

C Yuyan Shangji
Shiyan Zhidao yu Xitiji

C 语言上机 实验指导与习题集

主 编 王 晓 李 琦

副主编 宋 健 刘柏洋 成新文



西南交通大学出版社

C 语言上机实验指导与习题集

主 编 王 晓 李 琦

副主编 宋 健 刘柏洋 成新文

西南交通大学出版社
· 成 都 ·

内 容 简 介

本书是四川理工学院精品课程“C 语言程序设计”配套实验教材。编写组根据多年教学实践、上机实验、实训辅导经验编写而成,力求把 C 语言程序设计的知识点融入到具体的上机实验、实训中。

本书紧密配合理论教学,全书的实验、实训和习题内容丰富,不仅具有一定的启发性、综合性,还有较强的实用价值,因此本书可作为高等院校学生学习 C 语言的参考资料和 C 语言上机实验指导的教学参考书。

图书在版编目(CIP)数据

C 语言上机实验指导与习题集 / 王晓, 李琦主编. —
成都: 西南交通大学出版社, 2015.1
ISBN 978-7-5643-3685-1

I. ①C… II. ①王… ②李… III. ①C 语言—程序设计—高等学校—习题集 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2015) 第 005817 号

C 语言上机实验指导与习题集

主编 王 晓 李 琦

责任编辑 张 波
助理编辑 赵雄亮
特邀编辑 黄庆斌
封面设计 米迦设计工作室

出版发行 西南交通大学出版社
(四川省成都市金牛区交大路 146 号)

发行部电话 028-87600564 028-87600533

邮政编码 610031

网 址 <http://www.xnjdcbs.com>

印 刷 成都中铁二局永经堂印务有限责任公司

成品尺寸 185 mm × 260 mm

印 张 19

字 数 485 千

版 次 2015 年 1 月第 1 版

印 次 2015 年 1 月第 1 次

书 号 ISBN 978-7-5643-3685-1

定 价 30.00 元

课件咨询电话: 028-87600533

图书如有印装质量问题 本社负责退换

版权所有 盗版必究 举报电话: 028-87600562

前 言

在“C 语言程序设计”课程的学习中，上机实验、实训环节的教学、训练尤为重要。四川理工学院“C 语言程序设计”精品课程组在总结多年教学实践、上机实验、实训辅导经验的基础上编写了此书。

全书紧密结合“C 语言程序设计”课程的理论教学，精心设计的上机实验、实训内容，具有一定的启发性和综合性。另外，书中还配有丰富的练习题、程序设计题和等级考试模拟题，具有较强的实用性。因此本书对学生学习和掌握“C 语言程序设计”课程内容，提高编程能力以及顺利通过考试具有积极意义。

全书共由三部分组成：

第一部分：上机实验与指导。该部分先介绍 C 语言程序的上机步骤、程序运行过程、C 语言集成开发环境、编程规范。重点介绍了 VC++6.0 集成开发环境的使用，初学者可通过练习逐步熟悉，从而掌握在 VC++6.0 集成开发环境下编辑、修改、编译、链接、调试运行及排错全过程。然后安排了上机实验，内容的基本部分安排了 26 学时，综合部分安排了 10 学时，可视其具体情况选做。

第二部分：习题集与参考答案。该部分紧密配合“C 语言程序设计”课程各章节的教学内容，提供了大量练习题和参考答案，其目的是使初学者通过练习题进一步加深对所学知识的理解和掌握。

第三部分：计算机等级考试二级 C 语言上机试题。该部分以全国计算机等级考试二级 C 语言上机试题样题和四川省计算机等级考试二级 C 语言上机试题样题为例，介绍了二级 C 语言等级考试相关内容，力求使学生在掌握课程内容的同时，通过训练能顺利通过各类等级考试。

全书由王晓、李琦任主编，宋健、刘柏洋、成新文任副主编。由于作者水平有限，加之时间仓促，书中难免存在不当之处，敬请读者批评指正。

编 者

2015 年 1 月

目 录

第一部分 上机实验与指导

第 1 章 实验环境的构建	1
1.1 C 语言程序的上机步骤	1
1.2 VC++6.0 集成开发环境	3
1.3 C-Free 集成开发环境	10
1.4 其他 C 语言开发环境	16
第 2 章 编程规范及范例	19
2.1 排 版	19
2.2 注 释	22
2.3 标识符命名	23
2.4 可读性	24
2.5 函数、过程	26
2.6 规范练习	26
第 3 章 上机实验内容	29
3.1 实验一 C 程序运行环境和运行过程	29
3.2 实验二 结构化程序设计	31
3.3 实验三 数组与函数	43
3.4 实验四 指针、结构体、共用体与文件	58
3.5 综合实验 A1 C 语言程序的调试与编写 1	78
3.6 综合实验 A2 C 语言程序的调试与编写 2	80
3.7 综合实验 B1 小型文件管理系统的编写	85
3.8 综合实验 B2 小型学生成绩管理系统的编写	85
3.9 综合实验 B3 C 语言趣味程序	87

第二部分 习题集与参考答案

第 4 章 习题集	94
单元一 C 语言概述与数据类型	94
单元二 顺序结构程序设计	101
单元三 选择结构程序设计	107
单元四 循环结构程序设计	114
单元五 数组函数指针	126

目 录

单元六	编译预处理	176
单元七	结构体与共用体	178
单元八	位运算	188
单元九	文 件	190
单元十	程序设计题	197
第 5 章	习题集参考答案	227
单元一	C 语言概述与数据类型	227
单元二	顺序结构程序设计	228
单元三	选择结构程序设计	228
单元四	循环结构程序设计	229
单元五	数组函数指针	229
单元六	编译预处理	232
单元七	结构体与共用体	232
单元八	位运算	233
单元九	文 件	233
单元十	程序设计题	234

第三部分 计算机等级考试二级 C 语言上机试题

第 6 章	四川省计算机等级考试二级 C 语言上机试题举例	241
第 7 章	全国计算机等级考试二级 C 语言上机试题举例	256
第 8 章	C 程序设计算法总结举例	274

第一部分 上机实验与指导

第 1 章 实验环境的构建

1.1 C 语言程序的上机步骤

编写开发 C 语言程序，一般包括以下四步：

(1) 编辑。

可以用任何一种编辑软件将在纸上编写好的 C 语言程序输入计算机，并将 C 语言源程序文件“*.c”以纯文本文件形式保存在计算机的磁盘上（不能设置字体、字号等）。由于源程序是无法直接被计算机运行的，因为计算机的 CPU 只能执行二进制机器指令，因此，需要把源程序先翻译成机器指令，然后计算机的 CPU 才能运行翻译好的程序。

(2) 编译。

编译是用 C 语言编译程序对编辑好的源程序文件“*.c”进行编译处理，把每一条语句用若干条机器指令来实现，最后生成二进制目标代码文件“*.obj”。编译过程中，编译程序将对源程序逐句检查语法错误，当发现错误后，不仅会显示错误的位置（行号），还会告知错误类型信息。此时，读者需要再次回到编辑软件修改源程序的错误，然后进行编译，直至排除所有语法和语义错误。

(3) 连接。

程序编译后产生的目标文件是可重定位的程序模块，也不能直接运行，还需进行连接。连接是将编译生成的各个目标程序模块和系统或第三方提供的库函数“*.lib”连接在一起，生成可以脱离开发环境、直接在操作系统下运行的可执行文件“*.exe”。

(4) 运行程序。

如果经过测试，运行可执行文件达到预期设计目的，这个 C 语言程序的开发工作便完成了。如果运行出错，这说明程序处理的逻辑存在问题，需要再次回到编辑环境针对程序出现的逻辑错误进一步检查、修改源程序，重复编辑→编译→连接→运行的过程，直到取得预期结果为止。

C 语言程序的上机步骤（设 C 源程序文件名为：eg.c），如图 1.1 所示。

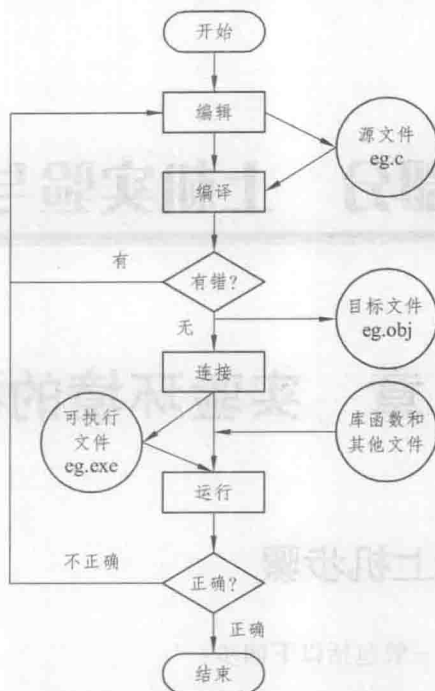


图 1.1 C 语言程序上机运行步骤

一般情况下，我们编写的 C 程序很难一次就能做到完全正确。因此，在上机过程中需要对程序进行调试，无论是出现编译错误、连接错误，还是运行结果不对（源程序中有语法错误或逻辑错误），都需要修改源程序，并对它重新编译、连接和运行，直至将程序调试正确为止。

编译错误通常是编程者违反了 C 语言的语法规则，如保留字输入错误、大括号不匹配、语句少分号等。对于编译连接错误，C 语言系统会提供出错信息，包括出错位置（行号）、出错提示信息。编程者可以根据这些信息，找出相应错误所在。有时系统提示的一大串错误信息，并不表示真的有这么多个错误，往往是因为前面的一两个错误带来的。所以，我们强烈建议初学者按从前往后的顺序纠正错误。当纠正了一个或者多个错误后，可再编译连接一次，然后根据最新的出错信息继续纠正，直到编译成功。

连接错误一般由未定义或未指明要连接的函数，或者函数调用不匹配等因素引起，如对系统函数的调用必须要通过“include”说明等。

运行结果出错往往是程序处理的逻辑存在问题，错误的原因有两部分：一部分是程序书写错误带来的，例如应该使用变量 x 的地方写成了变量 y，虽然没有语法错误，但意思完全错了；另一部分可能是程序的算法不正确，解题思路不对。还有一些程序有时计算结果正确，有时不正确，这往往是编程时对各种情况考虑不周所致，也就是算法对各种数据情况考虑不全面。解决办法最好多选几组典型的输入数据进行测试，除了普通的数据外，还应包含一些边界数据和 incorrect 的数据。比如确定正常的输入数据范围后，分别以最小值、最大值、比最小值小的值和比最大值大的值，多方面运行检查自己的程序。解决运行错误的首要步骤是错误定位，即找到出错的位置，才能予以纠正。通常先设法确定错误的大致位置，然后通过 C 语言提供的调试工具找出真正的错误。

为了确定错误的大致位置，可以先把程序分成几大块，并在每一块的结束位置，手工计

算一个或几个阶段性结果,然后用调试方式运行程序,到每一块结束时,检查程序运行的实际结果与手工计算是否一致,通过这些阶段性结果来确定各块是否正确。对于出错的程序块,可逐条仔细检查各语句,找出错误所在。如果出错块程序较长,难以一下子找出错误,可以进一步把该块细分成更小的块,按照上述步骤进一步检查。确定大致出错位置后,如果无法直接看出错误,可以通过单步运行相关位置的几条语句逐条检查,一定能找出错误的语句。

初学者在学习程序设计过程中,要特别注重培养自己调试程序的能力,它不可能靠几句话讲清楚,要靠自己在上机过程中不断摸索总结,它可以说是一种经验积累。

下面分别以 VC++6.0、C-Free 5.0 为上机平台,对 C 程序编译、连接和调试作简单介绍。建议一开始学习上机时,把注意力放在程序的编译、连接和运行,以能运行为目标,而把调试部分放到学习了循环控制结构后再看,只有具有一定的程序语句量,调试才有作用。

1.2 VC++6.0 集成开发环境

C++语言是在 C 语言的基础上发展而来的,它增加了面向对象的编程思想,已成为当今最流行的一种程序设计语言。Visual C++是微软公司开发的,面向 Windows 编程的 C++语言工具。它不仅支持 C++语言的编程,也兼容 C 语言的编程。由于 VC++被广泛地用于各种编程,其使用面很广。这里仅简要地介绍如何在 VC++下运行 C 语言程序。

1.2.1 启动 VC++

VC++是一个庞大的语言集成工具,经安装后将占用几百兆磁盘空间。从“开始”/“程序”/“Microsoft Visual Studio 6.0”/“Microsoft Visual C++6.0”,可启动 VC++ 6.0。VC++的窗口如图 1.2 所示的窗口。

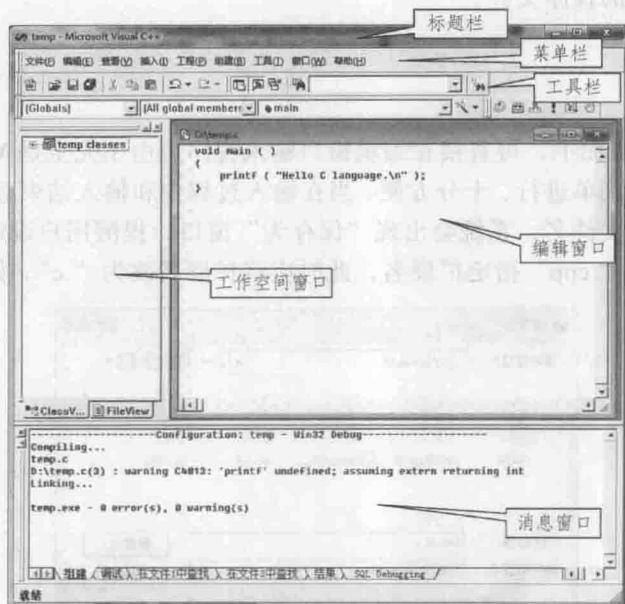


图 1.2 VC++6.0 窗口

1.2.2 新建/打开 C 程序文件

在 VC++6.0 中新建 C 语言程序, 可在启动 VC++ 时进行, 也可在启动后进行。如下所示为启动 VC++6.0 新建 C 语言程序的过程。选择“文件”/“新建”, 选择如图 1.3 所示“新建”窗口中的“文件”标签(默认的是“工程”标签), 选中“C++ Source File”选项, 设置 C 程序文件名和存放目录, 点击“确定”。之后就在编辑窗口中输入程序源码, 将 C 程序文件命名为 temp.c, 如图 1.3 所示, 存放位置为 D 盘根目录。

注意: C 程序文件名的扩展名建议取为“.c”。

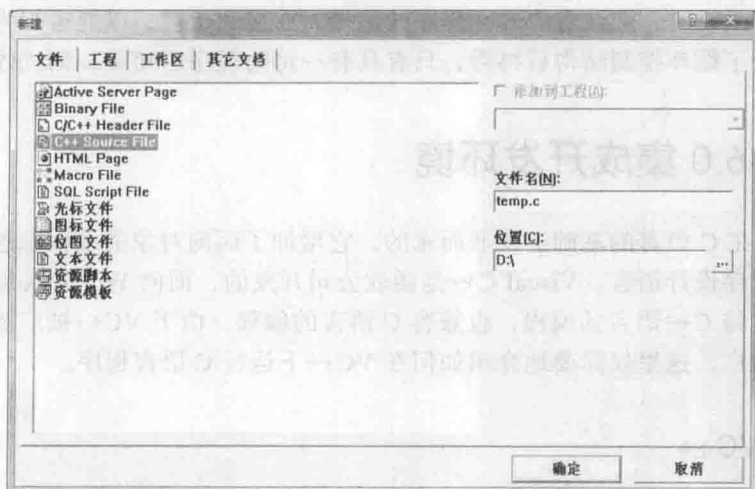


图 1.3 新建文件

如果程序已经存在, 可选择“文件”菜单的“打开”菜单项, 并在查找范围中找到正确的文件夹, 调入指定的程序文件。

1.2.3 程序保存

在打开的 VC++ 界面上, 可直接在编辑窗口输入程序。由于完全是 Windows 界面, 输入及修改可借助鼠标和菜单进行, 十分方便。当在输入过程中和输入结束后, 应注意保存文件。如果之前未设定程序文件名, 系统会出现“保存为”窗口, 提醒用户设定程序文件名。系统默认将按 C++ 扩展名“.cpp”指定扩展名, 此时应将扩展名改为“.c”。如图 1.4 所示。

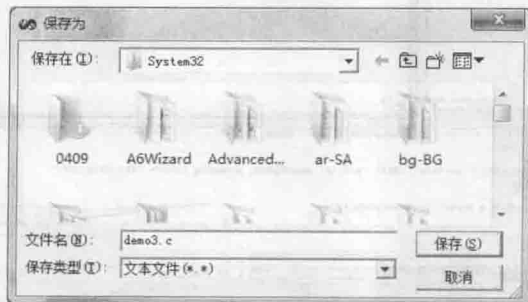


图 1.4 指定保存文件名

1.2.4 执行程序

首先要生成可执行文件。使用 VC++ “编译” 菜单中的 “组建” 菜单项, 如图 1.5 所示, 也可使用快捷键【F7】。在组建 (编译、链接) 过程中, VC++ 将保存该新输入的程序, 并生成一个同名的工作空间。



图 1.5 组建菜单

如果程序没有错误, 将在如图 1.6 所示消息窗口中显示内容:

0 error(s) 0 warning(s)



图 1.6 编译链接正确

消息窗口中有时出现几个警告性信息(warning), 这不影响程序执行。假如有致命性错误(error), 如图 1.7 所示, 通过双击某行出错信息, 程序窗口中会指示对应出错位置, 此时根据消息窗口的提示分别予以纠正。

组建正确后, 可选择 “编译” 菜单中的 “执行” 菜单项 (或快捷键【Ctrl】+【F5】、或工具栏上的 “红色叹号” 按钮) 执行程序。

当运行 C 程序后, 如果需要输入数据, VC++ 将自动弹出数据输入输出窗口。输入正确后, 系统将显示运行结果, 如图 1.8 所示。运行完毕后, 可按任意键关闭该窗口。



图 1.7 编译连接出错

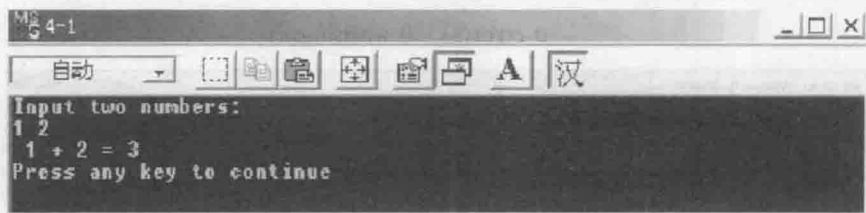


图 1.8 数据输入输出窗口

1.2.5 命令行参数处理

VC++是一个基于窗口操作的 C++ 系统，没有提供命令行参数功能。如果需要直接运行组建生成的可执行文件（扩展名为 .exe），则需要在 Windows 的“命令提示符”窗口里以命令方式实现。具体步骤如下：

- (1) 正确编译链接，生成可执行程序。
- (2) 通过“我的电脑”或“资源管理器”找到所运行的 C 源程序（设为 a.c）。
- (3) 进入 debug 文件夹（该文件夹和 a.c 在同一位置，它包含 a.c 程序的可执行文件 a.exe）。
- (4) 执行“开始”菜单的“运行”中，填入 command（或 cmd），然后“确定”。
- (5) 在打开的“命令提示符”窗口中输入：a.exe 参数 1 参数 2.....，带参数运行程序。

说明：该命令行中，可执行文件的扩展名.exe 可省略，命令与参数、参数与参数之间用空格分隔。

1.2.6 关闭程序工作空间

当一个程序编译链接后，VC++系统自动产生相应的工作空间，以完成程序的运行和调试。

若想执行第二个程序时,必须关闭前一个程序的工作空间,然后通过新的组建(编译连接),产生第二个程序的工作空间。否则将一直运行前一个程序。

“文件”菜单提供关闭程序工作空间功能,如图 1.9 所示,执行“关闭工作空间”菜单功能,在弹出的对话框中选择“是”,将清空工作空间窗口和消息窗口,同时关闭源程序编辑窗口。这时即可开始新程序的编写。



图 1.9 关闭程序工作空间

1.2.7 程序调试

如果只是对 C 程序进行组建(编译、连接)和运行,在简单的程序中遇到的错误还能比较容易地进行查找改错。如果遇到较为复杂和较长的程序,这时就需要对程序进行调试。VC++ 是一个完全基于 Windows 的系统,它的调试过程比较容易进行。

说明:在进行调试前,一般需要对程序进行组建。这就要求首先需要修改程序中的语法错误。也就是说,程序调试主要是用于修改程序运行过程中的错误,如算法错误。

1. 程序执行到中途暂停以便观察阶段性结果

(1) 方法一:使程序执行到光标所在的那一行暂停。

① 在需暂停的行上单击鼠标,定位光标。

② 如图 1.10 所示,分别点击菜单“编译”/“开始调试”/“Run to Cursor”,或按【Ctrl】+【F10】,程序执行到光标所在行将会暂停。如果把光标移动到后面的某个位置,再按【Ctrl】+【F10】,程序将从当前的暂停点继续执行到新的光标位置,第二次暂停。

(2) 方法二:在需暂停的行上设置断点。

① 在需设置断点的行上单击鼠标,定位光标。

② 按“编译微型条”中最右面的按钮,如图 1.11 所示,或按【F9】。被设置了断点的行前面会有一个红色圆点标志。

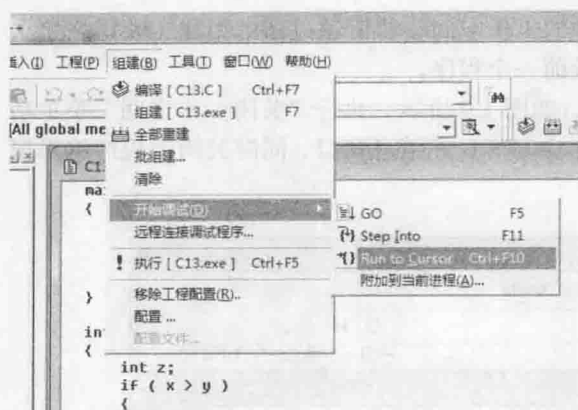


图 1.10 执行到光标所在行暂停

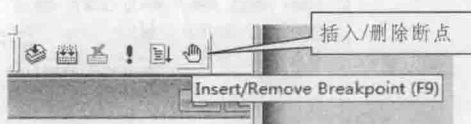


图 1.11 设置断点按钮

不管是通过光标位置还是断点设置，其所在的程序行必须是程序执行的必经之路，亦即不应该是分支结构中的语句，因为该语句在程序执行中受到条件判断的限制，有可能因条件的不满足而不被执行。这时程序将一直执行到结束或下一个断点为止。

2. 设置需观察的结果变量

按照上面的操作，使程序执行到指定位置时暂停，目的是为了查看有关的中间结果。在如图 1.12 所示的左下角窗口中，系统自动显示了有关变量的值，其中 value1 和 value2 的值分别是 123、456，而变量 i、sum 的值是不正确的，因它们还未被赋值。图中左侧的箭头表示当前程序暂停的位置。如还想增加观察变量，可在图中右下角的“名称”框中填入相应变量名。

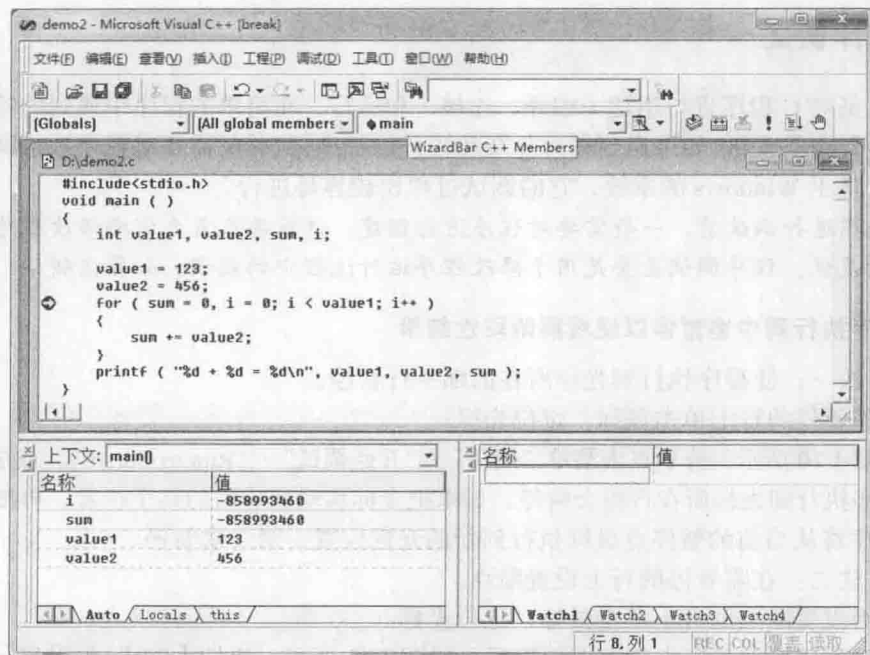


图 1.12 观察结果变量

3. 单步执行

当程序执行到某个位置时发现结果已经不正确了,说明在此之前肯定有错误存在。如果能确定一小段程序可能有错,先按上面步骤暂停在该小段程序的头一行,再输入若干个查看变量,然后单步执行,即一次执行一行语句,逐行检查下来,看看到底是哪一行造成结果出现错误的,从而能确定错误的语句并予以纠正。

单步执行按“调试条”中“Step Over”按钮或【F10】,如图1.13所示。如果遇到自定义函数调用,想进入函数进行单步执行,可按“Step Into”按钮或【F11】。当想结束函数的单步执行,可按“Step Out”按钮或【Shift】+【F11】。对不是函数调用的语句来说,【F11】与【F10】作用相同,但一般对系统函数不要使用【F11】。

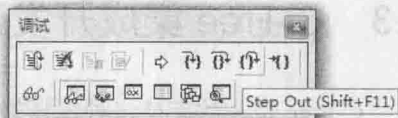


图 1.13 单步调试

4. 断点的使用

使用断点也可以使程序暂停。但一旦设置了断点,不管用户是否还需要调试程序,每次执行程序都会在断点上暂停。因此调试结束后应取消所定义的断点。方法是先把光标定位在断点所在行,再按“编译微型条”中最右面的按钮或【F9】,该操作是一个开关,按一次是设置,按两次是取消设置。如果有多个断点想全部取消,可执行“编辑”菜单中的“断点”菜单项,屏幕上会显示“Breakpoints”窗口,如图1.14所示,窗口下方列出了所有断点,按“Remove All”按钮,将取消所有断点。

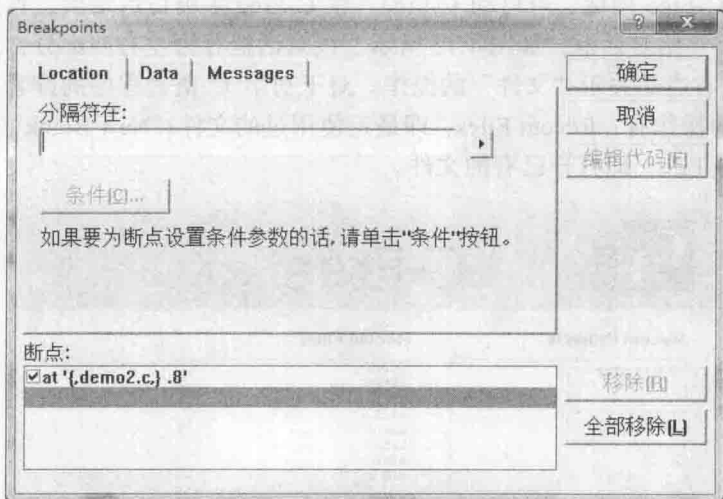


图 1.14 取消所有断点

断点通常用于调试较长的程序,可以避免使用“Run to Cursor”(运行程序到光标处暂停)或【Ctrl】+【F10】功能时,经常要把光标定位到不同的地方。而对于长度为上百行的程序,要寻找某位置并不太方便。

如果一个程序设置了多个断点,按一次执行键【Ctrl】+【F5】会暂停在第一个断点,再按一次【Ctrl】+【F5】会继续执行到第二个断点暂停,依次执行下去。

5. 停止调试

使用“Debug”菜单的“Stop Debugging”菜单项，或【Shift】+【F5】键可以结束调试，从而回到正常的运行状态。

到此为止，只对 VC++ 中的主要功能作了介绍，对于其他的操作读者可以自己试验，或参考有关 Visual C++ 手册。

1.3 C-Free 集成开发环境

C-Free 是一款支持多种编译器的专业化 C/C++ 集成开发环境 (IDE)。利用 C-Free，使用者可以轻松地编辑、编译、连接、运行、调试 C/C++ 程序。C-Free 提供了对目前业界主流 C/C++ 编译器的支持，在 C-Free 中可轻松切换编译器，可定制快捷键、外部工具以及外部帮助文档，使使用者在编写代码时得心应手。最新的 C-Free 5.0 还具有增强的 C/C++ 语法加亮器（可加亮函数名，类型名，常量名等），增强的智能输入功能，代码完成功能和函数参数提示功能，在调试时显示控制台窗口等功能。下面以 C-Free 5.0 为例，介绍如何在 C-Free 中编辑、调试和运行 C 语言程序。

1.3.1 启动 C-Free

双击桌面的 C-Free 图标，可启动 C-Free。在 C-Free 主窗口启动后，首先会出现一个名字叫“Star Here”开始对话框，如图 1.15 所示。该对话框分为左右两部分，左边主要是关于“工程”的操作，右边是关于“文件”的操作。对于初学 C 语言程序的读者来说，主要使用右边部分。主要的操作有：Recent Files，即最近使用过的文件；New Blank File，即新的空文件；Open Existing File，即打开已有的文件。

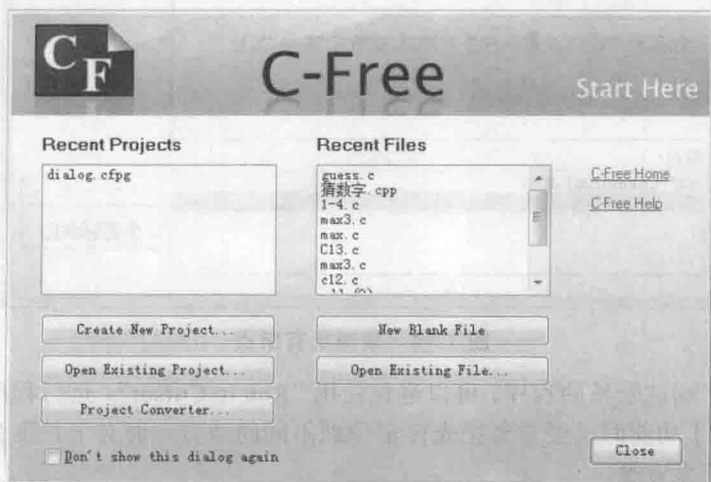


图 1.15 C-Free 开始对话框

选择其中一项后，将进入 C-Free 的主窗口。C-Free 的主窗口如图 1.16 所示。主要包括如下部分：主菜单栏、工具栏、标签栏、符号工具条、符号窗口、代码编辑器、文件列表窗

口、消息窗口和状态栏。其中，标签栏用于显示当前的源程序文件名，代码编辑器用于编写源程序代码，菜单栏菜单项或工具栏的工具按钮用于进行程序构建和运行，消息窗口中将出现相应的信息。

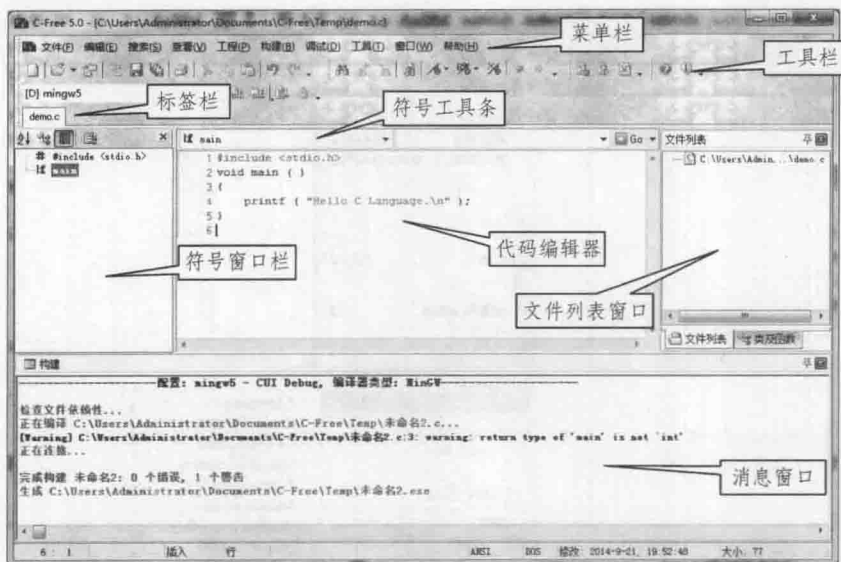


图 1.16 C-Free 的主窗口

1.3.2 新建/打开和保存 C 程序文件

除了可以在 C-Free 的开始对话框中新建和打开 C 程序文件外，还可以在 C-Free 主窗口的“文件”菜单中选择“新建”/“打开”命令来新建和打开 C 程序文件。

“新建”命令会让 C-Free 新建一个空的文档。新建的文件是一个未命名的文件，默认名字是“未命名 n.cpp”，这里 n 是一个整数。这个文件默认的存在位置是 C-Free 安装目录下的 temp 文件夹，默认的文件扩展名是.cpp。建议新建文件后，立即修改文件名和存放位置，文件扩展名修改为.c。

在 C-Free 中，默认情况下，新建文件扩展名是.cpp。在“工具”菜单中“环境选项”对话框中可以改变新建文件的扩展名。

在编写源程序文件时，一个好的习惯是经常性地保存文件。在 C-Free 中，可以使用“文件”菜单中的“保存”命令或工具栏中的“保存”按钮保存文件，保存命令的快捷键是 Ctrl+S。如果保存的是新建文档，C-Free 会显示“另存为”对话框，这时可以命名文档。如果想更改现有的名字和目录，可选择“另存为”命令。

1.3.3 编辑、构建和运行程序

源程序代码的输入，是在 C-Free 的代码编辑器中进行的。值得一提的是，C-Free 5.0 具有高度智能化的显示和输入功能。比如以不同颜色和增亮的方式显示函数名、类型名、常量名等，能预测输入的代码，自动缩进和能一次输入成对符号（如括号、双引号等）。