第一章

C语言概述

- 本章要点

  - C语言的特点
  - C程序的结构
  - 在计算机上运行C程序的方法

§1-1 C语言出现的历史背景

- C语言是国际上广泛流行的高级语言。
- C语言是在B语言的基础上发展起来的。
- B （BCPL）语言是1970年由美国贝尔实验室设计的，并用于编写了第一个UNIX操作系统,在PDP 7上实现。优点：精练,接近硬件，缺点：过于简单,数据无类型。
- 1973年贝尔实验室的D.M.Ritchie 在B语言的基础上设计出了C语言，对B取长补短，并用之改写了原来用汇编编写的UNIX，（即UNIX第5版），但仅在贝尔实验室使用。

§1-1 C语言出现的历史背景

- 1975年UNIX第6版发布,C优点突出引起关注。
- 1977年出现了《可移植C语言编译程序》，推动了UNIX在各种机器上实现，C语言也得到推广，其发展相辅相成。
- 1978年影响深远的名著《The C Programming Language》由 Brian W.Kernighan和Dennis M.Ritchie 合著,被称为标准C。
- 之后，C语言先后移植到大、中、小、微型计算机上,已独立于UNIX和PDP，风靡世界,成为最广泛的几种计算机语言之一。

§1-1 C语言出现的历史背景

- 1983年,美国国家标准化协会(ANSI)根据C语言各种版本对C的发展和扩充,制定了新的标准ANSI C，比标准C有了很大的发展。
- 1988年K & R按照 ANSI C修改了他们的《The C Programming Language》。
- 1987年,ANSI公布了新标准——87 ANSI C。
- 1990年，国际标准化组织接受了87 ANSI C为ISO C 的标准(ISO9899—1990)。
- 1994年，ISO又修订了C语言标准。
- 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。

§1-1 C语言出现的历史背景

说明：

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，因此读者应了解所用的C语言编译系统的特点(可以参阅有关手册)。本书的叙述基本上以ANSI C 为基础。

§1-2 C语言的特点

（1）语言简洁、紧凑,使用方便、灵活。 32个关键字、9种控制语句,程序形式自由

（2）运算符丰富。34种运算符

（3）数据类型丰富,具有现代语言的各种数据结构。

（4）具有结构化的控制语句 ，是完全模块化和结构化的语言。

（5）语法限制不太严格,程序设计自由度大。

§1-2 C语言的特点

（6）允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可直接对硬件进行操作。兼有高级和低级语言的特点 。

（7）目标代码质量高，程序执行效率高。只比汇编程序生成的目标代码效率低10%-20%。

（8）程序可移植性好(与汇编语言比)。基本上不做修改就能用于各种型号的计算机和各种操作系统。

§1-2 C语言的特点

问题： 既然有了面向对象的C++语言，为什么还要学习C语言？

解释1： C++是由于开发大型应用软件的需要而产生的，并不是所有的人都要去编写大型软件；

解释2： 面向对象的基础是面向过程。C++是面向对象的语言，C是面向过程的，学起来比C语言困难得多，所以不太适合程序设计的初学者。

§1-3 简单的C

说明： 本程序的作用是输出一行信息：
This is a C program.

```
#include <stdio.h>                    /*文件包含*/
void  main( )                         /*主函数 */
{                                     /*函数体开始*/
  printf ("This is a C program.\n");  /*输出语句*/
}                                     /*函数体结束*/
```

说明： main-主函数名， void-函数类型

- 每个C程序必须有一个主函数main
- { }是函数开始和结束的标志,不可省
- 每个C语句以分号结束
- 使用标准库函数时应在程序开头一行写：

  #include <stdio.h>

说明： 输出一行信息:sum is 579

例1.2 求两数之和

```
#include <stdio.h>
void main( )        /*求两数之和*/
{
   int a,b,sum;     /*声明，定义变量为整型*/
   /*以下3行为C语句 */
   a=123; b=456;
   sum=a+b;
   printf(" sum is %d\n" ,sum);
}
```

说明： /*……*/表示注释。注释只是给人看的,对编译和运行不起作用。所以可以用汉字或英文字符表示，可以出现在一行中的最右侧，也可以单独成为一行。

例1.3 求3个数中较大者

```
#include <stdio.h>
void main( )    /* 主函数*/
{
  int max(int x,int y); / 对被调用函数max的声明 */
  int a, b, c;          /*定义变量a、b、c */
  scanf(" %d,%d",&a,&b); /*输入变量a和b的值*/
  c=max(a,b); /*调用max函数,将得到的值赋给c*/
  printf(" max=%d\\n",c);
```

- 程序运行情况如下：
- 8,5 ↙（输入8和5赋给a和b）
- max=8 （输出c的值）

```
int max(int x,int y);
{
  int z;
  if (x>y) z=x;
  else z=y;
  return (z);
}
```

说明： 本程序包括main和被调用函数max两个函数。max函数的作用是将x和y中较大者的值赋给变量z。return语句将z的值返回给主调函数main。

§1-3 简单的C语言程序介绍

C程序：

(1) C程序是由函数构成的。 这使得程序容易实现模块化。

(2) 一个函数由两部分组成：

函数的首部：例1.3中的max函数首部

```
int max(int x,int y )
```

函数体：花括号内的部分。若一个函数有多个花括号,则最外层的一对花括号为函数体的范围。

函数体包括两部分 ：

声明部分：int a,b,c; 可缺省

执行部分：由若干个语句组成。可缺省

§1-3 简单的C语言程序介绍

注意：

函数的声明部分和执行部分都可缺省，例如：

```
void dump ( )
{
}
```

这是一个空函数,什么也不做,但是合法的函数。

§1-3 简单的C语言程序介绍

小结：

(3) C程序总是从main函数开始执行的，与main函数的位置无关。

(4) C程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行上，C程序没有行号。

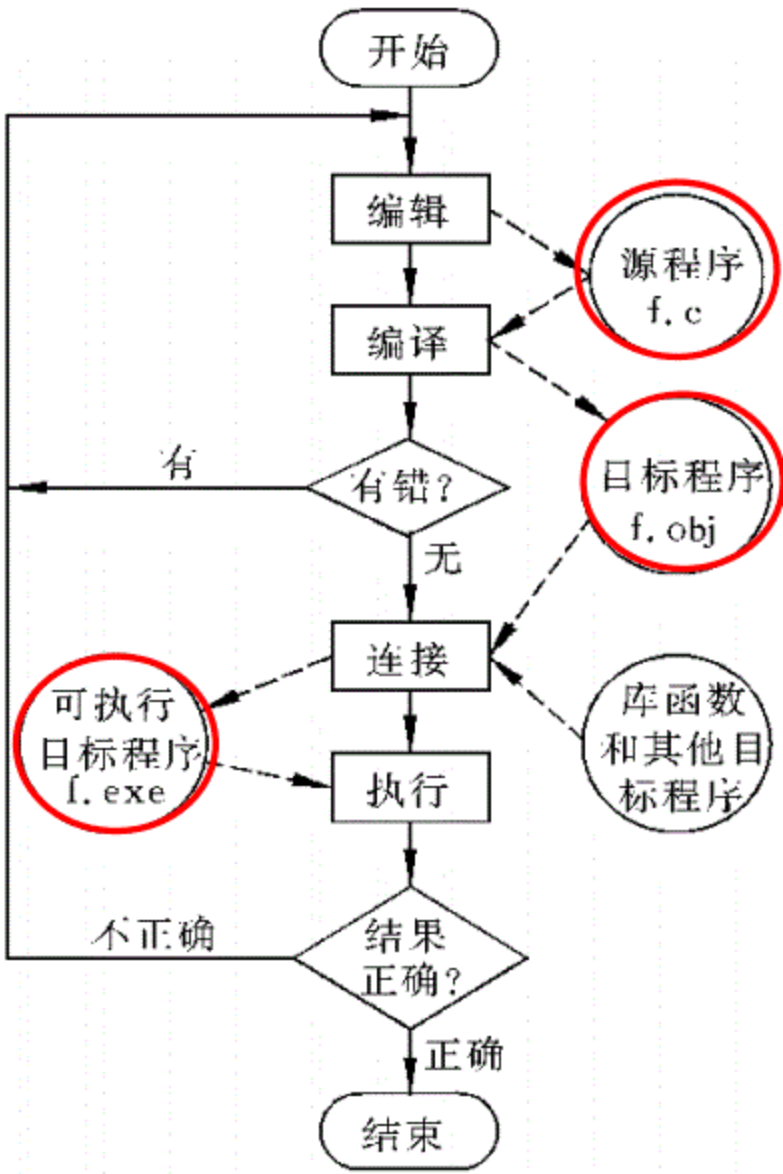
(5) 每个语句和数据声明的最后必须有一个分号。

(6) C语言本身没有输入输出语句。输入和输出的操作是由库函数scanf和printf等函数来完成的。C对输入输出实行“函数化”。

§1-4 运行C程序的步骤

一、运行C程序的步骤

- 上机输入与编辑源程序
- 对源程序进行编译
- 与库函数连接
- 运行目标程序



§1-4 运行C程序的步骤和方法

二、上机运行C程序的方法

- 目前使用的大多数C编译系统都是集成环境(IDE)的。可以用不同的编译系统对C程序进行操作
- 常用的有Turbo C 2.0、Turbo C++ 3.0、Visual C++等
- Turbo C++ 3.0：是一个集成环境，它具有方便、直观和易用的界面，虽然它也是DOS环境下的集成环境，但是可以把启动Turbo C++ 3.0 集成环境的DOS执行文件tc.exe生成快捷方式，也可以用鼠标操作。
- Visual C++：也可以用Visual C++对C程序进行编译。

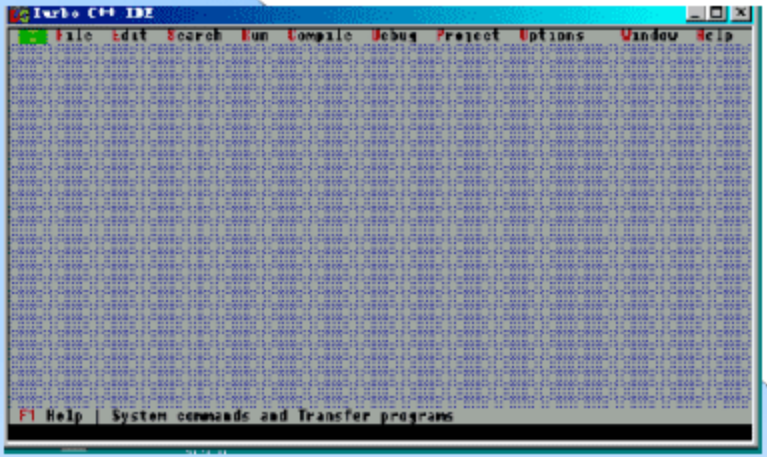
例：Turbo C++ 3.0的使用

将Turbo C++ 3.0编译程序装入磁盘某一目录下，例如：放在C盘根目录下一级TC3.0子目录下。

(1) 进入Turbo C++ 3.0集成环境

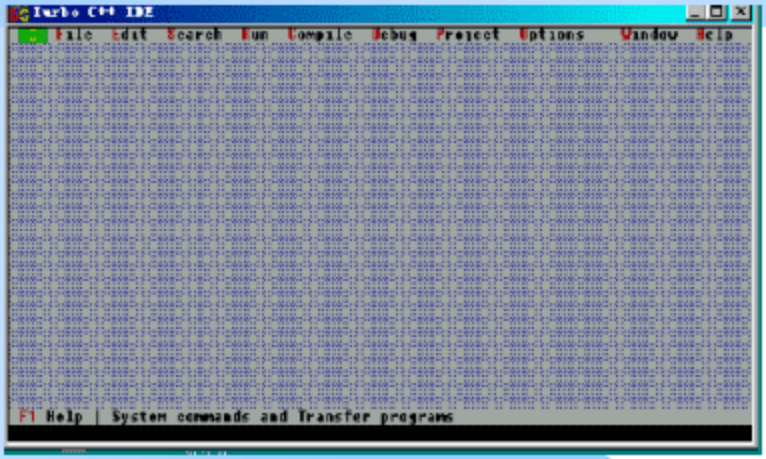
①在DOS环境下

C:\TC3.0>tc ↙



② 在Windows环境下

找到可执行文件tc.exe，执行该文件。

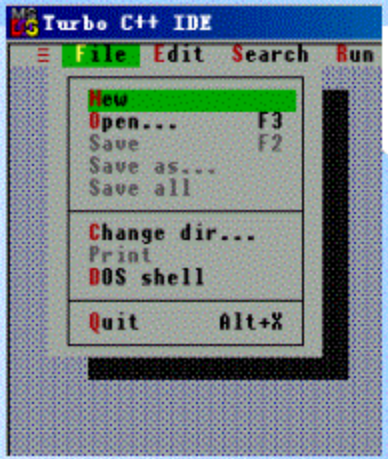


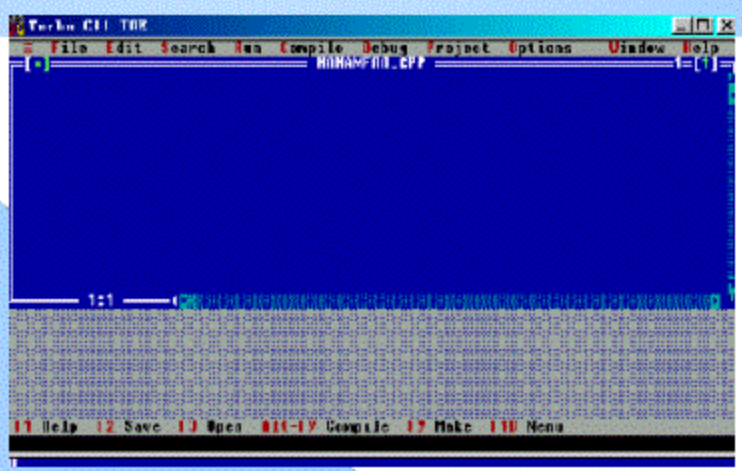
主菜单：11个菜单项:

File Edit Search Run Compile Debug Project Options Window Help

(2) 编辑源文件

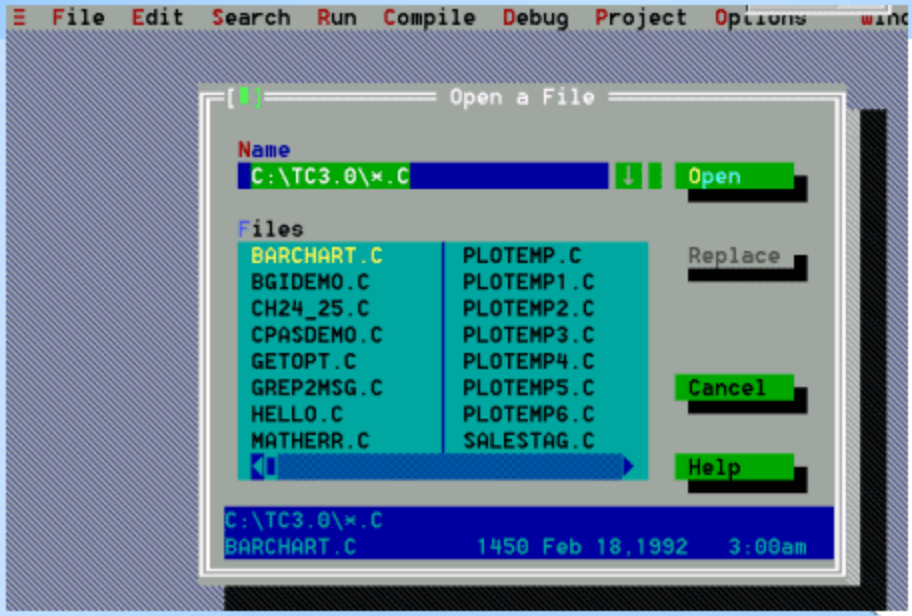
新建：单击“File”菜单下 的“New”，



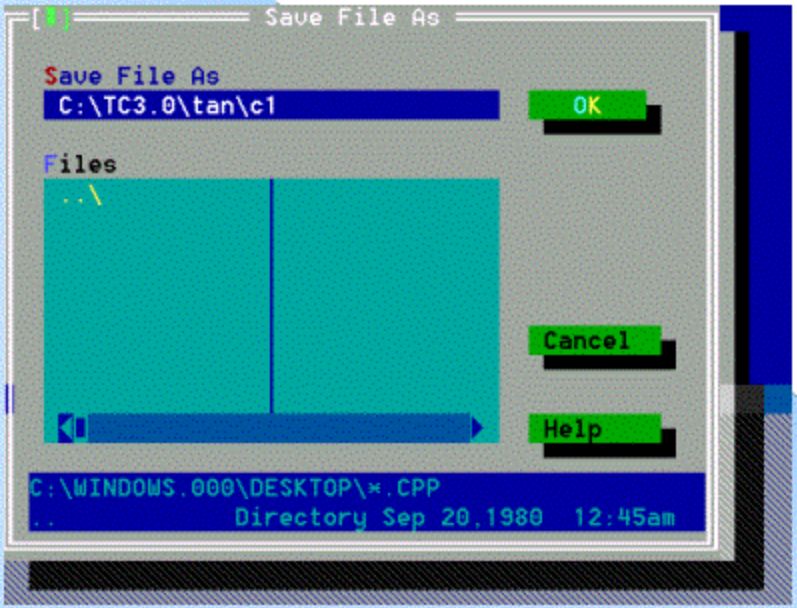


修改：选择“File”→“Open”(即单击“File”的下拉菜单中的“Open”项，修改已有的源程序。

在编辑(EDIT) 状态下光标表示当前进行编辑的位置，在此位置可以进行插入、删除或修改，直到自己满意为止。

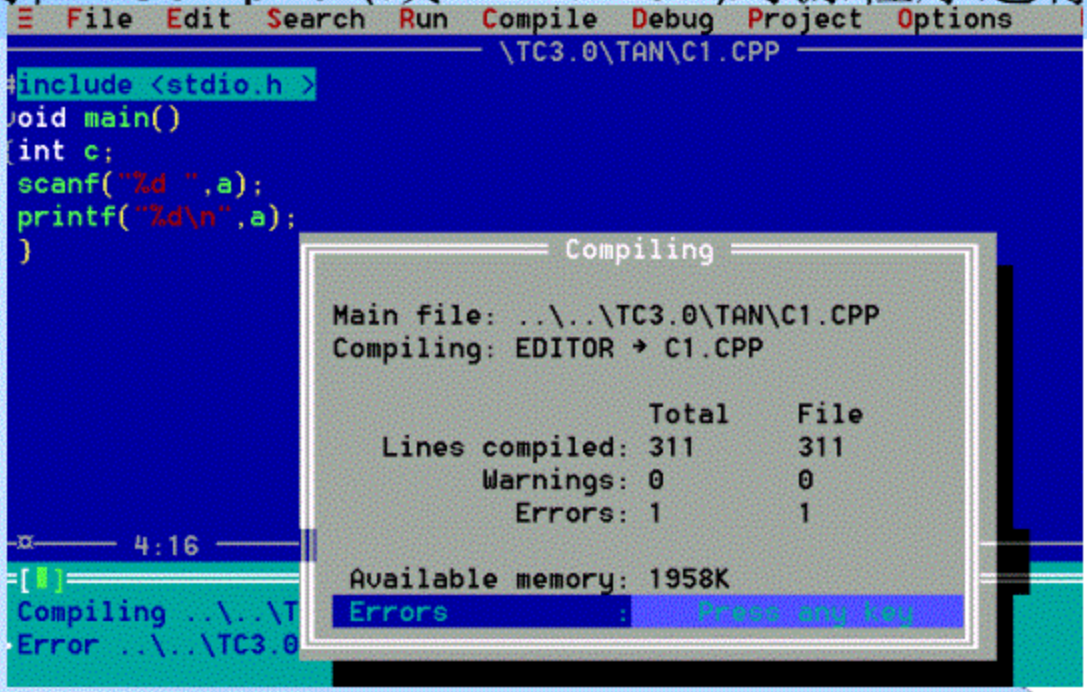


保存：在编辑(EDIT) 状态下光标表示当前进行编辑的位置，
在此位置可以进行插入、删除或修改，直到自己满意为止。



(3) 对源程序进行编译

选择“Compile”(或“Alt+F9”)对源程序进行编译。



c1.cpp源程序，出现1个错误(error)，0个警告(warming)。

(4) 将目标程序进行连接

选择菜单“Compile”→“Link”，如果不出现错误，会得到一个后缀为.exe的可执行文件。

(5) 执行程序。

选菜单“Run”→“Run”(或按“Ctrl+F9”键)。

(6) 退出Turbo C++ 3.0环境

选择“File”→“Quit”。

第二章

程序的灵魂——算法

- 本章要点

  - 算法的概念
  - 算法的表示
  - 结构化程序设计方法

- 主要内容

2.1 算法的概念

2.2 简单算法举例

2.3 算法的特性

2.4 怎样表示一个算法

2.5 化程序设计方法

一个程序应包括两个方面的内容:

- 对数据的描述：数据结构(data structure)
- 对操作的描述：算法(algorithm)

著名计算机科学家沃思提出一个公式:

数据结构＋算法＝程序

完整的程序设计应该是:

数据结构＋算法＋程序设计方法＋语言工具

§2.1 算法的概念

广义地说，为解决一个问题而采取的方法和步骤，就称为“算法”。

对同一个问题，可有不同的解题方法和步骤

例：求 $\sum_{n=1}^{100} n$

- 方法1：1+2，+3，+4，一直加到100 加99次
- 方法2：100+(1+99)+(2+98)+…+(49 +51)+50
  $= 100 + 49 \times 100 + 50$ 加51次

§2.1 算法的概念

为了有效地进行解题，不仅需要保证算法正确，还要考虑算法的质量，选择合适的算法。希望方法简单，运算步骤少。

计算机算法可分为两大类别：

- 数值运算算法：求数值解，例如求方程的根、求函数的定积分等。
- 非数值运算：包括的面十分广泛，最常见的是用于事务管理领域，例如图书检索、人事管理、行车调度管理等。

§2.2 简单算法举例

太繁琐

例2.1：求1×2×3×4×5

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再乘以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

如果要求1×2×…×1000，则要写999个步骤

可以设两个变量：一个变量代表被乘数，一个变量代表乘数。不另设变量存放乘积结果，而直接将每一步骤的乘积放在被乘数变量中。设p为被乘数，i为乘数。用循环算法来求结果，算法可改写：

S1：使p=1

S2：使i=2

S3：使p×i，乘积仍放在变量p中，可表示为：p×ip

S4：使i的值加1，即i+1i。

S5：如果i不大于5，返回重新执行步骤S3以及其后的步骤S4和S5；否则，算法结束。最后得到p的值就是5!的值。

如果题目改为：求1×3×5×……×1000

算法只需作很少的改动：

S1：1p
S2：3i
S3：p×ip
S4：i+2p
S5：若i≤11，返回S3。否则，结束。

算法简练

用这种方法表示的算法具有通用性、灵活性。S3到S5组成一个循环，在实现算法时 要反复多次执行S3，S4，S5等步骤，直到某一时刻，执行S5步骤时经过判断，乘数i已超过规定的数值而不返回S3步骤为止。此时算法结束，变量p的值就是所求结果。

例2.2 有50个学生，要求将他们之中成绩在80分以上者打印出来。设n表示学号，n_1代表第一个学生学号，代表第i个学生学号。用G代表学生成绩，g_i代表第i个学生成绩，算法表示如下：

S1：1i

S2：如果≥80，则打印和，否则不打印。

S3：i+1i

S4：如果i≤50，返回S2，继续执行。否则算法结束

变量i作为下标，用来控制序号(第几个学生，第几个成绩)。当i超过50时，表示 已对50个学生的成绩处理完毕，算法结束。

例2.3 判定2000～2500年中的每一年是否闰年，将结果输出。

分析： 闰年的条件是：(1) 能被4整除，但不能被100整除的年份都是闰年，如1996, 2004年是闰年；(2) 能被100整除，又能被400整除的年份是闰年。如1600, 2000年是闰年。不符合这两个条件的年份不是闰年。

变量i作为下标，用来控制序号(第几个学生，第几个成绩)。当i超过50时，表示 已对50个学生的成绩处理完毕，算法结束。

设y为被检测的年份，算法可表示如下：

S1：2000y

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，输出y“是闰年”，否则输出“不是闰年”。 然后转到S6。

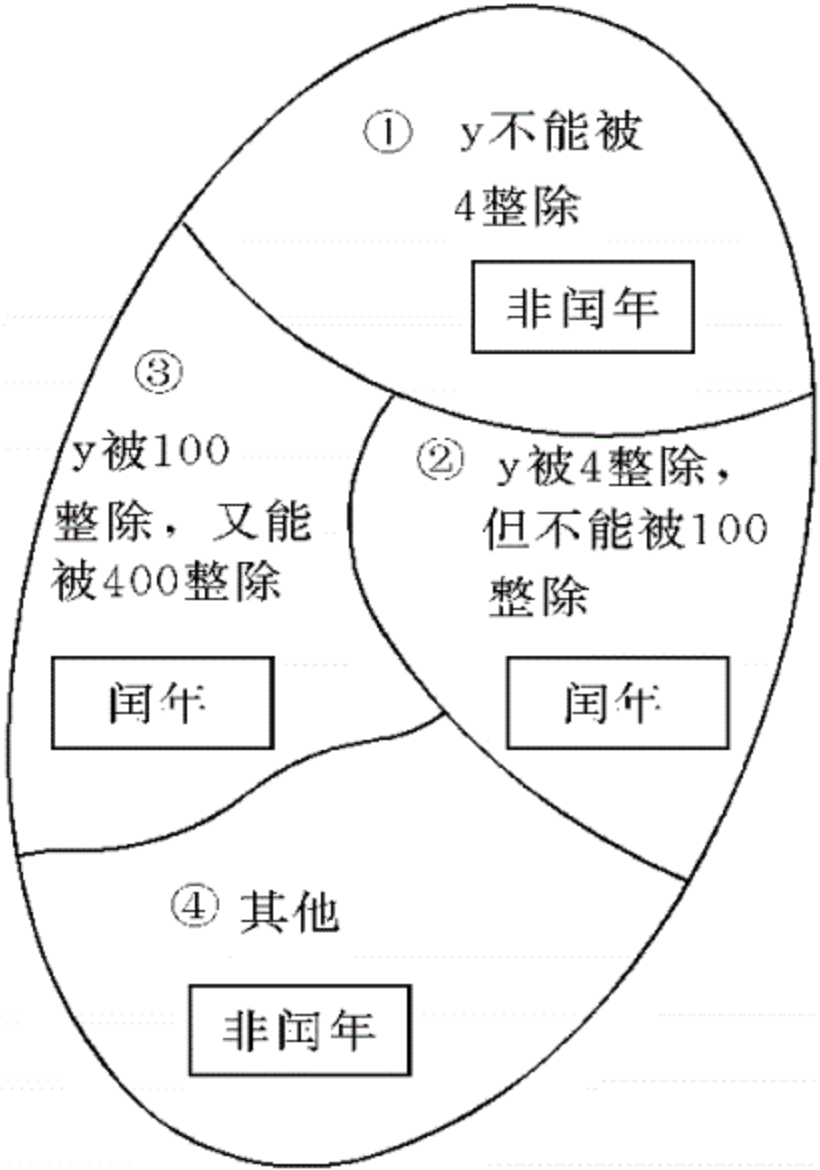
S5：输出y“不是闰年”。

S6：y+1y

S7：当y≤2500时，转S2继续执行，如y＞2500，算法停止。

以上算法中每做一步都分别分离出一些范围(已能判定为闰年或非闰年)，逐步缩小范围，直至执行S5时，只可能是非闰年。

“其它” 包括能被4整除，又能被100整除，而不能被400整除的那些年份(如1990) 是非闰年。



例2.4 求 $1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\cdots\cdots-\frac{1}{99}+\frac{1}{100}$ 算法如下：

S1: sign=1

S2: sum=1

S3: deno=2

S4: sign=(-1) × sign

S5: term=sign × (1/deno)

S6: sum=sum+term

S7: deno=deno+1

S8: 若deno≤100返回S4，否则算法结束。

单词作变量名，以使算法更易于理解：
sum表示累加和，deno是英文分母（denom inator）缩写，sign代表数值的符号，term代表某一项。

反复执行S4到S8步骤，直到分母大于100为止。
一共执行了99次循环，向sum累加入了99个分数。
sum最后的值就是多项式的值。

例2.5 对一个大于或等于3的正整数，判断它是不是一个素数。

概念： 所谓素数，是指除了1和该数本身之外，不能被其它任何整数整除的数。例如，13是素数。因为它不能被2，3，4，…，12整除。

分析： 判断一个数n (n ≥ 3) 是否素数的方法：

将n作为被除数，将2到 (n-1) 各个整数轮流作为除数，如果都不能被整除，则n为素数。

算法如下：

S1：输入n的值

S2：i=2 （i作为除数）

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 i

S6：如果i≤n-1，返回S3。否则打印 n “是素数”。然后结束。

实际上，n不必被2到(n-1)的整数除，只需被2到n/2间整数除，甚至只需被2到$\sqrt{n}$之间的整数除即可。

§2.3 算法的特性

一个算法应该具有以下特点：

- 有穷性：包含有限的操作步骤
- 确定性：算法中的每一个步骤都应当是确定的
- 有零个或多个输入：输入是指在执行算法时需要从外界取得必要的信息
- 有一个或多个输出：算法的目的是为了求解，“解”就是输出
- 有效性：算法中的每一个步骤都应当能有效地执行，并得到确定的结果。

§2.4 算法的表示

可以用不同的方法表示算法，常用的有：

- 自然语言
- 传统流程图
- 结构化流程图
- 伪代码
- PAD图

§2.4.1 用自然语言表示算法

自然语言就是人们日常使用的语言，可以是汉语或英语或其它语言。用自然语言表示通俗易懂，但文字冗长，容易出现“歧义性”。自然语言表示的含义往往不大严格，要根据上下文才能判断其正确含义，描述包含分支和循环的算法时也不很方便。因此，除了那些很简单的问题外，一般不用自然语言描述算法。

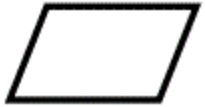
§2.4.2 用流程图表示算法

美国国家标准化协会ANSI(American National Standard Institute)规定了一些常用的流程图符号:

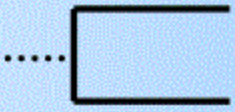
起止框

判断框

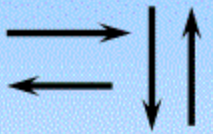
处理框



输入/输出框



注释框

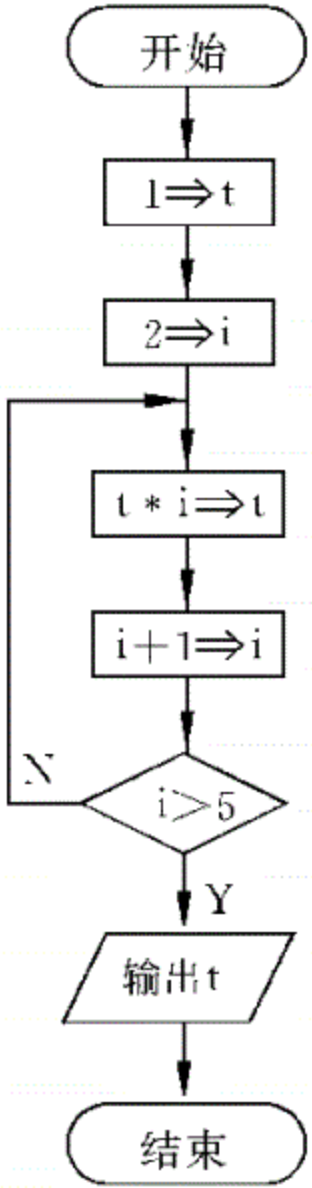


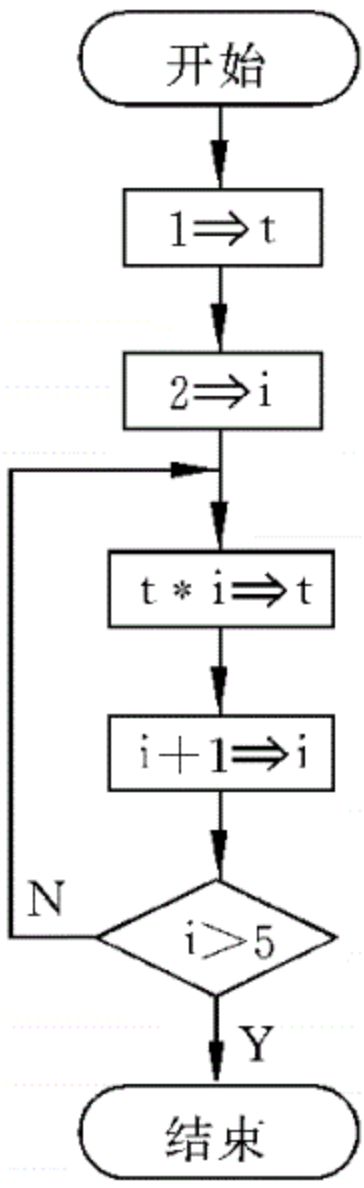
流向线

连接点

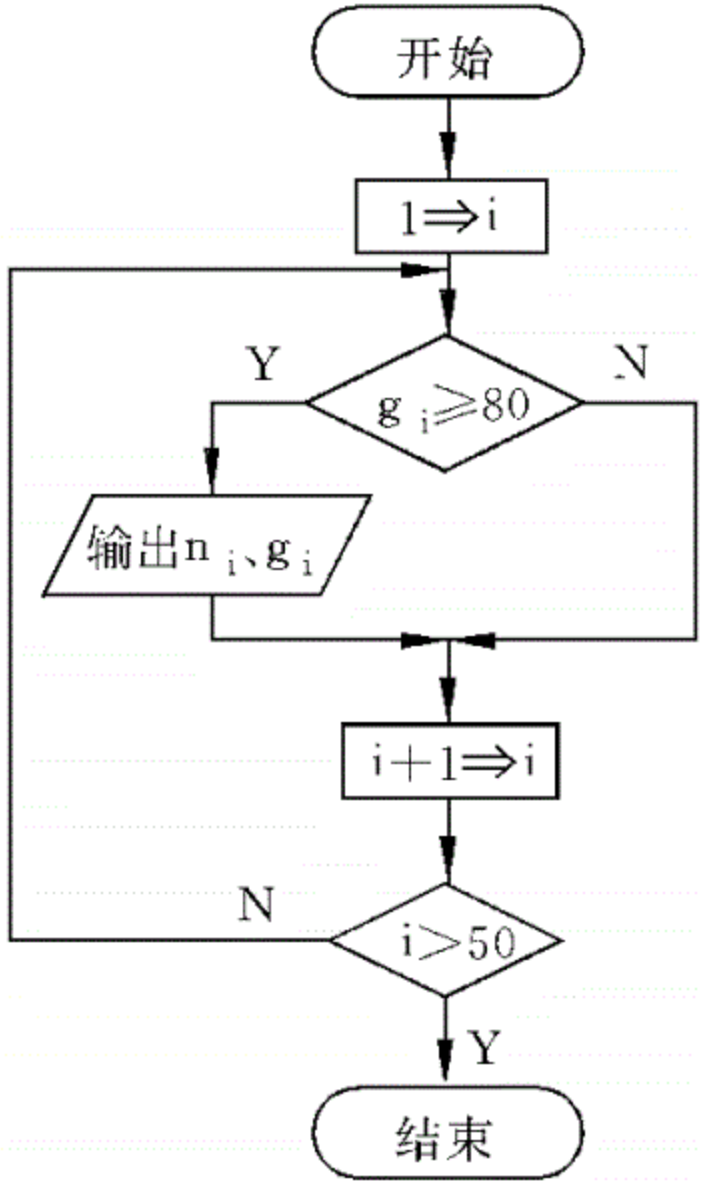
例2.6 将求5!的算法用流

如果需要将最后结果打印出来，可在菱形框的下面加一个输出框。

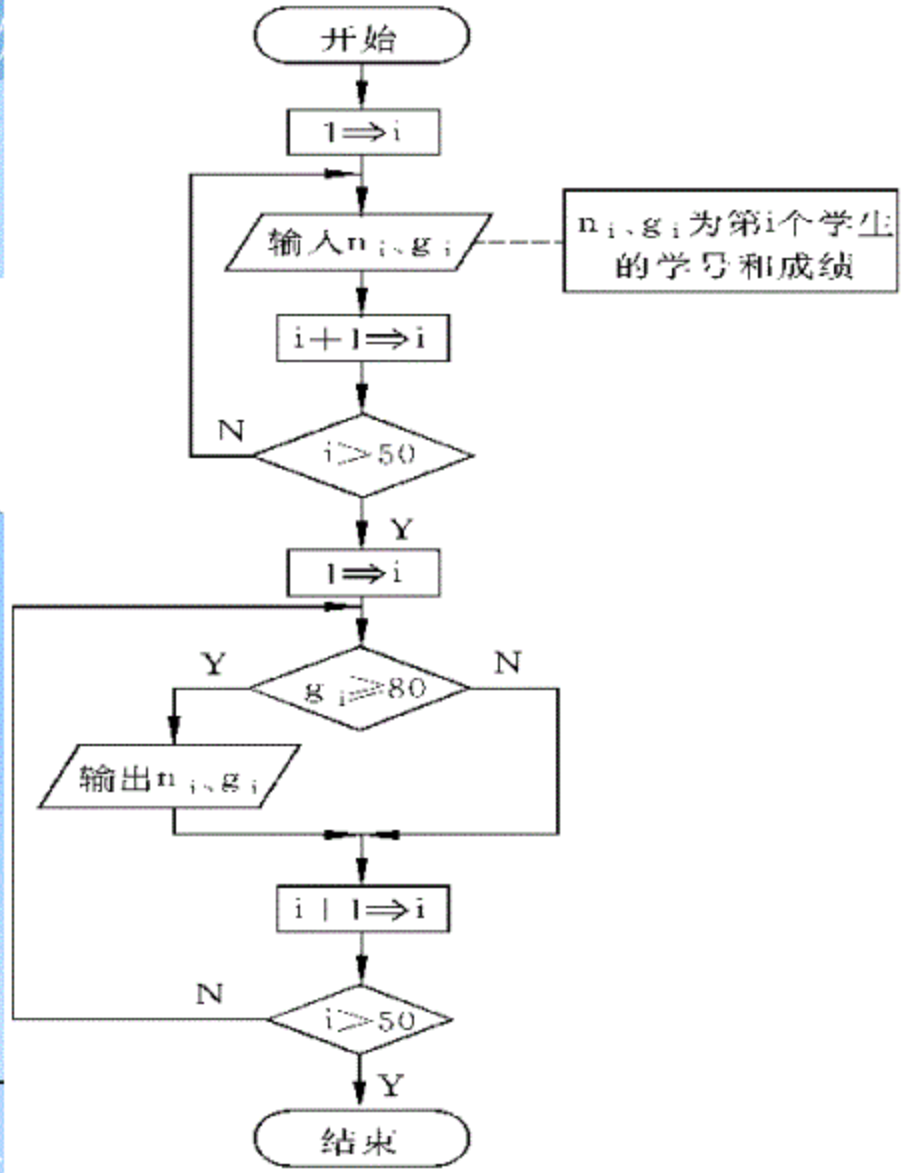




例2.7 将例2.2的算法用流程图表示。打印50名 学生中成绩在80分以上者的学号和成绩。

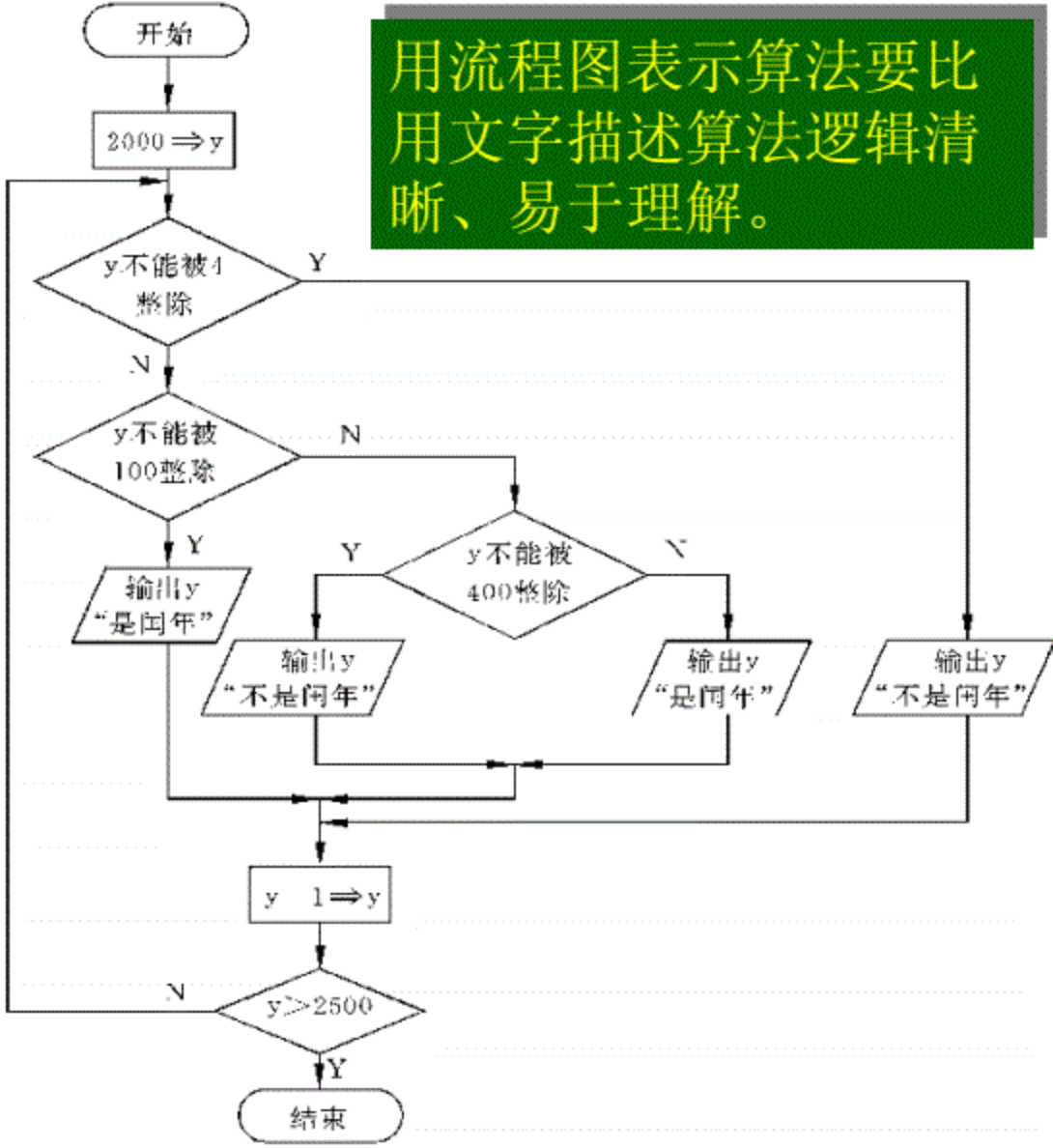


如果如果包括这个输入数据的部分，流程图为



例2.8 将例2.3判定闰年的算法用流程图表示

用流程图表示算法要比用文字描述算法逻辑清晰、易于理解。



例2.9 将例2.4的算法用流程图

$$1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\cdots\cdots-\frac{1}{99}+\frac{1}{100}$$