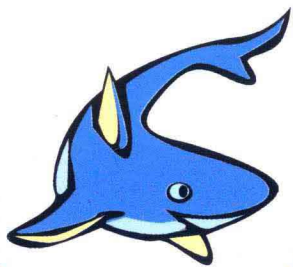




21世纪大学计算机基础规划教材

C/C++ 程序设计上机指导与测试

(第二版)



王连相 赵付青 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21 世纪大学计算机基础规划教材

C/C++程序设计上机指导与测试

(第二版)

王连相 赵付青 主编

内 容 简 介

本书是与《C/C++程序设计教程(第二版)》配套使用的学习用书。内容包括上机指导篇和测试篇:上机指导篇给出了每次的上机目的、上机内容、上机指导和思考题;测试篇根据知识点给出了大量的练习题,同时给出了几套非常实用的笔试综合测试题、上机练习题、上机考试模拟题和参考答案。

本书适合作为高校相关课程实践环节的教材,也可作为各种培训教材以及编程爱好者和参加全国计算机等级(二级C语言)考试人员的自学参考书。

图书在版编目(CIP)数据

C/C++ 程序设计上机指导与测试/王连相,赵付青主
编. —2 版. —北京:中国铁道出版社,2011.1 (2012.7 重印)
21 世纪大学计算机基础规划教材
ISBN 978-7-113-12492-2

I. ①C… II. ①王…②赵… III. ①
C 语言—程序设计—高等学校—教学参考资料 IV.
①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 009465 号

书 名: C/C++ 程序设计上机指导与测试 (第二版)
作 者: 王连相 赵付青 主编

策划编辑: 吴宏伟 辛 杰
责任编辑: 辛 杰
封面设计: 付 巍
责任印制: 李 佳

读者热线电话: 400-668-0820
编辑助理: 尚世博
封面制作: 白 雪

出版发行: 中国铁道出版社 (北京市宣武区右安门西街 8 号 邮政编码: 100054)
印 刷: 航远印刷有限公司
版 次: 2006 年 9 月第 1 版 2011 年 1 月第 2 版 2012 年 7 月第 7 次印刷
开 本: 787mm×1092mm 1/16 印张: 14.5 字数: 342 千
印 数: 20 601~24 100 册
书 号: ISBN 978-7-113-12492-2
定 价: 23.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社计算机图书批销部联系调换。

第二版前言

本书是根据教育部《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》中有关“程序设计基础”课程教学要求，在《C/C++程序设计上机指导与测试》（第1版，2006年9月）的基础上修订而成的。本次修订精简了综合练习题和上机练习题的内容，增加了C++基础的习题、上机考试模拟题及参考答案。

本书针对现代教育教学改革理念，在提高教学效率的同时，力求提高学生综合实践的能力。本书是在多年软件开发和C/C++程序设计教学实践经验的基础上，根据现代高校教学改革特有的情况及现代计算机教学的规律，收集分析了大量的教学文献，并基于实际应用而编写的，是与《C/C++程序设计教程（第二版）》配套使用的学习用书。

本书分为上机指导篇和测试篇。上机指导篇分别按简单程序设计，顺序结构，选择结构，循环结构，函数，编译预处理，数组，指针，结构体、共用体和枚举，文件等主要知识点，给出了每次上机目的、上机内容、上机指导和思考题，上机内容中强调基本知识的掌握和基本技能的训练，思考题则给出了较多、要求较高的练习题，供教学和学习时根据不同的要求和学时选择使用，且给出了编程分析、参考程序和调试每个程序时的主要观测点，通过调试，进一步理解和掌握相关知识。测试篇根据知识点给出了大量的练习题，部分练习题可以作为上机练习的补充题目，同时为了便于检查阶段性教学或学习的状况，给出了几套非常实用的笔试测试题、上机练习题、上机考试模拟和参考答案。

全书由王连相和赵付青编写并定稿。本书在编写过程中得到了於时才、丁政建、周小健、滕永晨、张宪立、邹晓、廖成斌、张墨逸和雷彩云的大力支持和帮助。本书还得到了中国铁道出版社的鼎力支持和帮助，在此一并表示衷心的感谢。

由于作者水平有限，因此难免会出现一些错误和不足之处，希望广大读者批评指正。

编 者
2010年11月

第一版前言

本书针对现代教育教学改革理念，在提高教学效率的同时，力求提高学生综合实践的能力。本书是在多年软件开发和 C/C++ 程序设计教学实践经验的基础上，根据现代高校教学改革特有的情况及现代计算机教学的规律，收集分析了大量的教学文献，并基于实际应用而编写的，是与《C/C++ 程序设计教程》配套使用的学习用书。

本书分为上机指导篇和测试篇。上机指导篇分别按简单程序设计，顺序结构、选择结构和循环结构，函数，编译预处理，数组，指针，结构体、共用体和枚举，文件等主要知识点，给出了每次上机目的、上机内容、上机指导和思考题，上机内容中强调基本知识的掌握和基本技能的训练，思考题则给出了较多、要求较高的练习题，供教学和学习时根据不同的要求和学时选择使用，且给出了编程分析、参考程序和调试每个程序时的主要观测点，通过调试，进一步理解和掌握相关知识。测试篇根据知识点给出了大量的练习题，部分练习题可以作为上机练习的补充题目，同时为了便于检查阶段性教学或学习的状况，给出了几套非常实用的笔试测试题、上机测试题和参考答案。

全书由王连相编写和定稿，管会生对本书的编写给予了很大的指导和帮助。本书还得到了李晓丽、李睿、赵付青和中国铁道出版社姜淑静、曹莉群和戴薇等的鼎力支持和帮助，在此表示衷心的感谢。

由于教学任务繁重，加之本书编写时间紧迫，因此在书中难免会出现一些错误和不足之处，在此恳请广大读者批评指正，并提出宝贵意见。

编 者

2006 年 6 月

目 录

上机指导篇

上机 1 简单程序的编辑和运行	1
上机 2 顺序结构程序设计.....	10
上机 3 选择结构程序设计.....	14
上机 4 循环结构程序设计.....	17
上机 5 函数	21
上机 6 编译预处理	25
上机 7 数组	26
上机 8 指针	31
上机 9 结构体、共用体和枚举类型	35
上机 10 文件	38
上机综合训练.....	40

测 试 篇

习题 1 C 语言基础知识	41
习题 2 顺序结构	45
习题 3 选择结构和循环结构	49
习题 4 函数	63
习题 5 编译预处理	75
习题 6 数组	77
习题 7 指针	86
习题 8 C++基础	104
综合练习题	112
综合练习题参考答案.....	163
上机练习题	173
上机练习题参考答案.....	184
上机考试模拟题及参考答案.....	195
参考文献.....	225

上机指导篇

上机 1 简单程序的编辑和运行

一、上机目的

- 熟悉 C 语言的系统环境，学习编辑运行 C 语言程序的基本方法。
- 进一步认识程序的两种错误类型对程序结果的影响，加深对程序测试重要性的认识。
- 认识 C 语言程序的结构特点，学习程序的基本编写方法。

二、上机内容

1. Turbo C 2.0 集成环境的使用

使用 Turbo C 2.0 集成环境，编辑并运行实现乘法运算的程序。乘法运算的源程序如下：

```
#include "stdio.h"
main()
{
    int a,b;
    a=137;
    b=4;
    printf("%d\n",a*b);
}
```

/* 给 a 赋值 137 */
/* 给 b 赋值 4 */
/* 计算并输出表达式 a*b */

2. 标识符和程序的逻辑错误

修改上面的程序，观察运行结果。

- (1) 将程序中的表达式“a*b”中的“a”改为“A”，然后运行程序。
- (2) 将程序中的表达式“a*b”修改为“a-b”，然后运行程序。

3. 编辑并运行能够对任意两个整数进行乘法运算的程序

源程序如下：

```
#include "stdio.h"
main()
{
    int a,b;
    printf("Input a,b:");
    scanf("%d,%d",&a,&b);
    printf("a*b=%d\n",a*b);
}
```

/* 在屏幕显示提示信息 Input a,b: */
/* 该语句的作用是实现数据的键盘输入 */
/* 输出乘法运算的结果 */

三、上机指导

Turbo C 集成编译环境是一个集程序编辑、编译、连接、调试为一体的 C 程序开发环境，它具有速度快、效率高、功能强、使用方便等优点。用户在这个集成环境下，可以利用内部的编辑

器进行全屏幕编辑，利用窗口功能进行编译、连接、调试、运行、环境设置等工作。

如果计算机系统已经安装了 Turbo C 编译系统，则在 DOS 命令状态下输入命令：

TC

或

TC filename

其中 filename 是用户需要进行编辑、编译、连接、运行的 C 程序文件名。在前者情况下，该文件名可以在进入集成环境后再指定。

如果是在 Windows 环境下启动 Turbo C，只需要找到 TC.exe 文件双击即可打开集成开发环境。进入 Turbo C 集成开发环境后，首先在屏幕上显示 Turbo C 主菜单窗口，如图 1-1 所示。



图 1-1 Turbo C 主菜单窗口

由图 1-1 可以看出，在该窗口中，有 8 个菜单条目，即提供了 8 种选择。

File: 处理文件（包括装入、存盘、选择、建立、换名写盘），目录操作（包括列表、改变工作目录），退出系统及调用 DOS。

Edit: 建立、编辑源文件。

Run: 控制运行程序。如果程序已经编辑连接好，且将“Debug→Source debugging”以及“Options→Compiler→Code generation→OBJ debug information”开关置为 On 状态，则可以用此菜单初始化调试阶段。

Compile: 编译并生成目标程序与可执行文件。

Project: 允许说明程序中包含哪些文件的管理选项（project）。

Options: 可以选择集成环境选项（如存储模式、编译时的选项、诊断及连接选项）及定义宏；也可以记录 include、output 及 library 文件目录，保存编译选项和从配置文件加载选项。

Debug: 检查、改变变量的值，查找函数程序运行时查看调用栈。选择程序编译时是否在执行代码中插入调试信息。

Break/watch: 增加、删除、编辑监视表达式，及设置、清除、执行至断点。

特别要指出的是，除了 Edit 菜单项外，每一个菜单项下都有子菜单。而选择 Edit 菜单项后，只是进入编辑器。

为了从主菜单中选择所需要的功能，有以下两种方式可以选择：

- 按【F10】键后，可以看到屏幕上部主菜单中的某个选项处出现亮块。此时，利用左、右光标键（【←】与【→】）将此亮块移到所要选择的选项位置处，然后按【Enter】键，即出现相应的子菜单。
- 直接按【Alt】键+主菜单条目中的首字母（分别为【F】、【E】、【R】、【C】、【P】、【O】、【D】、

【B】), 此时就会出现相应的子菜单。例如, 按【Alt+F】组合键表示选择“文件”菜单(File)。

当出现子菜单时, 其中某个选项是高亮度的, 此时可以利用上、下光标键(【↑】与【↓】)来移动该高亮度线, 从而选择所需要的功能。在主菜单或通过主菜单调用的任意一个子菜单中, 按【Esc】键后将直接返回到活动窗口。

下面简要介绍各菜单的功能。

1. “文件”菜单(File)

当选中“File”菜单后, 在“File”下方将出现一个子菜单, 如图 1-2 所示。在有的子菜单命令右边还标出了实现该功能的热键。“热键”是指为执行菜单中某一固定功能而设置的键。通过热键来实现某种功能, 一般要比通过菜单选择更简单直接, 但要求用户熟记这些热键。例如, 为了调用 Pick 命令, 除了通过“File→Rick”菜单以外, 还可以直接用【Alt+F】组合键来选择。



图 1-2 “File”菜单

下面简要说明各项的功能:

(1) Load (加载)

装入一个文件。当给定的文件名中有文件名通配符(“*”或“?”)时, 将进行列表选择。

(2) Pick (选择)

将最近装入编辑窗口的 8 个文件列成表, 供用户选择。选择后再装入编辑器, 光标置于上次修改过的地方。若选择了“Load File”命令, 屏幕上将出现“Load File Name”提示框。

(3) New (新文件)

新建一个文件, 默认文件名为 NONAME.C。

(4) Save (存盘)

将编辑器中的内容存盘。若文件名为 NONAME.C, 而又要存盘, 编辑器会询问是否要改名。

(5) Write to (存盘)

将编辑器中的内容写入指定的文件中。若该文件已经存在, 则会重写。

(6) Directory

显示目录与文件列表。若直接按【Enter】键则选择当前文件; 移动光标选择文件名后, 将该文件装入编辑器。

(7) Change dir (改变驱动器)

显示当前文件, 改变驱动器与目录。

(8) OS Shell (暂时退出)

暂时退出 Turbo C, 转到 DOS 状态, 在 DOS 状态下用 Exit 命令又可返回 Turbo C。此功能对于想运行 DOS 命令但又不想退出 Turbo C 时非常有用。

(9) Quit (退出)

退出 Turbo C, 返回到 DOS 状态。

2. “编辑”菜单 (Edit)

调用内部编辑器。在编辑器中按【F10】键或按【Alt】键加所需主菜单命令的首字母可返回主菜单, 但此时编辑器中的内容仍保持在屏幕上。在主菜单中按【Esc】或【E】键即可回到编辑器; 也可以按【Alt+E】组合键, 且在任何时候都起作用。

3. “运行”菜单 (Run)

当选中“Run”菜单后, 在“Run”下方将出现一个子菜单, 如图 1-3 所示, 其中也列出了对应的热键。



图 1-3 “Run”菜单

下面简要说明各项的功能:

(1) Run (运行)

运行当前程序。

(2) Program reset (程序重启动)

中止当前调试, 释放分配给程序的空间, 关闭已打开的文件。

(3) Go to cursor (执行到)

使程序从程序头运行到编辑窗口中光标所在行。若光标所在行不含可执行代码语句, 则显示一个提示框作为警告。

(4) Trace into (跟踪进入)

运行当前函数中的下一个语句。若此语句不含调试器可访问的函数调用, 则停在下一条可执行语句上; 但若此语句含有调试器可访问的函数调用, 则停在函数定义的开始。

(5) Step over (单步执行)

执行当前函数的下一语句, 即使遇到调试语句可访问的函数调用也不会跟踪进入下一级函数中。

(6) User screen (用户屏幕)

切换到用户屏幕。

4. “编译”菜单 (Compile)

当选中“Compile”菜单后，在“Compile”下方将出现一个子菜单，如图 1-4 所示。

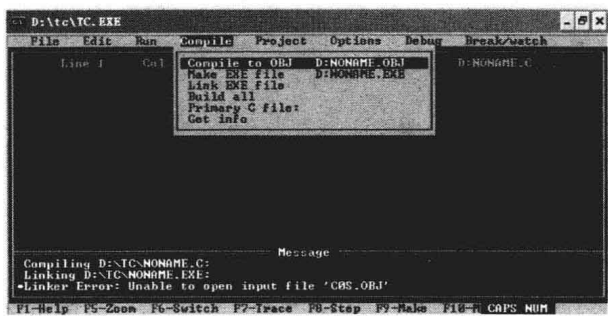


图 1-4 “Compile”菜单

下面简要说明各项的功能：

(1) Compile to OBJ (编译生成目标码)

本命令将一个.C 源文件编译成.OBJ 文件，同时显示生成的文件名。.OBJ 文件名由源.C 文件名产生；或在没有指定文件名时，由上次装入编辑器的文件名产生。Turbo C 在编译时弹出一个窗口，用于显示编译结果。在编译/组装完后，按任意键将清除编译窗口。此时若发现有错误，则转到消息窗口的第一个错误处（高亮显示）。

本命令的组合键为【Alt+F9】。

(2) Make EXE file (生成可执行文件)

本命令可生成.EXE 文件，并显示所生成的.EXE 文件名。.EXE 文件名是依次由下列文件名产生的：

“Project/Project Name”说明的文件名；或“Project C File”说明的文件名；或上次装入窗口的文件名。

本命令的热键为【F9】。

(3) Link EXE file (连接执行文件)

将当前文件与库文件（既可以是默认的，也可以是定义在当前项目文件中的）连接在一起，生成.EXE 文件。

(4) Build All (建立所有文件)

重建项目中的所有文件。本命令类似“Compile→Make EXE file”命令，只是它是无条件执行的，而“Compile→Make EXE file”命令只重建那些非过时的文件。本命令首先将所有的 Project 文件中的日期与时间置为 0，然后再组装。这样，若用户因按【Ctrl+Break】组合键中断了“Build All”命令，只要用“Compile→Make EXE file”即可恢复命令。

(5) Primary C file: (主 C 文件)

当编译多个.H 头文件和单个.C 文件时，Primary C file: 命令是很有用的（但并非必要的）。若在编译过程中发现错误，包含错误的文件（.C 或.H）将被自动装入编辑器，可对其修改。但必须注意，.H 文件只有在已将“Options→Environment→Message tracking”命令设置改为 All File 状态时才能自动装入，而原默认设置不会自动加载.H 文件。即使.C 文件不在编辑器，但只要按【Alt+F9】组合键，C 主文件即被重新编译。

(6) Get info (获得信息)

选择“Compile→Get info”命令开辟一个窗口，给出如下信息：

- 源文件。
- 与当前文件相联系的目标文件名。
- 当前源文件名。
- 当前源文件字节数。
- 程序退出码。
- 可用空间。

5. “项目”菜单 (Project)

当选中“Project”菜单后，在“Project”下方将出现一个子菜单，如图 1-5 所示。

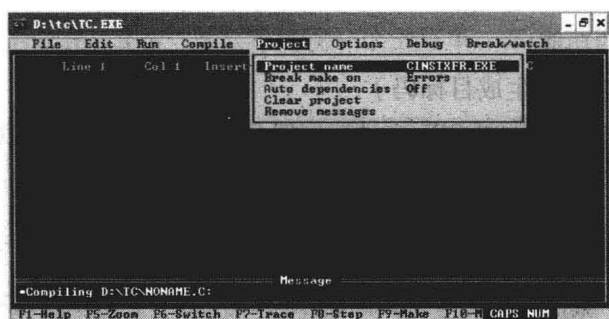


图 1-5 “Project”菜单

下面简要说明各项功能：

(1) Project name

选择一个包含将要编译连接的文件名的 Project 文件，项目名也将是以后要建立的可选择.EXE 或.MAP 文件。典型的项目文件扩展名为.PRJ。

(2) Break make on

提供用户说明中止 make 的默认条件，如警告 (Warnings)、错误 (Errors)、致命错误 (Fatal Error) 等。

(3) Auto dependencies (自动依赖)

这是一个开关。当置为 On 状态时，项目组装 (Project-Make) 自动检查每个项目表中在磁盘上相应.C 文件的那些.OBJ 文件的源文件日期/时间信息与.OBJ 文件的依赖关系。所谓自动依赖关系检查是指：项目组装打开.OBJ 文件，寻找包含在源代码中的那些文件的有关信息。此时，将每个组成.OBJ 文件的日期/时间信息与.OBJ 中的进行比较，若不同，则重新编译.C 源文件。若“Auto dependencies”开关置为 off 状态，则不进行这种检查。

(4) Clear project (清除 Project 文件)

执行该命令可以清除项目文件，重置消息窗口 (Message Window)。

(5) Remove message (删除信息)

执行该命令可以将错误信息从消息窗口中清除掉。

6. “设置”菜单 (Options)

当选中“Options”菜单后，在“Options”下方将出现一个子菜单，如图 1-6 所示。

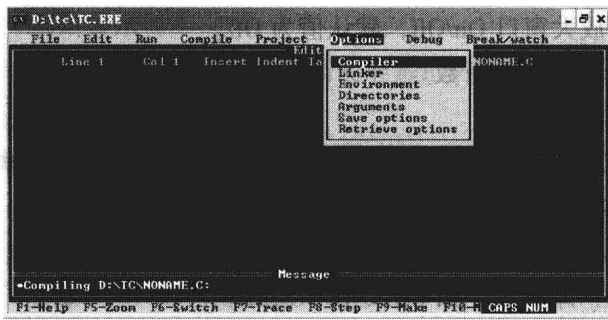


图 1-6 “Options” 菜单

下面简要说明各项的功能：

(1) Compiler (编译器)

本命令将产生一个子菜单，为用户提供说明硬件配置、存储模式、调试技术、代码优化、诊断消息控制以及宏定义等。各菜单条目如下：

- Model: 选择存储模型。
- Define: 打开一个宏定义框。
- Code generation: 代码生成。
- Optimization: 优化用户代码。
- Source: 处理源代码。
- Errors: 处理和响应诊断信息。
- Names: 改变代码、数据等。

其中每一个条目又对应一个子菜单，供用户选择各种功能。详细信息请参看 Turbo C 的用户手册。

(2) Linker (连接器)

本命令将产生有关连接器的设置。它包括以下内容：

- Map file: 选择映射文件的类型 On/Off，默认值为 Off。
- Initialize segments: 段初始化 On/Off，默认值为 Off。
- Default libraries: 默认库 On/Off，默认值为 On。
- Graphics library: 图形库 On/Off，默认值为 On。
- Warn duplicate symbols: 警告重复字符 On/Off，默认值为 On。
- Stack warning: 堆栈警告 On/Off，默认值为 On。
- Case-sensitive link: 大小写区别连接 On/Off，默认值为 On。

(3) Environment (环境设置)

本命令将产生编译环境的设置。它包括以下内容：

- Message tracking: 消息跟踪 “Current File/All Files/Off”，默认值为 Current File。
- Keeping message: 保存消息 Yes/No，默认值为 No。
- Config auto save: 配置自动保存 On/Off，默认值为 On。
- Edit auto save: 编辑自动保存 On/Off，默认值为 Off。
- Backup files: 备份文件 On/Off，默认值为 On。
- Tab size: 制表键大小，默认值为 8。

- Zoomed window: 放大窗口 On/Off, 默认值为 Off。
- Screen size: 选择屏幕显示行数。

(4) Directories (目录)

本命令告诉 Turbo C 到哪里去寻找编译连接所需的文件, 生成的可执行文件放到何处, 在哪里查找配置文件。具体内容如下:

- Include directories: 包含目录, 如 C:\TURBOC\INCLUDE。
- Library directories: 库目录, 如 C:\TURBOC\LIB。
- Output directory: 输出目录。
- Turbo C directory: Turbo C 目录。
- Pick file name: pick 文件名。
- Current pick file: 当前 pick 文件。

(5) Arguments (参数)

本设置允许用户给出运行程序命令行。

(6) Save options (保存任意项)

将选择的编辑器、连接器环境、调试和 Project 选项保存到一个配置文件中(默认文件名为 TCCONFIG.TC)。启动时, Turbo C 再到 TURBOC 目录中去寻找同样的文件。

(7) Retrieve options (恢复选项)

加载以前用“Option→Save options”命令保存的配置文件。

7. “调试”菜单(Debug)

当选中“Debug”菜单后, 在“Debug”下方将出现一个子菜单, 如图 1-7 所示。

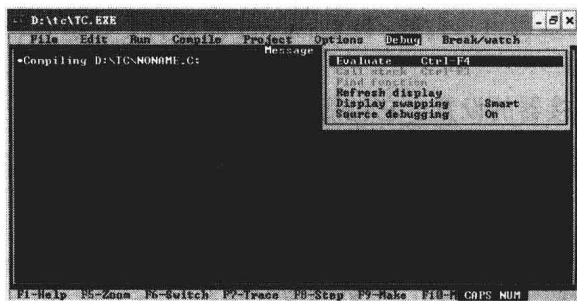


图 1-7 “Debug”菜单

下面简要说明各项功能:

(1) Evaluate (计算)

计算变量或表达式值, 并显示其结果。

(2) Call stack (调用栈)

本命令显示一个调用栈的弹出窗口。调用栈显示程序运行到正在运行时调用的函数序列。其中主函数 main()在栈底, 正在运行的函数在栈顶。调用函数的每一项显示了函数名以及传递给它的参数值。

(3) Find function (查找函数定义)

显示编辑窗口每一函数的定义, 只有在调试阶段才能使用本命令。

(4) Refresh display (刷新显示器)

万一编辑屏幕被重写, 使用本命令可以恢复当前屏幕的内容。

(5) Display swapping (显示转换)

本命令提供 3 种选择: On (默认值)、Always 和 None。

(6) Source debugging (源代码调试)

本命令提供 3 种选择 On (默认值)、Standalone 和 None。

8. “断点和监视”菜单 (Break/watch)

当选择 “Break/watch” 菜单后, 在 “Break/watch” 下方将出现一个子菜单, 如图 1-8 所示。使用本命令可以控制断点和监视表达式。

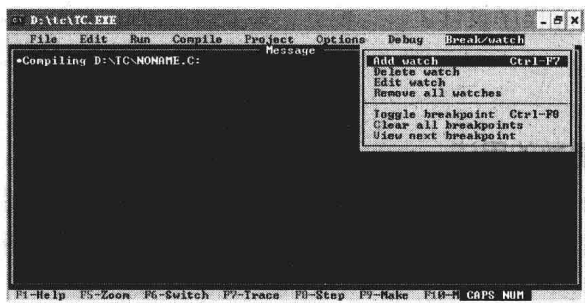


图 1-8 “Break/watch” 菜单

下面简要说明各项的功能:

(1) Add watches (增加监视表达式)

向监视窗口插入一个监视表达式。

(2) Delete watch (删除监视表达式)

从监视窗口中删除当前监视表达式。

(3) Edit watch (编辑监视表达式)

选择本命令后, 调试器弹出一个含有当前监视表达式复制的窗口。

(4) Remove all watches (删除所有监视表达式)

将所有监视表达式从监视窗口中删除。

(5) Toggle breakpoint (打开或关闭断点)

设置或去除光标所在断点。

(6) Clear all breakpoints (清除所有断点)

从程序中删除所有断点。

(7) View next breakpoint (显示下一个断点)

将光标移到程序中的下一个断点。

四、思考题

1. 选择一种较熟悉的 C 语言环境, 编辑运行能实现 $1+2+3+\cdots+100$ 的程序。
2. 了解 Turbo C 2.0 集成环境的功能。
3. 熟悉 Visual C++ 6.0 集成环境的功能。

上机 2 顺序结构程序设计

一、上机目的

- 掌握 printf()、scanf()、putchar()、getchar()等基本输入/输出函数的格式及主要用法。
- 熟练掌握顺序结构的程序设计。

二、上机内容

1. 基本输入/输出函数的用法

编辑运行下面的程序，并根据执行结果分析程序中各个语句的作用。

```
#include <stdio.h>
main()
{
    int a,b;
    float d,e;
    char c1,c2;
    double f,g;
    long m,n;
    unsigned int p,q;
    a=61;b=62;
    c1='a';c2='b';
    d=5.67;e=-6.78;
    f=1234.56789;g=0.123456789;
    m=50000;n=-60000;
    p=32768;q=40000;
    printf("a=%d,b=%d\nc1=%c,c2=%c\n",a,b,c1,c2);
    printf("d=%6.2f,e=%6.2f\n",d,e);
    printf("f=%15.6f,g=%15.10f\n",f,g);
    printf("m=%ld,n=%ld\np=%u,q=%u\n",m,n,p,q);
}
```

2. 顺序结构程序设计

① 已知圆柱体横截面半径 r ，圆柱高 h 。编写程序，计算圆周长 l 、圆面积 s 和圆柱体体积 v ，并输出计算结果。

② 编写一个程序，根据本金 a 、存款年数 n 和年利率 p 计算到期利息。计算公式如下：

$$\text{到期利息} = a \times (1 + p)^n - a$$

a^b 的计算公式为 $\exp(b \times \ln(a))$ 。

三、上机指导

1. 基本输入/输出函数的用法

(1) 运行所给出的源程序，对照结果分析各语句的作用。

(2) 将程序中的第2、第3个 printf 语句修改为如下形式，然后运行程序，查看结果。

```
printf("d=%-6.2f,e=%-6.2f\n",d,e);
printf("f=%-15.6f,g=%-15.10f\n",f,g);
```

(3) 将上述两个 printf 语句进一步修改为如下形式，然后运行程序，查看结果。

```
printf("d=%-6.2f\te=%-6.2f\n",d,e);
printf("f=%-15.6f\tg=%-15.10f\n",f,g);
```

(4) 将程序的第10~15行修改为如下语句：

```
a=61;b=62;
c1='a';c2='b';
f=1234.56789;g=0.123456789;
d=f;e=g;
p=a;m=50000;q=b;n=-60000;
```

运行程序，并分析结果。

(5) 修改(1)中的程序，不使用赋值语句，而用下面的 scanf 语句为 a、b、c1、c2、d、e 输入数据。

```
scanf("%d%d%c%c%f%f",&a,&b,&c1,&c2,&d,&e);
```

① 请按照程序原来的数据，选用正确的数据输入格式，为上述变量提供数据。

② 请思考使用如下数据输入格式，为什么得不到正确的结果。

输入数据：61, 62, a, b, 5.67, -6.78

(6) 进一步修改(5)中使用的程序，使 f 和 g 的值用 scanf() 函数输入。

(7) 进一步修改上面的程序，使其他所有变量的值都改用 scanf() 函数输入。

① 参考程序如下：

```
#include <stdio.h>
void main()
{
    int a,b;
    float d,e;
    char c1,c2;
    double f,g;
    long m,n;
    unsigned int p,q;
    printf("Input (a,b,c1,c2,d):");
    scanf("%d%d%c%c%f%f",&a,&b,&c1,&c2,&d,&e);
    printf("Input (f,g):");
    scanf("%lf%lf",&f,&g);
    printf("Input (m,n,p,q):");
    scanf("%ld%ld%u%u",&m,&n,&p,&q);
    printf("a=%d,b=%d\nc1=%c,c2=%c\n",a,b,c1,c2);
    printf("d=%-6.2f,e=%-6.2f\n",d,e);
    printf("f=%-15.6f,g=%-15.10f\n",f,g);
    printf("m=%ld,n=%ld\np=%u,q=%u\n",m,n,p,q);
}
```