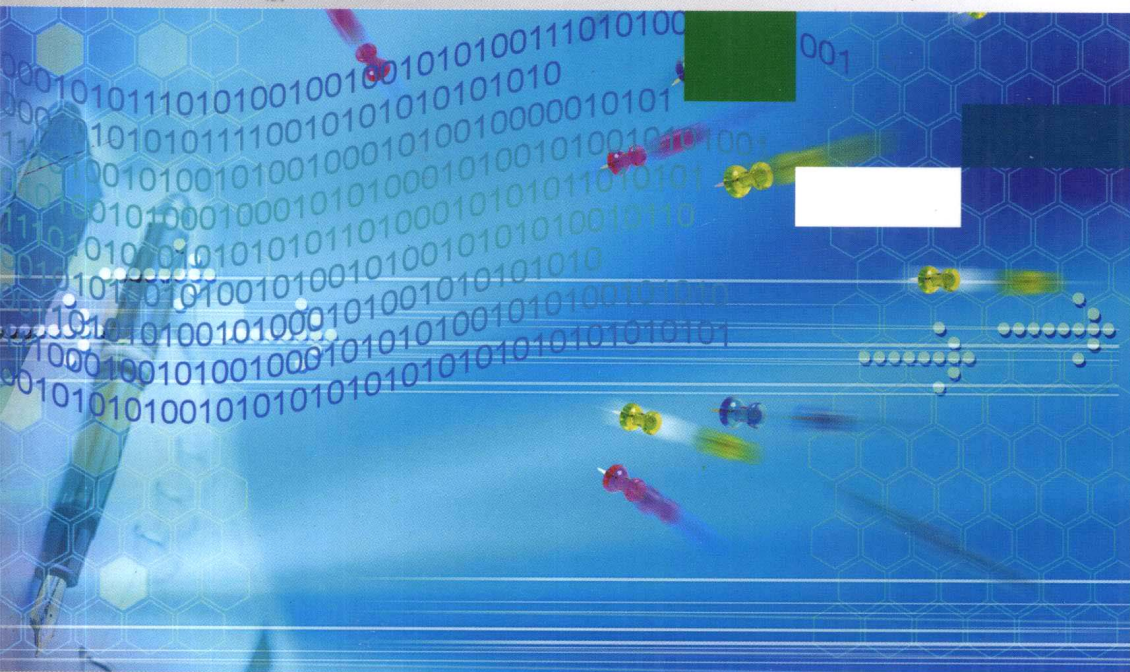




普通高等教育“十一五”规划教材



C语言程序设计

唐云廷 主编

 科学出版社
www.sciencep.com

提供
电子教案

普通高等教育“十一五”规划教材

C 语言程序设计

唐云廷 主 编

王 明 杨 柳 刘加海 副主编

科 学 出 版 社

北 京

内 容 简 介

本书以程序设计为主线,通过案例和问题引入学习内容,由浅入深、循序渐进地讲解程序设计的思想、方法和相关的语言知识,主要包括:程序的概念与算法、C 语言程序的基本构成、输入输出程序设计、while 循环程序设计、for 循环程序设计、if 语句的应用、多分支语句的应用、常用数据表达、数组的应用、函数及其应用、结构体的应用、链表及其应用、变量存储类型与多文件系统和文件程序的设计等。本书内容全面、层次清晰,编排合理,并对难点做了十分详尽的阐述,对重要知识点配有适量的例题,便于读者理解和掌握。

本书可作为高等院校各专业学生高级语言程序设计课程的教材。

图书在版编目(CIP)数据

C 语言程序设计/唐云廷主编. —北京:科学出版社, 2009

(普通高等教育“十一五”规划教材)

ISBN 978-7-03-023964-8

I. C… II. 唐… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 012722 号

责任编辑:陈晓萍/责任校对:赵 燕

责任印制:吕春珉/封面设计:耕者设计工作室

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

骏杰印刷厂印刷

科学出版社发行 各地新华书店经销

*

2009 年 2 月第 一 版 开本:787×1092 1/16

2009 年 2 月第一次印刷 印张:18 1/2

印数:1—4 500 字数:436 000

定价:30.00 元

(如有印装质量问题,我社负责调换(环伟))

销售部电话 010-62134988 编辑部电话 010-62135120-8003

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

程序设计是高等院校重要的一门计算机基础课程，它以编程语言为平台，介绍程序设计的思想和方法。通过该课程的学习，学生不仅要掌握高级程序设计语言的知识，更重要的是在实践中逐步掌握程序设计的思想和方法，培养问题求解和语言的应用能力。因此，这是一门以培养学生程序设计基本方法和技能为目标，以培养学生实践能力为重点的特色鲜明的课程。

C 语言是得到广泛应用的程序设计语言之一，它既具备高级语言的特性，又具有直接操纵计算机硬件的能力，并因其丰富灵活的控制和数据结构、简洁而高效的语句表达、清晰的程序结构和良好的可移植性而拥有大量的使用者。目前，C 语言被许多高校列为程序设计课程的首选语言。

“C 语言程序设计”是一门实践性很强的课程，该课程的学习有其自身的特点，听不会，也看不会，只能练会。学习者必须通过大量的编程训练，在实践中掌握语言知识，培养程序设计的基本能力，并逐步理解和掌握程序设计的思想和方法。因此，“C 语言程序设计”课程的教学重点应该是培养学生的实践编程能力，教材也要以程序设计为中心来组织内容。

本书以程序设计为主线，在案例和问题中导出内容和知识点，由浅入深、循序渐进地讲解程序设计的思想、方法和相关的语言知识。教材在教学内容编排上，力求突出设计思想，淡化设计手段，加强算法分析，减弱语言知识，通过全面、深入、系统地介绍程序设计方法和程序设计语言，帮助学生掌握 C 语言的基本语法、语句、控制结构以及结构化程序设计的基本思想和方法，使学生认识到算法、良好的程序设计风格以及实践在本课程学习中的重要性，培养学生熟练使用 C 语言分析和解决实际问题的能力，通过分析、分解，最终归纳整理出计算机能够实现的过程（算法），拓展学生的思维空间，训练学生的思维能力；同时也为今后应用程序设计解决相关专业领域内实际问题铺垫良好的程序设计基础。

本书共 14 章，主要内容安排如下：第 1 章为程序的概念与算法；第 2 章为 C 语言程序的基本构成；第 3 章为输入输出程序设计；第 4 章为 while 循环程序设计；第 5 章为 for 循环程序设计；第 6 章为 if 语句的应用；第 7 章为多分支语句的应用；第 8 章为常用数据表达；第 9 章为数组的应用；第 10 章为函数及其应用；第 11 章为结构体的应用；第 12 章为链表及其应用；第 13 章为变量存储类型与多文件系统；第 14 章为文件程序的设计。本书在结构上力求层次清晰，编排合理，便于读者自学；在内容上做到系统、全面，并对难点做了十分详尽的阐述，对重要知识点配有一定数量的例题，便于读者理解和掌握。

在编写本书的过程中，我们参阅了国内外大量有关 C 语言程序设计的书籍与文献，

在此向相关作者表示诚挚的感谢。

本书由唐云廷任主编，王明、杨柳、刘加海担任副主编，编者在总结多年的程序设计课程的教学经验及教改实践基础上，编写了本书。

限于时间、水平和能力，书中肯定会有不少缺点和不足，热切期望得到专家和读者的批评指正。

唐云廷

2008 年 12 月

目 录

前言

第 1 章 程序的概念与算法	1
1.1 程序的概念	2
1.2 算法与流程图	2
1.3 结构化程序的结构	3
1.3.1 顺序结构	4
1.3.2 分支结构	7
1.3.3 循环结构	10
练习与实践	12
第 2 章 C 语言程序的基本构成	13
2.1 C 源文件	14
2.2 C 函数	17
2.3 C 语句	17
2.4 保留字与标识符	18
2.4.1 保留字	18
2.4.2 标识符	18
2.5 运算符	19
2.5.1 算术运算符	19
2.5.2 关系运算符	20
2.5.3 逻辑运算符	20
2.5.4 位运算符	22
2.6 编辑 C 语言程序时应注意的问题	23
练习与实践	23
第 3 章 输入输出程序设计	28
3.1 字符串的输入与输出	29
3.1.1 字符串的输出	29
3.1.2 字符串的输入	31
3.2 整型数的输入与输出	32
3.2.1 整型数的输出	32
3.2.2 整型数的输入	33
3.3 字符的输入与输出	36
3.4 实型数的输入与输出	38

练习与实践	40
第 4 章 while 循环程序设计	42
4.1 循环的概念	43
4.1.1 while 循环	43
4.1.2 do-while 循环	48
4.1.3 break 语句和 continue 语句	50
4.2 循环嵌套的应用	55
练习与实践	57
第 5 章 for 循环程序设计	60
5.1 for 循环结构	61
5.2 循环嵌套	68
练习与实践	71
第 6 章 if 语句的应用	74
6.1 if 语句的应用	75
6.2 if-else 语句的应用	78
6.3 if-else if 语句的应用	82
6.4 if 嵌套语句的应用	85
练习与实践	87
第 7 章 多分支语句的应用	89
练习与实践	101
第 8 章 常用数据表达	103
8.1 数组	104
8.1.1 一维数组	104
8.1.2 字符数组	107
8.1.3 二维数组	108
8.2 结构体	110
8.3 指针	116
8.3.1 一维数组指针	116
8.3.2 数组指针	119
8.3.3 指针数组	120
练习与实践	122
第 9 章 数组的应用	124
9.1 数组元素的查找	125
9.1.1 无序数组的查找	125
9.1.2 有序数组的查找	129
9.2 数组的排序	131
9.2.1 冒泡排序	131

9.2.2 选择排序	135
9.3 数组元素的插入	139
9.3.1 绝对位置的插入	139
9.3.2 有序数据的插入	141
9.4 数组元素的删除	144
练习与实践	146
第 10 章 函数及其应用	147
10.1 函数的基本概念	148
10.2 函数的参数	148
10.3 系统函数的应用	151
10.3.1 数学函数	151
10.3.2 字符串处理函数	152
10.3.3 字符操作函数与转换函数	154
10.4 自定义函数	155
10.5 函数的应用	159
10.6 函数的嵌套与递归调用	171
10.6.1 函数的嵌套调用	171
10.6.2 函数递归调用	173
练习与实践	175
第 11 章 结构体的应用	182
11.1 结构体的基本概念	183
11.1.1 结构体类型定义	183
11.1.2 结构体变量的定义	183
11.1.3 结构体变量占据的内存空间	184
11.1.4 结构体变量对结构体成员的引用	184
11.1.5 结构体变量的赋值	184
11.2 结构体类型的数组	187
11.3 结构体变量的输入、输出与存储	190
11.4 结构体变量的查找与删除	197
11.5 结构体变量的排序	205
练习与实践	216
第 12 章 链表及其应用	221
12.1 链表的定义	222
12.2 堆栈	223
12.3 队列	225
12.4 链表的插入	228
12.5 链表的删除	230

12.6 链表的应用	231
练习与实践	244
第 13 章 变量存储类型与多文件系统	245
13.1 变量的存储类型	246
13.2 auto 存储类型的变量与作用范围	247
13.3 static 存储类型的变量与作用范围	249
13.4 register 存储类型的变量与作用范围	252
13.5 extern 存储类型的变量与作用范围	253
13.6 多文件系统中的函数调用	255
13.7 外部变量在多文件系统中的应用	260
练习与实践	261
第 14 章 文件程序的设计	264
14.1 文件概述	265
14.2 文件打开与关闭函数	266
14.2.1 文件打开函数 fopen	267
14.2.2 文件关闭函数 fclose	267
14.3 文件中字符的读/写函数	268
14.3.1 文件中字符读函数 fgetc	268
14.3.2 文件中字符写函数 fputc	268
14.4 文件的格式化读/写函数	270
14.4.1 文件的格式化读函数	270
14.4.2 文件的格式化写函数	271
14.5 文件中的数据块读/写函数	273
14.5.1 数据块读函数 fread	273
14.5.2 数据块写函数 fwrite	273
14.6 文件位置定位函数	276
14.6.1 位置指针复位函数 rewind	276
14.6.2 随机读写函数 fseek	277
练习与实践	278
附录	282
附录 A ASCII 表	282
附录 B 运算符及其优先级汇总表	283
附录 C C 语言的保留字	284
附录 D Visual C++ 6.0 上机步骤	284

第 1 章

程序的概念与算法



本章重点

- 程序的概念。
- 结构化程序的结构。
- 算法的概念。
- 流程图的表示。
- 程序的框架。



本章难点

- C 语言程序的结构。
- 源文件、目标文件、可执行文件的概念。
- 算法的理解。
- 流程图的画法。



本章学习建议

建议重点掌握对一个问题的分析及如何解决，为什么要采用这种算法？应正确理解对程序过程的描述、流程图的结构、流程图的画法，掌握程序的三种结构，对本章中的源程序只作一般性的了解即可，例如了解大致的程序结构，像 main 函数、程序开始、程序结束、语句、C 语言中区分大小写等，也可以暂时不学习源程序。

1.1 程序的概念

程序是指人们将需要计算机做的工作写成一定形式的指令，并把它们存储在计算机的内部存储器中。当人们给出命令之后，它就按指令操作顺序自动进行，把这种可以连续执行的一条条指令的集合称为“程序”。目前，正在使用的计算机程序设计语言有上百种，有些语言是面向机器的，如二进制语言，而多数是面向问题的语言。面向问题的语言都被称为计算机的“高级语言”，如 C 与 C++ 语言等。目前，程序设计一般可分为非结构化程序设计、结构化程序设计和面向对象的程序设计。常见的 C 语言是结构化程序设计语言，而 C++ 语言则是面向对象的语言。这些语言都是用接近人类习惯的自然语言和数学语言作为语言的表达形式，人们学习和操作起来会感到十分方便。

程序设计的过程一般包括：

- ① 问题的提出、要求及所采用的数据结构。
- ② 算法的确定、程序的编制。
- ③ 程序的调试及修改。
- ④ 整理并写出文档资料。

程序是为了某个方面的应用而设计的。用 C 语句所写的序列称为 C 源程序，它的后缀为 .c，C 源程序经过编译（compile）后生成一个目标文件（.obj），链接库函数后生成可执行文件（.exe）。因而 C 语言程序的设计过程如图 1.1 所示。

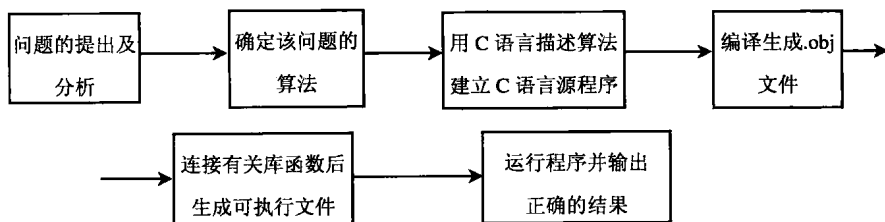


图 1.1 C 语言程序的设计过程

1.2 算法与流程图

什么是算法？算法是指解决问题所使用的一系列合乎逻辑的解题步骤，可以把算法定义为解一确定类问题的任意一种特殊的方法。在计算机科学中，算法要用计算机算法语言描述，算法代表用计算机解一类问题的精确、有效的方法。算法+数据结构=程序，求解一个给定的可计算或可解的问题，不同的人可以编写出不同的程序来解决同一个问题。

制定一个算法，一般要经过设计、确认、分析、编码、测试、调试、计时等阶段。要解决一个复杂的问题，需要仔细地分析与研究此问题，并进行精确的逻辑分析，找到解决问题的方法，再用计算机实现。例如判断一个整数的奇偶性，大致分成以下几步：

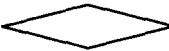
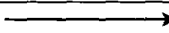
- ◇ 计算机读入该数
- ◇ 将该数除以 2
- ◇ 判断余数是否为 0
- ◇ 如果为 0, 该数为偶数
- ◇ 否则为奇数

这些解决问题的一系列步骤就是算法, 并且可以用 C 语言编写成程序, 解决数的奇偶性问题。

什么是流程图? 流程图是一种用图解方式说明实现解决一个方案所需完成的一系列操作。

流程图比文字更容易理解、更加直观, 在流程图中的符号含义如表 1.1 所示。

表 1.1 流程图的符号及含义

常用基本组件	含 义
	程序的入口和出口
	加工、处理
	条件判断
	控制流
	连接符
	输入、输出

例 1.1 问题的描述: 键盘输入两个数, 求这两个数之和, 并把结果输出, 用表 1.1 中的符号画出此问题的流程图。

分析: 程序的执行过程如下。

- ◇ 程序开始
- ◇ 分配数的存储空间
- ◇ 输入两个数
- ◇ 两数相加
- ◇ 显示结果
- ◇ 结束

程序流程图如图 1.2 所示。

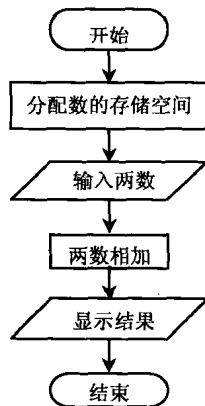


图 1.2 两数相加程序流程图

1.3 结构化程序的结构

结构化程序设计由三种结构组成: 顺序结构、选择结构和循环结构。例 1.1 中的结

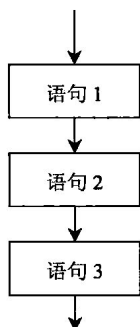


图 1.3 顺序结构的程序段

构是比较典型的顺序结构。

1.3.1 顺序结构

顺序结构的程序是一条语句接一条语句顺序地往下执行的。例如在图 1.3 中，先执行语句 1，再执行语句 2，最后执行语句 3，顺序结构的程序是最简单的程序。

例 1.2 设计一个程序，显示下列图形。

```

*****
C 语言的学习要努力!
*****
  
```

分析：这是一个典型的顺序执行的程序设计，输出三个字符串，输出字符串用函数 printf，具体的过程描述如下。

```

开始
显示： *****
换行后再显示： C 语言的学习要努力!
换行后再显示： *****
程序结束
  
```

程序流程图如图 1.4 所示。

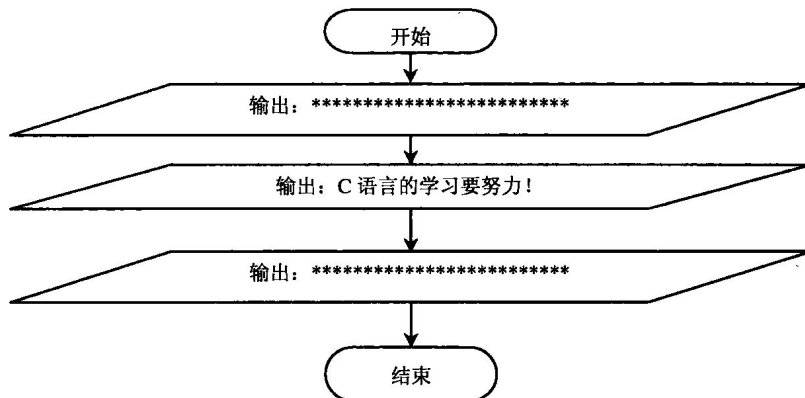


图 1.4 程序设计流程图

用 C 语言写成的源程序如下。

```

#include<stdio.h>                                /*包含标准输入输出库预处理命令*/
int main()                                       /*主函数开始*/
{                                              /*程序从此开始执行*/
    printf("*****\n");                        /*输出一行星号*/
    printf("C 语言的学习要努力! \n");         /*输出一行字符*/
    printf("*****\n");                        /*输出一行星号*/
    return 0;                                  /*与 main 函数中的 int 相对应*/
}                                              /*程序结束*/
  
```

注意 C 语言程序设计中往往要用到输入与输出函数 scanf、printf，因而要包含预处理命令#include<stdio.h>，程序从 main 函数开始执行。一个能运行的 C 程序必须要有一

个 main 函数，并且只能有一个 main 函数；printf 为系统输出函数，语句后要有“;”号，细节问题请读者通过第 2 章来学习，在此请你通过上机练习来加深理解。



上机练习

步骤 1：在 Visual C++ 环境下新建文件可单击菜单“开始→程序→Microsoft Visual C++ 6.0→Microsoft Visual C++ 6.0”。

如果出现“每日提示”，如图 1.5 所示，即单击“关闭”按钮。

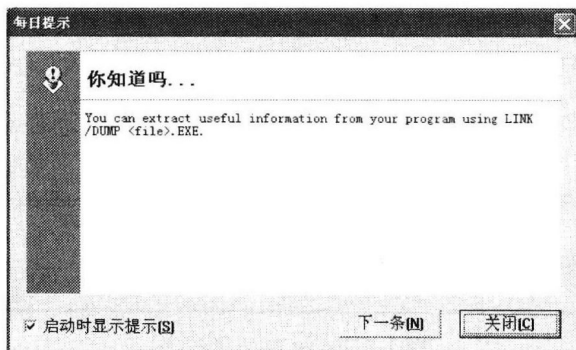


图 1.5 “每日提示”对话框

步骤 2：单击菜单“文件→新建→文件→C++ Source File”，输入文件名 1-1.c，如图 1.6 所示，单击“确定”按钮。

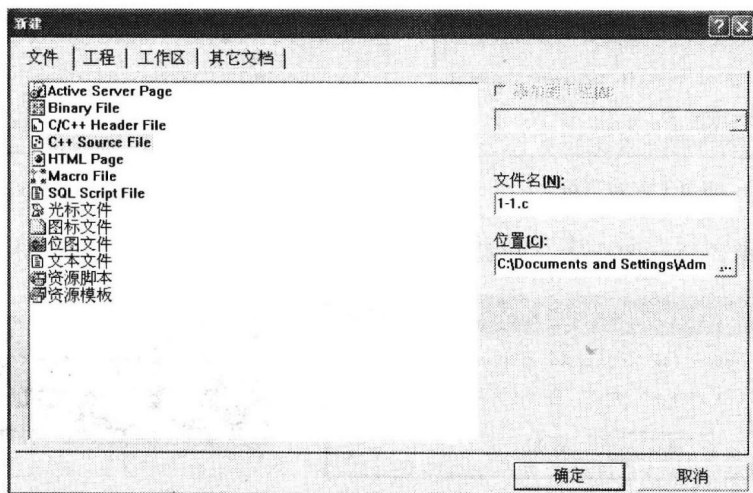


图 1.6 “新建”对话框

步骤 3：在编辑框内输入 C 源程序，如图 1.7 所示。

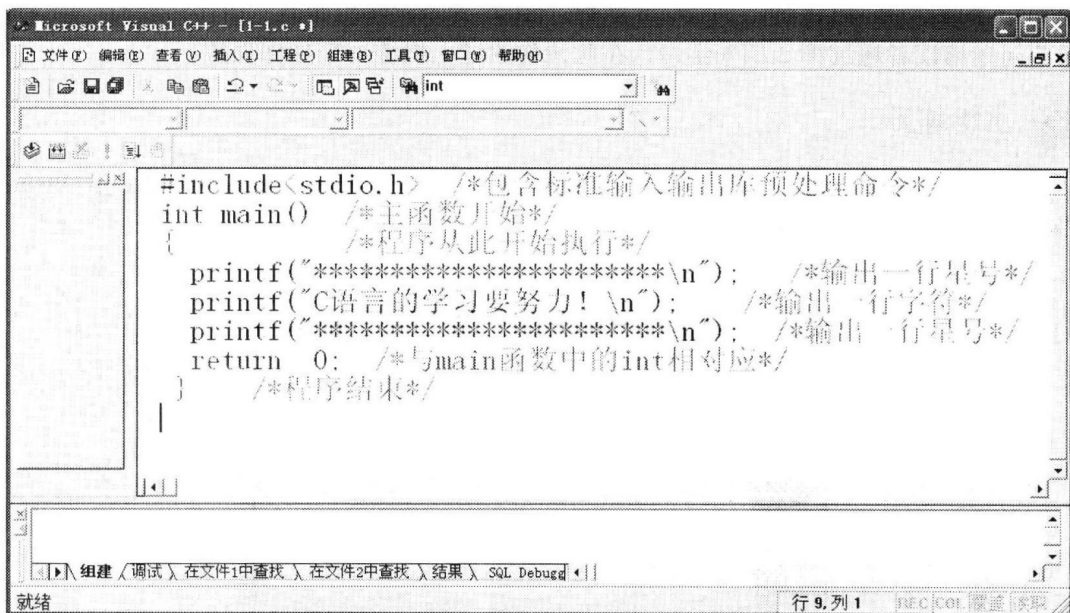


图 1.7 输入 1-1.c 源程序

步骤 4: 在 Visual C++ 环境下单击工具图标, 出现如图 1.8 所示对话框, 在图 1.8 中单击“是”按钮, 建立一个工程文件。然后在图 1.9 中单击“是”按钮, 将改动保存到文件中。在图 1.10 中单击“是”按钮, 建立应用程序文件。

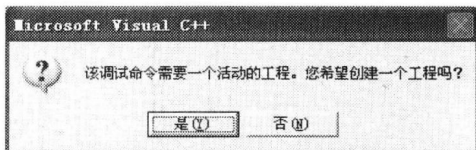


图 1.8 创建工程对话框

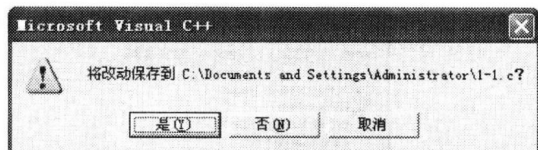


图 1.9 将改动保存到文件中

步骤 5: 单击执行图标, 执行程序 1-1.exe, 程序运行结果如图 1.11 所示。

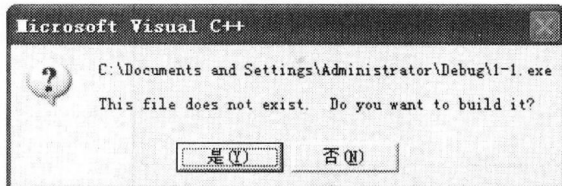


图 1.10 建立应用程序对话框

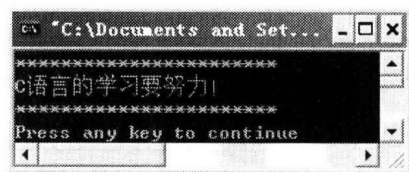


图 1.11 程序运行结果

思考

1. 在程序 1-1.c 中, 如果没有 main, 编译程序后出现的错误提示是什么? 在 C 语言程序设计中要注意什么问题?
2. 在程序 1-1.c 中, 如把 main 写成了 Main, 编译程序后出现的错误提示是什么?

在 C 语言程序设计中要注意什么问题？

3. 在程序 1-1.c 中，如 `printf ("*****\n");` 后少了 “;”，编译程序后出现的错误提示是什么？在 C 语言程序设计中要注意什么问题？

4. 在程序 1-1.c 中，如没有写预处理命令 `#include <stdio.h>`，编译程序后出现的错误提示是什么？想想这是为什么？

5. 在程序 1-1.c 中，任何一行的输出语句中少了符号 “\n”，程序运行后的结果有什么变化？问在 C 语言中符号 “\n” 的作用是什么？

6. 在程序 1-1.c 中，如 `printf ("*****\n");` 后少了 “'”，编译程序后出现的错误提示是什么？

1.3.2 分支结构

在程序执行过程中，程序的流程可由多路分支组成，根据不同的条件去执行不同的任务。例如在图 1.12 中程序执行到条件表达式时，首先判断条件是否为“真”，如为真执行语句 1，否则执行语句 2。

假如判断从键盘输入的两个数的大小，当 $x > y$ 时，提示“x 大于 y”，当 $x < y$ 时，提示“x 小于 y”，此时将会用到如图 1.13 所示的流程图。

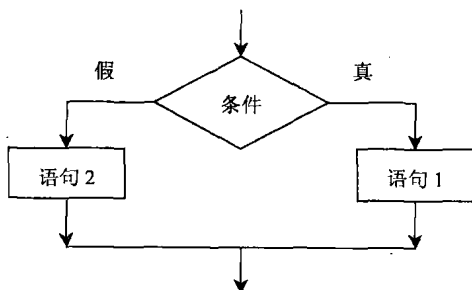


图 1.12 分支结构的程序段

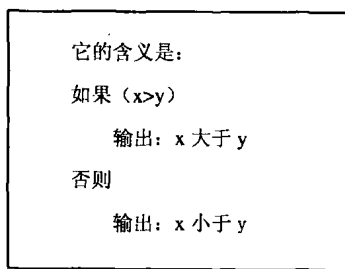
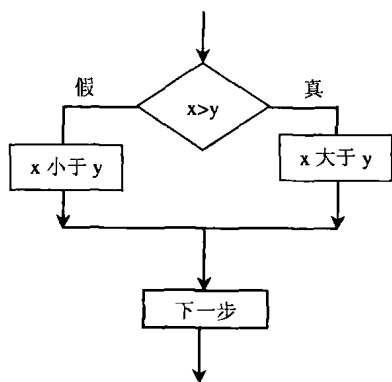


图 1.13 分支结构形式

例 1.3 键盘输入两个数，判断两个数的大小。

分析：这是一个分支程序设计，具体的过程描述如下。

- ◇ 程序开始
- ◇ 分配数的存储空间
- ◇ 输入两个数给 x、y
- ◇ 如果 x 大于 y
- ◇ 输出：x 大于 y

- ◇ 否则
- ◇ 输出: x 小于 y
- ◇ 程序结束

根据上述描述, 所设计的程序流程图如图 1.14 所示。

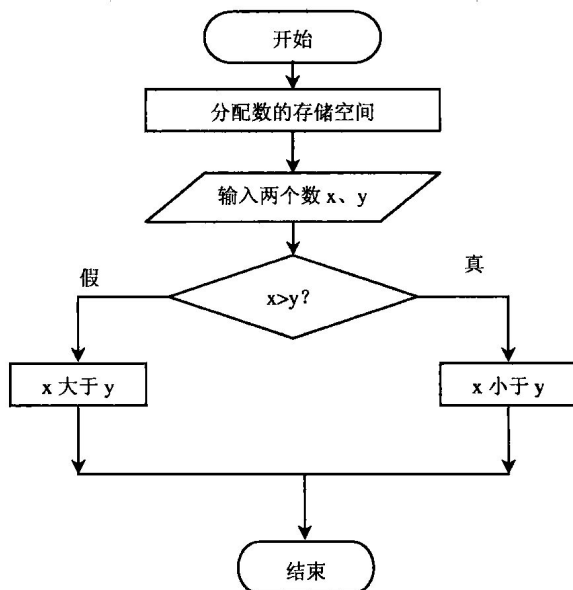


图 1.14 两数中求最大值流程图

用 C 语言写成的源程序如下。

```

#include<stdio.h> /*包含标准输入输出库预处理命令*/
int main() /*主函数开始*/
{
    /*程序从此开始执行*/
    int x,y; /*定义两个变量*/
    scanf("%d %d",&x,&y); /*读入两个整型数给变量 x、y*/
    if(x>y) /*如果 x 大于 y*/
        printf("x 大于 y\n"); /*输出: x 大于 y */
    else /*否则*/
        printf("x 小于 y\n"); /*输出: x 小于 y */
    return 0;
}
  
```



上机练习

步骤 1: 在 Visual C++ 环境下新建文件可单击菜单“开始→程序→Microsoft Visual C++ 6.0→Microsoft Visual C++ 6.0”来实现, 关闭“每日提示”对话框。

步骤 2: 单击菜单“文件→新建→文件→C++ Source File”, 输入文件名 1-2.c, 单击“确定”按钮。

步骤 3: 在编辑框内输入 C 源程序 1-2.c, 如图 1.15 所示。