《K&R》中并没有定义一个完整的标准 C 语言,后来由美国国家标准学会在此基础上制定了一个 C 语言标准,于 1983 年发表,通常称之为 ANSI C。

早期的 C 语言主要是用于 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、小和微型计算机上得到了广泛使用,并成为当代最优秀的程序设计语言之一。

2. C 语言的特点

C 语言是一种结构化语言。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。由于 C 语言实现了对硬件的编程操作,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 语言还具有效率高、可移植性强等特点。因此广泛地移植到了各类型的计算机上,从而形成了多种版本的 C 语言。

3. C 程序的结构

为了说明 C 语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些例子中了解到组成一个 C 源程序的基本部分和书写格式。

```
main()
{
    printf("c 语言世界 www.vcok.com,您好! \n");
}
```

main 是主函数的函数名,表示这是一个主函数。每一个 C 源程序都必须有,且只能有一个主函数(main 函数)。函数调用语句,printf 函数的功能是把要输出的内容送到显示器去显示。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。

```
#include "stdio.h"
#include "math.h"
main()
{
    double x,s;
    printf("input number: \n");
    scanf("%lf",&x);
    s=sin(x);
    printf("sine of %lf is %lf\n",x,s);
}
```

每行注释:

- (1) include 称为文件包含命令,扩展名为“.h”的文件,也称为头文件或首部文件。
- (2) 定义两个实数变量,以被后面程序使用。
- (3) 显示提示信息。
- (4) 从键盘获得一个实数 x。
- (5) 求 x 的正弦,并把它赋给变量 s。
- (6) 显示程序运算结果。
- (7) main 函数结束。

程序的功能是从键盘输入一个数 x,求 x 的正弦值,然后输出结果。在 main()之前的两行称为预处理命令(详见后面)。预处理命令还有其他几种,这里的 include 称为文件包含命令,其意义是把引号"或尖括号<>内指定的文件包含到本程序中,成为本程序的一部分。被包含的文件通常是由系统提供的,其扩展名为“.h”,因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此,凡是在程序中调用一个库函数时,都必须包含该函数原型所在的头文件。在本例中,使用了三个库函数:输入函数 scanf、正弦函数 sin 及输出函数 printf。sin 函数是数学函数,其头文件为 math.h 文件,因此在程序的主函数前用 include 命令包含了 math.h。scanf 和 printf 是标准输入、输出函数,其头文件为 stdio.h,在主函数前也用 include 命令包含了 stdio.h 文件。

需要说明的是,C 语言规定对 scanf 和 printf 这两个函数可以省去对其头文件的包含命令,所以在本例中也可以删去第二行的包含命令 #include。同样,在后面例 1-1 中使用了 printf 函数,也省略了包含命令。

在例题中的主函数体分为两部分,一部分为说明部分,另一部分为执行部分。说明是指变量的类型说明,第 6 个例题中未使用任何变量,因此无说明部分。C 语言规定,源程序中所有用到的变量都必须先说明后使用,否则将会出错。这一点是编译型高级程序设计语言的一个特点,与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分,第

2个例中使用了两个变量 x 、 s ，用来表示输入的自变量和 \sin 函数值。由于 \sin 函数要求这两个量必须是双精度浮点型，故用类型说明符 `double` 来说明这两个变量。说明部分后的是行为执行部分或称为执行语句部分，用以完成程序的功能。执行部分的第一行是输出语句，调用 `printf` 函数在显示器上输出提示字符串，请操作人员输入自变量 x 的值；第二行为输入语句，调用 `scanf` 函数，接受键盘上输入的数并存入变量 x 中；第三行是调用 `sin` 函数并把函数值送到变量 s 中；第四行是用 `printf` 函数输出变量 s 的值，即 x 的正弦值。程序结束。

```
printf("input number: \n");
scanf("%lf",&x);
s=sin(x);
printf("sine of %lf is %lf\n",x,s);
```

运行本程序时，首先在显示器屏幕上给出提示串 `input number`，这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数，如“5”，按下回车键，接着在屏幕上给出计算结果。

在前两个例子中用到了输入和输出函数 `scanf` 和 `printf`，在项目三中我们要详细介绍。这里我们先简单介绍一下它们的格式，以便下面使用。`scanf` 和 `printf` 这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的格式输入输出值。因此，这两个函数在括号中的参数表都由以下两部分组成：“格式控制串”、“参数表”。格式控制串是一个字符串，必须用双引号括起来，它表示了输入输出量的数据类型。各种类型的格式表示法可参阅项目三。在 `printf` 函数中还可以在格式控制串内出现非格式控制字符，这时在显示屏幕上将原文显示。参数表中给出了输入或输出的量，当有多个量时，用逗号间隔。例如：

```
printf("sine of %lf is %lf\n",x,s);
```

其中 `%lf` 为格式字符，表示按双精度浮点数处理。它在格式串中出现两次，对应了 x 和 s 两个变量。其余字符为非格式字符则照原样输出在屏幕上。

```
int max(int a,int b);
main()
{
    int x,y,z;
    printf("input two numbers: \n");
    scanf("%d%d",&x,&y);
    z=max(x,y);
    printf("maxmum=%d",z);
}
int max(int a,int b)
{
    if(a>b)return a;
    else return b;
```

}

此函数的功能是输入两个整数,并输出其中的大数。

```
/* 函数说明 */
/* 主函数 */
/* 变量说明 */
/* 输入 x,y 值 */
/* 调用 max 函数 */
/* 输出 */
/* 定义 max 函数 */
/* 把结果返回主调函数 */
```

上面例中程序的功能是由用户输入两个整数,程序执行后输出其中较大的数。本程序由两个函数组成,主函数和 max 函数。函数之间是并列关系,可从主函数中调用其他函数。max 函数的功能是比较两个数,然后把较大的数返回给主函数。max 函数是一个用户自定义函数。因此在主函数中要给出说明(程序第三行)。可见,在程序的说明部分中,不仅可以有变量说明,还可以有函数说明。关于函数的详细内容将在项目五中介绍。在程序的每行后用/*和*/括起来的内容为注释部分,程序不执行注释部分。

上例中程序的执行过程是,首先在屏幕上显示提示串,请用户输入两个数,回车后由 scanf 函数语句接收这两个数送入变量 x、y 中,然后调用 max 函数,并把 x、y 的值传送给 max 函数的参数 a、b。在 max 函数中比较 a、b 的大小,把大者返回给主函数的变量 z,最后在屏幕上输出 z 的值。

4. C 程序的结构特点

(1) C 源程序的结构特点:

- ① 一个 C 语言源程序可以由一个或多个源文件组成。
- ② 每个源文件可由一个或多个函数组成。
- ③ 一个源程序不论由多少个文件组成,都有一个且只能有一个 main 函数,即主函数。
- ④ 源程序中可以有预处理命令(include 命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。
- ⑤ 每一个说明、每一个语句都必须以分号结尾。但预处理命令,函数头和花括号“}”之后不能加分号。

⑥ 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,则可不再加空格间隔。

(2) 书写程序时应遵循的规则:

从书写清晰,便于阅读、理解、维护的角度出发,在书写程序中应遵循以下规则:

- ① 一个说明或一个语句占一行。
- ② 用{}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。

③ 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。以便看起来更加清晰,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。

【例 1-1】 第一个程序：Hello,World!

```
/* example1.1 The first C Program */
#include <stdio.h>
main()
{
    printf("Hello,World!");
}
```

输出：

Hello,World!

第二个程序：计算两数之和。

```
/* example1.1 calculate the sum of a and b */
#include <stdio.h>
/* This is the main program */
main()
{
    int a,b,sum;
    a=10;
    b=24;
    sum=add(a,b);
    printf("sum= %d\n",sum);
}
/* This function calculates the sum of x and y */
int add(int x,int y)
{
    int z;
    z=x+y; return(z);
}
```

运行结果：

sum=34

任务二

C 程序的上机步骤

一、Turbo C 2.0 简介和启动

我们上机实习和考试都将使用 Borland Turbo C 2.0 这个版本。该系统是 DOS 操作系统支持下的软件,在 Windows 98 环境下,可以在 DOS 窗口下运行。

我们机房是在 D 盘根目录下建立的一个 TC 子目录下安装 Turbo C 2.0 系统的。TC 下还建立了两个子目录 LIB 和 INCLUDE,在 LIB 子目录中存放库文件,在 INCLUDE 子目录中

存放所有头文件。

在 DOS 环境下或在 Windows 98 的 DOS 窗口下运行 Turbo C 2.0 时,只要在 TC 子目录下键入 TC 并回车即可进入 Turbo C 2.0 集成开发环境。

在 Windows 98 环境下,也可以选运行菜单,然后键入 d:\tc\tc 即可,或者可以在 tc 文件夹中找到 tc.exe 文件,然后用鼠标双击该文件名也可进入 Turbo C 2.0 集成开发环境。

Turbo C 是美国 Borland 公司的产品,Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件,如 Turbo BASIC、TurboPascal、Turbo Prolog,这些软件很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品,其中使用了全新的集成开发环境,即使用了一系列下拉式菜单,将文本编辑、程序编译、连接以及程序运行一体化,大大方便了程序的开发。1988 年,Borland 公司又推出了 Turbo C 1.5 版本,增加了图形库和文本窗口函数库等,而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能,并开始可以在 Tiny 模式下直接生成 .COM (数据、代码、堆栈处在同一 64k 内存中) 文件,还可对数学协处理器 (支持 8087/80287/80387 等) 进行仿真。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C ++,它继承发展 Turbo C 2.0 的集成开发环境,并包含了面向对象的基本思想和设计方法。1991 年为了适用 Microsoft 公司的 Windows 3.0 版本,Borland 公司又将 Turbo C ++ 做了更新,即 Turbo C 的新一代产品 Borlandc C ++。

二、Turbo C 2.0 集成开发环境

进入 Turbo C 2.0 集成开发环境中后,屏幕上显示如图 1-1 所示。



图 1-1 屏幕界面

其中顶上一行为 Turbo C 2.0 主菜单,中间窗口为编辑区,接下来是信息窗口,最底下一行为参考行。这四个窗口构成了 Turbo C 2.0 的主屏幕,以后的编程、编译、调试以及运行都将在这个主屏幕中进行。

主菜单在 Turbo C 2.0 主屏幕顶上一行,显示下列内容:

File Edit Run Compile Project Options Debug Break/watch

除 Edit 外,其他各项均有子菜单,只要用【Alt】加上某项中第一个字母,就可进入该项的子菜单中。

1. File 菜单

按【Alt+F】可进入 File 菜单,如图 1-2 所示。



图 1-2 File 菜单

File 菜单的子菜单共有 9 项,分别叙述如下:

(1) Load: 装入一个文件,可用类似 DOS 的通配符(如 *.c)来进行列表选择。也可装入其他扩展名的文件,只要给出文件名(或只给路径)即可。该项的热键为【F3】,即只要按【F3】即可进入该项,而不需要先进入 File 菜单再选此项。

(2) Pick: 将最近装入编辑窗口的 8 个文件列成一个表让用户选择,选择后将该程序装入编辑区,并将光标置在上次修改过的地方。其热键为【Alt】~【F3】。

(3) New: 新建文件,缺省文件名为“NONAME.C”,存盘时可改名。

(4) Save: 将编辑区中的文件存盘,若文件名是“NONAME.C”时,将询问是否更改文件名,其热键为【F2】。

(5) Write to: 可由用户给出文件名将编辑区中的文件存盘,若该文件已存在,则询问要不要覆盖。

(6) Directory: 显示目录及目录中的文件,并可由用户选择。

(7) Change dir: 显示当前默认目录,用户可以改变默认目录。

(8) Os shell: 暂时退出 Turbo C 2.0 到 DOS 提示符下,此时可以运行 DOS 命令,若想回到 Turbo C 2.0 中,只要在 DOS 状态下键入“EXIT”即可。

(9) Quit: 退出 Turbo C 2.0,返回到 DOS 操作系统中,其热键为【Alt+X】。

说明:

以上各项可用光标键移动色棒进行选择,回车则执行;也可用每一项的第一个大写字母直

接选择。若要退到主菜单或从它的下一级菜单列表框退回均可用【Esc】键,Turbo C 2.0 所有菜单均采用这种方法进行操作,以下不再说明。

2. Run 菜单

按【Alt+R】可进入 Run 菜单,该菜单有以下各项,如图 1-3 所示。



图 1-3 Run 菜单

(1) Run: 运行由 Project/Project name 项指定的文件名或当前编辑区的文件。如果对上次编译后的源代码未做过修改,则直接运行到下一个断点(没有断点则运行到结束)。否则先进行编译、连接后才运行,其热键为【Ctrl+F9】。

(2) Program reset: 中止当前的调试,释放分给程序的空间,其热键为【Ctrl+F2】。

(3) Go to cursor: 调试程序时使用,选择该项可使程序运行到光标所在行。光标所在行必须为一条可执行语句,否则提示错误,其热键为【F4】。

(4) Trace into: 在执行一条调用其他用户定义的子函数时,若用 Trace into 项,则执行长条将跟踪到该子函数内部去执行,其热键为【F7】。

(5) Step over: 执行当前函数的下一条语句,即使用户函数调用,执行长条也不会跟踪进函数内部,其热键为【F8】。

(6) User screen: 显示程序运行时在屏幕上显示的结果。其热键为【Alt+F5】。

3. Compile 菜单

按【Alt+C】可进入 Compile 菜单,该菜单有以下几个内容,如图 1-4 所示。

(1) Compile to OBJ: 将一个 C 源文件编译生成 .obj 目标文件,同时显示生成的文件名。其热键为【Alt+F9】。

(2) Make EXE file: 此命令生成一个 .exe 的文件,并显示生成的 .exe 文件名。其中 .exe 文件名是下面几项之一:

- ① 由 Project/Project name 说明的项目文件名。
- ② 若没有项目文件名,则为 Primary C file 说明的源文件。



图 1-4 Compile 菜单

③ 若以上两项都没有文件名,则为当前窗口的文件名。

(3) Link EXE file: 把当前. obj 文件及库文件连接在一起生成. exe 文件。

(4) Build all: 重新编译项目里的所有文件,并进行装配生成. exe 文件。该命令不作过时检查(上面的几条命令要作过时检查,即如果目前项目里源文件的日期和时间与目标文件相同或更早,则拒绝对源文件进行编译)。

(5) Primary C file: 当在该项中指定了主文件后,在以后的编译中,如没有项目文件名则编译此项中规定的主 C 文件,如果编译中有错误,则将此文件调入编辑窗口,不管目前窗口中是不是主 C 文件。

(6) Get info: 获得有关当前路径、源文件名、源文件字节大小、编译中的错误数目、可用空间等信息,如图 1-5 所示。

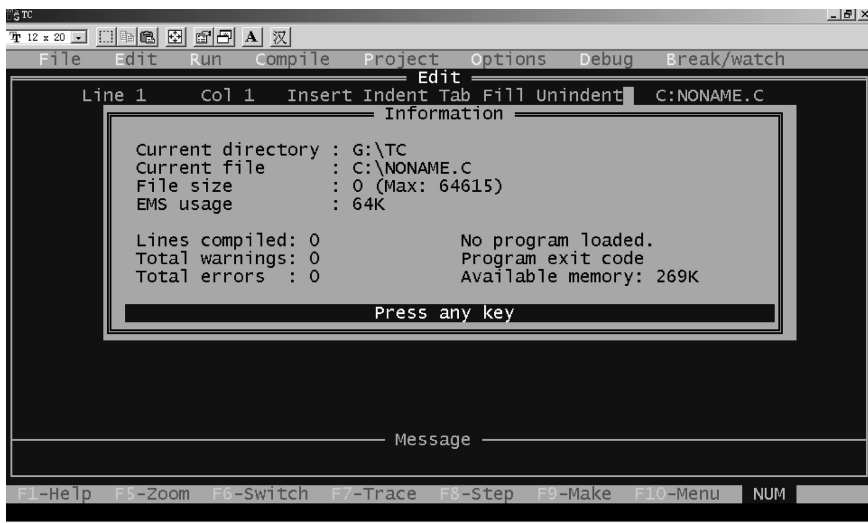


图 1-5 Get info 命令

4. Project 菜单

按【Alt+P】可进入 Project 菜单,该菜单包括以下内容,如图 1-6 所示。



图 1-6 Project 菜单

(1) Project name: 项目名具有 .PRJ 的扩展名,其中包括将要编译、连接的文件名。例如有一个程序由 file1.c、file2.c、file3.c 组成,要将这 3 个文件编译装配成一个 file.exe 的执行文件,可以先建立一个 file.prj 的项目文件,其内容如下:

```
file1.c
file2.c
file3.c
```

此时将 file.prj 放入 Project name 项中,以后进行编译时将自动对项目文件中规定的三个源文件分别进行编译,然后连接成 file.exe 文件。如果其中有些文件已经编译成 .obj 文件,而又没有修改过,可直接写上 .obj 扩展名。此时将不再编译而只进行连接。

例如:

```
file1.obj
file2.c
file3.c
```

将不对 file1.c 进行编译,而直接连接。

说明:

当项目文件中的每个文件无扩展名时,均按源文件对待,另外其中的文件也可以是库文件,但必须写上扩展名 .lib。

(2) Break make on: 由用户选择是否在有 Warning、Errors、Fatal Errors 时或 Link 之前退出 Make 编译。

(3) Auto dependencies: 当开关置为 on,编译时将检查源文件与对应的 .obj 文件日期和时间,否则不进行检查。

(4) Clear project: 清除 Project/Project name 中的项目文件名。

(5) Remove messages: 把错误信息从信息窗口中清除掉。

5. Debug 菜单

按【Alt+D】可选择 Debug 菜单,该菜单主要用于查错,它包括以下内容,如图 1-7 所示。



图 1-7 Debug 菜单

(1) Evaluate:

① Expression 要计算结果的表达式。

② Result 显示表达式的计算结果。

③ New value 赋给新值。

(2) Call stack: 该项不可接触,而在 Turbo C debugger 时用于检查堆栈情况。

(3) Find function: 在运行 Turbo C debugger 时用于显示规定的函数。

(4) Refresh display: 如果编辑窗口偶然被用户窗口重写了可用此恢复编辑窗口的内容。

任务三

程序的灵魂——算法



任务描述

从最简单的数学问题入手,通过对计算过程的模拟,理解算法的概念。



任务目标

1. 了解算法的含义。

2. 能用流程图描述算法。



知识储备

一个程序应包括：

(1) 对数据的描述。在程序中要指定数据的类型和数据的组织形式，即数据结构(data structure)。

(2) 对操作的描述。即操作步骤，也就是算法(algorithm)。

Nikiklaus Wirth 提出的公式：

$$\text{数据结构} + \text{算法} = \text{程序}$$

教材认为：

$$\text{程序} = \text{算法} + \text{数据结构} + \text{程序设计方法} + \text{语言工具和环境}$$

这 4 个方面是一个程序设计人员所应具备的知识。

本课程的目的是使同学知道怎样编写一个 C 程序，进行编写程序的初步训练，因此只介绍算法的初步知识。

一、算法的概念

做任何事情都有一定的步骤。为解决一个问题而采取的方法和步骤，就称为算法。

(1) 计算机算法：计算机能够执行的算法。

(2) 计算机算法可分为两大类：

- ① 数值运算算法：求解数值。
- ② 非数值运算算法：事务管理领域。

二、简单算法举例

【例 1-2】求 $1 \times 2 \times 3 \times 4 \times 5$ 。

最原始方法：

步骤 1：先求 1×2 ，得到结果 2。

步骤 2：将步骤 1 得到的乘积 2 乘以 3，得到结果 6。

步骤 3：将 6 再乘以 4，得 24。

步骤 4：将 24 再乘以 5，得 120。

这样的算法虽然正确，但太繁琐。

改进的算法：

S1：使 $t=1$ ；

S2：使 $i=2$ ；

S3：使 $t \times i$ ，乘积仍然放在在变量 t 中，可表示为 $t \times i \rightarrow t$ ；

S4：使 i 的值 +1，即 $i+1 \rightarrow i$ ；

S5: 如果 $i \leq 5$, 返回重新执行步骤 S3 以及其后的 S4 和 S5; 否则, 算法结束。

如果计算 $100!$ 只需将 S5: 若 $i \leq 5$ 改成 $i \leq 100$ 即可。

如果是求 $1 \times 3 \times 5 \times 7 \times 9 \times 11$, 算法也只需做很少的改动:

S1: $1 \rightarrow t$;

S2: $3 \rightarrow i$;

S3: $t \times i \rightarrow t$;

S4: $i+2 \rightarrow i$;

S5: 若 $i \leq 11$, 返回 S3; 否则, 结束。

该算法不仅正确, 而且是计算机较好的算法, 因为计算机是高速运算的自动机器, 实现循环轻而易举。

想一想 若将 S5 写成, S5: 若 $i < 11$, 返回 S3; 否则, 结束。

【例 1-3】 有 50 个学生, 要求将他们之中成绩在 80 分以上者打印出来。

如果, n 表示学生学号, n_i 表示第 i 个学生学号; g 表示学生成绩, g_i 表示第 i 个学生成绩; 则算法可表示如下:

S1: $1 \rightarrow i$;

S2: 如果 $g_i \geq 80$, 则打印 n_i 和 g_i , 否则不打印;

S3: $i+1 \rightarrow i$;

S4: 若 $i \leq 50$, 返回 S2; 否则, 结束。

【例 1-4】 判定 2000~2500 年中的每一年是否闰年, 将结果输出。

闰年的条件:

- (1) 能被 4 整除, 但不能被 100 整除的年份;
- (2) 能被 100 整除, 又能被 400 整除的年份(图 1-8)。

设 y 为被检测的年份, 则算法可表示如下:

S1: $2000 \rightarrow y$;

S2: 若 y 不能被 4 整除, 则输出 y “不是闰年”, 然后转到 S6;

S3: 若 y 能被 4 整除, 不能被 100 整除, 则输出 y “是闰年”, 然后转到 S6;

S4: 若 y 能被 100 整除, 又能被 400 整除, 则输出 y “是闰年”, 否则输出 y “不是闰年”, 然后转到 S6;

S5: 输出 y “不是闰年”;

S6: $y+1 \rightarrow y$;

S7: 当 $y \leq 2500$ 时, 返回 S2 继续执行; 否则, 结束。

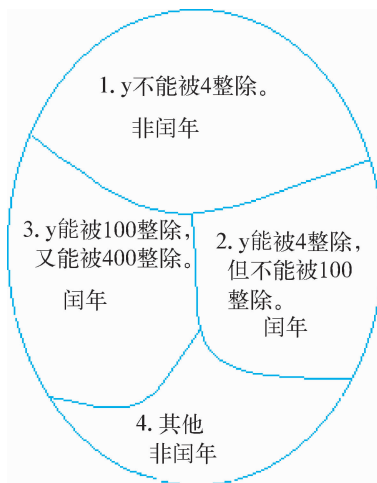


图 1-8 闰年判断

【例 1-5】 求 $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{99} - \frac{1}{100}$ 。

算法可表示如下：

- S1: $sigh=1$;
- S2: $sum=1$;
- S3: $deno=2$;
- S4: $sigh=(-1)\times sigh$;
- S5: $term=sigh\times(1/deno)$;
- S6: $term=sum+term$;
- S7: $deno=deno+1$;
- S8: 若 $deno\leq 100$, 返回 S4; 否则, 结束。

【例 1-6】 对一个大于或等于 3 的正整数, 判断它是否为素数。

算法可表示如下：

- S1: 输入 n 的值;
- S2: $i=2$;
- S3: n 被 i 除, 得余数 r ;
- S4: 如果 $r=0$, 表示 n 能被 i 整除, 则打印 n “不是素数”, 算法结束; 否则执行 S5;
- S5: $i+1\rightarrow i$;
- S6: 如果 $i\leq n-1$, 返回 S3; 否则打印 n “是素数”, 然后算法结束。

改进：

- S6: 如果 $i\leq \sqrt{n}$, 返回 S3; 否则打印 n “是素数”, 然后算法结束。

三、算法的特性

- (1) 有穷性：一个算法应包含有限的操作步骤而不能是无限的。
- (2) 确定性：算法中每一个步骤应当是确定的, 而不能是含糊的、模棱两可的。
- (3) 有零个或多个输入。
- (4) 有一个或多个输出。
- (5) 有效性：算法中每一个步骤应当能有效地执行, 并得到确定的结果。

对于程序设计人员, 必须会设计算法, 并根据算法写出程序。

四、怎样表示一个算法

- 1. 用自然语言表示算法
除了很简单的问题, 一般不用自然语言表示算法。
- 2. 用流程图表示算法
流程图表示算法, 直观形象, 易于理解(图 1-9)。

【例 1-7】 将例 1-2 中求 $5!$ 的算法用流程图表示(图 1-10)。

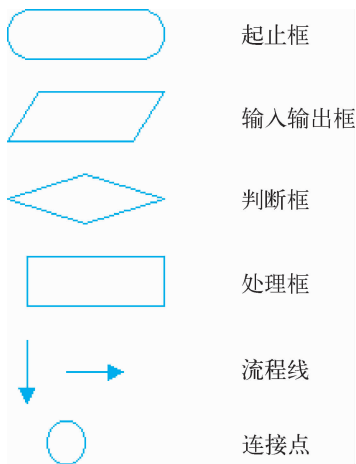


图 1-9 用流程图表示算法

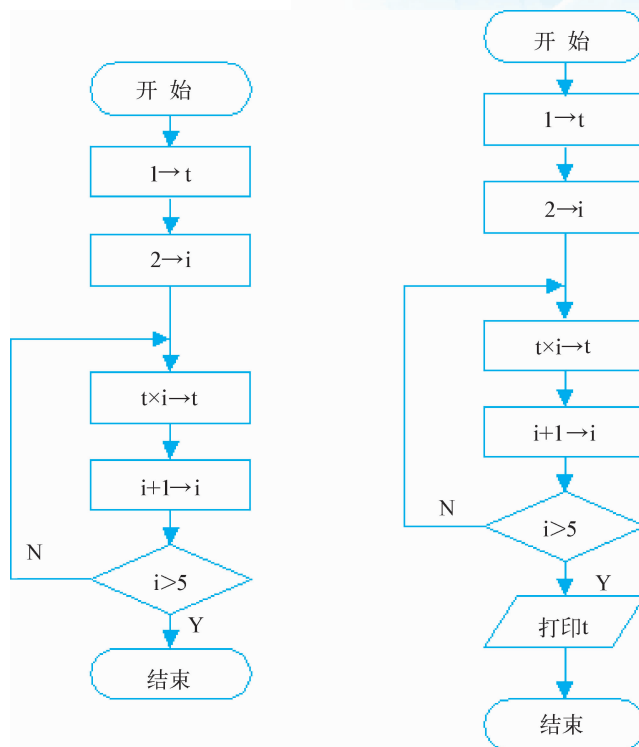


图 1-10 求 5! 的算法流程图

【例 1-8】将例 1-3 中的算法用流程图表示(图 1-11)。

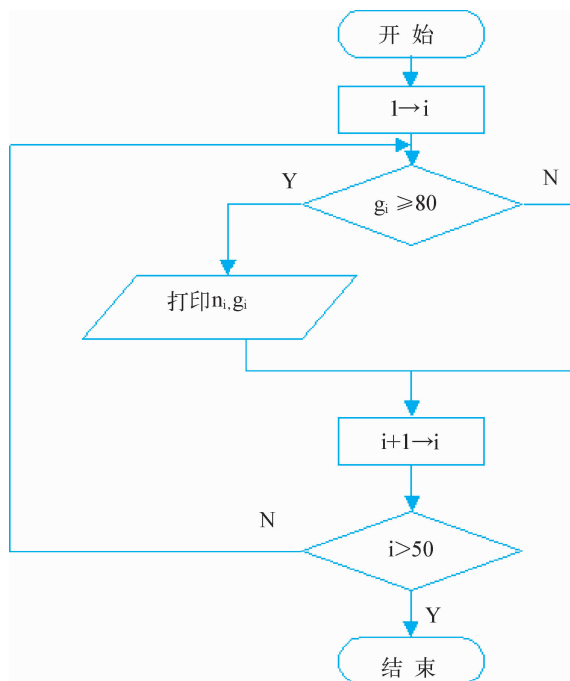


图 1-11 例 1-3 算法流程图

【例 1-9】将例 1-4 中判定闰年的算法用流程图表示(图 1-12)。

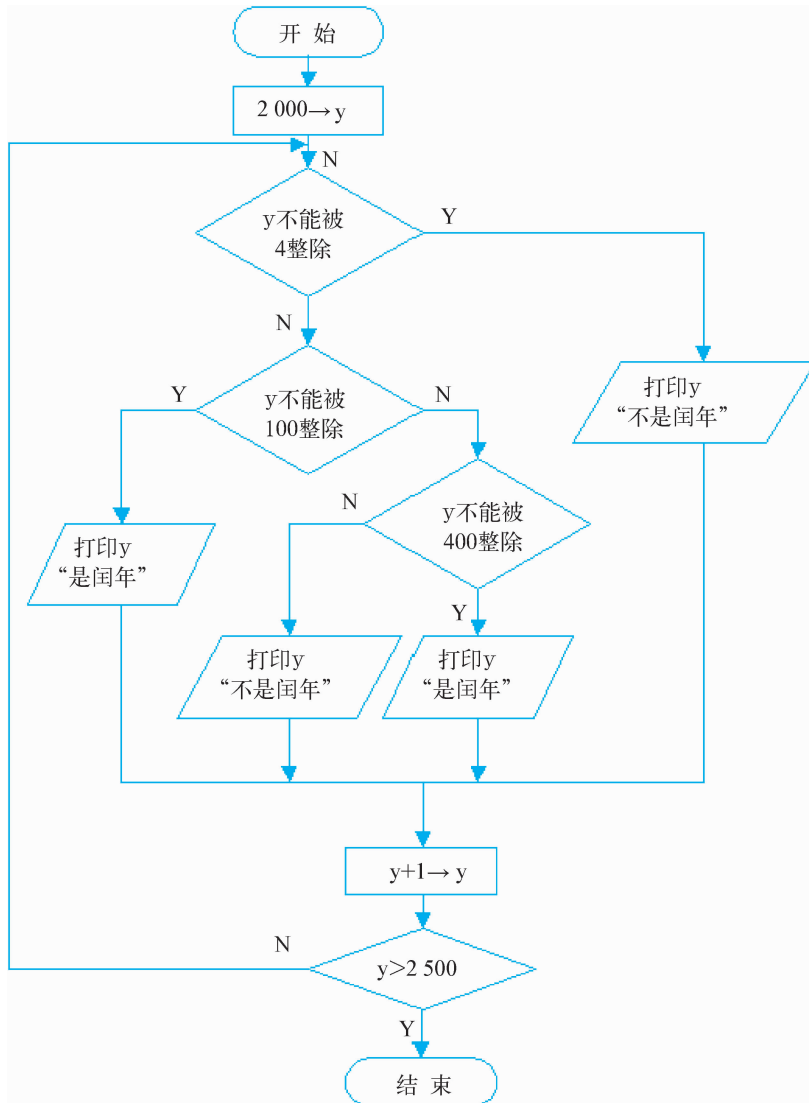


图 1-12 判定闰年的算法流程图

一个流程图应包括：

- (1) 表示相应操作的框。
- (2) 带箭头的流程线。
- (3) 框内外必要的文字说明。

3. 三种基本结构和改进的流程图

- (1) 顺序结构,如图 1-13 所示。
- (2) 选择结构,如图 1-14 所示。
- (3) 循环结构,如图 1-15 所示。

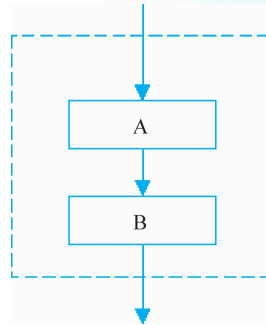


图 1-13 顺序结构

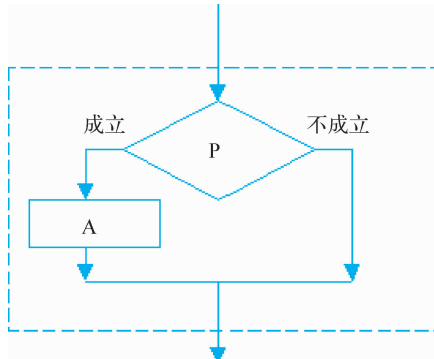
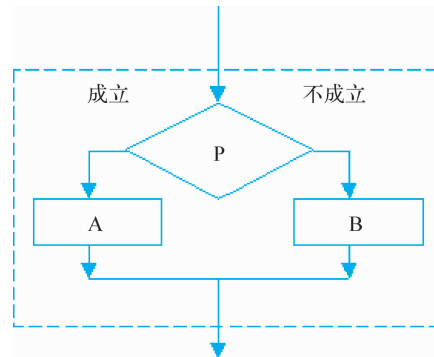


图 1-14 选择结构

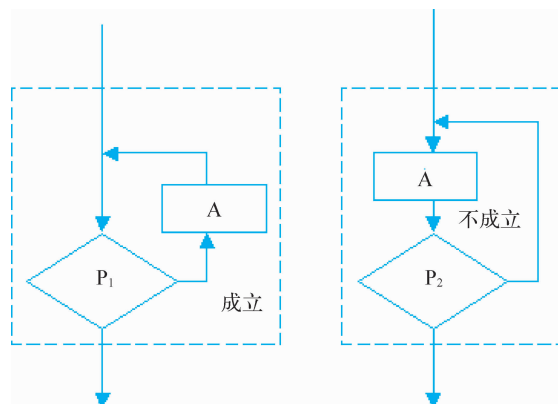


图 1-15-1 循环结构-1

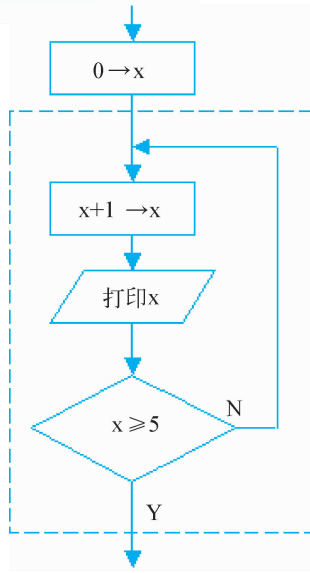


图 1-15-2 循环结构-2

三种基本结构的共同特点为：

- (1) 只有一个入口。
- (2) 只有一个出口。
- (3) 结构内的每一部分都有机会被执行到。
- (4) 结构内不存在“死循环”。

4. 用 N-S 流程图表示算法

1973 年美国学者提出了一种新型流程图：N-S 流程图。

(1) 顺序结构(图 1-16)：

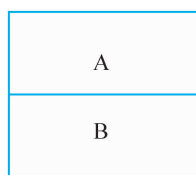


图 1-16 顺序结构

(2) 选择结构(图 1-17)：

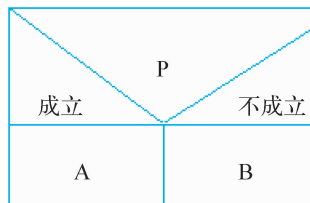


图 1-17 选择结构