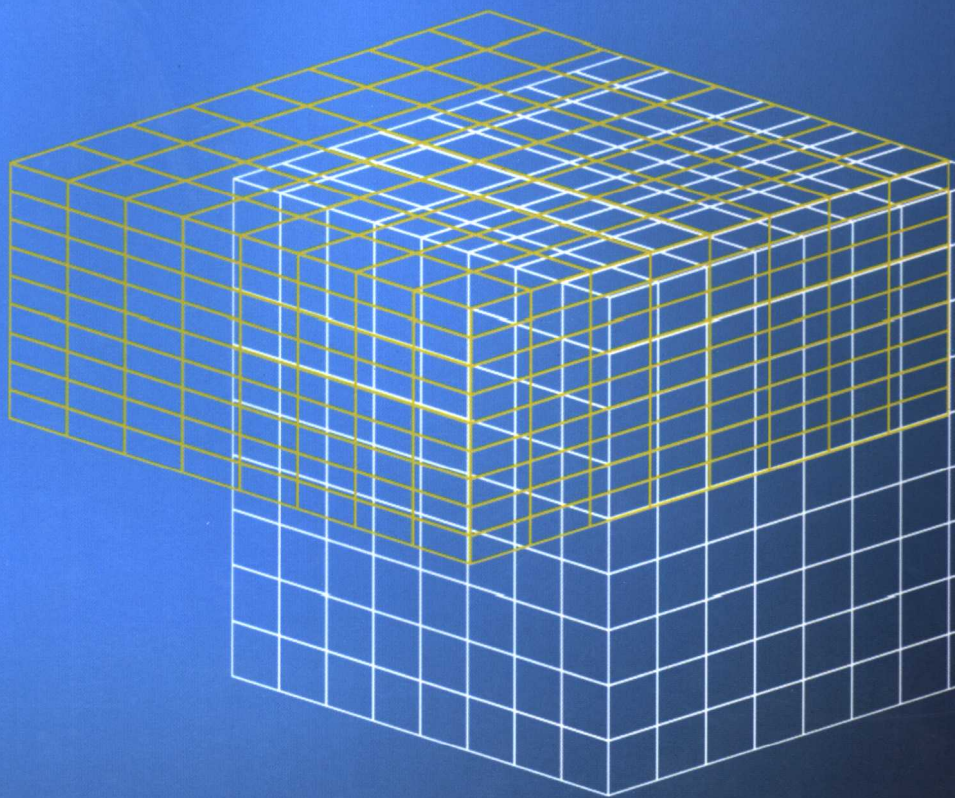




普通高等教育“十一五”国家级规划教材
(高职高专教育)



C语言程序设计

乌云高娃 温希东 王明福 主 编
沈翠新 杨淑萍 副主编



高等教育出版社

普通高等教育“十一五”国家级规划教材

(高职高专教育)

C 语言程序设计

乌云高娃 温希东 王明福 主 编

沈翠新 杨淑萍 副主编

高等教育出版社

内容提要

本书是普通高等教育“十一五”国家级规划教材（高职高专教育）。
本书以培养学生的C语言应用能力为主线，强调理论教学与实训密切结合。通过大量实际任务的分析和程序实现，学习C语言程序设计的知识点和语法。本书面向高等职业教育，教材编写充分考虑高等职业院校学生的学习基础、学习习惯与培养目标，案例取材于生产、生活实际，将流程图描述的算法贯穿全书，程序书写遵循企业规范，充分体现职业性。
本书根据ANSI C的新规则，引入了函数原型、void关键字等内容。书中C语言经典内容包括运算符和表达式、程序的三种结构、数组、函数、构造类型数据、指针、文件等均从任务解决中引入，大大降低学习门槛，在教材展开过程中步步深入，切合初学者学习专业基础课程课时少、台阶式进步的需求。
本书取材新颖、概念清楚、语言简洁流畅、结构合理、通俗易懂、适用性强，便于教师指导教学和学生自学，可作为应用性、技能型人才培养的各类教育“C语言程序设计”课程的教学用书，也可供各类培训、计算机从业人员和爱好者参考使用。

图书在版编目（CIP）数据

C语言程序设计 / 乌云高娃，温希东，王明福主编.
北京：高等教育出版社，2007.7
ISBN 978-7-04-021837-4

I. C... II. ①乌…②温…③王… 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（2007）第067268号

策划编辑 冯 英 责任编辑 关 旭 封面设计 于 涛 责任绘图 朱 静
版式设计 马静如 责任校对 王效珍 责任印制 韩 刚

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京汇林印务有限公司

开 本 787×1092 1/16
印 张 17.25
字 数 420 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 [http:// www.hep.edu.cn](http://www.hep.edu.cn)
[http:// www.hep.com.cn](http://www.hep.com.cn)
网上订购 [http:// www.landracom.com](http://www.landracom.com)
[http:// www.landracom.com.cn](http://www.landracom.com.cn)
畅想教育 [http:// www.widedu.com](http://www.widedu.com)

版 次 2007年7月第1版
印 次 2007年7月第1次印刷
定 价 23.60元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。
版权所有 侵权必究
物料号 21837-00

前 言

C 语言是目前世界上应用范围最广, 使用最多的高级程序设计语言。国内外各高等职业院校、本科院校中计算机、电子等相关专业均开设了 C 语言程序设计课程。

“C 语言程序设计”作为一门专业基础课, 必须理论与实践并重, 作为高职高专院校的基础课, 必须充分考虑高等职业院校学生的学习基础、学习习惯与培养目标, 在教材中体现职业性特色。本书在 4 个方面体现高职特色。

(1) 在案例选材方面, 大量选用与生产、生活比较贴近的实际问题, 将从真正意义上实现案例教学。精心进行实践性教学设计, 将在几届教学中实践成熟的项目拓展到教学中, 循序渐进、横向拓展、纵向深入。通过问题分析、算法描述和程序实现, 将实际问题的解决同 C 语言程序设计的知识和语法有机结合。由于案例取材于实际项目, 使得学生能够从分析常见问题入手, 逐步将其转化为程序描述, 符合高等职业院校教育的培养目标。

(2) 本书将流程图描述算法贯穿全书。算法是程序设计的灵魂, 是程序设计基础课程的核心内容。大量采用流程图描述算法, 对于初学者来说, 直观易懂、思路清晰、印象鲜明, 同时能训练读者的编程思路和逻辑思维, 便于教师引导教学。

(3) 本书所有程序按照企业规范书写。程序设计课程最终是要培养程序开发人员的编程能力, 本书作为程序设计的基础课程, 注重在程序设计学习的最初加强编程规范意识的培养, 通过将变量命名规则、程序注释、程序书写的缩进格式等方面融入教学的各个环节, 使读者熟悉企业编程角色, 养成专业、规范的编程习惯。

(4) 在教材结构方面, 开门见山, 深入浅出, 注重学习的连贯性和渐进性。章节之间的实例具有连续性, 在实例后提出拓展功能和程序改造设想, 有助于学生进一步自主学习和教师根据教学情况进行引导、深化。通过『学一学』、『试一试』、『想一想』、『练一练』四个层次进行知识点的引入、技能应用、编程实践与深入思考的规划, 有效实现深度与广度的拓展、能力与思维的提升, 对复杂一些的问题在『试一试』中按照『解题步骤』、『流程图』、『程序代码』和『说明』四个步骤进行问题的分析与解决。

本书第 1 章的“程序算法基础”和“软件编程规范”两小节可以在课程开始时介绍, 也可以在课程的进行中根据任务解决的需要介绍; 本书第 7 章“位运算”可供电子、控制类专业选用; 第 11 章“图形显示技术的应用”一节可供图形处理、动画类专业选用; 第 5 章中的冒泡法排序和第 9 章的链表可以根据实际情况进行选学。另外, 第 12 章常见错误分析与程序调试, 适合以上各专业的读者提高编程实践能力。

本教材的编写既注重知识的深度与广度, 又兼顾能力的提升与拓展, 层次分明, 重点突出, 适合按需取舍, 其中加“*”号的部分为选学内容。

本书由主编进行规划与统稿, 由乌云高娃(第 1、5 章)、温希东(第 7、8 章)、王明福(第 9、11 章)、沈翠新(第 4、6、12 章)、杨淑萍(第 2、3、10 章)合作编写, 邓雪峰也参与了

前言

本书的编写工作。安志远教授和樊月华教授审阅了全书，提出了许多宝贵的意见和建议，在此对他们的工作表示衷心的感谢。本教材在构思和编写过程中得到赵杰、石光华等老师的指导和帮助，特此致谢。由于时间紧迫和编者水平的限制，书中不足甚至错误在所难免，敬请读者多提宝贵意见。

本书依托国家精品课程，围绕教材和课程建设的相关网络资源极其丰富，相关内容可参阅C语言国家精品课程网站 (<http://jpkc.szpt.edu.cn/cyy/>)。

编 者

2007年2月于深圳职业技术学院

目 录

第1章 C语言程序设计基础.....1	
1.1 C语言的产生和发展.....1	
1.2 C语言的特点.....1	
1.3 C语言程序的结构.....2	
1.4 C语言程序的上机步骤.....3	
1.4.1 编译环境的准备.....4	
1.4.2 编译环境的设置.....5	
1.4.3 使用Turbo C 2.0.....6	
1.5 程序算法基础.....8	
1.5.1 程序设计基本概念.....8	
1.5.2 算法的特性.....8	
1.5.3 算法的描述.....10	
1.5.4 程序设计方法.....13	
1.6 软件编程规范.....14	
1.6.1 程序的可读性原则.....15	
1.6.2 匈牙利命名法.....16	
1.7 编译预处理.....16	
1.7.1 文件包含.....16	
1.7.2 宏定义#define命令.....17	
习题一.....18	
第2章 顺序结构流程及应用.....20	
2.1 基本字符、标识符和关键字.....20	
2.1.1 基本字符.....20	
2.1.2 标识符.....20	
2.1.3 关键字.....21	
2.2 数据类型及常量、变量.....21	
2.2.1 常量.....22	
2.2.2 变量.....22	
2.3 数据的格式化输出和输入.....25	
2.3.1 数据的格式化输出.....25	
2.3.2 数据的格式化输入.....27	
2.3.3 单个字符的输出和输入.....28	
2.4 运算符与表达式.....30	
2.4.1 算术运算符和算术表达式.....30	
2.4.2 赋值运算符和赋值表达式.....35	
2.4.3 逗号运算符和逗号表达式.....36	
2.5 顺序结构程序设计.....38	
习题二.....40	
第3章 选择结构流程及应用.....43	
3.1 选择结构判定条件的构成.....43	
3.1.1 关系运算符与关系表达式.....43	
3.1.2 逻辑运算符与逻辑表达式.....44	
3.2 单分支和双分支选择结构 程序设计.....46	
3.2.1 简单if语句.....47	
3.2.2 if-else语句.....48	
3.2.3 条件运算符与条件表达式.....49	
3.3 多分支选择结构程序设计.....51	
3.3.1 嵌套if-else语句.....51	
3.3.2 switch语句.....55	
习题三.....59	
第4章 循环结构的流程及应用.....62	
4.1 while语句的流程与应用.....62	
4.2 do-while语句的流程与应用.....69	
4.3 for语句的流程与应用.....71	
4.4 循环结构的综合实例.....77	
习题四.....82	
第5章 数组及其应用.....84	
5.1 一维数组的应用.....84	
5.2 二维数组的应用.....97	
5.3 字符数组与字符串的应用.....104	
习题五.....114	

第 6 章 函数	117	9.3 指向结构体的指针变量	191
6.1 函数的定义、调用及简单应用	117	9.4 线性链表	196
6.2 数组做函数参数的应用	127	习题九	205
6.3 函数的嵌套调用与递归调用	135	第 10 章 文件及其应用	209
6.4 变量的作用域和生存期	140	10.1 文本文件的读写	209
习题六	146	10.2 二进制文件的读写	216
第 7 章 位运算	148	10.3 文件的定位与随机读取	223
7.1 位运算和位运算符	148	习题十	230
7.2 位段	155	第 11 章 文本窗口与图形显示技术	232
习题七	158	11.1 文本窗口的程序设计	232
第 8 章 结构体与共用体及其应用	160	11.2 图形显示技术的应用	235
8.1 结构体变量的应用	160	11.3 汉字显示技术及其应用	242
8.2 结构体数组的应用	164	习题十一	247
8.3 结构体在函数中的应用	166	第 12 章 常见错误分析与程序调试	249
8.4 共用体的应用	171	12.1 常见错误分析	249
习题八	174	12.2 程序调试	258
第 9 章 指针	176	附录 A ASCII 码表	261
9.1 指向变量的指针变量	176	附录 B C 语言运算符的优先级和结合性	262
9.2 指向数组的指针变量	183	附录 C 常用库函数	263

学习目标

- C 语言的产生、发展和特点。
- C 程序的结构和 C 程序的上机步骤。
- 程序算法基础和软件编程规范。

1.1 C 语言的产生和发展

C 语言是目前极为流行的一种高级语言，它既具有高级语言的功能，又具有机器语言的一些特性。它是 1972 年由美国贝尔实验室的 Dennis Ritchie 和 Brian Kernighan 等人首先推出的。之后，C 语言又经过不断改进，使其逐步完善。直到 1978 年 Brian Kernighan 和 Dennis Ritchie（简称 K&R）合著了影响深远的名著 *The C Programming Language*，首次向世人系统介绍了 C 语言。建立了所谓的 C 语言的 K&R 标准，它一度成为 C 语言的事实标准。

此后，C 语言的发展非常迅速，各种版本的 C 语言相继涌现出来。由于没有统一的标准，使得这些 C 语言之间出现了一些不统一的地方，为了改变这种情况，美国国家标准学会（american national standards institute, ANSI）于 1983 年制定了一套标准，称为 ANSI C（标准 C），成为各种 C 语言版本的基础。

20 世纪 80 年代中期，出现了面向对象程序设计的概念，贝尔实验室的 B. Stroustrup 博士借鉴了 Simula 67 中的类的概念，将面向对象的语言成分引入到 C 语言中，设计出了 C++ 语言，C++ 语言赢得了广大程序员的喜爱，不同的机器不同的操作系统几乎都支持 C++ 语言，如 PC 上，微软公司先后推出了 MS C++、Visual C++ 等产品，Borland 公司先后推出了 Turbo C++、Borland C++、C++ Builder 等产品，同时，C++ 语言也得到了国际标准化组织（international organization for standardization, ISO）的认可，并已对 C++ 语言实现标准化。

目前，微机中使用的 C 语言版本很多，比较经典的有 Turbo C、Borland C、Microsoft C 等。近年来，又推出了包含面向对象程序设计思想和方法的 C++，它们均支持 ANSI C，本书主要介绍 ANSI C 中的基础部分，同时兼顾各种版本的通用性和一致性。

1.2 C 语言的特点

C 语言之所以能存在和发展并具有强大的生命力，成为程序员的首选语言之一，是因为它具有如下特点：

- ① C 语言既具有高级语言的通用性及易写易读的特点，又具有汇编语言的“位处理”、“地

址操作”等能力。这使得 C 语言不仅像 Pascal、FORTRAN、BASIC 等高级语言那样用于应用软件的设计，还能像汇编语言那样用于计算机系统软件和控制软件的开发。

② C 语言是一种结构化程序设计语言，具有丰富的数据类型、众多的运算符，这使得程序员能轻松地实现各种复杂的数据结构和运算；C 语言所具有的体现结构化程序设计的控制结构和具备抽象功能及体现信息隐蔽思想的函数，可以实现程序的模块化设计。

③ 语句简练、紧凑，语法规则少，使用方便灵活。编译后生成的代码质量高，运行速度快。

④ C 语言具备良好的可移植性。若程序员在书写程序时严格遵循 ANSI C 标准，则其源代码基本上可不做修改，就能用于各种型号的计算机和各种操作系统。

⑤ 语言功能丰富。它不仅提供了丰富的运算符，还提供了各种功能强大的系统函数。

尽管 C 语言有很多优点，但也存在一些缺点和不足。比如它的类型检验和转换比较随便，优先级太多不便记忆。这些都对程序设计者提出了更高的要求，也给初学者增加了难度。

C 语言的上述特点，读者可在学习过程中逐渐体会，加深理解。

1.3 C 语言程序的结构

用 C 语言编写的源程序，简称 C 程序。C 程序是一种函数结构，一般由一个或若干个函数组成，其中必有一个名为 `main()` 的函数，程序的执行就是从这里开始的。

下面先介绍简单的 C 程序，然后从中分析 C 程序的结构。

问题 1.1 在屏幕上输出一行文本信息“Hello,world!”。

【程序代码】

```
#include "stdio.h"
void main()           /*主函数*/
{ printf("Hello,world!"); /*在屏幕上输出 Hello,world!*/
}
```

【说明】

① 本程序的功能是在屏幕上显示一行文本信息“Hello,world!”。

② `main()` 为主函数名。每个 C 程序都必须有一个 `main()` 函数，这是 C 程序执行的入口地址。

③ 大括号“{ }”是函数体界定符，位于大括号中的内容称为函数体，每个函数都必须用一对大括号将函数体括起来。

④ 输出语句 `printf("Hello,world!");` 的作用是将引号中的内容“Hello,world!”原样输出。`printf()` 为 C 语言的标准输出函数，是系统提供的库函数。

⑤ 语句后面有一个分号“;”，这是 C 语言的语句结束符。

⑥ 位于“/*...*/”之间的内容是注释语句，用来帮助读者阅读程序，在程序编译运行时这些内容是不起作用的，注释语句可写在程序中的任何位置。

问题 1.2 计算一个学生数学和英语成绩的平均分，并输出结果。

【程序代码】

```

#include "stdio.h"                /*编译预处理命令*/
void main()
{ int iMath,iEnglish;
  float fAverage;                /*定义变量*/
  iMath = 80;
  iEnglish = 88;                 /*给变量赋值*/
  fAverage=(iMath+iEnglish)/2.0; /*计算平均分并存入变量 fAverage*/
  printf("The average is %0.1f",fAverage); /*输出结果值*/
  getch();
}

```

程序运行结果如下:

The average is 84.0

【说明】

- ① `#include` 是编译预处理命令,放在源程序的最前面,编译预处理语句后面不加分号。
- ② 程序中变量在使用前要定义,定义语句放在可执行语句之前。
- ③ `printf()` 是系统提供的函数,在将 “The average is %0.1f” 输出时, “%0.1f” 由 `fAverage` 的值取代。

- ④ C 程序的一般形式如下:

```

预处理命令序列
void main()
{
    变量定义序列
    执行语句序列
}

```

其中:

- 预处理命令序列: 书写程序相关的预处理文件。
- 变量定义序列: 是声明部分,用来定义程序中所用到的变量。
- 执行语句序列: 是程序的执行部分,由若干语句组成,完成对数据的运算及各种处理。

以上编译预处理、变量定义、执行语句这 3 个序列可称为 C 程序结构上的三大区域,这三大区域在程序中是不可调换位置的,程序也将按这个顺序执行。

⑤ C 程序中除了可用库函数,也可使用程序员自行编写的用户函数,这将在有关函数的章节中学习。

⑥ C 程序的书写格式自由,一行内可以写一条或几条语句,一条语句也可以分写在多个行上。C 程序没有行号,每条语句和变量定义必须用一个分号结尾。

1.4 C 语言程序的上机步骤

前面已经看到了一些用 C 语言编写的程序,但它是不能直接运行的。因为计算机只能识别和执行由 0 和 1 组成的二进制的指令,而不能识别和执行用高级语言写的程序。为了使计算机

能执行高级语言所写的程序，必须先用一种称为“编译程序”的软件，把程序翻译成二进制形式的“目标程序”(target program)，然后将该目标程序与系统的函数库和其他目标程序连接起来，形成可执行的目标程序才能被机器所执行。相对于目标程序，我们用高级语言编写的程序被称为“源程序”(source program)。

在此，介绍利用 Turbo C 2.0 编译程序，把 C 语言源程序编译、连接生成可执行程序(*.exe 文件)的方法。假设 C 源程序名为 MY.c，具体操作步骤如下：

① 编辑源程序，并以扩展名.c 的文件“MY.c”存盘。

② 对源程序进行编译，将源程序转换为扩展名为.obj 的目标程序“MY.obj”，但目标程序仍不能运行。若源程序有错，必须予以修改，然后重新编译。

③ 对编译通过的源程序进行连接，即加入库函数和其他二进制代码生成可执行程序。连接过程中，若出现未定义的函数等错误，必须修改源程序，并重新编译和连接。

④ 执行生成的可执行代码，若不能得到正确的结果，必须修改源程序，重新编译和连接。若能得到正确结果，则整个编辑、编译、连接、运行过程顺利结束。

以上步骤用流程图表示如图 1-1 所示。下面介绍 Turbo C 2.0 集成开发环境的安装与使用。

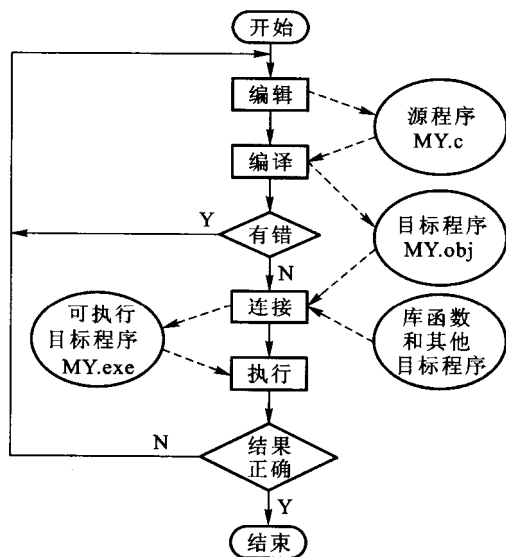


图 1-1 MY.c 程序流程图

1.4.1 编译环境的准备

1. 安装 Turbo C 2.0

Turbo C 2.0 是在微机上广泛使用的一个 C 程序开发环境，它具有方便、直观、易用的界面和丰富的库函数。它把程序的编辑、编译、连接和运行等操作全部集中在一个界面上进行，使用非常方便。

Turbo C 2.0 不需特别安装，只要将其复制至硬盘即可运行。

2. 建立工作目录

为了方便管理用户的程序文件和维护 Turbo C 2.0 的运行环境,应在机器的磁盘上建立用户自己的工作目录,以便用来存放自己所开发的源程序文件,例如,在 D 盘根目录下建立子目录 D:\MYC。

1.4.2 编译环境的设置

1. 设置工作目录

设置工作目录应按如下步骤操作:

- ① 启动 Turbo C 2.0 (简称 TC), 进入 TC 编辑界面。
- ② 按下功能键 F10, 这时光条就会跳到主菜单。
- ③ 用左、右方向键移动光带, 定位于“File”菜单, 按回车键。
- ④ 用上、下方向键移动光带, 定位于“Change dir”子菜单, 按回车键, 进入“New Directory”编辑框。
- ⑤ 编辑工作目录为“D:\MYC”, 如图 1-2 所示。

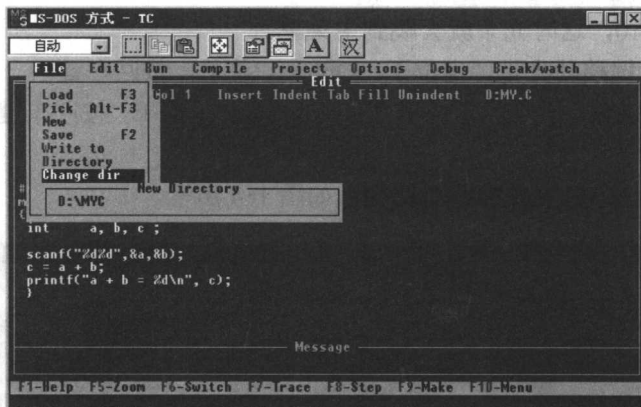


图 1-2 设置工作目录

2. 设置编译环境

设置编译环境应按如下步骤操作:

- ① 按下功能键 F10, 这时光条就会跳到主菜单。
- ② 用左、右方向键移动光带, 定位于“Options”菜单, 按回车键。
- ③ 用上、下方向键移动光带, 定位于“Directories”子菜单, 按回车键, 进入如图 1-3 所示编辑框。
- ④ 编辑框中可分别设置:
 - Include directories: /* Turbo C 的包含文件所在目录*/
 - Library directories: /* Turbo C 库函数所在目录*/
 - Output directory: /* 输出目录*/
 - Turbo C directory: /* Turbo C 所在目录*/

- Pick file name: /* 环境设置的保存文件（绝对路径）*/

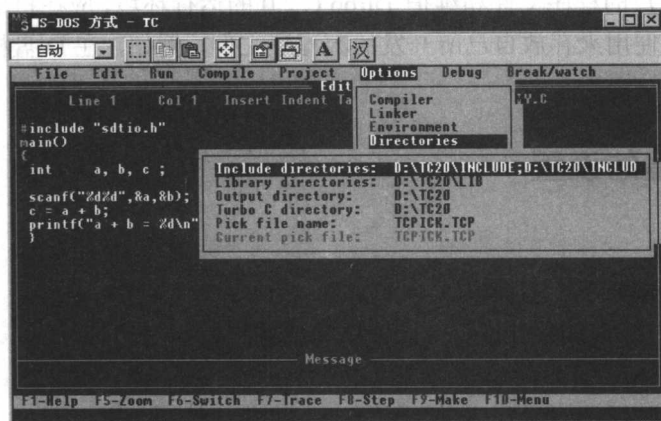


图 1-3 设置编译环境

3. 保存环境设置文件

保存环境设置文件按如下步骤操作：

- ① 按下功能键 F10，这时光条就会跳到主菜单。
- ② 用左、右方向键移动光带，定位于“Options”菜单，按回车键。
- ③ 用上、下方向键移动光带，定位于“Save options”子菜单，按回车键，进入“Config File”编辑框。
- ④ 在编辑框中输入 TC 文件的绝对路径，如图 1-4 所示。

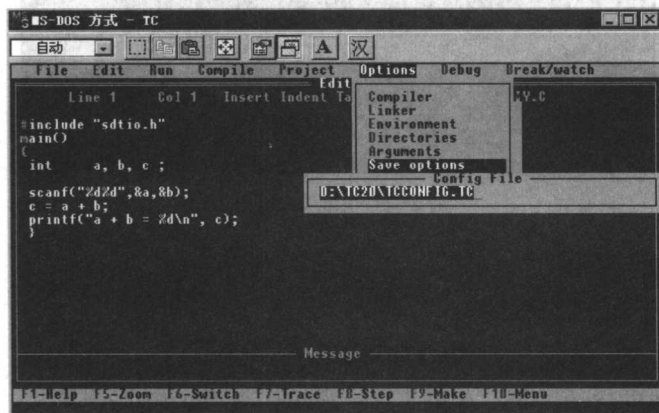


图 1-4 保存环境设置

1.4.3 使用 Turbo C 2.0

Turbo C 是一个在 DOS 环境下运行的集成开发环境。关于 Turbo C 2.0 环境中各项菜单的功能和用法，可查阅相关的 Turbo C 2.0 用户手册，这里不再赘述。下面，将通过一个求和的程

序来讲解一下在 Turbo C 2.0 中编辑、编译、连接、运行一个 C 程序的一般步骤。

1. 编辑一个 C 源程序文件

Turbo C 2.0 启动后, 在 Edit 状态下输入或修改源程序代码。输入完毕后, 选择“File/Save”菜单项, 并将程序名命名为 MY.c, 按回车键, 将源程序文件存盘保存, 如图 1-5 所示。

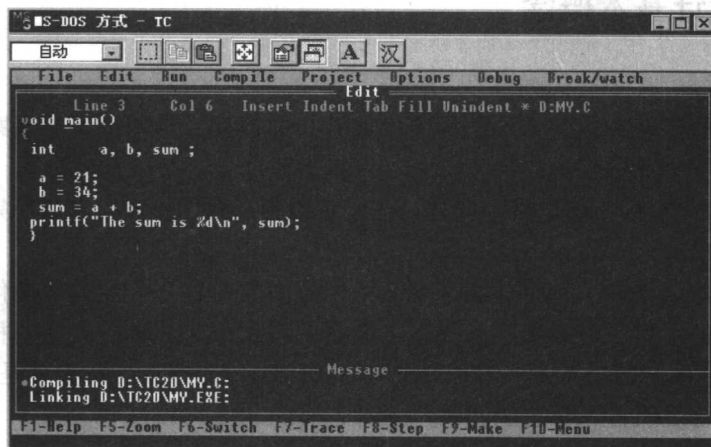


图 1-5 编辑一个 C 程序

一般情况下, 在输入完源程序代码后, 最好先将其存盘, 以防调试过程中出现死机而导致程序文件丢失。

2. 编译、连接源程序文件

选择“Compile/Compile to OBJ (或 Compile/Build all)”菜单项, 编译源文件, 生成目标代码文件 MY.obj (和可执行文件 MY.exe)。如果程序没有错误, 则编译后的屏幕显示如图 1-6 所示; 若有错误, 则需要修改源程序后, 再重新进行此过程。

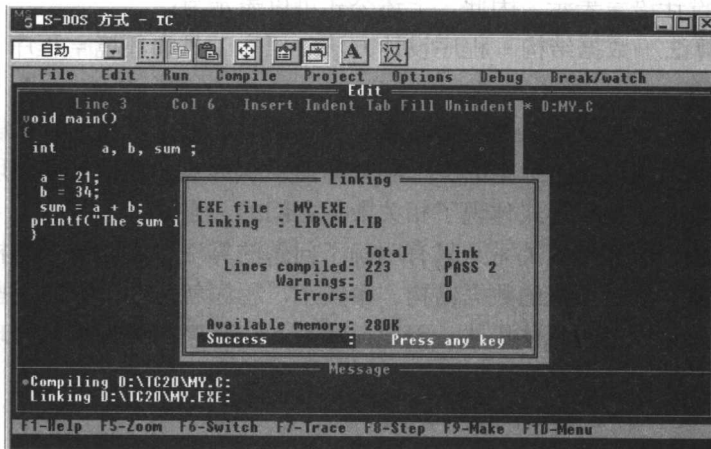


图 1-6 编译、连接

3. 运行可执行文件

执行“Run/Run”命令或按 Ctrl+F9 组合键运行该程序, 即可得到程序运行后显示的结果。

1.5 程序算法基础

1.5.1 程序设计基本概念

1. 程序与算法

人们做任何事情都有一定的方法和程序。如开会的议程、老师上课的教案、春节联欢晚会的节目单等都是程序。在程序的指导下，人们可以有秩序地、有效地完成每一项工作。随着计算机的问世和普及，“程序”逐渐被专业化，它通常特指：为让计算机完成特定任务（如解决某一问题或控制某一过程）而设计的指令序列。

从程序设计的角度来看，每个问题都涉及两个方面的内容——数据和操作。所谓“数据”泛指计算机要处理的对象，包括数据的类型、数据的组织形式和数据之间的相互关系，这些又被称为“数据结构”(data structure)；所谓“操作”是指处理的方法和步骤，也就是算法(algorithm)。而编写程序所用的计算机语言称之为“程序设计语言”。

换言之，一个程序应包括以下两方面的内容：

- ① 对数据的描述，即数据结构。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。
- ② 对数据处理的描述，即算法。算法是为了解决一个问题而采取的方法和步骤。

算法反映了计算机的执行过程，是对解决特定问题的操作步骤的一种描述。数据结构是对参与运算的数据及它们之间关系所进行的描述，算法和数据结构是程序的两个重要方面。因此，著名的计算机科学家沃斯(Niklaus Wirth)提出过一个经典公式：

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

实际上，一个程序除了以上两个要素之外，还应当采用结构化程序设计方法进行程序设计，并且用某一种程序设计语言表示。因此，上述公式可以表示为：

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

2. 数据结构

计算机处理的对象是数据，数据是描述客观事物的数、字符以及计算机能够接收和处理的信息符号的总称。数据结构是指数据的类型和数据的组织形式。数据类型体现了数据的取值范围和合法的运算，数据的组织形式体现了相关数据之间的关系。

数据结构与算法有着密切的关系，只有明确了问题的算法，才能更好地构造数据结构；但选择好的算法，又常常依赖于好的数据结构。事实上，程序就是在数据的某些特定的表示方式和结构的基础上对抽象算法的具体描述。因此，编写一个程序的关键就是合理地组织数据和设计好的算法。

1.5.2 算法的特性

下面看两个简单的实例。

问题 1.3 比较 3 个人的身高，将个子最高的那个人的身高输出到屏幕上。

【解题步骤】

问题分析：设变量 fTall1、fTall2 和 fTall3 分别存放 3 个人的身高，fMax 存放身高的最大值。为求最大值，要对 3 个数进行比较，具体步骤如下：

① 输入 3 个人的身高数据 fTall1、fTall2 和 fTall3。

② 把第 1 个人的身高 fTall1 的值赋给 fMax。

③ 将第 2 个人的身高 fTall2 与 fMax 的值进行比较。如果 $fTall2 > fMax$ ，则把第 2 人的身高数据 fTall2 的值赋给 fMax；否则，跳过本步骤，执行第④步。

④ 将第 3 个人的身高 fTall3 与 fMax 的值进行比较。如果 $fTall3 > fMax$ ，则把第 3 个人的身高 fTall3 的值赋给 fMax；否则，跳过本步骤，执行第⑤步。

⑤ 输出 fMax 的值，即个子最高的人的身高数据。

问题 1.4 本题是一个经典的数学问题。对一个大于 2 的正整数，判断它是不是一个素数。

问题分析：所谓素数，是指除 1 和该数本身之外，不能被其他任何整数整除的数。例如，7 是素数，因为它不能被 2、3、4、5、6 整除。

判断一个数 $n(n > 2)$ 是否为素数的方法很简单：将 n 作为被除数，将 $2 \sim n-1$ 各个整数轮流作为除数，如果都不能整除 n ，则 n 为素数。

可按如下步骤去做：

① 输入 n 的值。

② 设 $i=2$ (i 作为除数)。

③ n 除以 i ，求得余数 r 。

④ 判断：如果 $r=0$ ，表示 n 能被 i 整除，则 n 不是素数，算法结束；否则执行第⑤步。

⑤ $i+1$ 赋值给 i 。

⑥ 如果 $i < n$ ，返回第③步；否则打印 n 是素数，算法结束。

实际上， n 不必除以 $2 \sim n-1$ 之间的全部整数，只需除以 $2 \sim \sqrt{n}$ 之间的整数即可。例如，判断 13 是否素数，只需将 13 除以 2、3 即可。如果除不尽， n 必为素数。因此，第⑥步可改为：

如果 $i \leq \sqrt{n}$ ，返回第③步；否则打印 n 是素数，算法结束。

从以上两例可以看出，遇到问题时首先分析题目，然后寻找一种实现这个问题所要完成功能的方法，这种方法的具体化称为算法。因此可以说，算法是由一套明确的规则组成的一些步骤，它指定了操作顺序并通过有限个步骤把问题解决，得出结果。

一个算法应具有以下 5 个特性：

(1) 有穷性

一个算法应包含有限个操作步骤，而不能是无限的。即对于一个算法，要求其在时间和空间上是有穷的。例如，问题 1.4 判断任一正整数 n 是不是一个素数的算法。因为 n 是确定的，最多经过对它前面的 $(n-2)$ 个数轮流作为除数，就能得出结论，所以具有“有穷性”。

(2) 确定性

算法中的每一步都应当是确定的，而不应当是含糊的、模棱两可的，也就是要求必须有明确的含义，不允许存在二义性。例如，“将成绩优秀的同学名单打印输出”，在这一描述中，“成绩优秀”这种描述方式就很不明确，是每门功课必须在 95 分以上？还是指总成绩在 85 分以上？不明确。

(3) 有效性

有效性也叫可执行性。算法中描述的每一步操作都应该能有效地执行，并得到确定的结果。例如，当 $y=0$ 时， x/y 是不能有效执行的。又如，C 语言中的求余运算“%”，要求两个操作数都必须是整型数，否则求余数运算不能有效执行。

(4) 输入

一个算法有 0 个或多个输入数据。例如，计算 1~10 的累加和的算法无需输入数据；而求 $n!$ 的算法，一般来说需要从键盘输入 n 的值。

(5) 输出

算法的目的是为了求解，而“解”要输出。所以，一个算法应该有一个或多个输出，例如，问题 1.4（判断任一正整数 n 是不是一个素数）的算法，最后打印出的 n “是素数”或“不是素数”就是输出。没有输出的算法是毫无意义的。

1.5.3 算法的描述

算法的表示方法很多，常见的有自然语言、传统流程图、N-S 图、伪代码、PAD 图等。

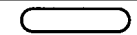
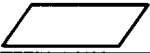
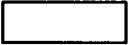
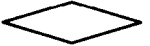
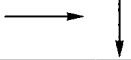
1. 用自然语言描述

自然语言就是人们日常使用的语言，可以是中文、英文等。用自然语言表示的算法通俗易懂，但一般篇幅较冗长，表达上往往不易准确，容易引起理解上的“歧义性”。所以，除了很简单的问题外，一般不使用这种描述。

2. 用传统流程图描述

传统流程图是用一组规定的图形符号、流程线和文字说明来表示各种操作、算法的方法，它直观形象，易于理解。ANSI 规定了一些常用的流程图符号，如表 1-1 所示。

表 1-1 传统流程图常用符号

符 号	符 号 名 称	含 义
	起止框	表示算法的开始和结束
	输入输出框	表示输入输出操作
	处理框	表示对框内的内容进行处理
	判断框	表示对框内的条件进行判断
	流程线	表示流程的方向
	连接点	通常用于换页处，表示两个具有同一标记的“连接点”应连接成一个点
	显示内容	表示输出到屏幕上的内容
	预先定义的进程	表示预先定义的函数、子例程等

用传统流程图描述问题 1.3 的算法如图 1-7 所示。

理论已经证明，任何复杂的算法都可以用顺序、分支和循环这 3 种基本结构组合、嵌套来描述。由于传统流程图用流程线（箭头）指明了各操作执行的方向和顺序，充分体现了结