

# 《C程序设计》

主讲：王 新 教授

云南民族大学应用技术学院





教材：《C程序设计》（第五版）

谭浩强 著

清华大学出版社 2017年8月

参考书：

《C程序设计（第五版）学习辅导》谭浩强 主编

清华大学出版社 2017年8月



# 学习课程基本要求：

- 1、课前预习、上课听讲，课后复习，掌握所讲授内容；
- 2、上机实践（Visual C++环境或C-free或Dev-C++等），验证教材例子、完成作业的编程、调试、运行；
- 3、完成作业、实验，并提交作业或实验报告；
- 4、根据自己的学习情况，选学《学习辅导》中的内容；
- 5、成绩评定：包括考勤、实验、期中考试、期末考试。





# 课程联系与交流方式

---

1. 个人邮箱: wxkmyn@163.com

2 . 钉钉课程群: 扫码加入!

1) 发布通知

2) 分享电子资源

3) 提交作业等。





# 第1章 程序设计和C语言

---

1.1 什么是计算机程序

1.2 什么是计算机语言

1.3 C语言的发展及其特点

1.4 最简单的C语言程序

1.5 运行C程序的步骤与方法

1.6 程序设计的任务



# 1.1 什么是计算机程序

- **程序**：一组计算机能识别和执行的**指令**
- 只要让计算机执行这个程序，计算机就会**自动地、有条不紊地**进行工作
- 计算机的一切操作都是由**程序**控制的，离开程序，计算机将一事无成

冯·诺依曼：数字计算机的数制采用二进制；计算机应该按照程序顺序执行。 数字计算机之父。





## 1.2 什么是计算机语言

---

- **计算机语言**：人和计算机交流信息的、  
**计算机和人**都能识别的语言



# 1.2 什么是计算机语言

低级语言

## ■ 计算机语言发展阶段：

- 机器语言（由**0**和**1**组成的指令）。
- 符号语言（用英文字母和数字表示指令）、也称汇编语言
- 高级语言（接近于人的自然语言和数学语言）
  - 面向过程的语言  
（非结构化的语言、如：**BASIC**、**FORTRAN**、**ALGOL**；结构化语言、如：**QBASIC**、**FORTRAN77**、**C**）
  - 面向对象的语言如：**C++**、**C#**、**Visual Basic**、**Java**等。





# 1.3 C语言的发展及其特点

- C语言是国际上广泛流行的计算机高级语言。
- C语言的发展:

具有多种数据类型

BCPL语言



B语言



C语言

1967英国  
剑桥大学  
推出

精练、接近硬件，但  
过于简单，无数据类型

1973年美国贝尔实  
验室的D.M.Ritchie

1970年由美国贝  
尔实验室设计



## 1.3 C语言的发展及其特点

- 最初的**C**语言只是为描述和实现**UNIX**操作系统提供一种工作语言而设计的。
- **1983**年，美国国家标准协会(**ANSI**)成立了一个委员会，根据**C**语言问世以来各种版本对**C**语言的发展和扩充，制定了**第一个C语言标准草案('83 ANSI C)**。





## 1.3 C语言的发展及其特点

---

- 1989年，**ANSI**公布了一个完整的C语言标准—**ANSI X3.159-1989**(常称**ANSI C**，或**C89**)。





## 1.3 C语言的发展及其特点

---

- 1990年，国际标准化组织  
**ISO**(International Standard Organization) 接受**C89**作为国际标准  
**ISO/IEC 9899:1990**，它和**ANSI**的  
**C89**基本上是相同的。



## 1.3 C语言的发展及其特点

- **1995年，ISO对C90作了一些修订，1999年，ISO又对C语言标准进行修订，在基本保留原来的C语言特征的基础上，针对应用的需要，增加了一些功能，尤其是C++中的一些功能，命名为ISO/IEC 9899:1999。**



## 1.3 C语言的发展及其特点

- **2001、2004**年先后进行了两次技术修正（**TC1**和**TC2**）。

**ISO/IEC 9899:1999**(及其技术修正)被称为 **C99**。

- **C99**是**C89**(及**1995**基准增补**1**)的扩充。



## 1.3 C语言的发展及其特点

- 本书的叙述以**C99标准**为依据（对**C99**新增加的功能作特别的说明）。
- 目前不同软件公司提供的各**C语言**编译系统多数**并未完全实现C99**建议的功能
- 本书中程序基本上都可以在目前所用的编译系统(如**VC++ 6.0**，**Turbo C++ 3.0**,**GCC**)上编译和运行。



## 1.3 C语言的发展及其特点

- C语言是一种用途广泛、功能强大、使用灵活的**过程性(procedural)**编程语言，既可用于**编写应用软件**，又能用于**编写系统软件**。因此C语言问世以后得到迅速推广。







# 1.3 C语言的发展及其特点

---

- C语言主要特点：
  - 语言简洁、紧凑，使用方便、灵活。
    - 只有**37**个关键字、**9**种控制语句
    - 程序书写形式自由，源程序短





# 1.3 C语言的发展及其特点

- C语言主要特点：
  - 运算符丰富。
    - 有**34**种运算符(见附录C)
    - 把括号、赋值、强制类型转换等都作为运算符处理
    - 表达式类型多样化



# 1.3 C语言的发展及其特点

- C语言主要特点:

- 数据类型丰富。

- 包括:整型、浮点型、字符型、数组类型、指针类型、结构体类型、共用体类型
    - **C99**又扩充了复数浮点类型、超长整型(**long long**)、布尔类型(**bool**)
    - 指针类型数据, 能用来实现各种复杂的数据结构(如链表、树、栈等)的运算。



# 1.3 C语言的发展及其特点

- C语言主要特点：
  - 具有结构化的控制语句
    - 如**if...else**语句、**while**语句、**do...while**语句、**switch**语句、**for**语句
    - 用函数作为程序的模块单位，便于实现程序的模块化
    - C语言是完全模块化和结构化的语言



# 1.3 C语言的发展及其特点

- C语言主要特点:

- 语法限制不太严格，程序设计自由度大。
  - 对数组下标越界不做检查，由编写程序者保证程序正确
  - 对变量的类型使用比较灵活，例如，整型量与字符型数据可以通用
  - C语言允许程序编写者有较大的自由度，因此放宽了语法检查



# 1.3 C语言的发展及其特点

- C语言主要特点：
  - 允许直接访问物理地址，能进行位操作，可以直接对硬件进行操作
    - C语言具有高级语言的功能和低级语言的许多功能，可用来编写系统软件
    - 这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言



# 1.3 C语言的发展及其特点

- C语言主要特点:

- 用C语言编写的程序可移植性好。

- C的编译系统简洁，很容易移植到新系统
- 在新系统上运行时，可直接编译“标准链接库”中的大部分功能，不需要修改源代码
- 几乎所有计算机系统都可以使用C语言





# 1.3 C语言的发展及其特点

---

- C语言主要特点：
  - 生成目标代码质量高，程序执行效率高。







# 1.4最简单的C语言程序

---

1.4.1 最简单的C语言程序举例

1.4.2 C语言程序的结构





## 1.4.1 最简单的C语言程序举例

**例1.1** 要求在屏幕上输出以下一行信息。

**This is a C program.**

➤ 解题思路：

在主函数中用**printf**函数原样输出以上文字。



# 1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

C程序必须有一个 main 函数

```
{
```

函数的名字，表示主函数

```
printf("this is a C program.\n");
```

```
return 0;
```

```
}
```



# 1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

主函数类型

```
printf("this is a C program.\n");
```

```
return 0;
```

```
}
```





## 1.4.1 最简单的C语言程序举例

---

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    printf ("This is a C program.\n");
```

```
    return 0;
```

```
}
```

函数体



## 1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

输出函数

```
printf("This is a C program.\n");
```

```
return 0;
```

```
}
```

输出语句



## 1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
printf ("This is a C program.\n");
```

```
return 0;
```

```
}
```

```
This is a C program.  
Press any key to continue.
```

输出语句



## 1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
printf ("This is a C program.\n");
```

```
return 0;
```

```
}
```

A screenshot of a terminal window with a black background and white text. The first line displays "This is a C program." and the second line displays "Press any key to continue." with a cursor at the end.

This is a C program.  
Press any key to continue.

换行符





## 1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    printf ("This is a C program.\n");
```

```
    return 0;
```

```
}
```

当main函数执行结束前  
将整数0作为函数值

## 1.4.1 最简单的C语言程序举例

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

用到函数库中的输入输出函数时

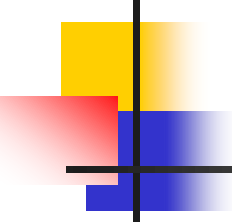
```
printf ("This is a C program.\n");
```

```
return 0;
```

```
}
```

表示语句结束





## P15,习题5

---

```
#include <stdio.h>
int main ( )
{ printf ("*****\n\n");
  printf("    Very Good!\n\n");
  printf ("*****\n");
  return 0;
}
```



## 1.4.1 最简单的C语言程序举例

C语言允许用两种注释方式:

- `//`: 单行注释
  - 可单独占一行
  - 可出现在一行中其他内容的右侧
- `/*.....*/`: 块式注释
  - 可包含多行





## 例1.2 求两个整数之和。

---

➤ 解题思路：

◆ 设置3个变量

◆ **a**和**b**用来存放两个整数

◆ **sum**用来存放和数

◆ 用赋值运算符“=”把结果传送给**sum**





```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    int a,b,sum;
```

定义整型变量a,b,sum

```
    a = 123;
```

```
    b = 456;
```

} 对变量a,b赋值

```
    sum = a + b;
```

将a与b的和赋给sum

```
    printf("sum is %d\n",sum);
```

```
    return 0;
```

```
}
```



```
sum is 579
```





```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    int a,b,sum;
```

```
    a = 123;
```

```
    b = 456;
```

```
    sum = a + b;
```

```
    printf("sum is %d\n",sum);
```

```
    return 0;
```

```
}
```

%d是指定的输出格式;  
\n为换行符

用sum的值替代

希望输出的字符



```
sum is 579
```





## 例1.3 求两个整数中的较大者。

➤ 解题思路:

- ◆ 用一个函数实现求两个整数中的较大者
- ◆ 在主函数中调用此函数并输出结果





```
#include <stdio.h>

int main( )
{
    int max(int x,int y);
    int a,b,c;
    scanf("%d,%d",&a,&b);
    c = max(a,b);
    printf("max=%d\n",c);
    return 0;
}
```

主函数

max函数

```
int max(int x,int y)
{
    int z;
    if (x > y) z = x;
    else z = y;
    return(z);
}
```

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    int max(int x,int y);
```

```
    int a,b,c;
```

```
    scanf("%d,%d",&a,&b);
```

```
    c = max(a,b);
```

```
    printf("max=%d\n",c);
```

```
    return 0;
```

```
}
```

将x和y中较大者  
值返回给主函数

```
int max(int x,int y)
```

```
{
```

```
    int z;
```

```
    if (x > y) z = x;
```

```
    else z = y;
```

```
    return(z);
```

```
}
```

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    int max(int x,int y);
```

```
    int a,b,c;
```

```
    scanf("%d,%d",&a,&b); {
```

```
    c = max(a,b);
```

```
    printf("max=%d\n",c);
```

```
    return 0;
```

```
}
```

```
int max(int x,int y)
```

```
    int z;
```

```
    if (x > y) z = x;
```

```
    else z = y;
```

```
    return(z);
```

```
}
```

```
#include <stdio.h>
```

```
int main( )
```

```
{ 因max函数的定义在main函数之后，需声明
```

```
int max(int x,int y);
```

```
int a,b,c;
```

```
scanf("%d,%d",&a,&b); {
```

```
c = max(a,b);
```

```
printf("max=%d\n",c);
```

```
return 0;
```

```
}
```

```
int max(int x,int y)
```

```
{
```

```
int z;
```

```
if (x > y) z = x;
```

```
else z = y;
```

```
return(z);
```

```
}
```

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    in  (int x,int y);
```

```
    int a,b,c;
```

```
    scanf("%d,%d",&a,&b);
```

```
    c = max(a,b);
```

```
    printf("max=%d\n",c);
```

```
    return 0;
```

```
}
```

```
int max(int x,int y)
```

```
{
```

```
    int z;
```

```
    if (x > y) z = x;
```

```
    else z = y;
```

```
    return(z);
```

```
}
```

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
int max(
```

输入语句

```
int a,b,c;
```

```
scanf("%d,%d",&a,&b); {
```

```
c = max(a,b);
```

```
printf("max=%d\n",c);
```

```
return 0;
```

```
}
```

8,5

```
int max(int x,int y)
```

```
{
```

```
int z;
```

```
if (x > y) z = x;
```

```
else z = y;
```

```
return(z);
```

```
}
```

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
int x, y);
```

```
int a,b,c;
```

```
scanf("%d,%d",&a,&b);
```

```
c = max(a,b);
```

```
printf("max= ");
```

```
return 0;
```

```
}
```

8,5

输入格式

输入的数据  
放到a,b中

a的地址

```
int max(int x,int y)
```

```
{
```

```
int z;
```

```
if (x > y) z = x;
```

```
else z = y;
```

```
return(z);
```

```
}
```

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    int max(int x,int y);
```

调用max函数

```
    scanf("%d,%d",&a,&b);
```

```
    c = max(a,b);
```

```
    printf("max=%d\n",c);
```

```
    return 0;
```

```
}
```

8,5

```
int max(int x,int y)
```

```
{
```

```
    int z;
```

```
    if (x > y) z = x;
```

```
    else z = y;
```

```
    return(z);
```

```
}
```



```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
    int max(int x,int y);
```

```
    int a,b,c;
```

```
    scanf("%d,%d",&a,&b);
```

```
    c = max(a,b);
```

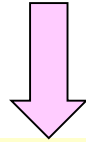
```
    printf("max=%d\n",c);
```

```
    return 0;
```

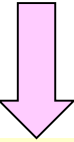
```
}
```

8,5

8



5



```
int max(int x,int y)
```

```
{
```

```
    int z;
```

```
    if (x > y) z = x;
```

```
    else z = y;
```

```
    return(z);
```

```
}
```

8



```

#include <stdio.h>
int main( )
{
    int max(int x,int y);
    int a,b,c;
    scanf("%d,%d",&a,&b);
    c = max(a,b);
    printf("max=%d\n",c);
    return 0;
}

```

```

8,5
max=8

```

8 5

↓ ↓

```

int max(int x,int y)
{
    int z;
    if (x > y) z = x;
    else z = y;
    return(z);
}

```

8

```
#include <stdio.h>
```

```
int main( )
```

```
{
```

```
int max(int x, int y);
```

```
int a, b;
```

```
scanf("%d,%d",&a,&b);
```

```
c = max(a,b);
```

```
printf("max=%d\n",c);
```

```
return 0;
```

```
}
```

实际参数

```
8,5  
max=8
```

形式参数

```
int max(int x, int y)
```

```
{
```

```
int z;
```

```
if (x > y) z = x;
```

```
else z = y;
```

```
return(z);
```

```
}
```

## 1.4.2 C语言程序的结构

C语言程序的结构特点:

**1.**一个程序由一个或多个源程序文件组成(后部份自学)

- 小程序往往只包括一个源程序文件
- 例1.1, 例1.2只有一个函数
- 例1.3有两个函数

只包括一个源程序文件



## 1.4.2 C语言程序的结构

C语言程序的结构特点:

- 一个源程序文件中可以包括三个部分:
  - 预处理指令 `#include <stdio.h>`等
  - 全局声明 在函数之外进行的数据声明
  - 函数定义 每个函数用来实现一定的功能





## 1.4.2 C语言程序的结构

C语言程序的结构特点：

### 2.函数是C程序的主要组成部分

- 一个C程序是由一个或多个函数组成的
- 必须包含一个**main**函数（只能有一个）
- 每个函数都用来实现一个或几个特定功能
- 被调用的函数可以是库函数，也可以是自己编制设计的函数



## 1.4.2 C语言程序的结构

C语言程序的结构特点:

3. 一个函数包括两个部分:

- 函数首部

int max ( int x, int y )

函数的第1行

函数类型

函数名

参数类型

参数名

## 1.4.2 C语言程序的结构

C语言程序的结构特点:

3. 一个函数包括两个部分:

- 函数首部

**int max ( int x, int y )**

若函数无参, 在括弧中写**void**或空括弧

**int main( void) 或 int main()**





## 1.4.2 C语言程序的结构

C语言程序的结构特点:

3. 一个函数包括两个部分:

- 函数体

- 声明部分

可以没有声明部分

- 定义在本函数中所用到的变量

- 对本函数所调用函数进行声明

- 执行部分: 由若干个语句组成, 指定在函数中所进行的操作



## 1.4.2 C语言程序的结构

C语言程序的结构特点:

3. 一个函数包括两个部分

- 函数体

可以是空函数

```
void dump ( )  
{  
}
```

## 1.4.2 C语言程序的结构

C语言程序的结构特点：

4. 程序总是从**main**函数开始执行

5. C程序对计算机的操作由C语句完成

- C程序书写格式是比较自由的
  - 一行内可以写几个语句
  - 一个语句可以分写在多行上
- 为清晰起见，习惯上每行只写一个语句





## 1.4.2 C语言程序的结构

C语言程序的结构特点：

- 4. 程序总是从**main**函数开始执行
- 5. C程序对计算机的操作由C语句完成
- 6. 数据声明和语句最后**必须有分号**
- 7. C语言本身不提供输入输出语句
- 8. 程序应当包含注释，增加可读性

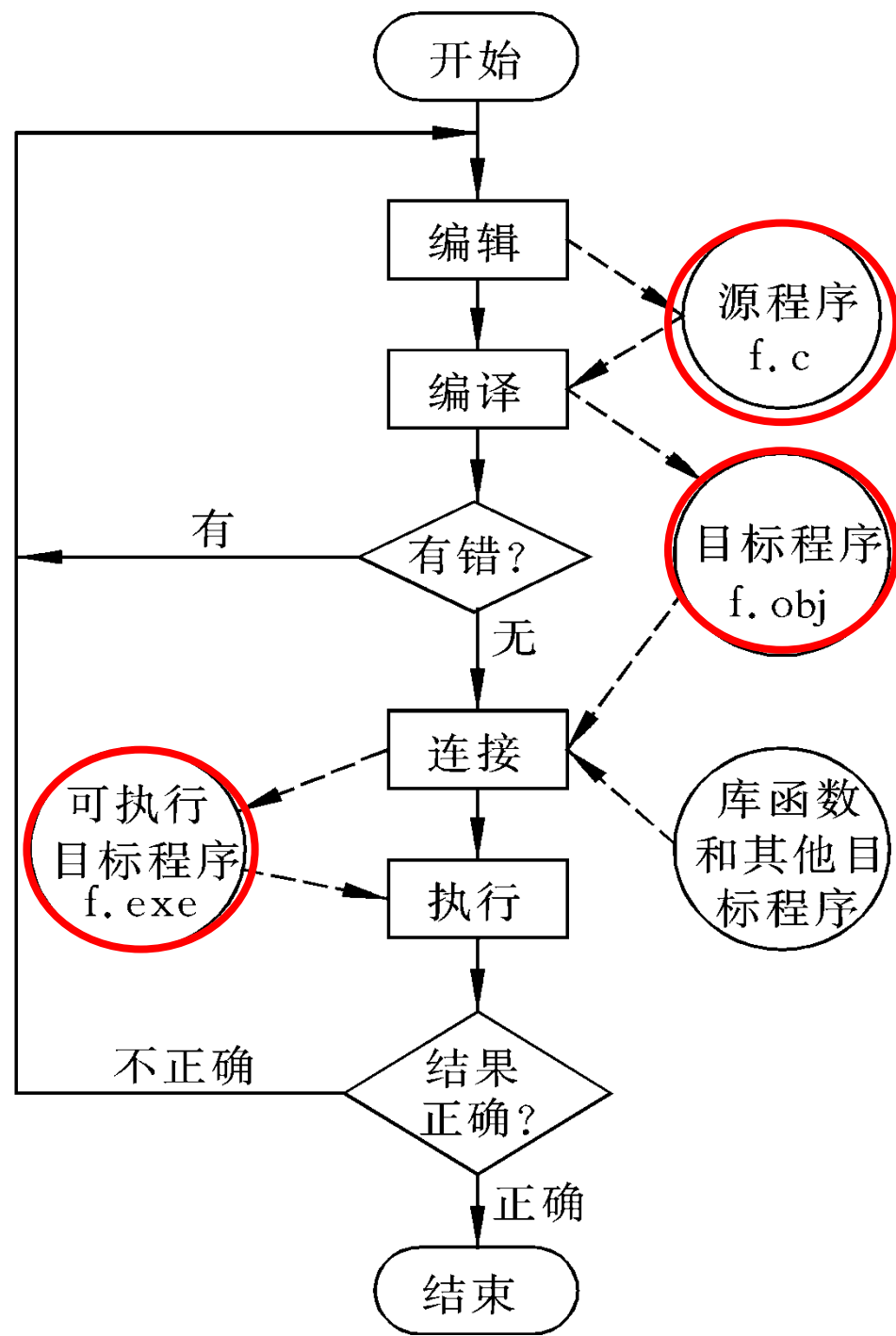
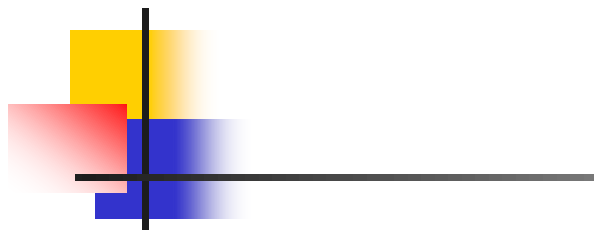


# 1.5 运行C程序的步骤与方法

- 1.上机输入和编辑源程序（**.c**文件）
  - 2.对源程序进行编译（**.obj**文件）
  - 3.进行连接处理（**.exe**文件）
  - 4.运行可执行程序，得到运行结果
- 说明：以上过程参见教材中图**1.2**

**参考：**C程序设计（第五版） 学习辅导 第**15**章  
怎样使用 **Visual C++6.0**运行程序







# VC++上机运行C程序的方法

- 可用不同的编译系统对C程序进行操作  
如：C-free、Dev-C++等。
- 本节介绍在Visual C++ 6.0 中怎样编辑、编译、连接和运行C程序



# VC++上机运行C程序的方法

- 安装VC++ 6.0
  - VC++ 6.0有英文版和中文版
  - 必须事先安装VC++ 6.0系统
  - 安装后最好在桌面上设立快捷方式图标，以方便使用
- VC++6.0集成环境





文件、编辑、查看、插入、项目、构建、工具、窗口、帮助

File Edit View Insert Project Build Tools Window Help

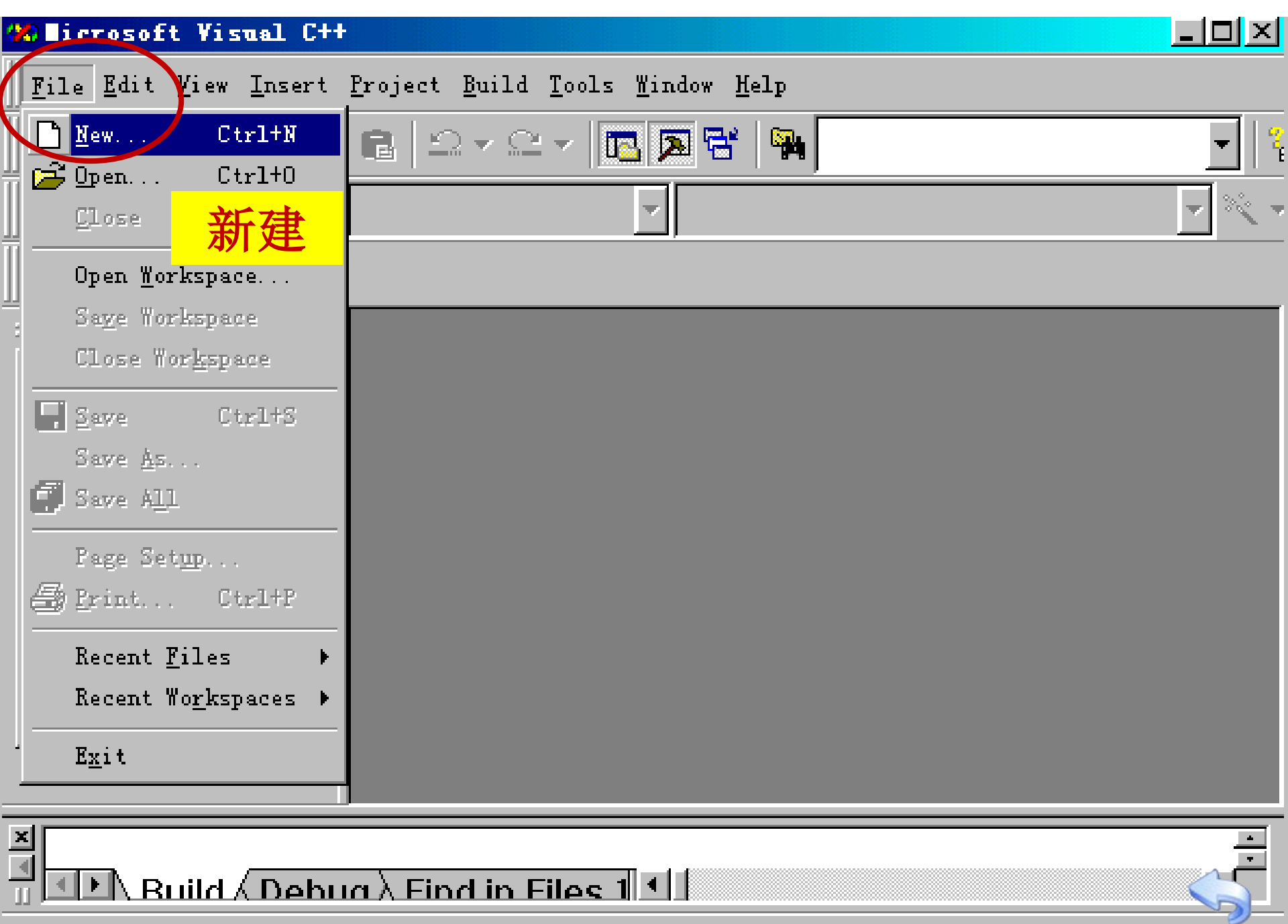


项目工作  
区窗口

程序编辑  
窗口



Ready



新建



Creates a new document, project or workspace

1

New



Files

Projects

Workspaces

Other Documents

- Active Server Page
- Binary File
- Bitmap File
- C/C++ Header File
- C++ Source File**
- Cursor File
- HTML Page
- Icon File
- Macro File
- Resource Script
- Resource Template
- SQL Script File
- Text File

2

☐ Add to project:

4

File

c1-1.c

Location:

D:\CC



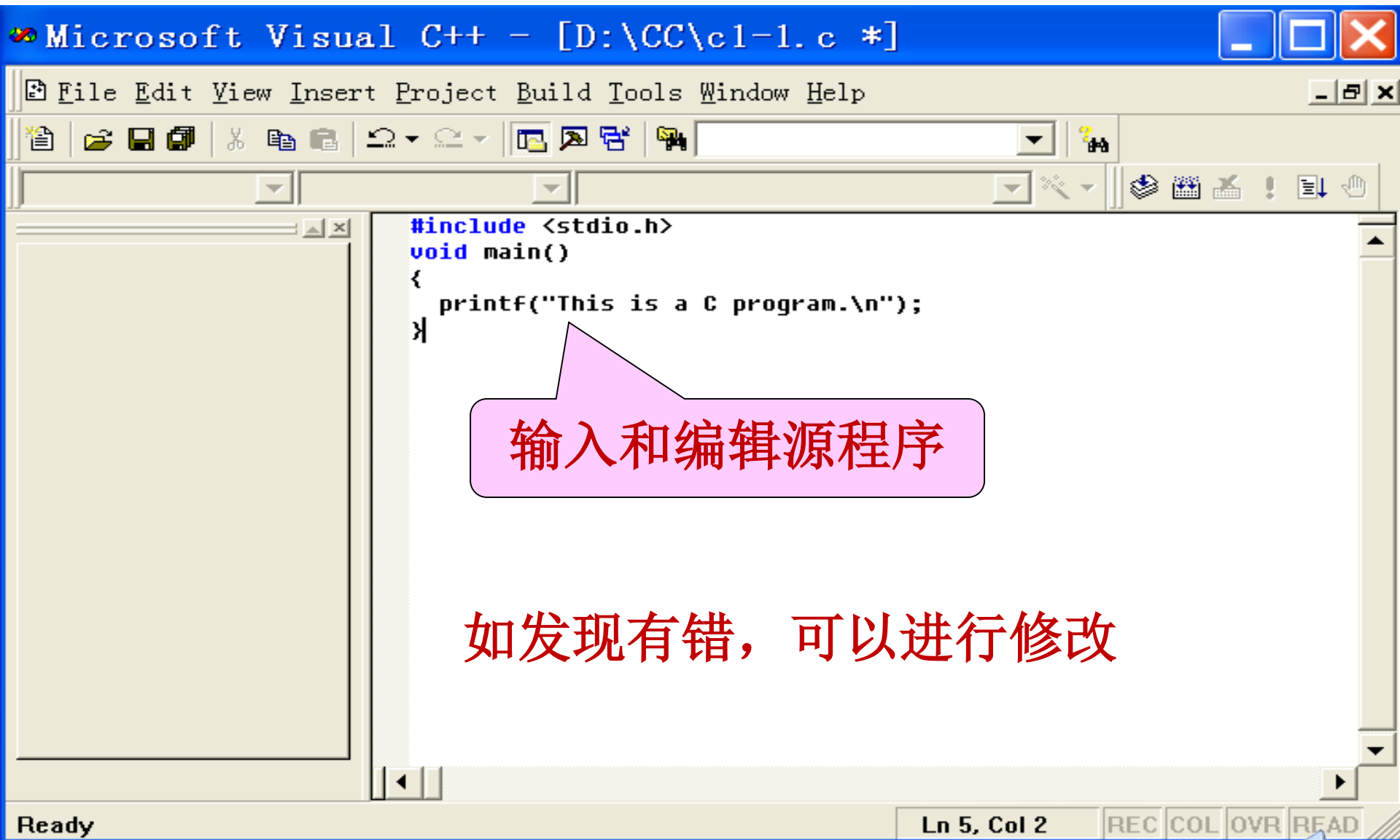
3

5

OK

Cancel

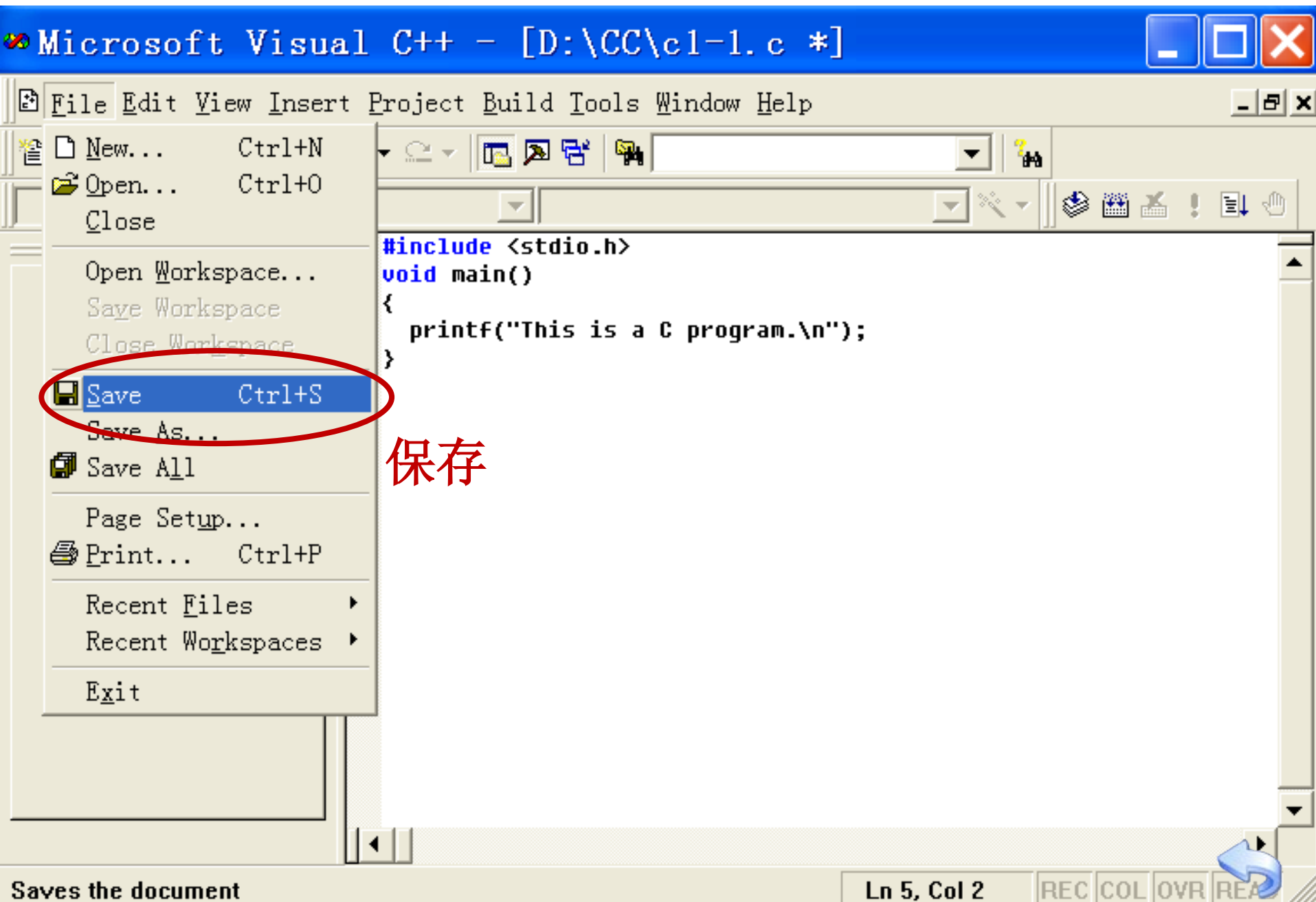


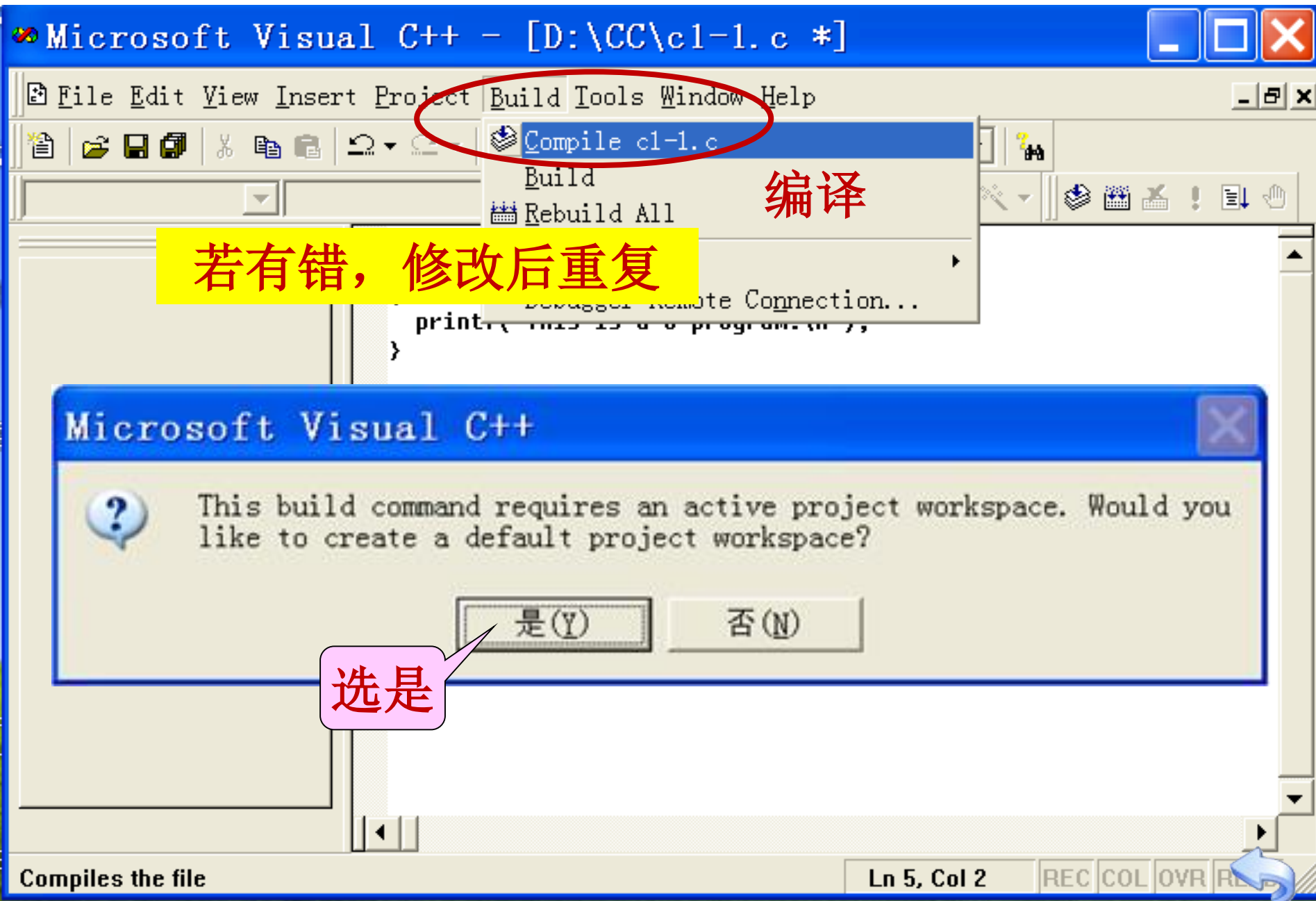


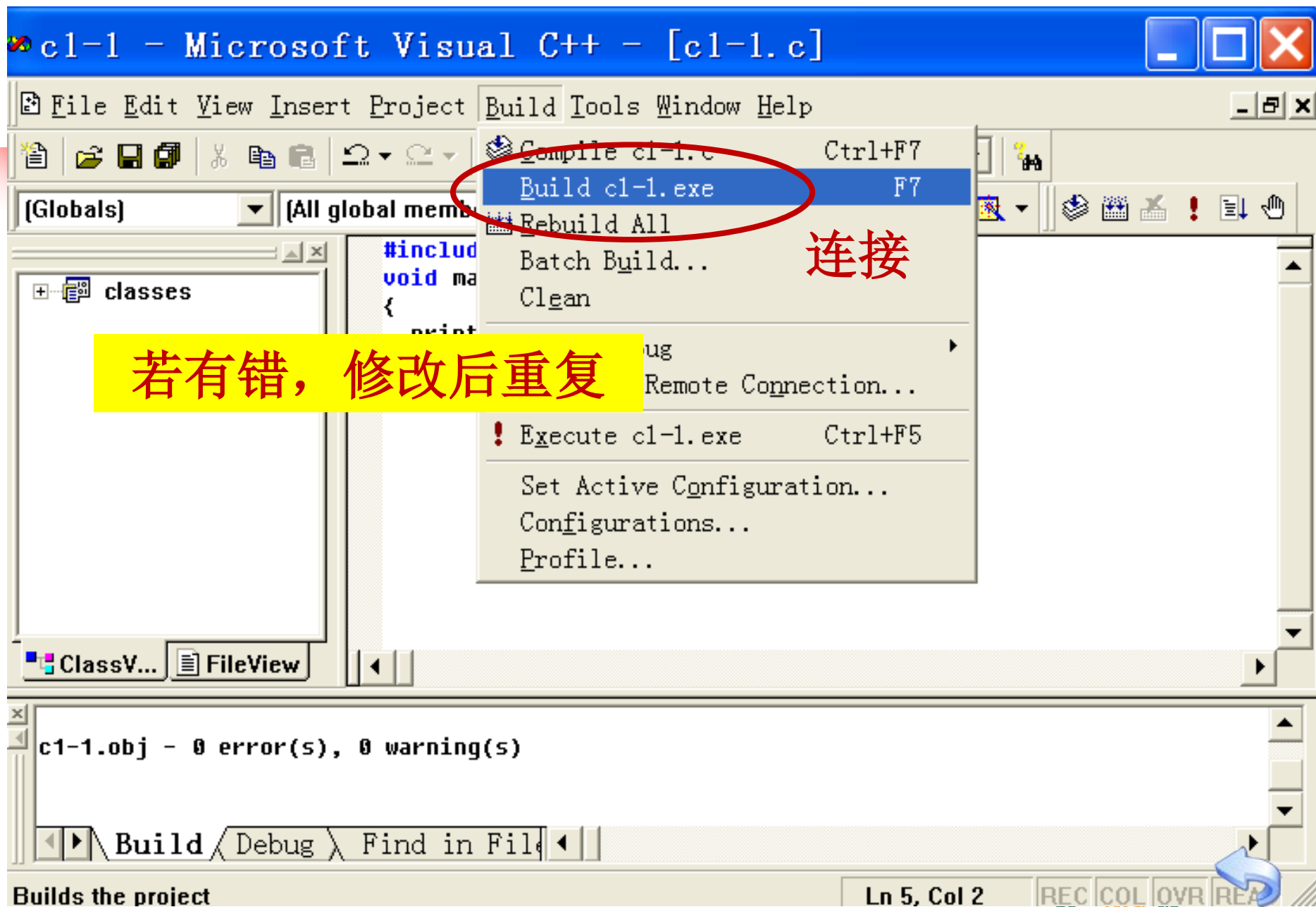
输入和编辑源程序

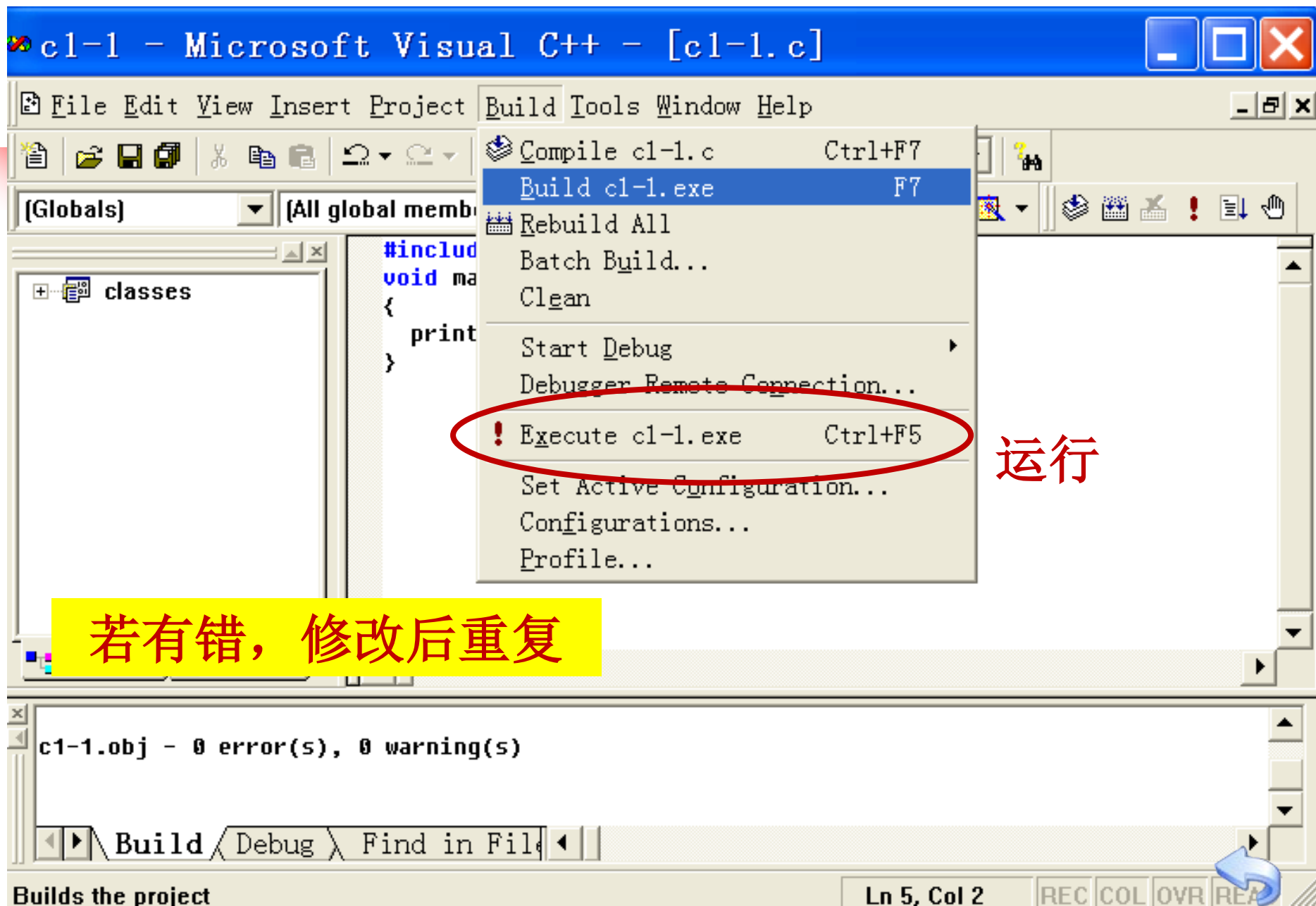
如发现有错，可以进行修改



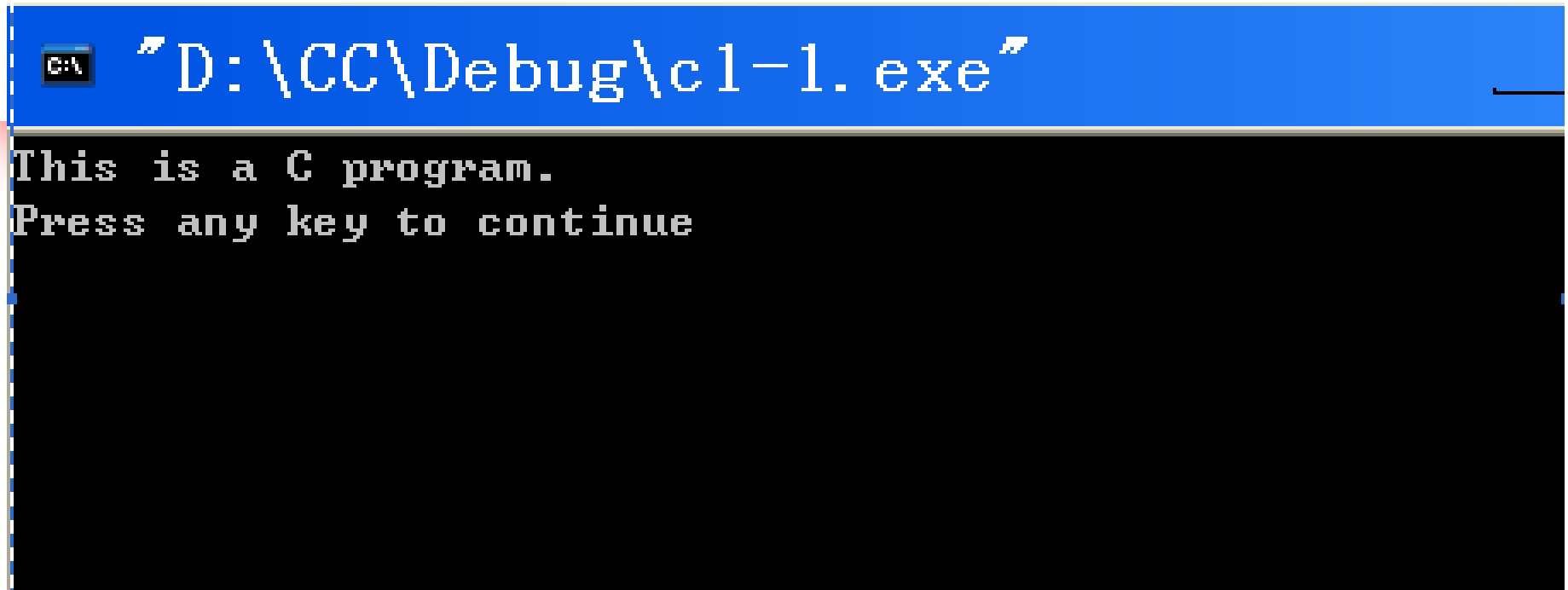












```
C:\> "D:\CC\Debug\cl-1.exe"

This is a C program.
Press any key to continue
```

如果不再进行其他的处理，应当选择文件→关闭工作区，结束对该程序的操作





# 1.6 程序设计的任务

---

## 1. 问题分析

- 对于接手的任务要进行认真的分析
- 研究所给定的条件
- 分析最后应达到的目标
- 找出解决问题的规律
- 选择解题的方法





# 1.6 程序设计的任务

---

**1. 问题分析**

**2. 设计算法**

■ 设计出解题的方法和具体步骤





# 1.6 程序设计的任务

---

**1. 问题分析**

**2. 设计算法**

**3. 编写程序**

**4. 对源程序进行编辑、编译和连接**

**5. 运行程序，分析结果**

- 结果错了，程序肯定错
- 结果对了，程序未必对





# 1.6 程序设计的任务

---

1. 问题分析
2. 设计算法
3. 编写程序
4. 对源程序进行编辑、编译和连接
5. 运行程序，分析结果
6. 编写程序文档



# 作业

## 4. 编写一个 C 程序,运行时输出

```
*****
```

```
Hello World!
```

```
*****
```

这个程序是一些常见的国外 C 教材中作为第一个程序例子介绍的,一般称为 Hello 程序。

解: 程序如下:

```
#include <stdio.h>

int main ( )
{
    printf ("*****\n\n");
    printf("      Hello  World!\n\n");
    printf ("*****\n");
    return 0;
}
```

运行结果:

```
*****
      Very  Good!
*****
```



5. 编写一个 C 程序,运行时输出以下图形:

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

解: 程序如下:

```
#include <stdio.h>
int main ( )
{
    printf ("*****\n");
    printf (" *****\n");
    printf ("      *****\n");
    printf ("          *****\n");
    return 0;
}
```



6. 编写一个 C 程序,输入 a,b,c 三个值,输出其中最大者。

解: 程序如下:

```
#include <stdio.h>

int main( )
{int a,b,c,max;
 printf("please input a,b,c: \n");
 scanf("%d,%d,%d",&a,&b,&c);
 max=a;
 if (max<b)
    max=b;
 if (max<c)
    max=c;
 printf("The largest number is %d\n",max);
 return 0;
}
```

运行结果:

```
please input a,b,c:
18,-43,34
The largest number is 34
```

输入 3 个数: 18, -43, 34, 输出最大数 34。

注意: 输入的 3 个数以逗号分隔, 如果以空格分隔, 会出错, 读者可试一下。请思考为什么。