

21世纪高等学校新理念教材建设工程

C语言学习指导

主编 赵颖 李丽萍 李昕



NEUPRESS
东北大学出版社



21 世纪高等学校新理念教材建设工程

TP312
1882

C 语言学习指导

主 编 赵 颖 李丽萍 李 昕

编 者 李 昕 赵 莉 孙莹光 赵 颖
徐琳娜 崔 杰 刘 艺 张 颖
李丽萍 王丽艳 陈殿华 刘晓宁
李风华

东北大学出版社

• 沈 阳 •

© 赵颖, 李丽萍, 李昕 2005

图书在版编目 (CIP) 数据

C 语言学习指导 / 赵颖, 李丽萍, 李昕主编. — 沈阳 : 东北大学出版社, 2005.11
(21 世纪高等学校新理念教材建设工程)

ISBN 7-81102-199-4

I . C… II . ①赵… ②李… ③李… III . C 语言—程序设计 IV . TP312

中国版本图书馆 CIP 数据核字 (2005) 第 117213 号

出 版 者: 东北大学出版社

地址: 沈阳市和平区文化路 3 号巷 11 号

邮编: 110004

电话: 024—83687331 (市场部) 83680267 (社务室)

传真: 024—83680180 (市场部) 83680265 (社务室)

E-mail: neuph @ neupress.com

http: // www. neupress. com

印 刷 者: 沈阳市政二公司印刷厂

发 行 者: 东北大学出版社

幅面尺寸: 184mm×260mm

印 张: 13.75

字 数: 352 千字

出版时间: 2005 年 11 月第 1 版

印刷时间: 2005 年 12 月第 1 次印刷

责任编辑: 王兆元

封面设计: 唐敏智

责任校对: 章 力

责任出版: 杨华宁

定 价: 16.00 元

序

C 语言之所以被称作出色的程序设计语言，是因为它精练、灵活、应用领域广泛。虽然走过多年，至今依然在计算机基础课教学和专业课教学中起着重要的作用。

计算机既是现代科学技术的结晶，又是大众化的工具。非计算机专业中的计算机教育，无论是目的、内容、教学体系，还是教材、教学方法等各方面都有相应的特点和规律，掌握计算机知识和应用，无疑是培养新型人才的一个重要环节。

在多年的教学实践中，我们认为：

(1) 非计算机专业中的计算机教育，与计算机专业有很大的不同，决不能简单搬用计算机专业的一套，也不能采取简单的浓缩处理。应该强调以应用为目的，以应用为出发点。如果不注意这个特点，将事倍功半。应该找到适合自己特点的教学体系、教材和教学方法。

(2) 计算机基础教育的基本模式是层次结构，也就是说计算机的应用是分层次的。同样计算机人才培养也是分层次的，不同领域、不同专业情况各不相同，应该区别对待。对一个初步接触计算机知识的学生来说，也有一个由浅入深、逐步提高的过程。全国高等学校计算机基础教育研究会提出了按层次来进行教学的方案，在实践中证明它是行之有效的。

(3) 计算机技术发展非常迅速，需要学习和掌握的新知识愈来愈多，对于计算机基础课教学来说，总学时是有限的，因此必须分清主次，确定每一门课的性质和要求。应当考虑在有限的学时内使学生接受到更多的、新的、有用的内容。提倡“精讲多练”、多上机的教学方法。

开设结合专业的计算机课是今后非计算机专业计算机基础教学的重点，是各专业培养合格人才的重要内容之一。它的目标是使各专业的大学生毕业后，都能熟练地使用计算机从事本专业的业务和科研工作。

我们编写这套教材的目的是：提供与教材相配套的大量例题、习题，提高学生解决实际问题的能力；启发学生学习兴趣，将教材中的理论融入实际例题、习题中。通过大量做题，消化理解 C 语言理论知识。参加本书编写的人员全部为多年从事计算机基础课教学的、并具有丰富教学经验的一线教师。他们注意从实际出发，力求用读者容易理解的体系和叙述方法，深入浅出，循序渐进，帮助读者更好地掌握课程的基本内容。本书遵循的方针是“内容新颖、实用性强、概念清晰、通俗易懂、层次配套”。

由于作者的水平有限，编写时间仓促，书中难免存在许多缺点和不足之处，恳请广大读者和同行批评指正。

编 者
2005 年 9 月

前 言

本书是根据教育部提出的非计算机专业计算机基础教学多层次的要求，参照《全国计算机等级考试大纲》组织编写的。全书共分 13 章，具体包括：C 语言概述；算法；数据类型、运算符与表达式；最简单 C 程序设计；选择结构程序设计；循环控制；数组；函数；编译预处理；指针；结构体与共用体；位运算；文件。

参加编书的教师都长期从事高校 C 语言教学，亲身感受到学生在学习过程中遇到的各种困难。为了使學生能够在 C 语言的学习过程中始终保持强烈的学习兴趣，领悟到程序设计的奥妙，把教材中的理论知识融入本书的例题、习题之中，使其在应用中学会新知识，作者对书中的内容和写法作了精心考虑，使得本书具有以下特色：

(1) 系统全面。内容安排由浅入深，循序渐进。全书围绕结构化程序设计方法，全面展开 C 语言教学内容，例题丰富，讲解透彻，习题难易适中，既有帮助理解基础知识的内容，又有提高学习兴趣和实践编程能力的例子。书中同时引用了许多算法实例，有利于读者深入学习计算机相关课程。

(2) 通俗易懂。在写作方式上既注意到概念的严谨，又考虑到语言叙述的通俗易懂，对一些难以理解的算法和容易混淆的概念使用了图解，使得本书不仅适用于课堂讲授，也便于自学。

(3) 实用性强。现在各种大规模的等级考试和学生平时考试都采取标准形式的多题型进行，学生如果不进行多题型的训练，难以通过考试。本书包括几千道例题和习题，其中主要有选择题、填空题、阅读程序、完善程序、改错题、编程题等多种题型，并在例题中，对典型题作了细致周到的讲解，对学生应试能力提高有实际性的帮助。

本书既可作高校计算机基础教学所用的辅助教材，也可作广大编程爱好者的自学读物。书中难免存在不足之处，望读者批评指正。

编 者
2005 年 9 月

目 录

第 1 章 C 语言概述	1
一、重点、难点	1
(一) C 语言的特点	1
(二) 源程序的书写规则	1
(三) C 程序的组成	1
(四) 函数的组成	1
二、例题	2
三、习题	3
习题答案	4
第 2 章 程序的灵魂——算法	5
一、重点、难点	5
(一) 算法的概念及描述	5
(二) 结构化程序设计	5
二、例题	6
三、习题	8
习题答案	8
第 3 章 数据类型、运算符与表达式	9
一、重点、难点	9
(一) 关键字	9
(二) 标识符	9
(三) C 语言的数据类型	9
(四) 常量和变量	10
(五) 变量的初值和类型转换	13
(六) 运算符	14
二、例题	16
三、习题	23
习题答案	30
第 4 章 最简单的 C 程序设计——顺序程序设计	31
一、重点、难点	31
(一) C 语言中的语句	31
(二) 常用的顺序执行语句	31

二、例题	35
三、习题	40
习题答案	47
第5章 选择结构程序设计	48
一、重点、难点	48
(一) 关系运算符和关系表达式	48
(二) 逻辑运算符和逻辑表达式	48
(三) 条件运算符	49
(四) if 语句	49
(五) switch 语句	50
二、例题	51
三、习题	55
习题答案	62
第6章 循环控制	65
一、重点、难点	65
(一) while 语句	65
(二) do—while 语句	65
(三) for 语句	65
(四) continue 语句和 break 语句	66
(五) goto 语句	66
二、例题	66
三、习题	74
习题答案	81
第7章 数 组	84
一、重点、难点	84
(一) 一维数组	84
(二) 二维数组	85
(三) 字符数组	85
(四) 字符串的处理函数	86
二、例题	87
三、习题	91
习题答案	97
第8章 函 数	99
一、重点、难点	99
(一) 库函数	99
(二) 函数的定义方法	99
(三) 函数的类型和返回值	99

(四) 参数值的传递	100
(五) 函数的调用	100
(六) 局部变量和全局变量	101
(七) 变量的存储类别、作用域和生存期	101
(八) 内部函数与外部函数	102
二、例题	102
三、习题	110
习题答案	120
第 9 章 编译预处理	122
一、重点、难点	122
(一) 编译预处理的含义、作用以及编译预处理命令行的正确表示形式	122
(二) 不带参数的宏的定义及使用方法	122
(三) 带参数的宏的定义及使用形式	122
(四) 带参数的宏与函数调用的区别	123
(五) 文件包含命令的正确使用	123
二、例题	123
三、习题	126
习题答案	129
第 10 章 指 针	130
一、重点、难点	130
(一) 指针的概念	130
(二) 指针变量	130
(三) 一维数组和指针	131
(四) 二维数组和指针	132
(五) 通过指针数组引用二维数组的元素	132
(六) 通过建立一个行指针来引用二维数组元素	133
(七) 指向指针的指针变量	133
(八) 返回指针值的函数	133
(九) 指向函数的指针变量	133
(十) 传给 main() 的参数	134
(十一) 字符指针变量	134
(十二) 二维数组名作为实参时, 实参和形参之间的数据传递	135
(十三) 指针数组作为实参时, 实参和形参之间的数据传递	135
二、例题	136
三、习题	147
习题答案	161
第 11 章 结构体与共用体	162
一、重点、难点	162

(一) 结构体类型	162
(二) 共用体类型	163
(三) 用 typedef 定义类型	164
二、例题	164
三、习题	177
习题答案	191
第 12 章 位运算	192
一、重点、难点	192
二、例题	192
三、习题	195
习题答案	197
第 13 章 文 件	199
一、重点、难点	199
(一) 文件的概念及分类	199
(二) 文件类型指针	199
(三) 文件的打开与关闭	199
(四) 文件的读写函数的使用	200
(五) 文件的定位函数的使用	200
(六) 文件操作的出错检测	200
二、例题	200
三、习题	204
习题答案	209

第 1 章 C 语言概述

一、重点、难点

(一) C 语言的特点

- (1) 语言简洁、紧凑、使用方便灵活。
- (2) 运算符丰富。
- (3) 具有结构化的控制语句。
- (4) 语法限制不太严格，程序设计自由度大。
- (5) C 语言允许用户直接访问物理地址，能进行位操作，可以直接对硬件进行操作。

(二) 源程序的书写规则

- (1) C 语言书写自由，一行内可以写几个语句，并且一个语句可以写在多行上。
- (2) C 程序没有行号，每个语句和数据定义的最后必须有一个分号。
- (3) C 语言分号是语句中必不可少的。即使是程序的最后一个语句也应该包含分号。
- (4) C 语言中的注释可以用“/*”开始，用“*/”结束，注释可以在任何允许插入空格符的地方插入。且“/”与“*”间不允许有空格。
- (5) C 语言的注释不允许嵌套，注释可以用西文，也可以用中文。

(三) C 程序的组成

- (1) 一个 C 程序由一个或多个源程序文件组成。
- (2) 一个 C 源程序文件是由若干个函数组成的。函数是 C 程序的基本单位。在这些函数中有且只有一个主函数 `main()`，主函数由系统提供。各个函数在程序中所处的位置不是固定的。
- (3) 一个源程序文件是一个编译单位，即以源文件为单位进行编译，而不是以函数为单位进行编译。C 语言源程序文件的扩展名为 `.C`。
- (4) 任何 C 程序都是从主函数开始执行的，调用其他函数后，回到 `main()` 主函数，在 `main()` 主函数中结束程序运行。

(四) 函数的组成

一个函数由以下两部分组成。

(1) 函数的首部。

函数类型、函数名、函数参数类型、函数参数名。

函数名后必须是一对圆括弧，但可以没有函数参数。

(2) 函数体。

函数体是最外层由 `{}` 括起来的部分，包括变量的声明和执行两部分。

二、例题

【例 1-1】 判断：

(1) C 语言是以函数为程序的基本单位，便于实现程序的模块化。()

解析：C 程序是由函数组成的，一个 C 源程序至少包含一个主函数 `main()`。函数是 C 程序的基本单位。C 语言的这种特点使 C 程序容易实现模块化。

答案：正确。

(2) C 程序的执行总是从程序第一句开始。()

解析：一个 C 程序总是从 `main()` 函数开始执行，而不论 `main()` 函数处在程序中的任何位置。

答案：错误。

(3) C 程序中可以不使用函数。()

解析：C 语言是函数式语言，一个 C 源程序至少包含有一个 `main()` 函数。

答案：错误。

(4) C 语言提供了一个输入语句 `scanf()` 和一个输出语句 `printf()`。()

解析：C 语言本身没有输入输出语句，输入输出操作由库函数 `scanf()` 和 `printf()` 等函数来完成。

答案：错误。

(5) C 程序的一条语句必须写在一行中。()

解析：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。

答案：错误。

(6) C 程序中注释说明只能写在一条语句的后面。()

解析：C 语言中的注释可以用 `/*` 开始，用 `*/` 结束，注释可以在任何允许插入空格符的地方插入。且 `/*` 与 `*/` 间不允许有空格。

答案：错误。

【例 1-2】 下列程序中格式错在何处？

- A. `main()`
- B. `{int a, b, z;`
- C. `a=2: b=3;`
- D. `z=a+b;}`

解析：A, B, D 没有错误。C 中，在 `a=2` 后应该是分号 `“;”` 或逗号 `“,”`，不应该是冒号 `“:”`，因此 C 处有错误。

答案：C。

【例 1-3】 C 语言的程序一行写不下时，可以

- A. 用逗号换行
- B. 用分号换行
- C. 在任意空格处换行
- D. 用回车符换行

解析：C 语言可以在任何一个分隔符或空格处换行。

答案：C。

【例 1-4】 下列程序段中哪一个是错误的注释方法？

A. `# in/* 包含 */clued<stdio.h>`
 `void main()`
 `{`
 `}`

B. `# include<stdio.h>`
 `void main/* */(/* 主函数 */)`
 `{`
 `}`

C. `# include<stdio.h>`
 `void main()`
 `{ int x/* 初始化 */=10;`
 `/* 打印 */printf("%d",x);`
 `}`

D. `# include <stdio.h>`
 `void main( )`
 `{ int x=10;`
 `printf("%d",x);`
 `/*打印 x 的值 * " " = 10 * /`
 `}`

解析：根据 C 语言的规定：注释不能插在标识符的中间，在其他有分隔符的地方都可以插入注释，上面四个程序段中只有 A 是将注释插入一标识符的中间了，因此 A 的注释是错误的。

答案：A。

【例 1-5】 下列程序中的错误在()。

```
main()
{.....
{
{.....}}
```

解析：从上面的 C 程序的结构中，可以明显看出花括号不是成对出现的，必须在上面的程序中再加上对应的花括号，可以加到最后，也可以加在第三行或第四行，视程序的具体情况而定。

答案：程序中少了一个花括号“}”。

三、习题

1. 下列说法中正确的是：

- A. C 语言总是从第一个定义的函数开始执行
- B. 在 C 语言程序中，要调用的函数必须在 main()函数中定义
- C. C 语言中的 main()函数必须放在程序的开始部分
- D. C 语言程序总是从 main()函数开始执行

2.()是 C 程序的基本构成单位。

- A. 函数 B. 函数和过程 C. 超文本过程 D. 子程序

3. 下列说明正确的是：

- A. main()函数必须放在 C 程序的开头
- B. main()函数必须放在 C 程序的最后
- C. main()函数可以放在 C 程序的中间部分，即在一些函数之前在另一些函数之后，但在执行 C 程序时是从 main()函数开始的
- D. main()函数可以放在 C 程序的中间部分，即在一些函数之前在另一些函数之后，但在执行 C 程序时是从程序开头执行的

4. C 语言规定，必须用()作为主函数名。

- A. function B. include C. main D. stdio

5. 下列说法正确的是:

- A. 一个函数的函数体必须要有变量定义和执行部分, 二者缺一不可
- B. 一个函数的函数体必须要有执行部分, 可以没有变量定义
- C. 一个函数的函数体可以没有变量定义和执行部分, 函数可以是空函数
- D. 以上都不对

6. 在 C 语言中, 每个语句和数据定义是用()结束。

- A. 分号 B. 逗号 C. 句号 D. 括号

7. 下列说法正确的是:

- A. 在执行 C 程序时不是从 main()函数开始的
- B. C 程序书写格式自由, 一个语句可以分写在多行上
- C. C 程序书写格式严格限制, 一行内必须写一个语句
- D. C 程序书写格式严格限制, 一行内必须写一个语句, 并要有行号

8. 下列叙述中正确的是:

- A. C 语言比其他语言高级
- B. C 语言可以不用编译就能被计算机识别执行
- C. C 语言以接近英语国家的自然语言和数学语言作为语言的表达形式
- D. C 语言出现得最晚, 具有其他语言的一切优点

9. 下列叙述中正确的是:

- A. C 程序中注释部分可以出现在程序中任何允许插入空格符的地方插入的地方
- B. 花括号 “{” 和 “}” 只能作为函数体的定界符
- C. 构成 C 程序的基本单位是函数, 所有函数名都可以由用户命名
- D. 分号是 C 语句之间的分隔符, 不是语句的一部分

10. 填空:

- (1) 主函数名后面的一对圆括号中间可以为空, 但一对圆括号不能()。
- (2) 函数体一般包括()、()。
- (3) C 语言的关键字都用() |大写或小写|。
- (4) 一个函数由两部分组成, 它们是()和()。
- (5) 在 C 语言中, 凡在一个标识符后面紧跟着一对圆括弧, 就表明它是一个()。
- (6) 一个 C 源程序至少包含一个(), 即()。
- (7) 函数体的范围是()。
- (8) C 语言是通过()来进行输入和输出的。

习题答案

1. D 2. A 3. C 4. C 5. C 6. A 7. B 8. C 9. A

10. (1) 省略 (2) 变量定义部分 执行部分 (3) 小写 (4) 函数说明 函数体
(5) 函数 (6) 主函数 main() (7) 最外层的一对大括号内的部分
(8) 输入和输出函数

第2章 程序的灵魂——算法

一、重点、难点

(一) 算法的概念及描述

算法是对解题过程准确而完整的描述，因此不知道解题的算法也就不可能编写出程序来解题。

描述算法的方法有自然语言表示法、程序流程图、N-S图、PAD图、伪代码等方式，但目前应用最广泛的是程序流程图和N-S图。图2-1表示程序流程图的基本符号。图2-2表示计算 $m!$ 的程序流程图求解算法。

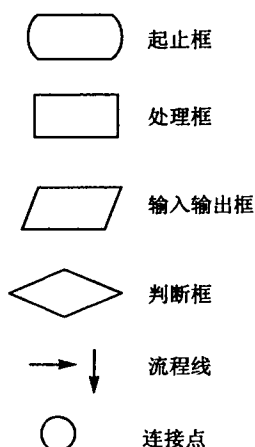


图 2-1 程序流程图的基本符号

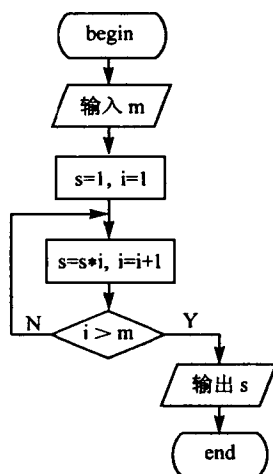


图 2-2 计算 $m!$ 的程序流程图求解算法

人们普遍采用程序流程图是因为它简单、直观，能将执行的流程表达得一清二楚，易于理解。然而，它也有不少缺陷，程序流程图中的箭头十分灵活，使用不当会影响到程序结构的清晰度。N-S图是另一种表达算法的工具，它不采用任何箭头，保证编写的程序完全是结构化的，程序结构清晰，容易理解和修改。图2-3表示N-S图的基本符号，图2-4表示计算 $m!$ 的N-S图求解算法。

(二) 结构化程序设计

结构化程序设计的提出是从 goto 语句(转向语句)的使用引起的，goto 语句可以使程序员随心所欲地从程序的一处转到另一处，充分发挥程序员的“技巧”，但过多使用 goto 语句会使程序结构十分零乱，程序一会儿向前，一会儿向后，令人眼花缭乱，顾此失彼，程序可读性差，难以阅读和理解，因此有的学者建议禁用 goto 语句。

1966年C.Bohm提出任何程序都可以用顺序、选择、循环三种基本结构来组合，这样

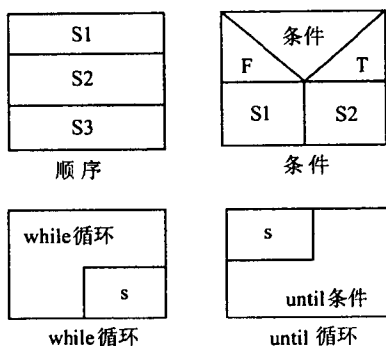


图 2-3 N-S 图的基本符号

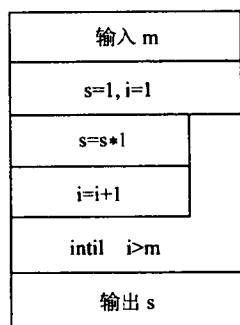


图 2-4 计算 $m!$ 的 N-S 图求解算法

编写出来的程序易懂、易读、易修改，提高了程序可靠性。这样的程序称为结构化程序，编写这样的程序称为结构化程序设计。结构化程序设计不是禁用 goto 语句，而是少用或不用，只限于在一个结构内部使用，不允许从一个结构跳转到另一个结构。

结构化程序设计的另一个概念是模块化设计，把一个大的、复杂的问题逐层分解为一系列小的、简单的模块来进行处理，每个小模块只完成单一的、具体的任务。模块内部联系紧密，而与其他模块之间联系较弱，这样的模块称为独立性高的模块。

模块之间是一种层次结构，上层模块对下层模块进行调用。如图 2-5 所示是报表生成程序的层次结构，以矩形框表示结构，框中的模块名称表明模块功能。顶层的模块只提出“做什么”而不涉及“怎样做”，最下层的模块十分具体，涉及“怎样做”的精确描述，易于编程。

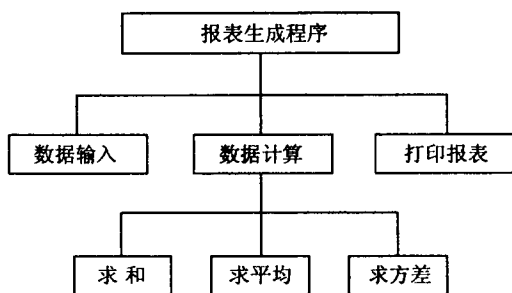


图 2-5 报表生成程序层次结构

结构化程序设计的特点：

- (1) 以三种基本结构的组合进行程序的编写；
- (2) 程序设计采用自顶向下、逐步细化、模块化的方法，以模块作为程序的编写单位；
- (3) 不用或少用 goto 语句，通常只允许从一个循环跳出而不允许从外部跳入一个循环；
- (4) 每一个结构只允许有一个入口和一个出口，各部分之间接口简单，逻辑清晰；
- (5) 采用结构化语言编写程序，程序结构清晰，易于阅读和修改；
- (6) 采用良好的编程风格。

二、例题

【例 2-1】 计算机结构化程序设计的三种基本结构是什么？对每种结构给出一个简单的例子，并画出 N-S 图或程序流程图。

解析： 本题考查对结构化程序的理解和对 N-S 图和程序流程图的掌握。

画 N-S 图或程序流程图时，不要忘记画输入和输出部分。

画程序流程图时，注意流程线一定是带箭头的，不能忘记，也不能忘记判断框的条件 T 和 F 或者 Y 和 N；输入输出框也可画成矩形框，有时可省略 begin 和 end 框。

答案：计算机结构化程序设计的三种基本结构是：顺序、分支(条件、选择)和循环，每一种结构都是单入口、单出口。

(1) 顺序结构举例：计算 3 个数 a , b , c 之和。其 N-S 图和程序流程图如图 2-6 所示。

(2) 分支结构举例：打印 a , b , c 三个数中的最大数。

解析：预先设定一个变量 $\max$ ，用它存放三个数中的最大值， a 与 b 直接比较即可找到 a 与 b 的最大值，将其放入 $\max$ 中，再用 $\max$ 和 c 比较。其 N-S 图和程序流程图如图 2-7 所示。

(3) 循环结构举例：计算 $1+2+3+\cdots+n$ 的值。

解析：设定变量 s 表示累加和，初值为 0；设定变量 i

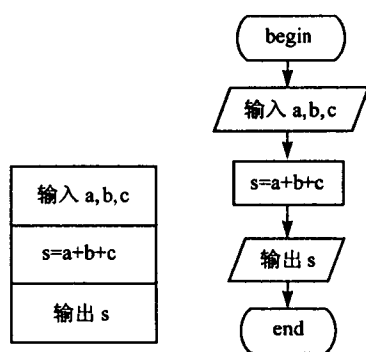


图 2-6 计算 3 个数 a , b , c 之和的 N-S 图和程序流程图

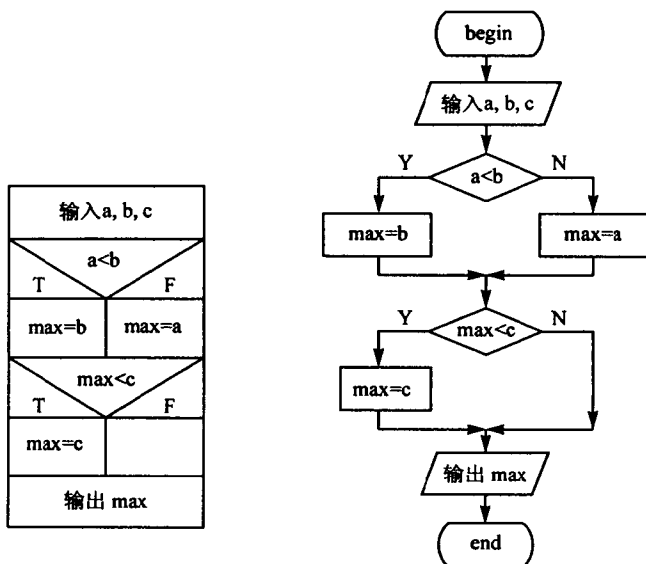


图 2-7 打印 a , b , c 三个数中的最大数的 N-S 图和程序流程图

表示即将加入的加数(1, 2, 3, 4, ..., n)，初值为 1；

求累加和的过程是：在上一次求和的基础上，不断累加新的加数，是一个重复的过程，用循环解决，重复的步骤是累加新的加数，准备下一个加数(i 自身增 1)，以此作为需重复执行的循环体。其 N-S 图和程序流程图如图 2-8 所示。

注意：不能构成死循环，即循环条件永远成立。本题要注意循环体中不能忘记 $i=i+1$ ，否则， i 始终为 1， $1 \leq n$ ，循环条件始终成立，形成死循环。

【例 2-2】 图 2-9 是求 $m!$ 的 N-S 图，请填空。

解析：从 N-S 图可看出，①处应填循环条件，循环体是在原来 t 的基础上乘以 i ，而 i 又不断加 1，显然一直增加到 $i > m$ 时为止，那么循环条件应为当 $i \leq m$ 时，要不断地执行循环体，①处应填“当 $i \leq m$ ”。②处是输出部分，应填输出结果 $m!$ ，本题用 t 表示，即②处应填“输出 t ”。

答案：①处应填“当 $i \leq m$ ”， ②处应填“输出 t ”。

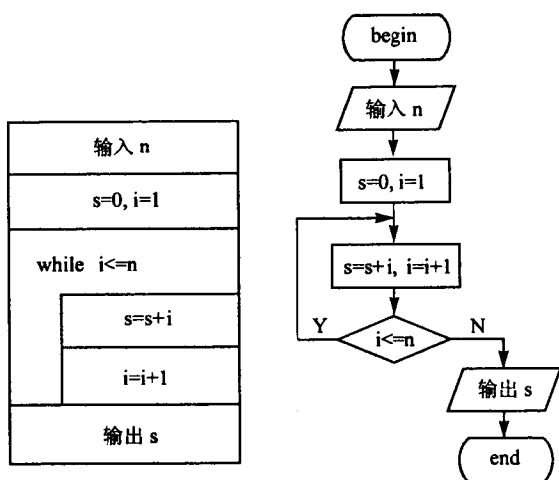


图 2-8 计算 $1+2+3+\dots+n$ 的值的 N-S 图和程序流程图

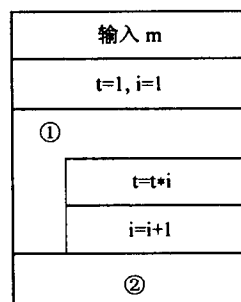


图 2-9 求 $m!$ 的 N-S 图

三、习题

1. 判断正误：

- (1) 自顶向下逐步求精设计是结构化程序设计的一个组成部分。()
- (2) 只要程序正确即可，不必考虑书写是否规范。()
- (3) 每一个结构只能有一个入口，出口可以有一个或多个。()
- (4) 模块之间联系越弱越好，模块内部联系越紧密越好。()
- (5) 结构化程序设计不能使用 goto 语句。()

2. 画出以下习题的 N-S 图。

(1) 分段函数

$$y = \begin{cases} 2x - 1 & (x < 0) \\ 0 & (x = 0) \\ 2x - 3 & (x > 0) \end{cases}$$

- (2) 查找所有的两位数中既是 3 的倍数，又是 5 的倍数的数。

习题答案

1. (1) 正 (2) 误 (3) 误 (4) 正 (5) 误
 2. (1) 如图 2-10 所示 (2) 如图 2-11 所示

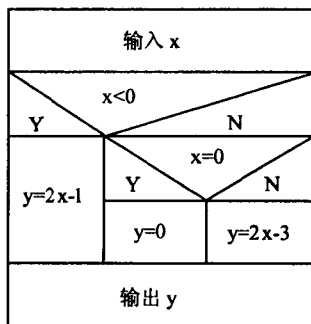


图 2-10 程序 N-S 图

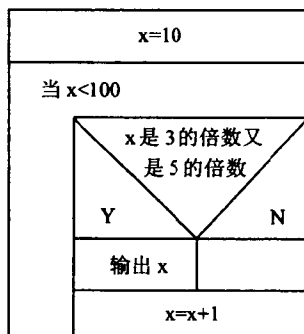


图 2-11 程序 N-S 图