



计算机导论与程序设计——第5篇

程序设计概述

Computer Introduction and Programming

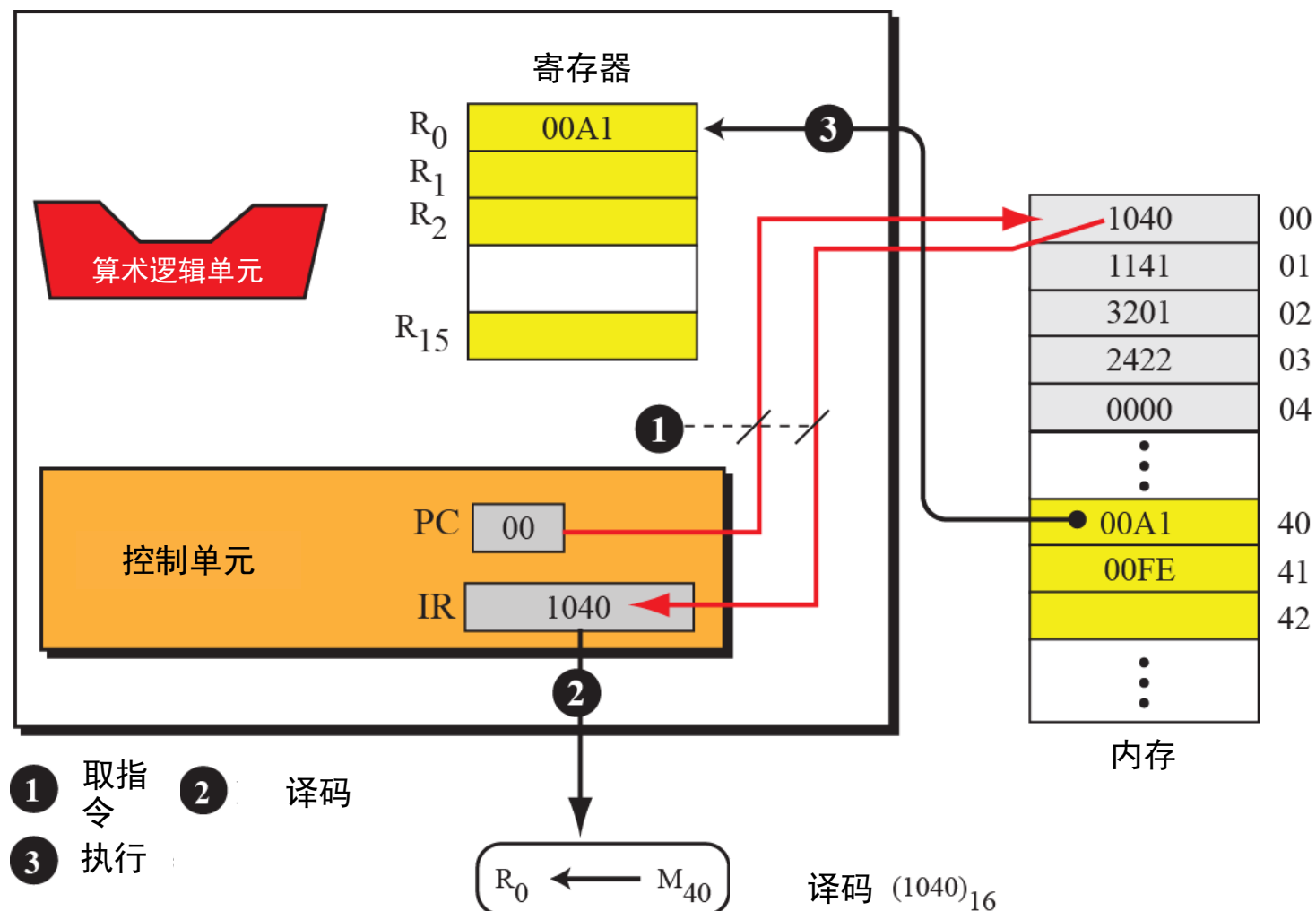
学习目标



- 了解程序设计语言的发展历史及趋势，了解不同类型语言的特点及优势
- 熟悉程序设计基本概念
- 掌握程序执行的原理和过程，对程序在计算机上的执行过程建立基本概念
- 掌握C语言程序的框架结构和创建简单C程序的方法

计算机程序

回顾



计算机程序



指令

可以被计算机理解并执行的基本操作命令。



程序

一组计算机能识别和执行的指令。
一个特定的指令序列用来完成一定的功能。



软件

与计算机系统操作有关的计算机程序、规程、规则，以及可能有的文件、文档及数据。

计算机语言

机器语言

计算机能直接识别和接受的二进制代码称为**机器指令**。机器指令的集合就是该计算机的**机器语言**。

特点：难学，难记，难检查，难修改，难以推广使用。依赖具体机器难以移植。

```
B8 7F 01
BB 21 02
03 D8
B8 1F 04
2B C3
```

汇编语言

机器语言的符号化。用英文字母和数字表示指令的**符号语言**。

特点：相比机器语言简单好记，但仍然难以普及。汇编指令需通过**汇编程序**转换为机器指令才能被计算机执行。依赖具体机器难以移植。

```
MOV AX 383
MOV BX 545
ADD BX AX
MOV AX 1055
SUB AX BX
```

高级语言

高级语言更接近于人们习惯使用的自然语言和数学语言。

特点：功能强大，不依赖于具体机器。用高级语言编写的**源程序**需要通过**编译程序**转换为机器指令的**目标程序**。

```
S=1055-(383+545)
```

高级语言的发展

01

非结构化的语言

02

结构化语言

规定：

程序必须由具有良好特性的基本结构(顺序结构、选择结构、循环结构)构成，程序中的流程不允许随意跳转，程序总是由上而下顺序执行各个基本结构。

特点：

程序结构清晰，易于编写、阅读和维护。

03

面向对象的语言

C语言的发展



D.M.Ritchie

- 1972–1973年间，美国贝尔实验室的D.M.Ritchie设计出了C语言。
- 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。
- 1978年以后，C语言便很快风靡全世界，成为世界上应用最广泛的程序设计高级语言。
- 1978年，Brian W.Kernighan和Dennis M.Ritchie 合著了影响深远的名著The C Programming Language，这本书中介绍的C语言成为后来广泛使用的C语言版本的基础，它是实际上第一个C语言标准。

C语言的发展

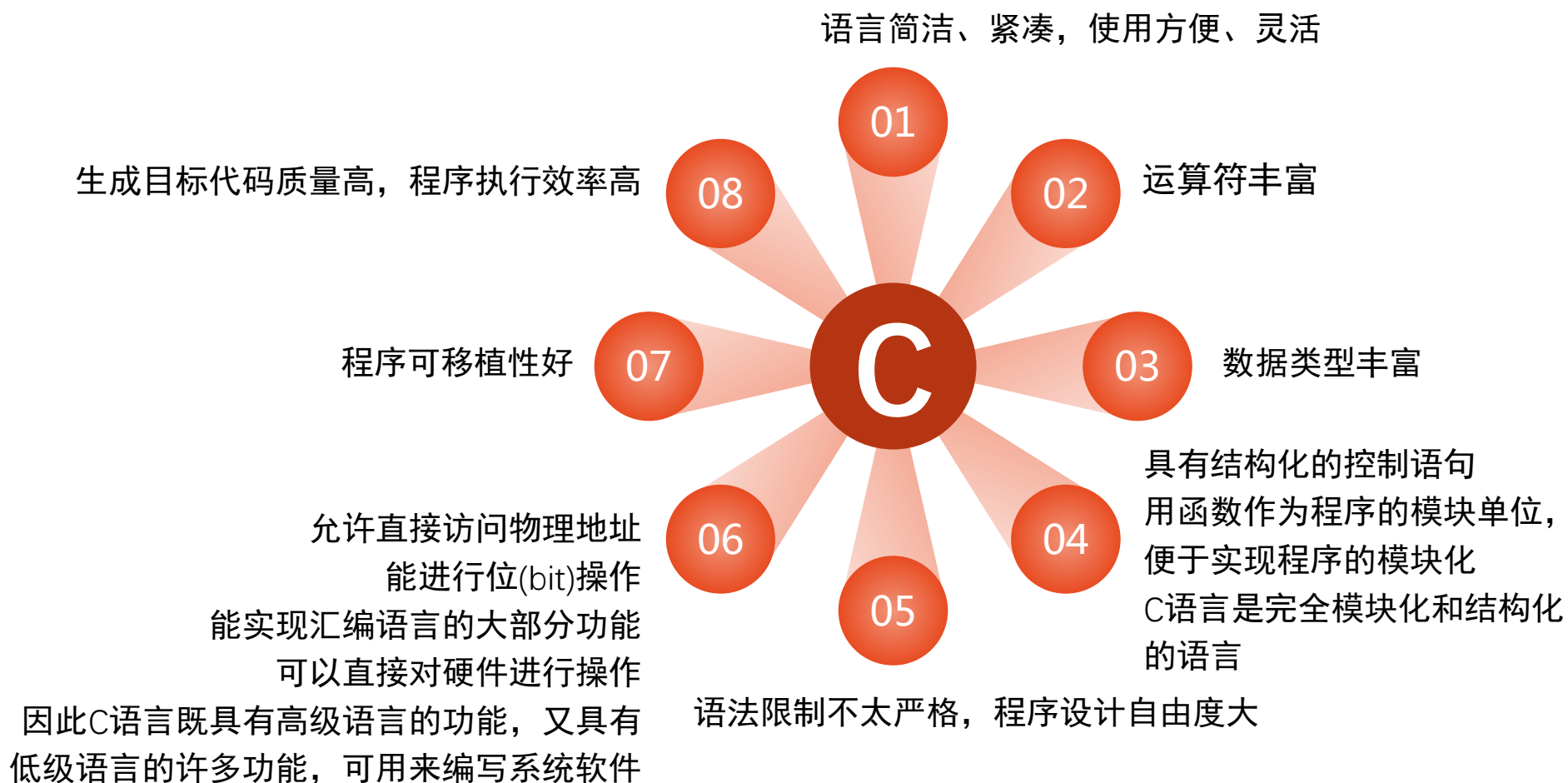


D.M.Ritchie

- 1983年，美国国家标准协会(ANSI)，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)。
- 1989年，ANSI公布了一个完整的C语言标准—ANSI X3.159-1989(常称为ANSI C或C 89)。
- 1990年，国际标准化组织ISO(International Standard Organization)接受C 89作为国际标准ISO/IEC 9899:1990，它和ANSI的C 89基本上是相同的。
- 1999年，ISO又对C语言标准进行了修订，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，并在2001年和2004年先后进行了两次技术修正，它被称为C 99，C 99是C 89的扩充。

目前由不同软件公司所提供的一些C语言编译系统并未完全实现C 99建议的功能，它们多以C 89为基础开发。

C语言的特点

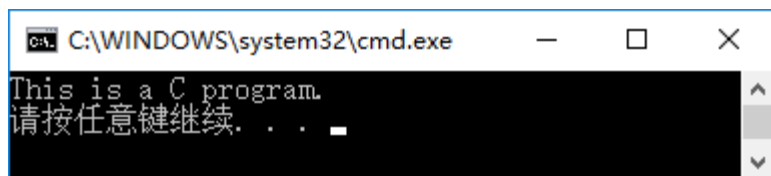


最简单的C语言程序

【例5.1】要求在屏幕上输出：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 #include <stdio.h> //这是编译预处理指令
2
3 int main()          //定义主函数
4 {                  //函数开始的标志
5     printf("This is a C program.\n"); //输出所指定的一行信息
6     return 0;       //函数执行完毕时返回函数值0
7 }                  //函数结束的标志
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window has a black background with white text. The first line of output is "This is a C program." followed by a newline. The second line is "请按任意键继续. . . ." followed by a cursor. The window has standard Windows window controls (minimize, maximize, close) in the title bar.

最简单的C语言程序

```
1 #include <stdio.h> //这是编译预处理指令
2
3 int main()          //定义主函数
4 {                  //函数开始的标志
5     printf("This is a C program.\n"); //输出所指定的一行信息
6     return 0;       //函数执行完毕时返回函数值0
7 }                  //函数结束的标志
```



程序分析

- **main**是函数的名字，表示“主函数”；每一个C语言程序都必须有一个main 函数。
- main前面的**int**表示此函数的类型是int类型(整型)，即在执行主函数后会得到一个值(即函数值)，其值为整型。

最简单的C语言程序

```
1 #include <stdio.h> //这是编译预处理指令
2
3 int main()          //定义主函数
4 {                  //函数开始的标志
5     printf("This is a C program.\n"); //输出所指定的一行信息
6     return 0;       //函数执行完毕时返回函数值0
7 }                  //函数结束的标志
```



程序分析

- **return 0;**的作用是当main函数执行结束前将整数0作为函数值，返回到调用函数处。
- 函数体由花括号**}**括起来。

最简单的C语言程序

```
1 #include <stdio.h> //这是编译预处理指令
2
3 int main()          //定义主函数
4 {                  //函数开始的标志
5     printf("This is a C program.\n"); //输出所指定的一行信息
6     return 0;       //函数执行完毕时返回函数值0
7 }                  //函数结束的标志
```



程序分析

- **printf**是C编译系统提供的函数库中的输出函数。printf函数中**双引号**内的字符串"This is a C program."按原样输出。**\n**是换行符，即在输出"This is a C program."后，显示屏上的光标位置移到下一行的开头。
- 每个语句最后都有一个**分号**，表示语句结束。

最简单的C语言程序

```
1 #include <stdio.h> //这是编译预处理指令
2
3 int main()          //定义主函数
4 {                  //函数开始的标志
5     printf("This is a C program.\n"); //输出所指定的一行信息
6     return 0;       //函数执行完毕时返回函数值0
7 }                  //函数结束的标志
```



程序分析

- 在使用函数库中的输入输出函数时，编译系统要求程序提供有关此函数的信息，程序第1行“**#include <stdio.h>**”的作用就是用来提供这些信息的。**stdio.h**是系统提供的一个文件名，**stdio**是standard input & output的缩写，文件后缀**.h**的意思是头文件(header file)。输入输出函数的相关信息已事先放在stdio.h文件中。

最简单的C语言程序

```
1 #include <stdio.h> //这是编译预处理指令
2
3 int main()          //定义主函数
4 {                  //函数开始的标志
5     printf("This is a C program.\n"); //输出所指定的一行信息
6     return 0;       //函数执行完毕时返回函数值0
7 }                  //函数结束的标志
```



程序分析

- `//`表示从此处到本行结束是“注释”，用来对程序有关部分进行必要的说明。在写C程序时应当多用注释，以方便自己和别人理解程序各部分的作用。在程序进行预编译处理时将每个注释替换为一个空格，因此在编译时注释部分不产生目标代码，注释对运行不起作用。注释只是给人看的，而不是让计算机执行的。

注释

以//开始的单行注释

这种注释可以单独占一行，也可以出现在一行中其他内容的右侧。此种注释的范围从//开始，以换行符结束。如果注释内容一行内写不下，可以用多个单行注释。

```
//第一行注释  
//继续注释
```

```
/*一整块都是  
注释*/
```

以/*开始，以*/结束的块式注释

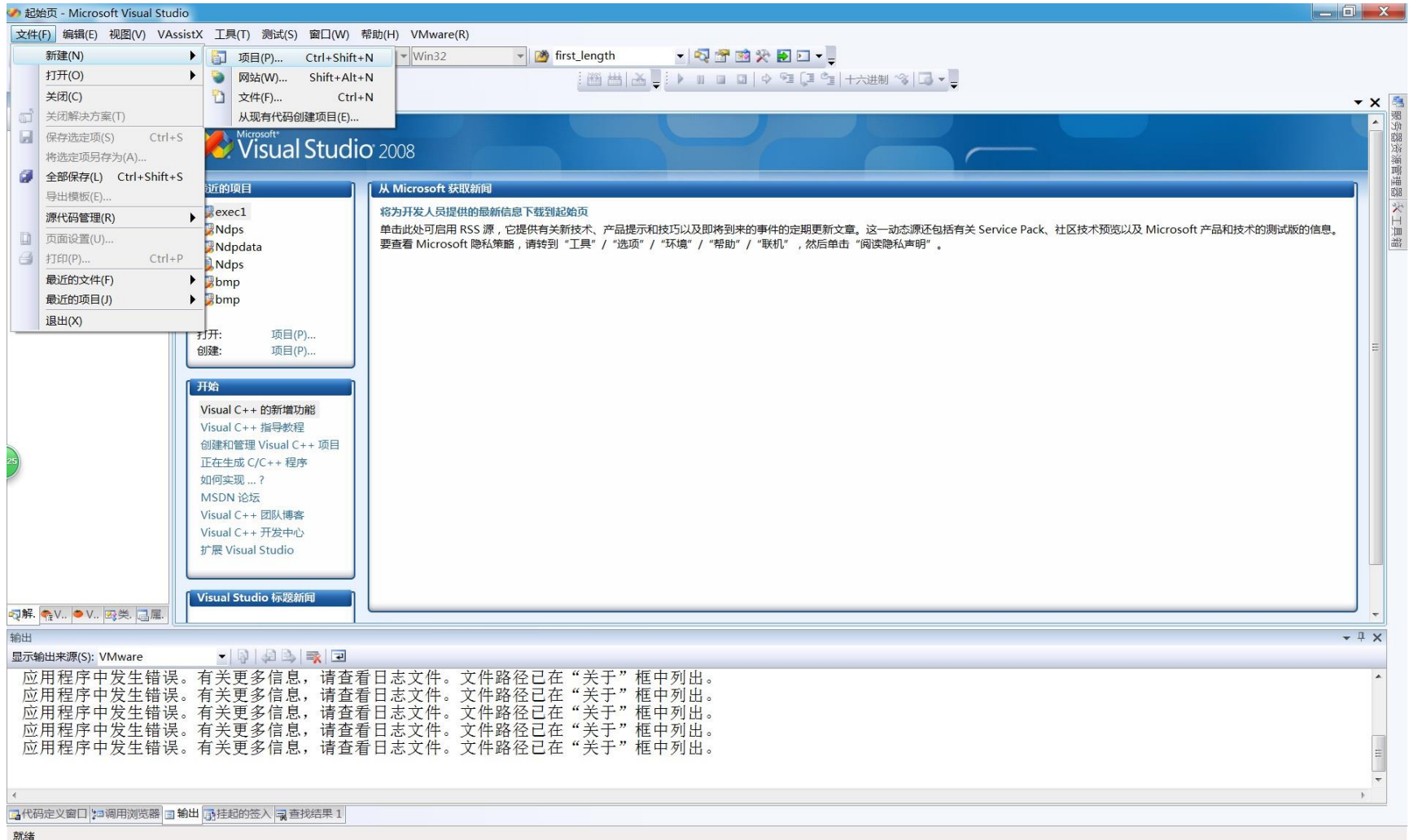
这种注释可以包含多行内容。它可以单独占一行(在行开头以/*开始，行末以*/结束)，也可以包含多行。编译系统在发现一个/*后，会开始找注释结束符*/，把二者间的内容作为注释。

注意：在字符串中的//和/*都不作为注释的开始。而是作为字符串的一部分。

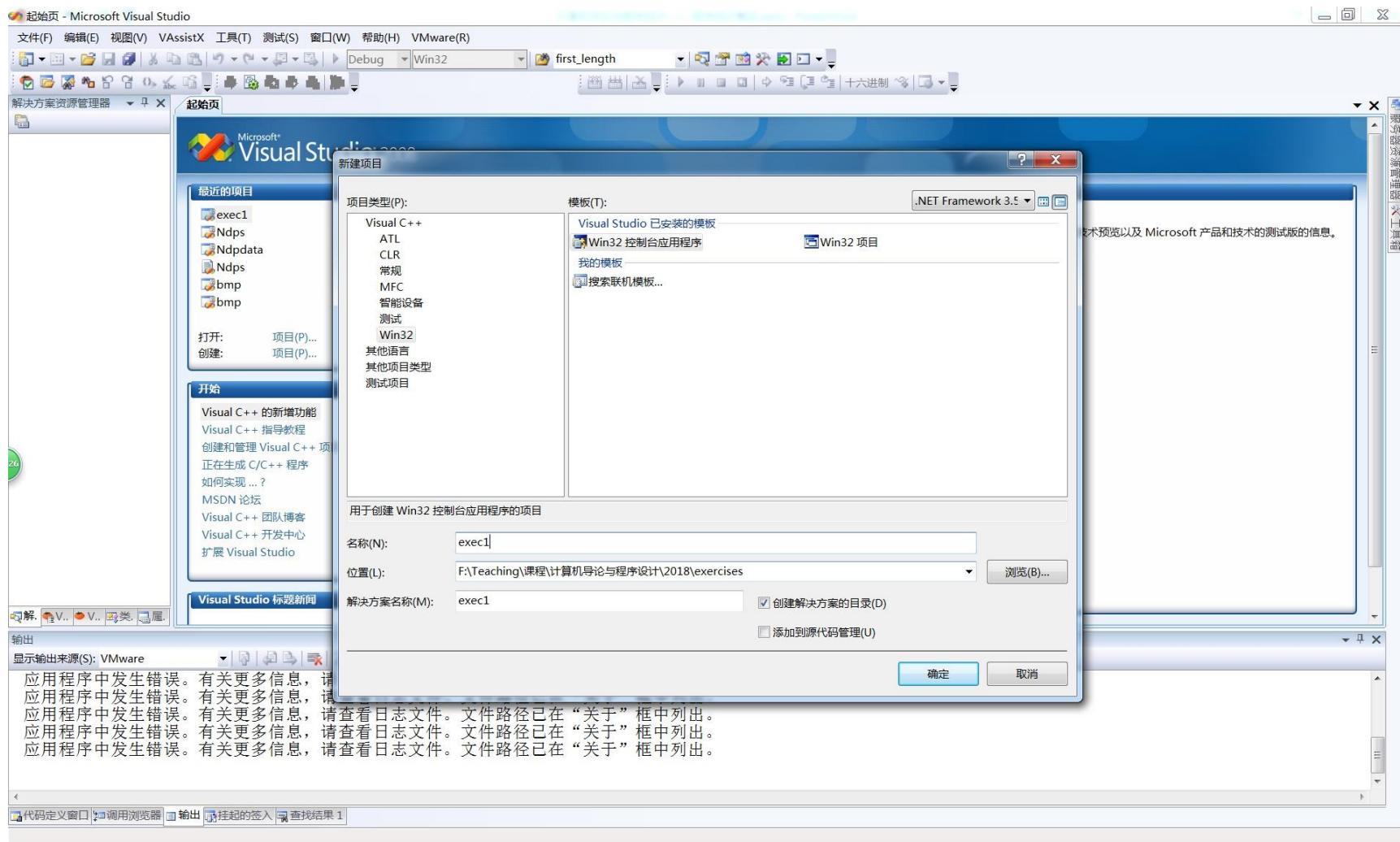
认识编程环境

- 编程环境也叫**集成开发环境**(Integrated Developing Environment, IDE), 是一个综合性的工具软件, 它把程序设计全过程所需的各项功能集合在一起, 为程序设计人员提供完整的服务。
- 例如, Visual C++ 6.0, Visual Studio 2010, Code Blocks, Dev-C++

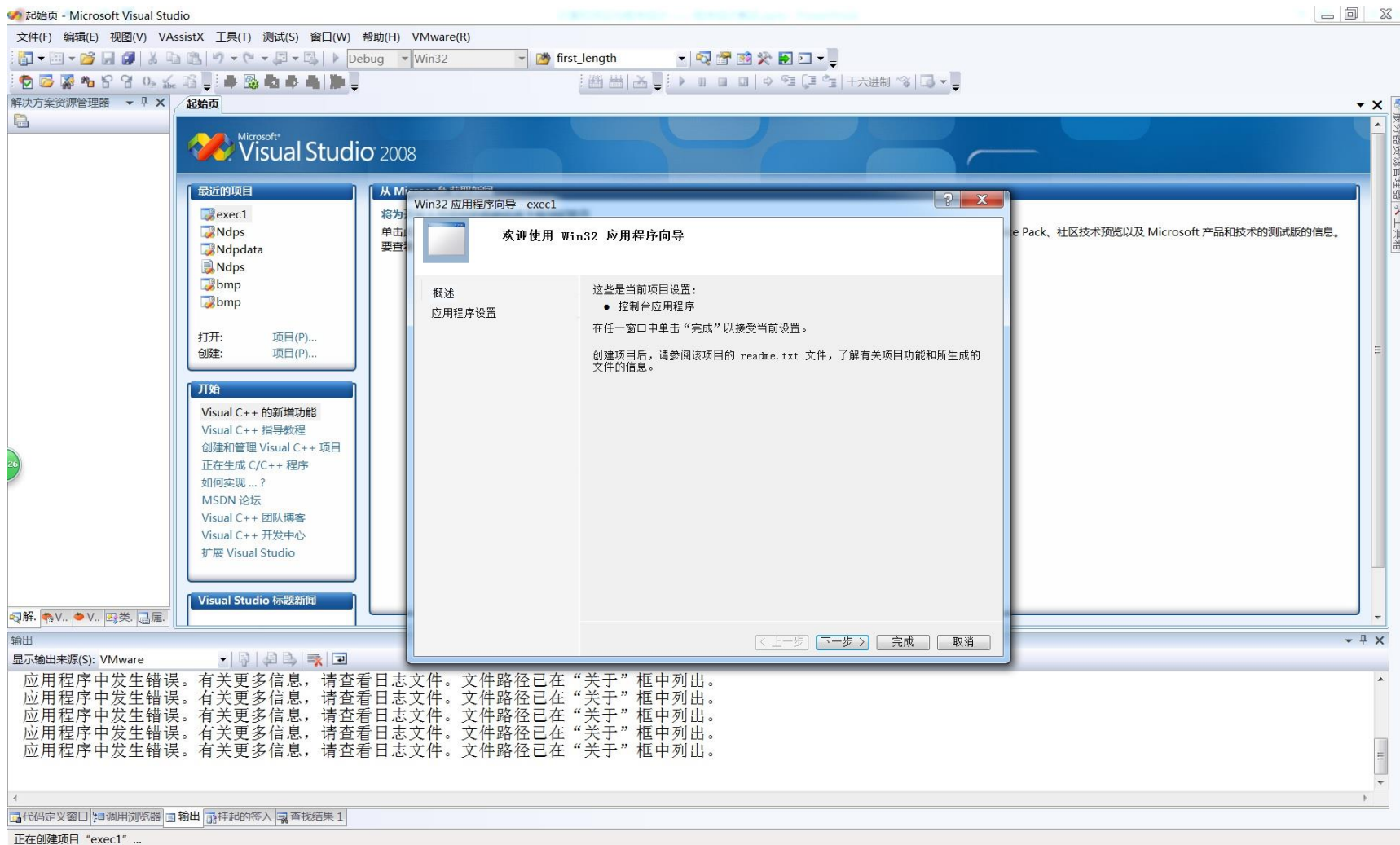
Visual Studio 编程 - 步骤1



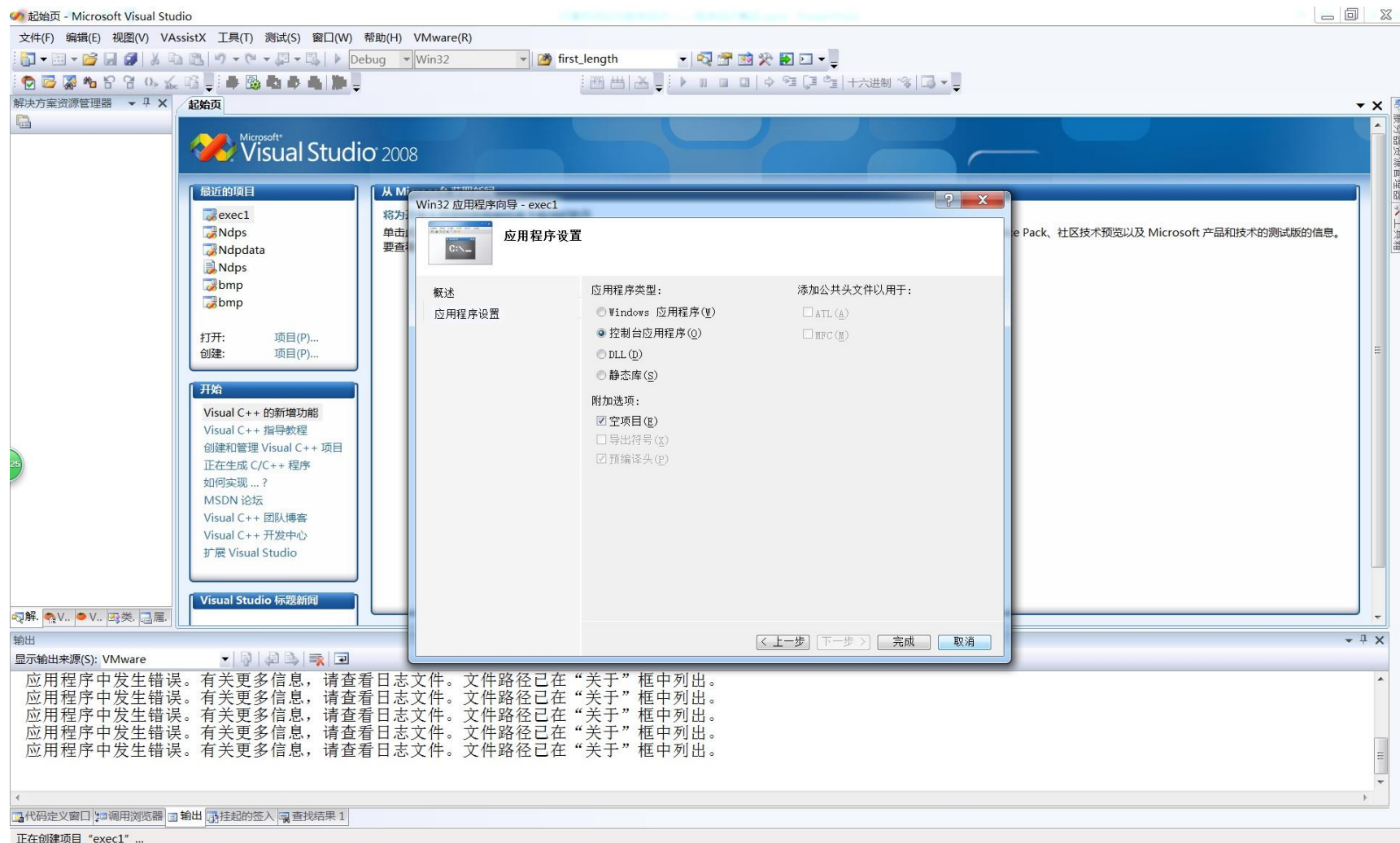
Visual Studio 编程-步骤2



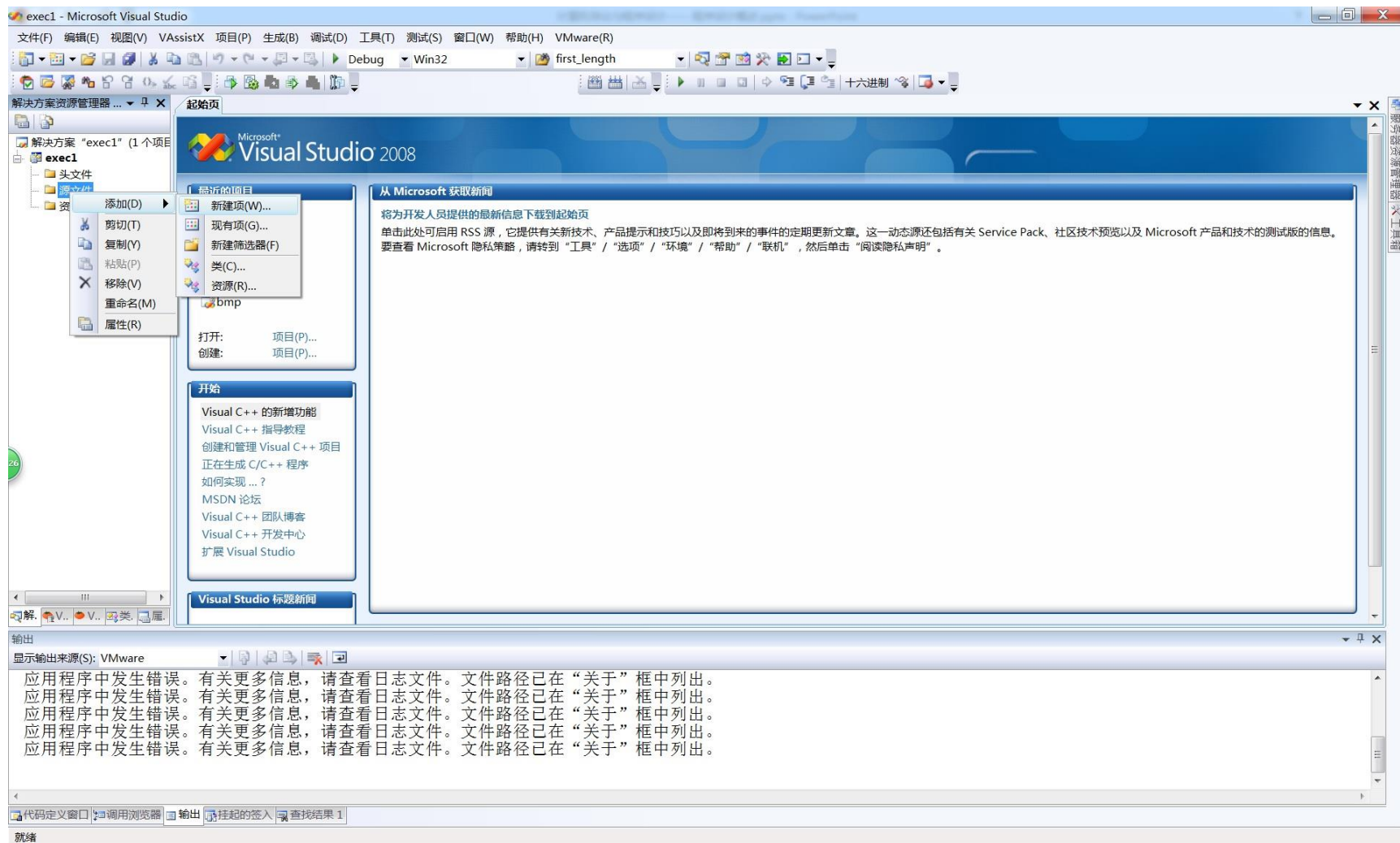
Visual Studio 编程-步骤3



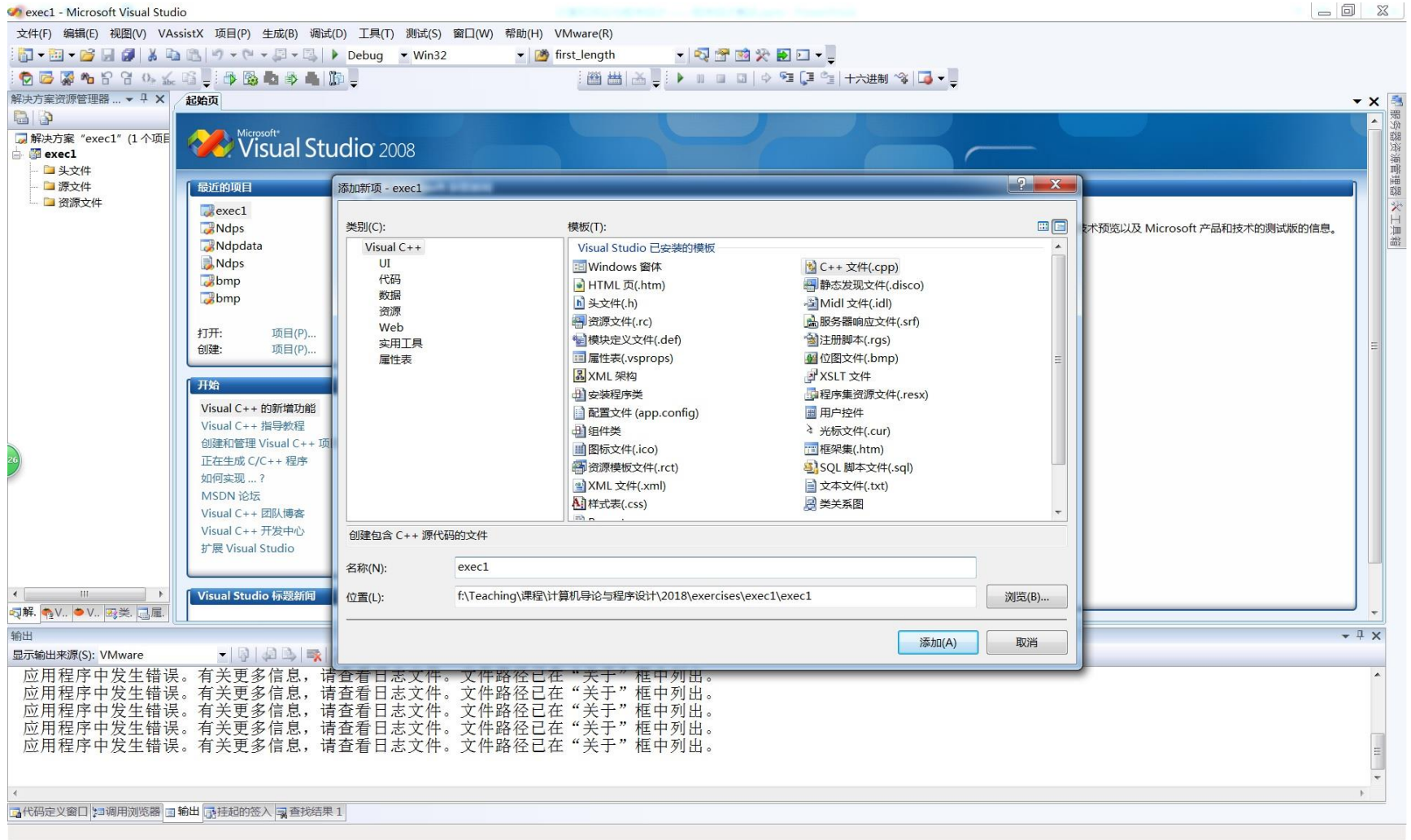
Visual Studio 编程-步骤4



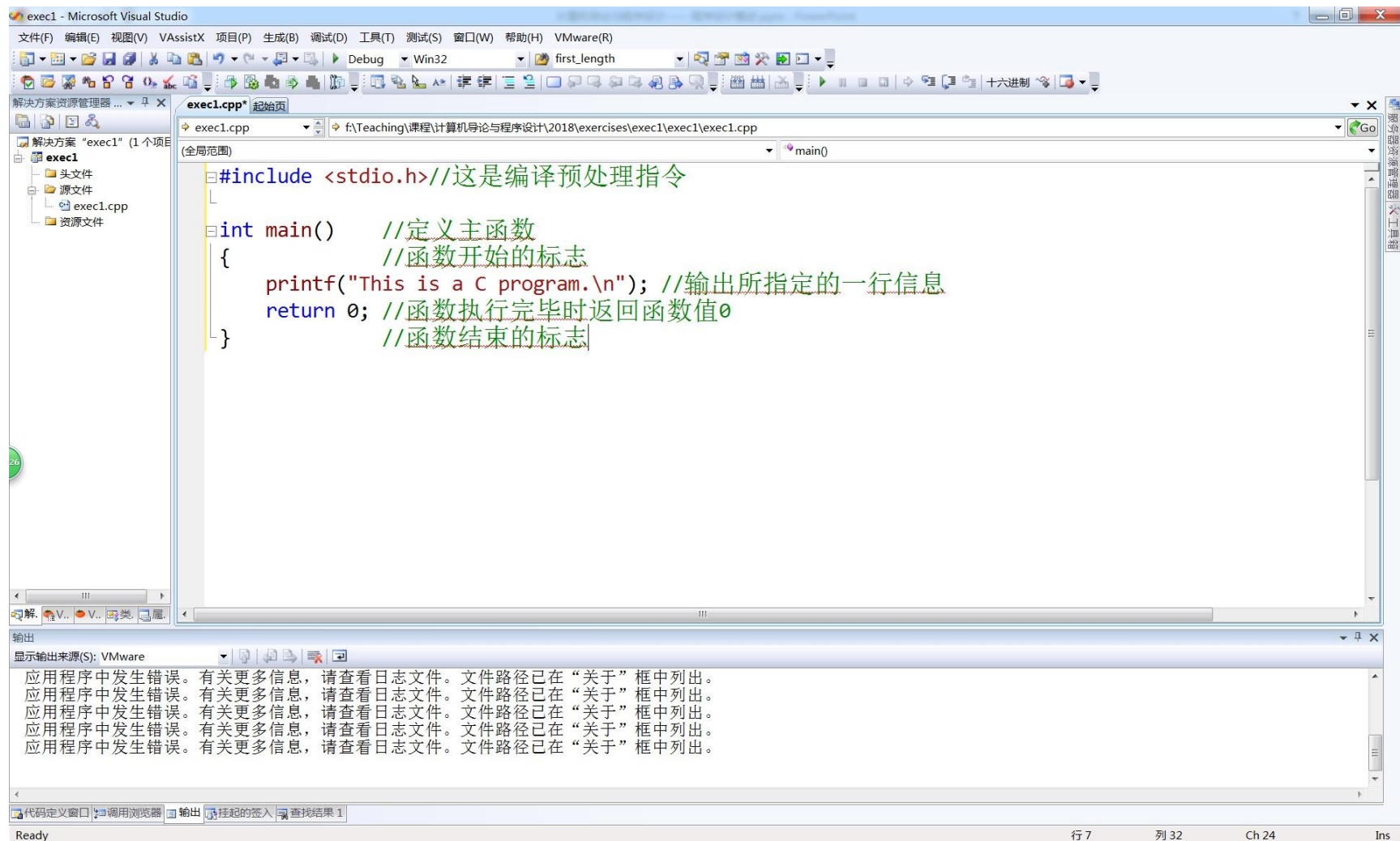
Visual Studio 编程-步骤5



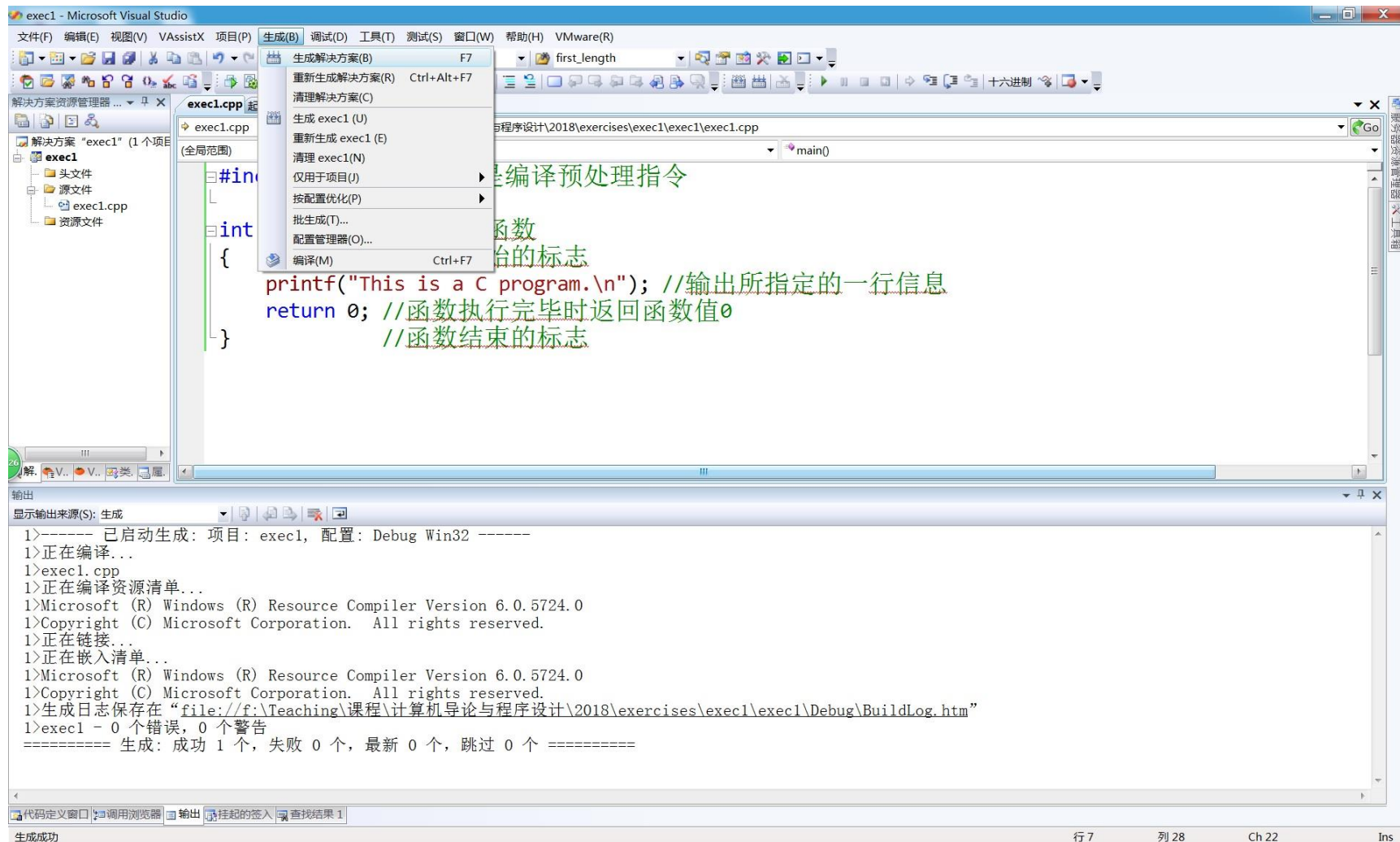
Visual Studio 编程-步骤6



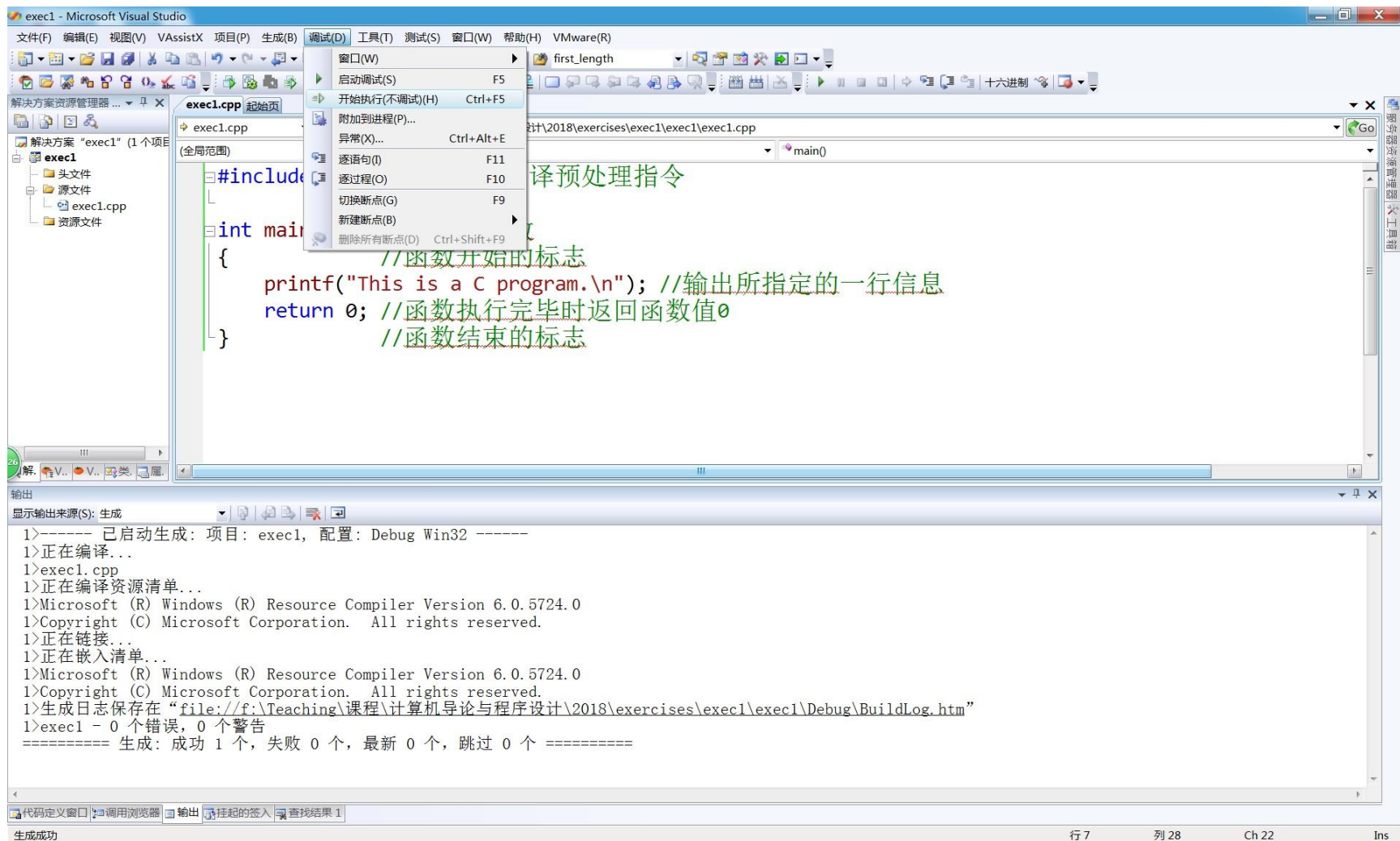
Visual Studio编程-步骤7



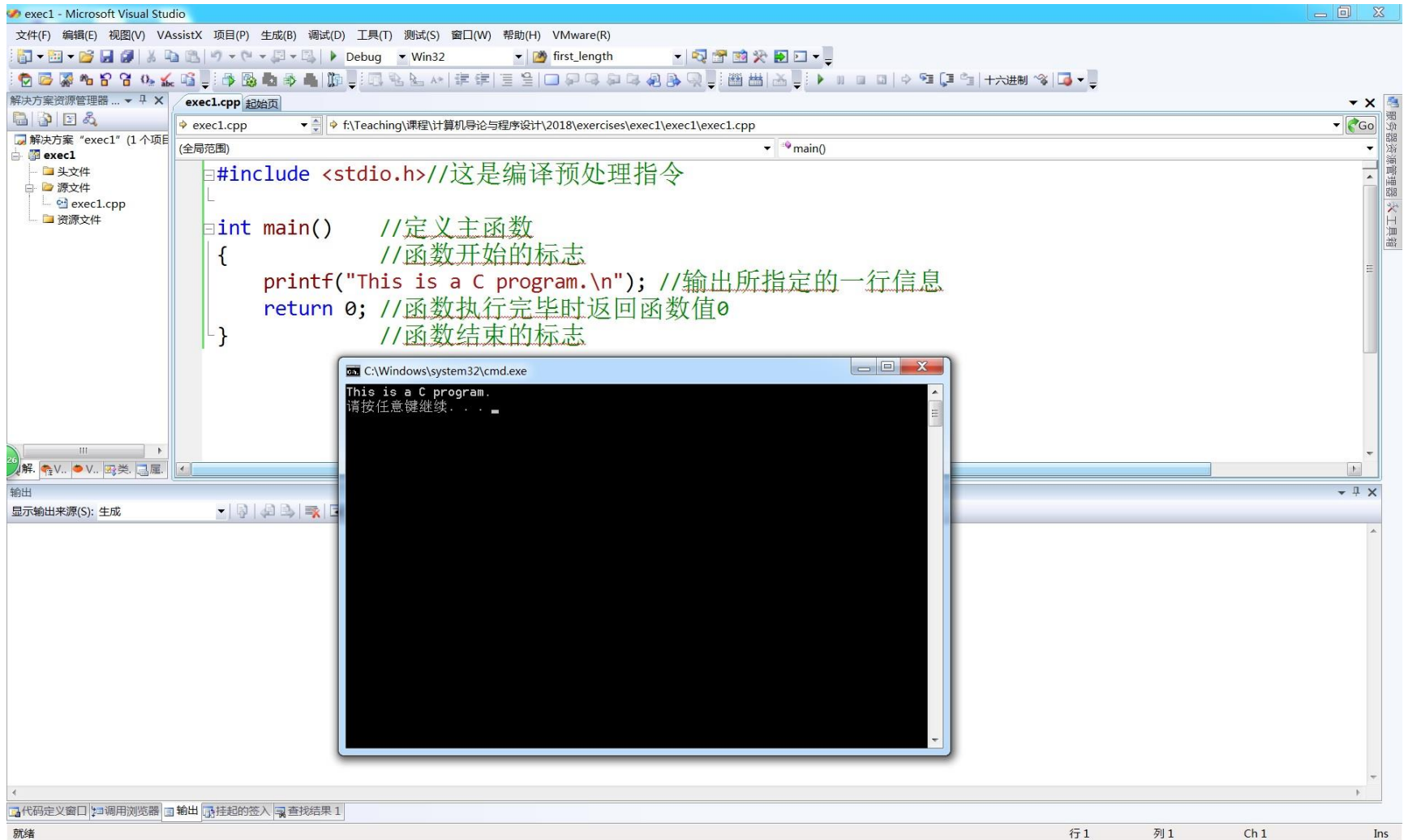
Visual Studio 编程-步骤8



Visual Studio 编程-步骤9



Visual Studio 编程-步骤10

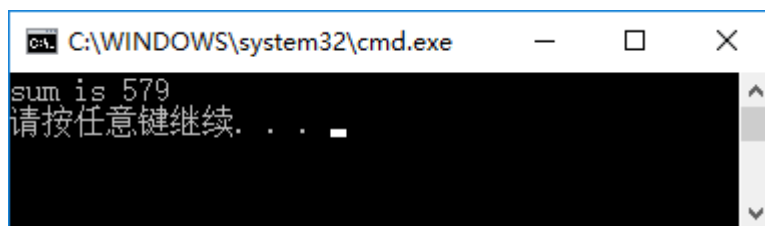


最简单的C语言程序

【例5.2】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“=”把相加的结果传送给sum。

```
1 #include <stdio.h> //这是编译预处理指令
2 int main( )         //定义主函数
3 {                   //函数开始
4     int a,b,sum;     //本行是程序的声明部分，定义a,b,sum为整型变量
5     a = 123;         //对变量a赋值
6     b = 456;         //对变量b赋值
7     sum = a + b;     //进行a+b的运算，并把结果存放在变量sum中
8     printf("sum is %d\n",sum); //输出结果
9     return 0;       //使函数返回值为0
10 }
```



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The output of the program is displayed as "sum is 579" followed by a prompt "请按任意键继续. . .".

最简单的C语言程序

```
1 #include <stdio.h> //这是编译预处理指令
2 int main( )         //定义主函数
3 {                   //函数开始
4     int a,b,sum;     //本行是程序的声明部分，定义a,b,sum为整型变量
5     a = 123;         //对变量a赋值
6     b = 456;         //对变量b赋值
7     sum = a + b;     //进行a+b的运算，并把结果存放在变量sum中
8     printf("sum is %d\n",sum); //输出结果           输出时用sum的值取代%d
9     return 0;       //使函数返回值为0
10 }                  //函数结束
```

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



程序分析

- `printf("sum is %d\n",sum);` printf函数圆括号内有两个参数。第一个参数是双引号中的内容sum is %d\n，它是输出格式字符串，作用是输出用户希望输出的字符和输出的格式。其中sum is是用户希望输出的字符，%d是指定的输出格式，d表示用“十进制整数”形式输出。圆括号内第二个参数sum表示要输出变量sum的值。在执行printf函数时，将sum变量的值(以十进制整数表示)取代双引号中的%d。

最简单的C语言程序

【例5.3】求两个整数中的较大者

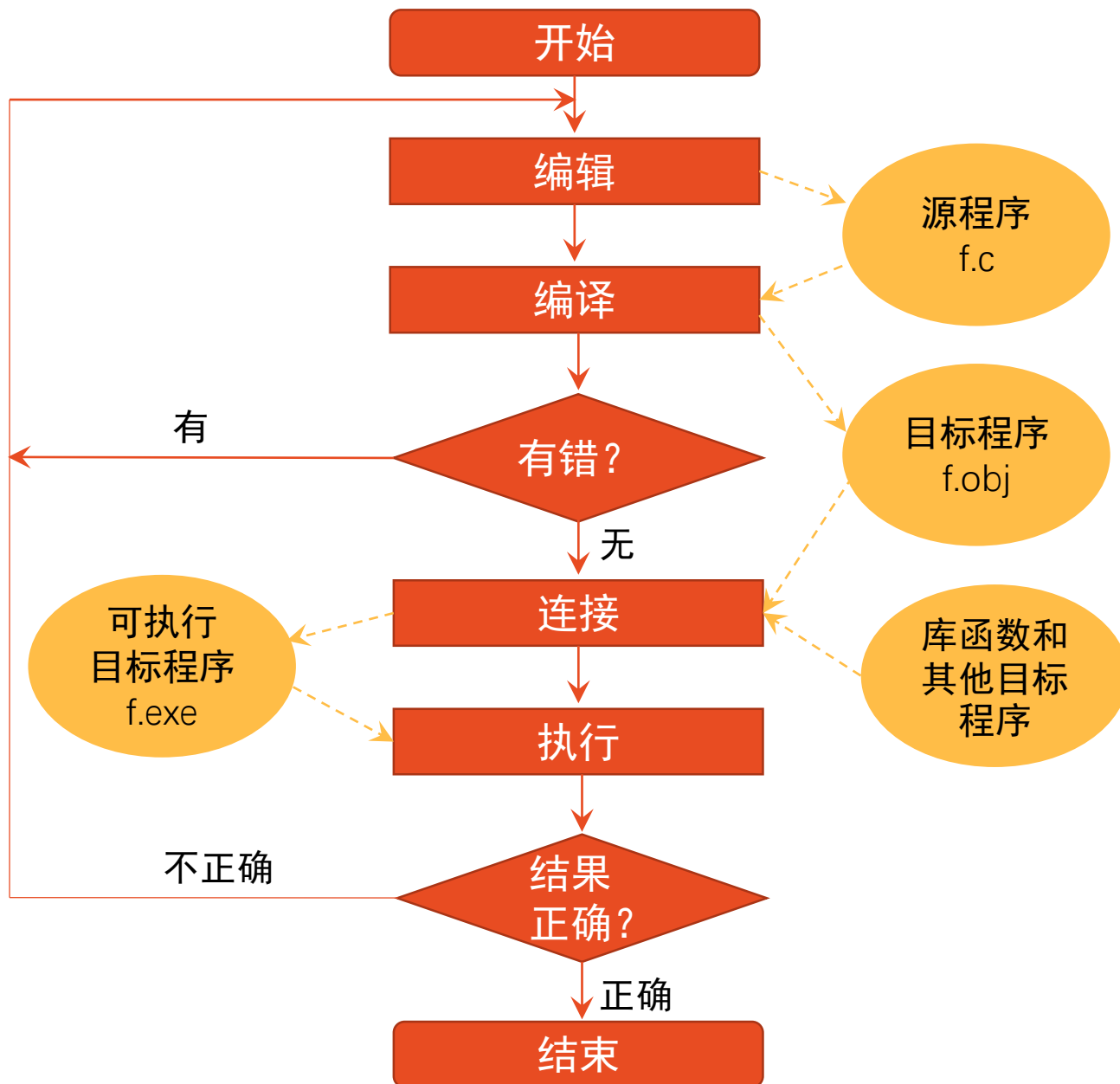
解题思路:设置3个变量, a和b用来存放两个整数, c用来存放两个数中的较大者。使用分支结构判断。

```
1 #include <stdio.h>
2 //主函数
3 int main()      //定义主函数
4 {
5     int a,b,c; //定义变量a, b, c
6     scanf("%d%d",&a,&b); //输入变量a和b的值
7     if(a > b)
8         c = a;
9     else
10        c = b;
11    printf("max = %d\n",c); //输出c的值
11    return 0; //返回函数值为0
11 } //主函数体结束
12
```

C语言程序的结构

- 一个程序由一个或多个源程序文件组成
- 函数是C程序的主要组成部分
 - 一个C语言程序是由一个或多个函数组成的，其中必须包含唯一的一个main函数
 - 程序中被调用的函数可以是系统提供的库函数，也可以是由用户根据需要自己编制设计的函数
- 程序总是从main函数开始执行
- 程序中的操作是由函数中的C语句完成的
- 在每个数据声明和语句的最后必须有一个分号
- C语言本身不提供输入输出语句，输入输出操作由函数完成
- 程序应当包含注释

运行C程序的步骤与方法



程序设计的任务

