

第 1 章

程序设计基础

本章介绍程序设计语言的分类及特点、C 语言的产生、算法的概念及特性、算法的描述方法，以及软件的编制步骤等。在本章实验中，将了解 Visual C++6.0 的编程环境，掌握 C 程序的编译过程，通过简单实例，用流程图设计算法，根据算法描述编制出 C 源程序，进一步编译、链接、运行，掌握 C 语言程序的基本结构及编译运行流程。

实验 1 Visual C++ 6.0 运行环境

一、实验学时

2 学时

二、实验目的和要求

- (1) 熟悉 Visual C++6.0 的运行环境。
- (2) 学习 Visual C++6.0 程序的编译过程。
- (3) 掌握用程序流程图描述算法。

三、实验内容与操作步骤

Visual C++6.0 是 Microsoft 公司推出的基于 Windows 环境的 C/C++ 集成开发工具，通常可以被单独安装。它功能强大，不仅可以用来开发 Windows 应用程序，还能直接编辑、运行 C++ 程序。其对下兼容，使得在 DOS 环境下(例如 Turbo C)开发的普通 C 程序也能在 Visual C++6.0 平台上方便地实现编辑、编译、链接与运行，因此 Visual C++6.0 作为一种 C 语言编译软件或者开发工具被广泛使用。

1. 【实验内容 1】

认识 Visual C++6.0 开发环境，了解 C 源程序从创建到运行的过程。

实验内容 1 的操作步骤如下。

(1) 在 Windows 桌面上, 单击“开始”菜单“程序”“Microsoft Visual C++6.0”程序项或直接双击桌面上 Visual C++6.0 的图标(如图 1-1 所示), 即可启动 Visual C++6.0 开发环境。

(2) Visual C++6.0 启动后, 主窗口界面如图 1-2 所示, 单击“文件”菜单“新建...”, 打开新建工程对话框, 创建 Visual C++6.0 工程。



图 1-1 Visual C++6.0 图标

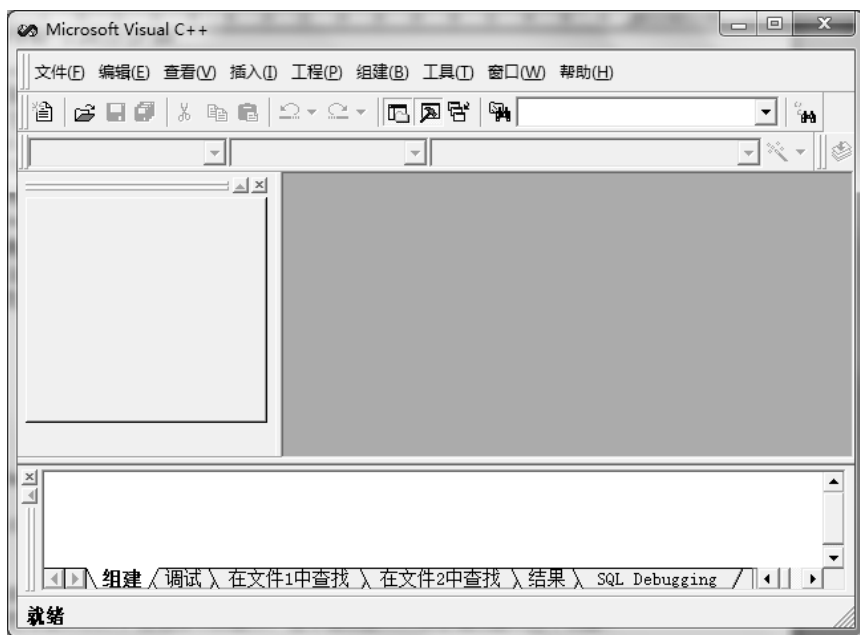


图 1-2 Visual C++6.0 主窗口界面

在 Visual C++6.0 中有“独立文件模式”和“项目管理模式”两种方式来编辑、编译、运行 C 语言源程序。当需要编写的源程序文件比较简单时可采用独立文件模式直接创建、录入、编辑、链接、运行；当一个程序由多个源程序文件组成时, 可使用项目管理模式将全部源程序文件合在一起, 构成一个整体程序, 该程序在 Visual C++6.0 中称为项目。两种模式在启动开发环境、编译、连接、运行等方面是相同的, 主要区别在于运行环境的建立、源程序的录入与编辑。下面介绍的是在“项目管理模式”下运行 C 语言, “独立文件模式”运行 C 语言比较简单, 此处不再赘述。

(3) 在“新建”对话框中, 选择“工程”选项卡, 如图 1-3 所示, 从中选择“Win32 Console Application”(32 位控制台应用程序)项, 在屏幕右侧“位置”框中输入新建工程的存放路径(本例为 d:\student), 最后在“工程名称”文本框中输入新建工程名(本例为 ch01)。此时系统将在 d:\student 文件夹下自动生成一个名字为 ch01 的文件夹, 并在该文件夹中自动生成 ch01.dsp (工程文件/项目文件)、ch01.dsw (工作区文件)和 debug 文件夹(用于存储编译、链接过程中产生的文件), 单击“确定”按钮, 完成设置。



图 1-3 Visual C++6.0 新建对话框

(4) 显示如图 1-4 所示的对话框，要求选择工程文件的类别，有 4 个可选项目。

- “一个空工程 (An empty project.)” 表示系统仅仅生成一个空白的工程项目，连 main() 函数也不提供，一切文件都由用户自己输入；
- “一个简单的应用程序 (A simple application.)” 表示系统生成一个仅带 main() 函数的工程项目，但该 main() 函数仅有框架无函数体。
- “一个 Hello World! 程序 (A “Hello, World!” application.)” 功能与 “简单的程序” 基本相同，只是 main() 函数的函数体中有一句 printf("Hello World! ") 语句。
- “一个支持 MFC 的程序 (An application that supports MFC.)” 表示创建一个 MFC 类型的应用程序。

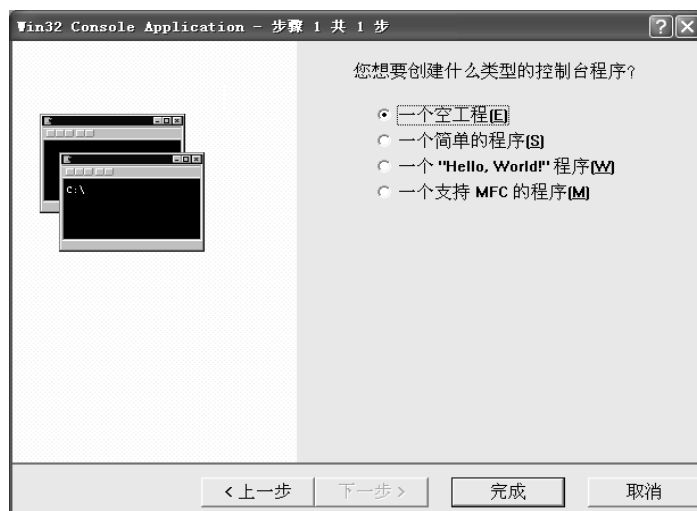


图 1-4 选择工程类别对话框

(5) 选择 “一个空工程[E]” 选项，单击 “完成” 按钮。显示即将新建的 Win32 控制台应用程序框架说明，如图 1-5 所示。



图 1-5 Win32 位控制台应用程序框架说明

(6) 在确认 Win32 控制台应用程序新建工程信息无误后, 单击“确定”按钮, 弹出 ch01 工程编辑窗口, 如图 1-6 所示, 至此“工程管理模式”的运行环境已经建立。

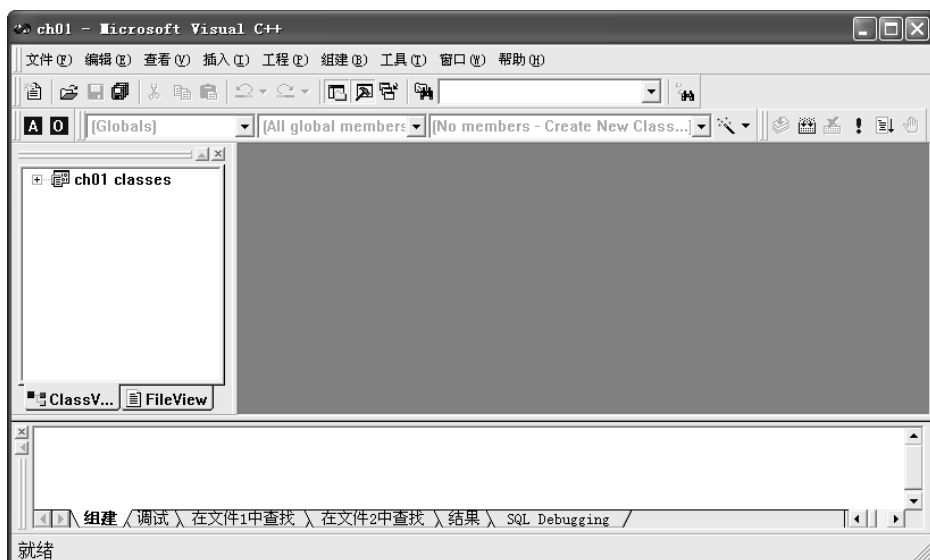


图 1-6 新创建的空工程窗口

一个新建的工程运行环境建立完毕后, 系统将会在项目存放路径生成一组相关的文件夹和文件, 打开资源管理器窗口, 单击“d:\student\ch01”文件夹, 其内容显示如图 1-7 所示。

在创建一个 Visual C++6.0 工程时, 系统会自动产生许多相关的文件, 这些文件不同的类型和作用简单介绍如下:

.cpp 文件: Visual C++6.0 源程序文件, 即程序代码, 可以单独用记事本打开编辑。

.dsw 文件: 工作区 (Workspace) 文件, 用它可以直接打开工程, 属于级别最高的 Visual C++6.0 文件。

.dsp 文件: 项目文件, 主要用来存放应用程序的有关信息。

.opt 文件：是工程关于环境的选项设置文件，当运行的机器环境发生了变化，该文件删除后也将自动重建。

debug 文件夹：在刚刚建立工程时里面还没有任何文件，只有当程序编译、链接、运行以后，程序的可执行文件等其他相关文件会放在其中。



图 1-7 新建工程文件夹中的文件

“项目管理模式”下，一个项目由多个源程序文件组成，应该分别对源程序进行创建、录入、编辑，上一步仅仅创建了一个空的工程，必须将 C 源程序文件和头文件添加到工程中去，才能运行工程。接下来我们将在上面创建的工程 ch01 中添加一个 C 源程序。

(7) Visual C++6.0 主窗口中打开“文件”菜单，选中“新建”命令，打开如图 1-8 所示的“新建”对话框。

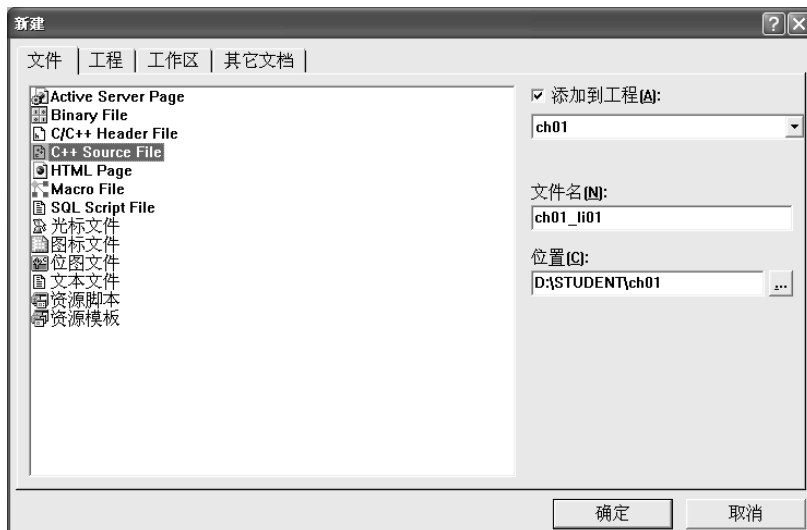


图 1-8 新建 C++源文件对话框

(8) 单击“文件”选项卡，在左侧列表中选择“C++ Source File”选项，选中“添加到工

程”复选框，再在右侧“文件名”文本框中填写要新建的源程序文件名，本例输入 ch01_li01，即在工程 ch01 中新建源程序文件 ch01_li01.cpp。这里没有输入文件的扩展名，系统会自动添加默认的扩展名“.cpp”，如果用户想创建扩展名为“.c”源程序文件，在文件名框中输入“ch01_li01.c”即可。工程下拉框和位置下拉框中已显示创建工程时的设置，这里不要更改，单击“确定”按钮，进入 Visual C++6.0 的集成开发环境主窗口，如图 1-9 所示。

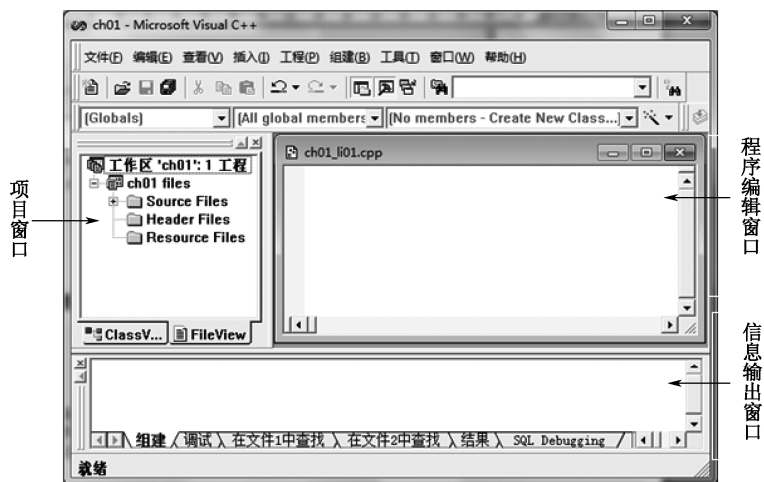


图 1-9 Visual C++6.0 集成环境主窗口

Visual C++6.0 的集成开发环境中，除包含一般应用程序窗口所共有的标题栏、菜单栏、工具栏、状态栏等部件以外，还包含项目窗口、程序编辑窗口和信息输出窗口三个视窗。

项目窗口。

项目窗口也称为项目工作区，主要显示开发的工程项目中的全部信息，包括类名、文件名及其项目文件等文件和函数列表，项目工作区文件的扩展名是.dsw，如果要打开项目，可以直接打开项目对应的工作区文件即可。在 Windows 的 32 位应用程序中，项目窗口包含有 ClassView 和 FileView 两个页面显示标签，单击项目窗口底部的页面选项卡可以实现两标签的切换。

● ClassView 页面

用于显示当前项目中的类以及该项目所包含的全局变量、函数等相关信息。若从该页面窗口中单击某个函数名，该函数的源代码就会显示在右边的程序窗中，图 1-10 中所示的就是项目窗口的 Classview 页面。

● FileView 页面

用于分类显示当前项目中的所有文件列表。包括：源文件、头文件、资源文件和帮助文件等。图 1-11 显示的是 ch01 项目所包含的源文件、头文件和资源文件信息。图中该项目仅包含一个 ch01_li01.cpp 源文件。

程序编辑窗口。

程序编辑窗口又称为代码窗口，用于编辑和显示当前项目的源程序代码，我们后续的程序代码将在这个窗口编辑。

信息输出窗口。

信息输出窗口主要用于显示程序编译和链接过程的信息。如果程序在编译、链接时没有错误，就会在该窗口显示程序编译和链接过程和对应的程序的名字等信息；若出错就显示出

错信息。



图 1-10 项目窗口“Class View”选项卡



图 1-11 项目窗口“File View”选项卡

(9) 在 Visual C++6.0 的程序编辑窗口中可以输入、编辑源程序代码，也可以用“文件”“打开”菜单命令打开以前的程序。本例输入以下的 C 程序，如图 1-12 所示。

```
#include <stdio.h>

int main()
{

    printf(" 这是我编写的第一个C语言程序！");

    return 0;

}
```

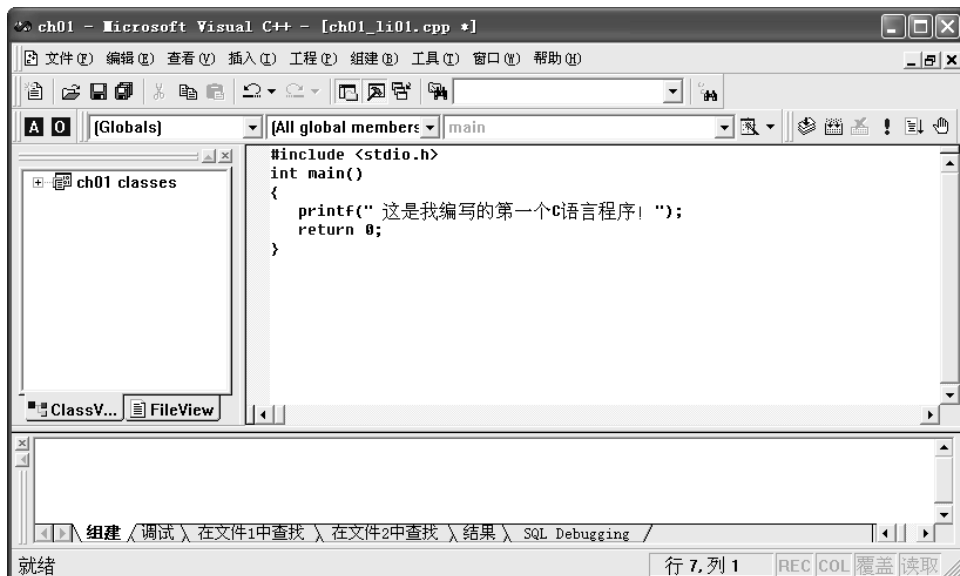


图 1-12 源程序输入窗口

源代码说明：

main 是主函数的函数名，表示这是一个主函数。每一个 C 源程序都必须有且只能有一个

主函数 (main()函数)。“return 0;”表示 int main()函数结束,返回 0。

主函数的说明也可定义为 void main()或 main(),此时可省略 return 语句。

函数调用语句 printf 函数的功能是把要输出的内容送到显示器中显示。printf 函数是一个在 stdio.h 文件中定义的标准函数,可在程序中直接调用,因此每个源程序首部要书写预处理语句 #include <stdio.h>或 #include"stdio.h"。

至此,我们就在 d:盘的 student\ch01 文件夹下创建了 ch01_li01.cpp 源程序文件。

使用 C 语言编写的源程序,是不能直接运行的。因为计算机只能识别和执行由 1 和 0 组成的二进制代码指令,不能识别和执行由高级语言编写的源程序。源程序就是用某种程序设计语言编写的程序,其中的程序代码称为源代码。因此,一个高级语言编写的源程序,必须用编译程序把高级语言程序翻译成机器能够识别的二进制目标程序,通过和系统提供的库函数和其他目标程序的连接,形成可以被机器执行的目标程序。所以一个 C 语言源程序到扩展名为.exe 可执行的文件,一般需要经过:编辑 编译调试 链接 运行四个步骤,上面我们编辑的源程序 ch01_li01.cpp 要想让计算机执行,需要经过如图 1-13 所示的步骤进行编译链接。

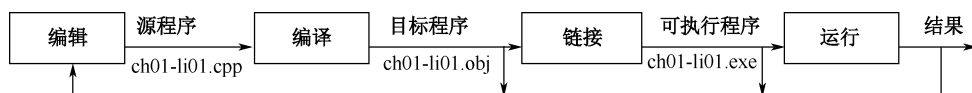



图 1-13 C 语言源程序编译链接流程图

编译时,会对源程序文件 ch01_li01.cpp 中的语法错误进行检测,并在信息输出窗口中给出反馈,编程者根据提示将错误一一纠正后完成编译,形成目标文件 ch01_li01.obj。链接是将程序中所加载的头函数及其他文件链接在一起,形成完整的可执行文件 ch01_li01.exe。

项目管理模式下源文件输入、编辑完成后,单击“文件” “保存”命令,对文件进行保存,然后按下面的步骤对其进行编译、链接和运行。

(10) 单击“组建” “编译[ch01_li01.cpp]”菜单命令,或按 Ctrl+F7 快捷键,或单击工具面板中的按钮,编译源程序。

源程序编译信息将会在信息输出窗口中出现。如果程序有语法错误,出错信息就显示在信息输出窗口中,包括错误的个数、位置、类型,可以直接用鼠标双击错误信息,系统可以实现错误的自动定位,方便程序员对程序的错误进行修改。对源文件出错信息修改后再编译,一直到源程序正确为止。

在图 1-14 所示的信息输出窗口中,看到了源程序 ch01_li01.cpp 的编译错误有“1 error(s), 0 warning(s)”的提示,错误信息为:“d:\student\ch01_li01\ch01_li01.cpp(5) : error C2143: syntax error : missing ';' before 'return'”,此行信息可以确定错误发生在 ch01_li01.cpp 文件的第 5 行,并且是语法错误,根据提示信息分析出是在“return”之前丢失了分号“;”,可以直接用鼠标双击错误信息行,系统会定位到发生错误的位置,即程序中的第 5 行,在“return”之前补写上分号“;”,即在程序中的第 4 行语句最后结束的位置补写上分号“;”,再次编译即可。如果程序中没有错误编译正确,在输出窗口中的信息“ch01_li01.obj - 0 error(s), 0 warning(s)”表示编译完成,如图 1-15 所示,系统已生成目标文件 ch01_li01.obj,并存于工程下的 debug 文件夹下。

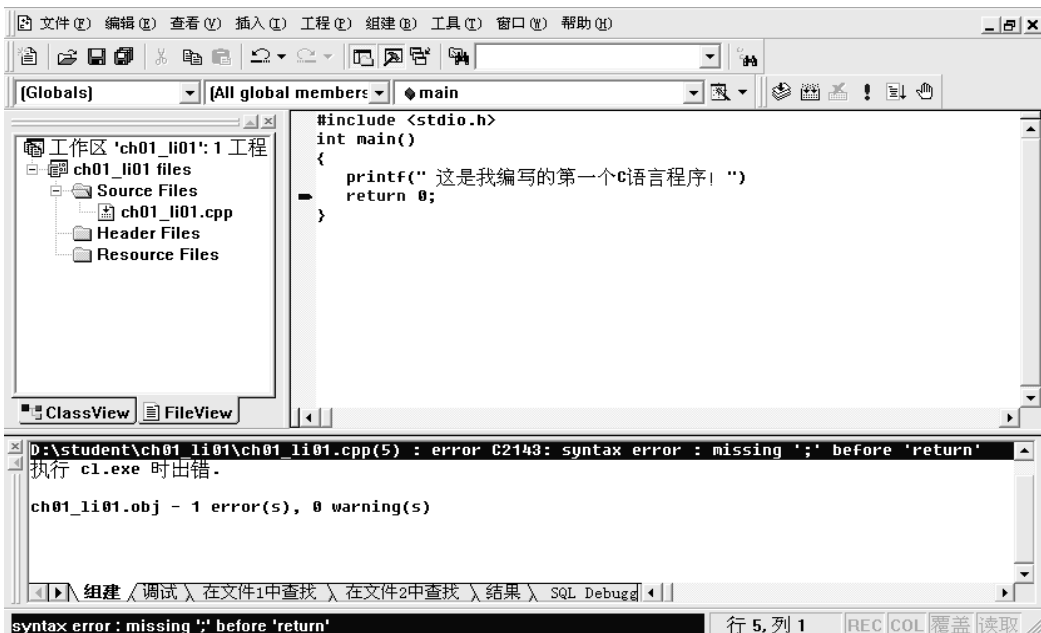


图 1-14 在编译时出错时输出的信息

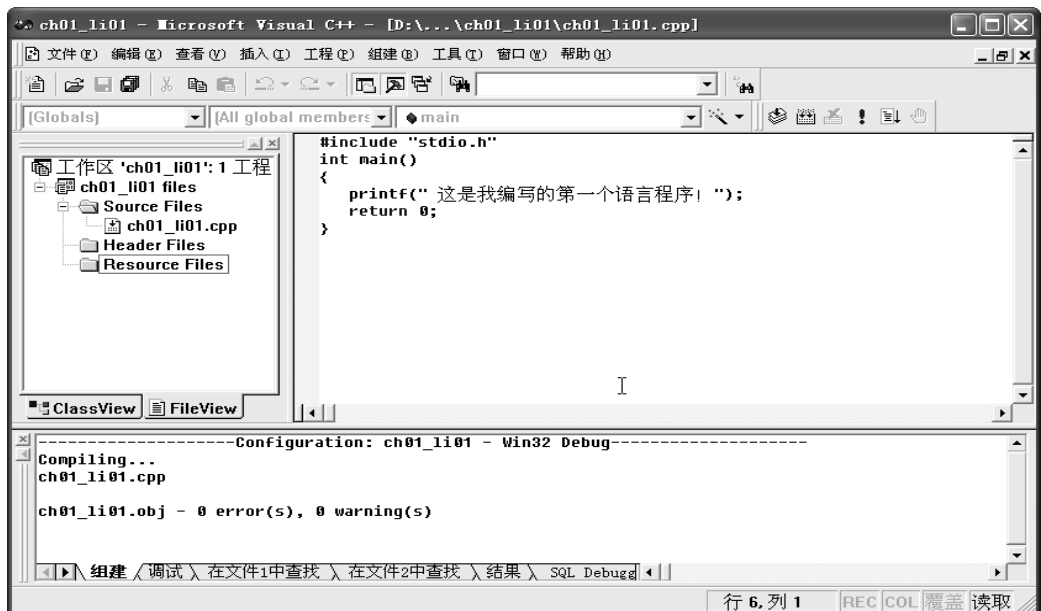




图 1-15 在编译正确时输出的信息

要注意的是：C 语言源程序的每一条语句需以“；”作为语句结束，但预处理命令、函数头和花括号“}”之后不能加分号。

(11) 编译通过后，单击“组建”菜单，选择“组件[ch01_li01.exe]”命令或按 F7 键，或单击工具面板中的  按钮，即开始进行链接。链接成功与否会在输出窗口中显示信息，如果链接成功，则生成可执行文件 ch01_li01.exe，存于 d:\student\ch01_li01\debug 文件夹下。

(12) 链接成功后，从“组建”菜单中选择“执行[ch01_li01.exe]”命令或按 Ctrl+F5 快捷键，或单击工具面板中的  按钮运行 ch01_li01.exe 文件，系统自动打开一个模拟 DOS

状态窗口，如图 1-16 所示，程序需要的输入数据与输出结果都在该窗口中进行。因此，与 Turbo C 类似，在 Visual C++6.0 中，程序的输出并不显示在输出窗口。程序运行完毕后，按任意键可返回到 Visual C++6.0 程序编辑窗口。



图 1-16 程序运行的 DOS 窗口

以上就是在 Visual C++6.0 中创建 C 程序运行环境的方式，实现了 C 程序的编辑 编译 链接 运行的全过程。也可以采用编译 链接 运行三个步骤合为直接“运行”一步完成方式，在执行“组建”菜单的“执行[ch01_li01.exe]”命令时，系统会自动检查程序是否已经编译、链接。若没有，就先对程序执行编译和链接，然后再运行。

说明：

① 一个工程可以包含多个源程序文件和头文件，但是源程序文件至少有一个，而头文件可以允许没有；当一个工程包含多个源程序文件时，只能有一个源程序文件包含 main() 函数，也就是说一个工程文件只能有一个 main() 函数，否则将会发生编译错误。

② 若打开原来已存盘的工程项目，在“文件”菜单中选择“打开工作区”命令，打开“打开工作区”对话框，选择工程项目所在的路径，从对话框选择项目的.dsw 文件（该文件是在创建项目时自动生成的），单击“打开”按钮，编辑、链接、运行等步骤与前面项目管理模式相同。

③ 在 Visual C++6.0 环境下编辑 C 程序，对于单行的注释允许惯用的简化标记符“//”，对于多行注释，使用“/*-----*/”标记形式。

从书写清晰，便于阅读，理解，维护的角度出发，在书写程序时应遵循以下规则：

① 一个说明或一个语句占一行。

② 用 {} 括起来的部分，通常表示程序的某一层结构。{} 一般与该结构语句的第一个字母对齐，并单独占一行。

③ 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写。以便看起来更加清晰，增加程序的可读性。在编程时应力求遵循这些规则，以养成良好的编程风格。

(13) 退出 Visual C++6.0 开发环境，单击“文件”菜单中的“退出”按钮，或者单击开发环境右上角的“关闭”按钮退出 Visual C++6.0。

2. 【实验内容 2】

根据命题要求“输入任意三个整数，求它们的和及平均值”，绘制程序流程图，在 Visual C++6.0 中输入程序，验证程序的运行结果。

实验内容 2 的操作步骤如下。

(1) 绘制流程图。

此问题是一个简单的输入、求解、输出的过程，是典型的顺序算法，流程图用到的基本组件有起止框、输入/输出框、处理框、流程线。绘制出程序流程图如图 1-17 所示。

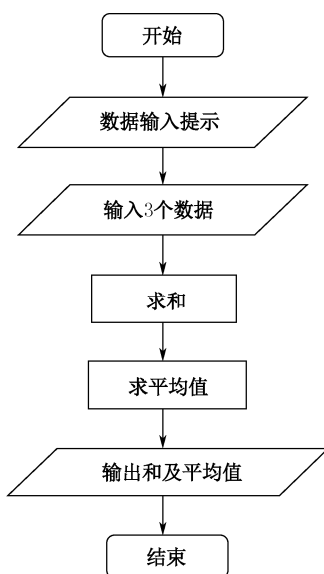


图 1-17 程序运行流程图

(2) 执行“开始”“程序”“Microsoft Visual Studio 6.0”“Microsoft Visual C++ 6.0”，菜单命令启动 VC++6.0。

(3) 执行“文件”“新建”命令，新建“Win32 Console Application”，选择存储路径及设定项目名称创建一个空工程。

(4) 再次执行“文件”“新建”菜单命令，新建一个“C++ Source File”，输入文件名，添加到步骤(2)创建的工程中。

(5) 在程序编辑窗口中输入如下代码，如图 1-18 所示。

```

#include <stdio.h>

void main()
{
    int num1,num2,num3,sum;
    float aver;

    printf("Please input three numbers:");
    scanf("%d,%d,%d",&num1,&num2,&num3);/*输入三个整数*/

    sum=num1+num2+num3;                /*求累计和*/
    aver=sum/3.0;                       /*求平均值*/

    printf("num1=%d,num2=%d,num3=%d\n",num1,num2,num3);
    printf("sum=%d,aver=%7.2f\n",sum,aver);

}
  
```

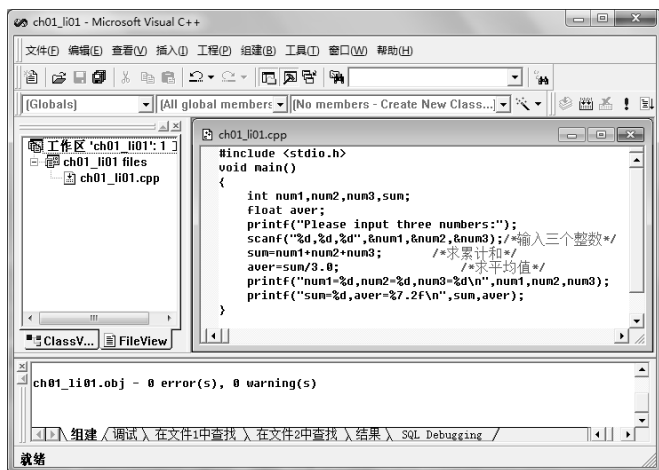





图 1-18 实验 2 程序编辑窗口

(6) 单击工具面板上的 ，进行编译，如没有语法错误，下面的“组建”窗口将输出“0 error(s), 0 warning(s)”的信息，如出现错误，则根据错误提示修改源程序，直到编译成功为止；编译没有语法错误后，单击工具面板上的 ，进行连接，此时依然是通过“组建”窗口查看连接信息，若出现错误，一般是库函数连接不成功，要检查开发环境，若对基层环境不是很熟悉，可重新新建项目来重启环境。连接正确后，单击工具面板上的 ，执行程序，转入如图 1-19 所示的运行界面。

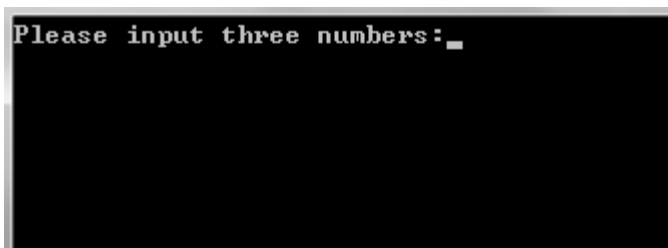


图 1-19 程序运行界面

该程序是一个典型的顺序结构流程，要得到运算结果，必须先有操作数据，界面上的提示信息是程序中的“printf("Please input three numbers:");”语句执行的结果，提示要求用户输入 3 个数据。接下来执行到“scanf("%d,%d,%d",&num1,&num2,&num3);”语句，用户在界面上输入 3 个数据，将会被分别存放在 3 个变量 num1、num2 和 num3 中。通过运行“sum=num1+num2+num3;”和“aver=sum/3.0;”语句得到和及平均值存放在变量 sum 和 aver 中。最后执行两条输出语句“printf("num1=%d,num2=%d,num3=%d\n",num1,num2,num3);”和“printf("sum=%d,aver=%7.2f\n",sum,aver);”输出运行结果，程序运行结果如图 1-20 所示。

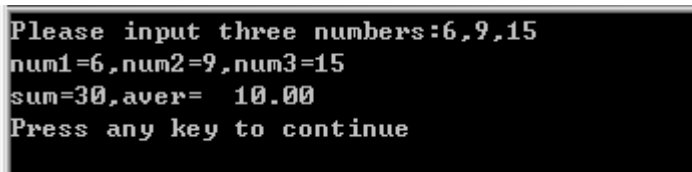


图 1-20 程序运行结果图

四、实验作业

- (1) 用程序流程图设计算法：输入一个数 n ，求出 $n!$ 。
- (2) 创建程序，在显示器屏幕上输出如图 1-21 所示的欢迎图案。



图 1-21 实验作业 2 运行界面

五、问题思考

- (1) 如果把 `main()` 函数前的 `void` 改为 `int`，程序能否正常运行，若出错，如何修改？
- (2) 如果去掉 `#include <stdio.h>`，程序能否正常运行，为什么？
- (3) 如果去掉每一个 `printf` 语句后的分号“`;`”，观察编译错误提示。
- (4) 如果去掉每一个 `printf` 语句中的“`\n`”，观察程序运行情况，思考“`\n`”的作用。