信息与电气工程学院

C程序设计案例教程

第1章 计算机程序设计算法与实现

本章主要内容有

- 程序设计语言
- 程序设计过程
- 程序设计算法与
- 计算机程序算法的表示
  - -自然语言描述
  - -程序流程图描述
  - -N-S图描述
  - -结构化程序设计和模块化结构
  - -程序设计语言描述
- 程序算法实现案例分析

1.1程序设计概述

计算机技术应用领域博大广泛，程序设计算法实现是基础

程序设计不仅是学习计算机语言的语义语法规则本身，更重要的是

学会用计算机程序设计语言解决实际问题的过程，即程序算法分析与实现过程。

其**核心**是掌握用计算思维的方法。实现问题的求解的过程。

1.1.1程序设计语言分类

TIOBE编程语言排行榜

（2021.09）

Sep 2021	Sep 2020	Change	Programming Language	Ratings	Change
1	1		C	11.83%	-4.12%
2	3	↑	Python	11.67%	+1.20%
3	2	↓	Java	11.12%	-2.37%
4	4		C++	7.13%	+0.01%
5	5		C#	5.78%	+1.20%
6	6		Visual Basic	4.62%	+0.50%
7	7		JavaScript	2.55%	+0.01%
8	14	⇈	Assembly language	2.42%	+1.12%
9	8	↓	PHP	1.85%	-0.64%
10	10		SQL	1.80%	+0.04%
11	22	⇈	Classic Visual Basic	1.52%	+0.77%
12	17	⇈	Groovy	1.46%	+0.48%
13	15	↑	Ruby	1.27%	+0.03%
14	11	↓	Go	1.13%	-0.33%
15	12	↓	Swift	1.07%	-0.31%
16	16		MATLAB	1.02%	-0.07%
17	37	⇈	Fortran	1.01%	+0.65%
18	9	⇊	R	0.98%	-1.40%
19	13	⇊	Perl	0.78%	-0.53%
20	29	⇈	Delphi/Object Pascal	0.77%	+0.24%

1.1.2 算法实现-程序设计过程

计算机解决实际问题的一般过程如图1.1所示。

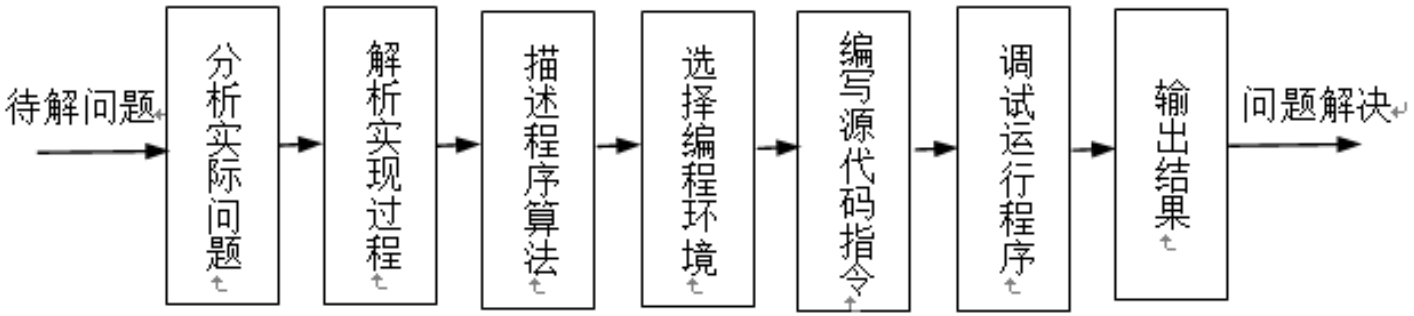


图 1.1　计算机程序设计实现过程

分析问题及建立算法模型是实现具体算法的重要步骤，一般应有必要的专业基础与专业应用背景，最好还应具备一定的实际经验，因此常常由具有专业技术背景人员与程序编写员共同协作完成

你的第一个C语言程序

- Hello world!

```
#include "stdio.h"
int main()
{
    printf("Hello world!");
    return 0;
}
```

1.2 程序设计算法-结构与组成

★程序包括的内容：

- ❖数据结构：数据的类型和组织形式
- ❖算法：操作步骤的描述

★ Nikiklaus Wirth提出：

程序 = 数据结构+算法

★ 教材认为：

程序 =算法+数据结构+程序设计方法+语言工具和环境

灵魂　加工对象　工具

算法的概念

★为解决一个问题而采取的方法和步骤，就成为算法。例如：歌曲的乐谱，建造房子等。

★算法核心是解决“做什么”和“怎么做”的问题。

- ❖求1.....5之和
- ❖求1.....10000之和
- ❖判断一个数字是否是质数
- ❖找到100以内的所有质数
- ❖判断闰年

- ❖可以有多种方法，一般采用简单和运算步骤少的。

准确、高效

简单程序设计案例

例如，编程实现计算任意圆半径和高度的圆柱体体积。

由数学模型（数学语言）

$$v=\pi\times r^2\times h$$

计算得到圆柱体体积值，将数据存入变量v，获得体积v值。其中符号以字母标识。

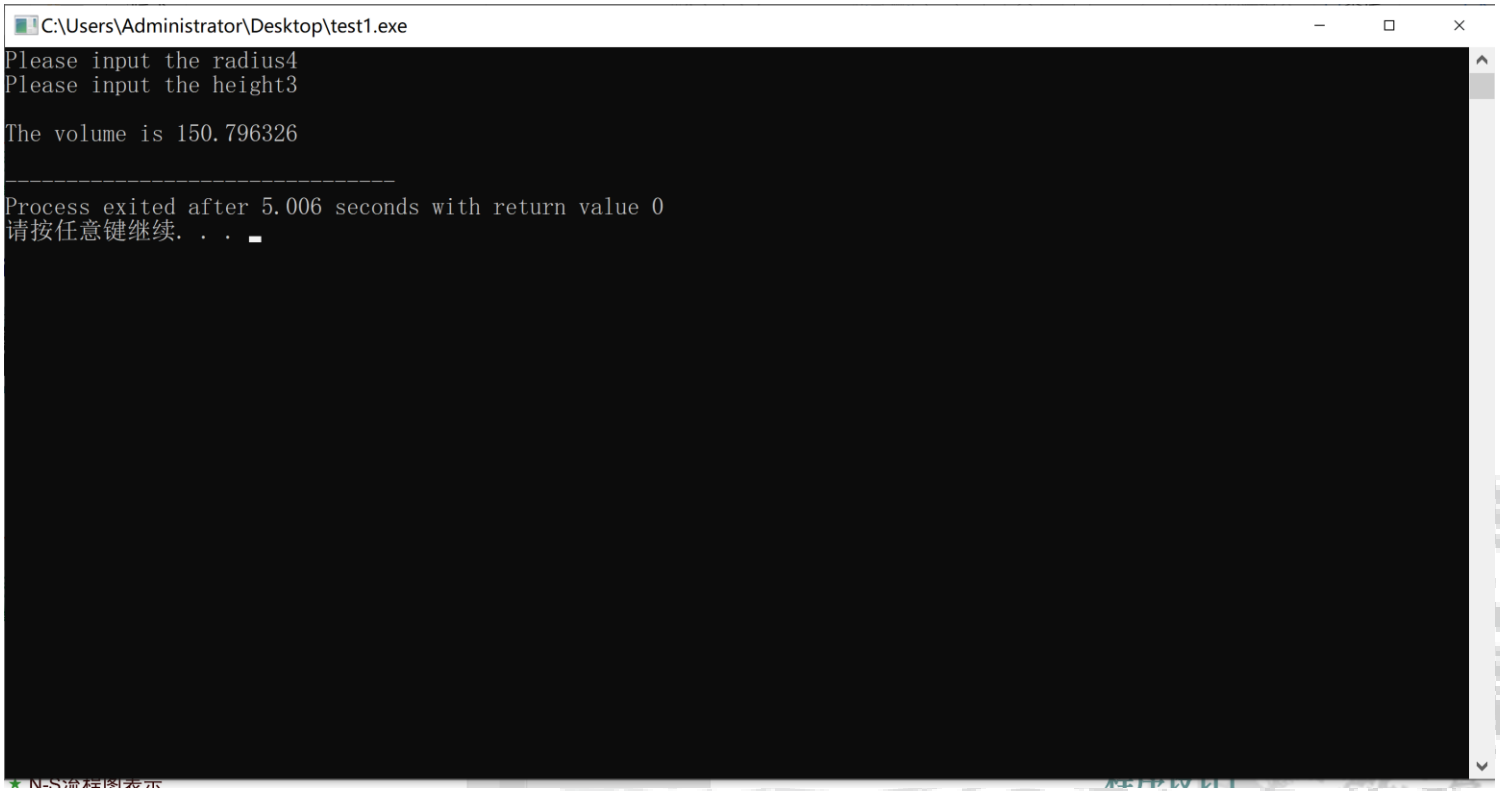
又如，若π值运算精度确定时，可用常量定义代换：

```
#define  PI  3.14159
```

例1 编程实现计算任意圆半径和高度的圆柱体体积。

```
#include "stdio.h"                /*宏定义说明使用输入输出函数*/
#define  PI  3.14159              /*宏定义PI宏代换常量*/
int main()
{
 float r,h,v;                     /*定义3个浮点型数值变量*/
 printf("Please input the radius");    /*输出显示提示信息*/
 scanf("%f",&r);                  /*程序运行时从键盘输入半径值*/
 printf("Please input the height");    /*输出显示提示信息*/
 scanf("%f",&h);                  /*程序运行时从键盘输入高度值*/
 v= PI*r*r*h;                     /*计算体积*/
 printf("\nThe volume is %f\n",v);   /*输出圆柱体积计算结果*/
  return 0;
}
```

源程序分别在Dev C++环境下载入程序运行、输入数据及输出结果如图1.2所示.



1.3计算机算法的表示

- ★自然语言表示
  - ❖易懂，文字冗长，易歧义性
- ★流程图表示
  - ❖用流程图符号构成，直观，易懂
- ★ N-S流程图表示
- ★伪代码表示
- ★计算机语言表示

1.3计算机算法的表示

1.3.1 自然语言描述

例如，执行求和运算；完成按班级排序的数据操作；执行按人名检索数据等。

1.3计算机算法的表示

1.3.2 程序流程图描述

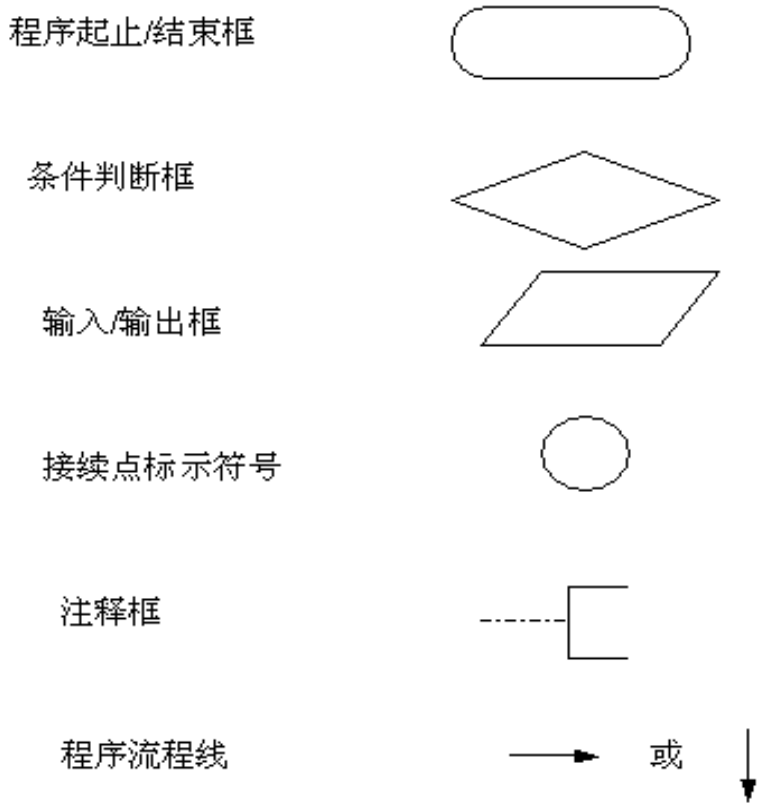
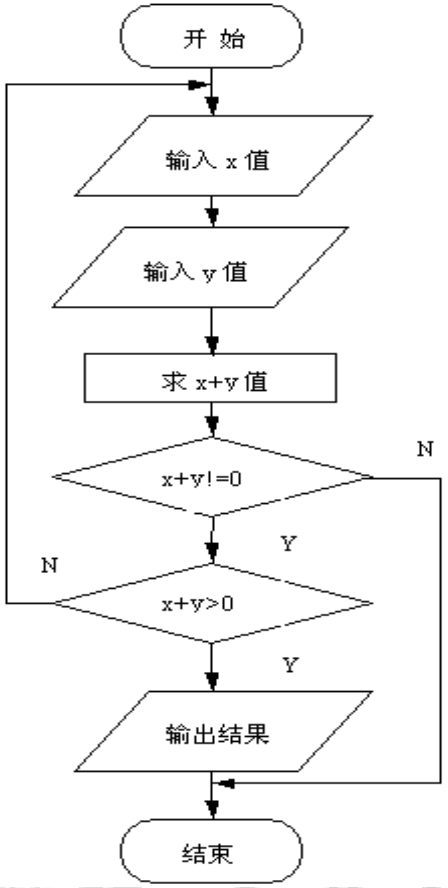


图 1.2 程序流程图基本符号



1.3.2 程序流程图描述

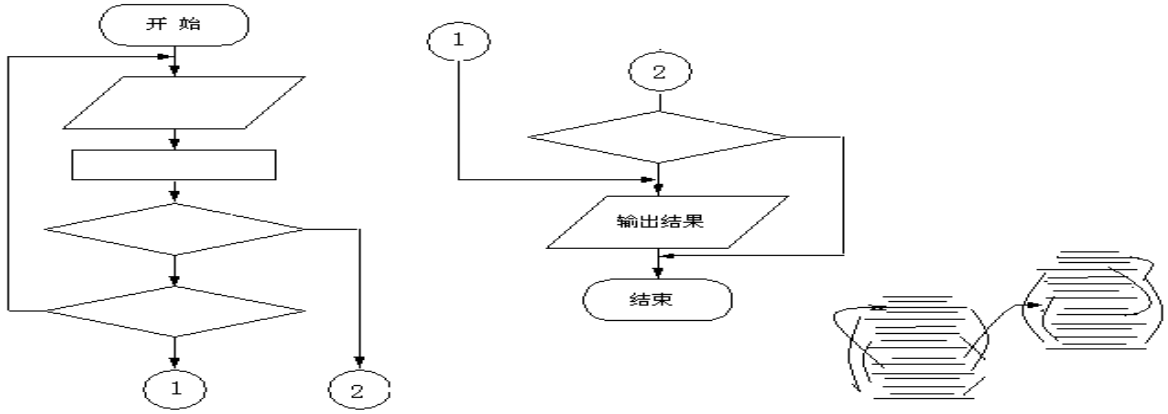


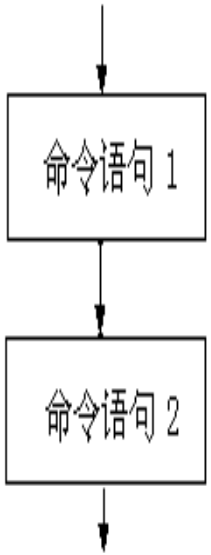
图1.4 表示复杂的算法看似乱麻

传统流程图流向混乱、可读性差，所以应该采用结构化流程图。

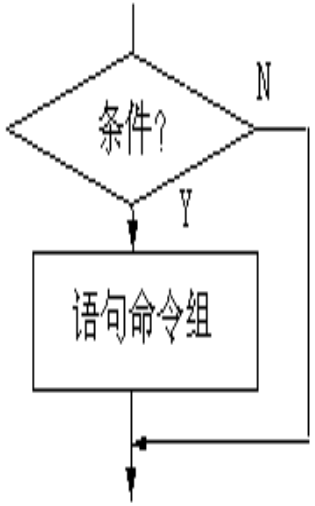
★ 结构化程序设计

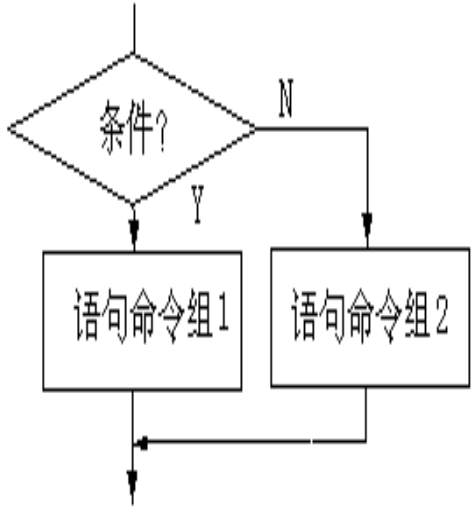
- ❖ 基本思想：任何程序都可以用三种基本结构表示，限制使用无条件转移语句（goto）
- ❖ 结构化程序：由三种基本结构反复嵌套构成的程序
- ❖ 优点：结构清晰，易读，提高程序设计质量和效率

1.3.2 程序流程图描述

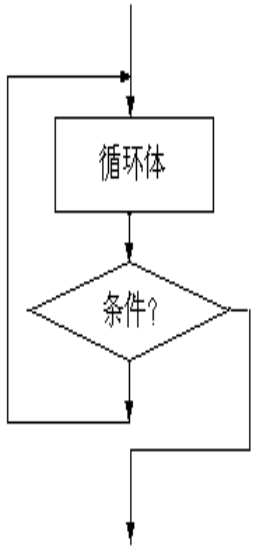


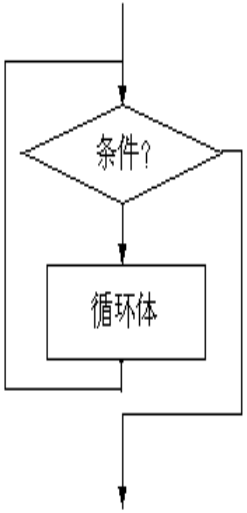
顺序结构





条件判断分支结构





循环控制结构

1.3.3 N-S图描述

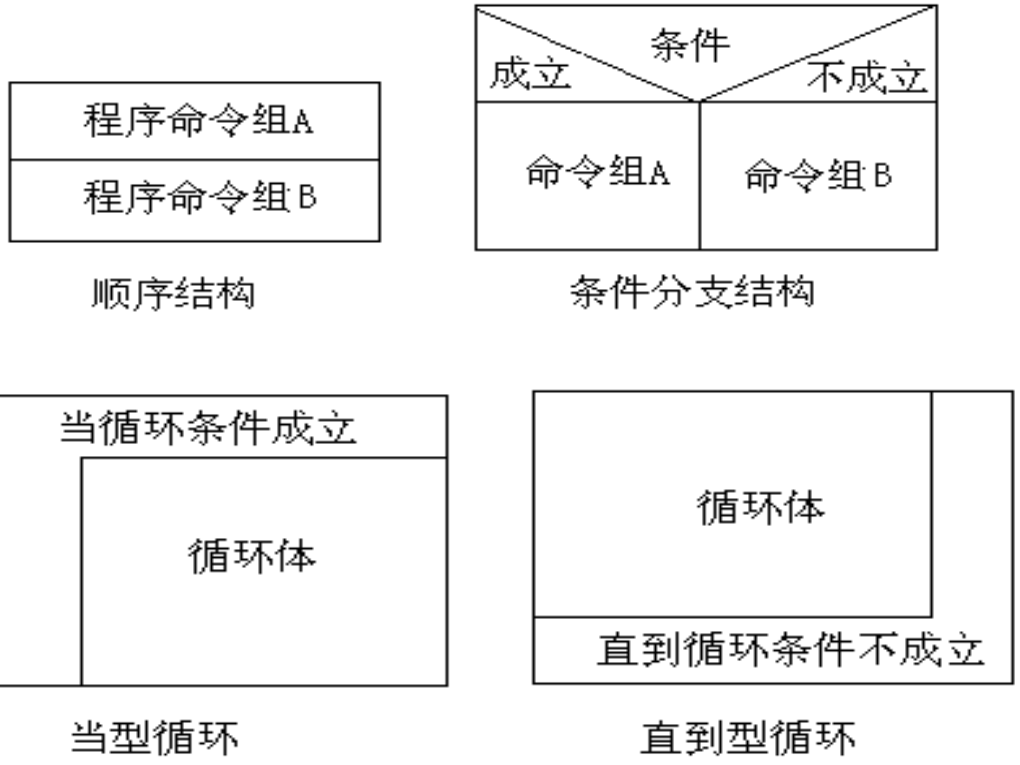


图1.3 N-S图表示三种基本结构

1.3.4 结构化程序设计方法

1. 三种基本结构的共同特点：

(1) 只有一个入口。

(2) 只有一个出口。请注意，一个菱形判断框有两个出口，而一个选择结构只有一个出口。不要将菱形框的出口和选择结构的出口混淆。

(3) 结构内的每一部分都有机会被执行到。对每一个框来说，都应有一条从入口到出口的路径通过它。

(4) 结构内不存在“死循环”(无终止的循环)。图1.16就是一个死循环。

其实，基本结构不一定只限于上面三种，只要具有上述4个特点的都可以作为基本结构。人们可以自己定义基本结构，并由这些基本结构组成结构化程序。

但是，人们都普遍认为最基本的是本节介绍的三种基本结构。

1.3.4 结构化程序设计方法

★结构化程序：

★用三种基本结构组成的程序

★已经证明，由以上三种基本结构顺序组成的算法结构，可以解决任何复杂的问题。

★基本设计思路：

❖复杂问题分解成 几个最基本问题，再分别处理。

★采用的方法：

❖自顶向下；逐步细化；

❖模块化设计：复杂问题按功能分成多个子模块

❖结构化编码：正确采用三种基本结构实现

算法举例

计算1+2+3+4+5+.....+100之和，可表示为

$$sum = \sum_{i=1}^{100} i$$

分析：这是一个自然数升序连续加法运算问题，设一个变量i作为存放加数的变量，同时可用来累计加法运算的次数，再设一个变量sum用来存放连续加法运算的累加值。

算法步骤分析：

S1: 累加器变量sum 赋初值0 ， 即sum=0

S2: 计数器变量i 赋初值1 ， 即i=1

S3: 使累加器变量值sum加计数器变量值i, 结果仍放在sum中, 即sum=sum+i, 此时sum值为 sum=sum+i=0+1=1

S4: 使计数器变量i加1,结果仍放在i中, 即 i=i+1, 此时i值为

i=i+1=1+1=2

S5: 使累加器变量值sum加计数器变量值i,,结果仍放在sum中， 即sum=sum+i, 此时sum值为sum=sum+i=1+2=3

S6: 使i加1,结果仍放在i中, 即i=i+1, 此时i值为

i=i+1=2+1=3

S7: 使sum加i,结果仍放在sum中， 可表示为sum=sum+i, 此时sum值为

sum=sum+i=3+3=6

S8: 使i加1,结果仍放在i中， 可表示为i=i+1, 此时i值为

i=i+1=3+1=4

S9: 使sum加i,结果仍放在sum中., 可表示为sum=sum+i, 此时sum值为

sum=sum+i=6+4=10

.....

程序流程图如图1.4所示，N-S图如图1.5所示。

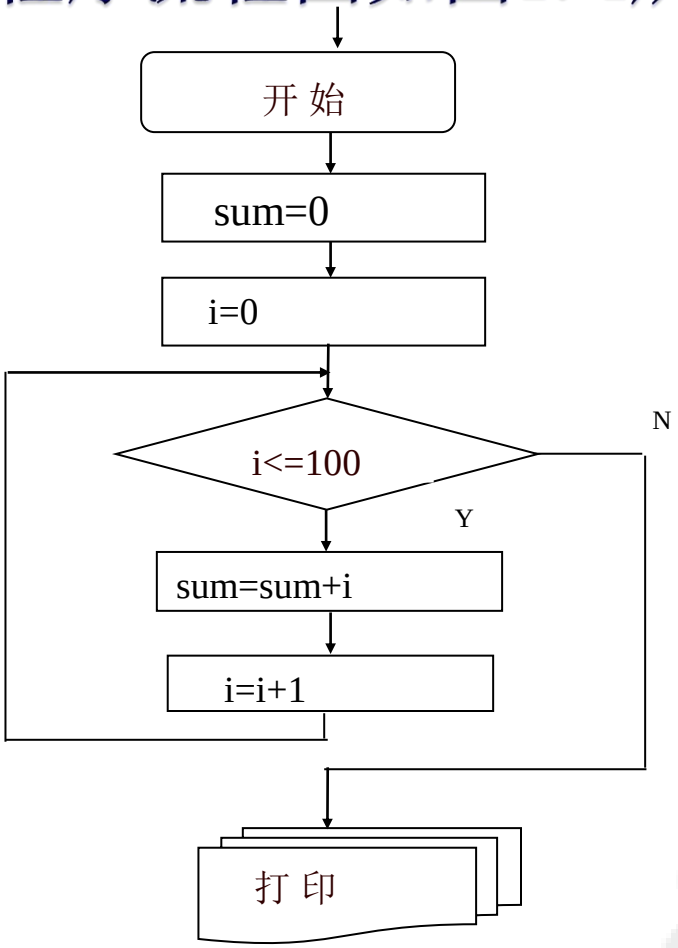


图1.4 累加运算程序流程图

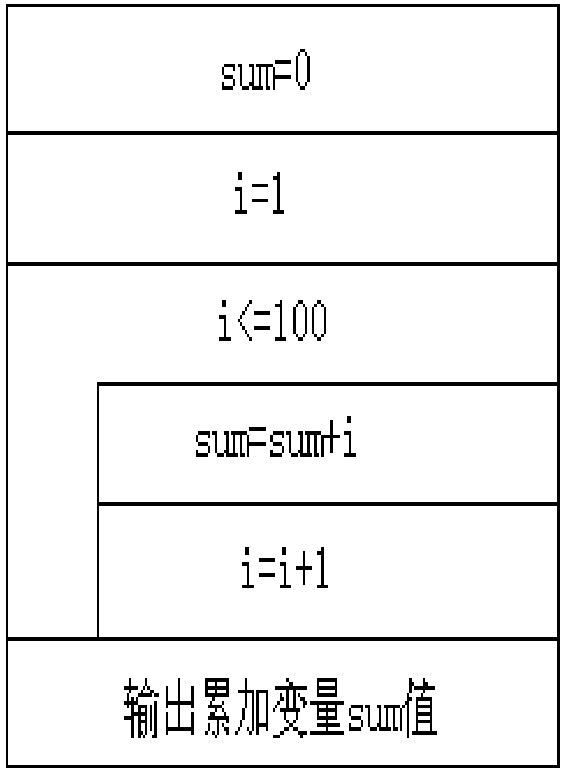


图1.5 累加运算N-S图

C语言程序算法如下：

```
#include "stdio.h"
 int main()
 {
    int i=1,sum=0;          /*定义变量及其数据类型*/
    while(i<=100)           /*循环控制结构*/
           {sum+=i;
            i=i+1;
            }               /*循环体结束*/
    printf("sum=%d\n",sum);  /*输出累加结果*/
    return 0;
 }
```

程序算法不是唯一的，对这个问题求解还有其它的算法

程序算法不是唯一的：

算法1：

```
#include "stdio.h"
int main()
{int i=1,sum=0;
  while (i<=100)
     {sum=sum+i ;
         i++;
      }
printf("the sum is %d",sum);
return 0;}
```

算法 2：

```
#include "stdio.h"
int main()
{ int i=1,sum=0;
  do
   {sum=sum+i;
       i=i+1;
    } while(i<=100);
printf("the sum is %d",sum);
return 0;}
```

以此类推，可以很容易表示出计算之和的算法

$$1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}\ldots\frac{1}{n}\quad (n<=100)$$