

第二章

C语言的基本概念

本章要点

- **C程序的基本结构**
- **在计算机上运行C程序的方法**

简单的C语言程序

例1: pun.c

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

```
int main(void)
```

```
{
```

```
    printf ("To C, or not to C: that is the question.\n");
```

```
    return 0;
```

```
}
```

- C程序文件以.c为文件名后缀
- 以#开头的是指令。#include包含标准输入输出库stdio.h
- 每个C程序必须有一个主函数main(), int是函数返回类型
- 每个C语句以分号结束
- return 0: 表示程序结束时向操作系统返回值0

编译和链接

- 程序要执行，需要三个步骤：
- 预处理
 - 预先处理程序前面以 #开头的命令
- 编译
 - 编译器把程序翻译成机器指令（目标代码）
- 链接
 - 把编译后的目标代码和所需的其他附加代码（如库函数printf）整合在一起，产生可执行程序。
- 预处理器与编译器通常集成在一起

程序编写及运行过程

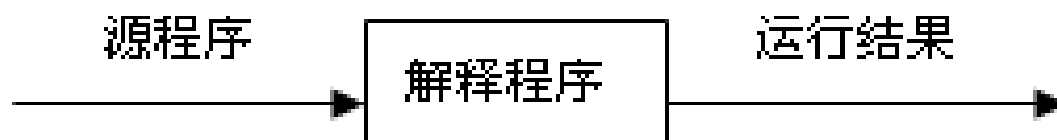


图 1-1-2 解释过程

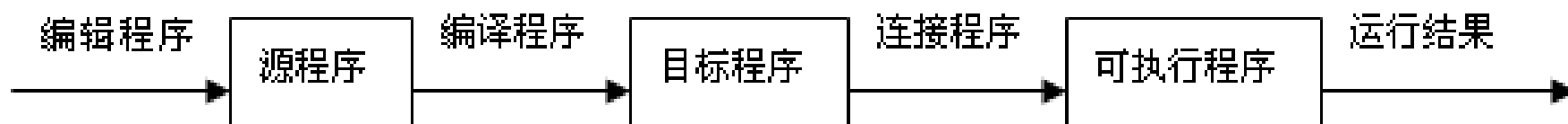


图 1-1-3 编译过程

温故而知新——低级、高级语言

- 低级语言
 - 面向机器（小孩）
 - 高级语言
 - 面向用户（大人）
-
- 机器语言
 - 汇编语言
 - C语言
 - C++、JAVA

温故而知新——C语言特点

- 高效
 - 贴近低级语言，地址访存
- 可移植性强
 - 标准C，编译器小
- 功能强大
 - 丰富的数据类型和运算符
- 灵活
 - 语法灵活，约束少。
- 标准库
 - 常用功能以库函数方式提供（成语）

温故而知新——编译和链接

- 程序要执行，需要三个步骤：
- 预处理
 - 预先处理程序前面以 #开头的命令
- 编译
 - 编译器把程序翻译成机器指令（目标代码）
- 链接
 - 把编译后的目标代码和所需的其他附加代码（如库函数printf）整合在一起，产生可执行程序。
- 预处理器与编译器通常集成在一起

集成开发环境IDE

- 集编辑、编译、链接、执行甚至调试为一体的软件包
- 如，visual C 6.0、2010，C-free；
- driond 4 C
- 同学们发掘：
 - Codevs
 - VSCode：配置VSCode+WSL开发环境
 - Dev-c++



答疑和谈论

- 直接@我
- 班群提问
 - 大家问题相通
 - 同学相互作答
 - 带动学习氛围
- 上机验证
- 百度/google

培养独立思考
和自学能力

示例程序演示

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

```
int main(void)
```

```
{
```

```
    printf ("To C, or not to C: that is the  
question.\n");
```

```
    return 0;
```

```
}
```

2.2 简单程序的一般形式

指令

```
int main(void)
```

```
{
```

语句

```
}
```

- { } 是函数开始和结束的标志, 不可省
- C程序至少包含三部分: 指令、函数和语句

指令

- 程序编译前由预处理器进行预先处理——预处理指令
 - **#include; #define**, 准备工作, 如指明库函数出处, 定义常量等
 - 类似处理缩写或专有名词, 类比淘米洗菜等准备工作
- 例如: `#include <stdio.h>`
 - **stdio.h**包含或定义**C**标准输入/输出库信息的头文件
 - **#include <stdio.h>**将标准输入/输出库信息包含到我们的程序中
- 通常占一行, 没有结尾符

函数

- 一系列组合在一起并赋予名字的语句
 - 用于构造代码块，类似过程，便于模块化
 - 对数学函数概念的借用，可计算数值（返回结果 **return**），也可不计算（无结果返回）
 - 类似文章的章节
- 两类：
 - 自编函数，库函数
- C程序可包含多个函数，但有且仅有一个 **main()**
 - **main()**位置可任意位置，程序总是从**main**函数开始执行
- 在程序终止时向操作系统返回状态码
 - **return 0;**

函数

```
int main()
{
    tmt=2; egg=2; salt=5g; soy=5ml;
    con=10g; oil=50ml;//变量声明
    clean tmt;//洗净番茄
    flay tmt;//去皮
    cut tmt;//切块;
    cut con;//葱花切片
    egg=egg+salt;//鸡蛋加少许盐
    mix egg;//打散
    hot oil;//烧热油;
    fry egg;
    .....
    .....
    return();//盛出;
}
```

```
int main()
{
    tmt=2; egg=2; salt=5g; soy=5ml; con=10g;
    oil=50ml;//变量声明
    deal_tmt(tmt);//处理番茄
    deal_egg(egg);//处理鸡蛋
    fry_egg(egg);
    fry_tmt(tmt);
    mix_fry(egg, tmt);
    return();//盛出;
}

deal_tmt(tmt)
{
    clean(tmt);
    flay(tmt);
    cut(tmt);
    return tmt;
}
```

封装

main()抓大放小

掌控整个程序脉络，具体细节交给其他函数

语句

- 程序要让计算机具体执行的操作
- 程序的基本组成单位
- 以分号结尾
- eg:
 - **printf(“hello!\n”);**
 - //函数调用语句，输出字符串“hello!”，“\n”换行
 - **y=3*x+5;**
 - //算术运算语句
 - **return 0;**
 - //返回语句；可返回也可不返回

C-Free(轻量C程序集成开发环境)

- 工程 (project)
 - 方便管理和设置
- 控制台程序 (console)
 - 没有独立窗口，一般在命令行运行。
- GCC (GNU Compiler Collection)
 - Linux环境下最流行的编译器。
- MinGW (Minimalist GNU on Windows)
 - 在Windows平台上模拟了Linux下GCC的开发环境
 - 收集了一系列免费的Windows 使用的头文件和库文件；
 - 整合了GNU的工具集，如经典gcc, g++, make等。

控制台程序

- 控制台程序（console）
 - 在32位windows环境中模拟16位DOS程序，在DOS窗口中执行。
 - 没有可视化界面，通过字符串来显示或者监控程序。常应用于测试、监控、计算等只关心数据、功能，不在乎界面。
- 没有独立窗口，一般在命令行运行。
- 输入输出通过标准IO进行，不象界面程序可以通过鼠标点击进行操作。
- 一般后台运行的程序可作为控制台应用程序。
- return后程序结束，自动关闭
 - 让计算机等待输入：
 - scanf();getchar();
 - **system（“pause”）；#include <stdlib.h>**

程序练习I（5分钟）

- 输出自己姓名、年龄
- 开发工具C-Free

练习（5分钟）

- 攻略：

- 1、打开C-Free
- 2、新建project
- 3、new *.cpp源程序文件
- 4、编写代码
- 5、编译源代码
- 6、构建程序
- 7、运行&调试

2.3 注释

- 对程序或代码进行注解，便于程序阅读和维护，对编译和运行不起作用。
- “//” 开头或 “/*” 和 “*/” 包含
- 可以用汉字或英文字符表示。

◦ 放油少许酱油；继续翻炒；/*翻炒时动作要轻柔，频率不能太高，否则西红柿太烂了反而不好吃*/

```
/* Name: pun.c  
   Purpose: Prints a bad pun.  
   Author: K. N. King */
```

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

```
int main()
```

```
{ printf("Hello Tom!"); //greeting  
}
```

程序
注解

语句
注解

语句
注解

2.4 变量和赋值

- 变量
 - 存放程序执行过程中会产生变化的数据，eg. $f(x)$, $egg =$
- 每个变量都有一个类型，eg.
 - `int n;` // 人数
 - `float t;` // 温度，浮点数，
 - eg, $t = 36.8 = 3.68 \times 10 = 0.368 \times 10^2$
- 不同类型有不同取值范围（以及存放方式），由类型的长度决定
 - `int` 两或四字节——16/32bit，最大32767

声明

- 声明（介绍）变量
 - 说明变量的类型和名字
 - 内存分配空间
 - 先声明，后使用
- 单独声明：
 - `int height;`
 - `float profit;`
- 合并声明：
 - `int height, length, width, volume;`
 - `float profit, loss;`

赋值

- 赋予变量值
 - **height = 8;**
 - **length = 12;**
 - **width = 10;**
 - **volume = height*length*width;**
 - **profit = 2150.48(f);**

显示变量值

- `printf()` 格式化输出函数，f表示format
`printf("Height: %d\n", height);`
`printf("Profit: $%.2f\n", profit);`
`printf("%d\n", height * length * width);`
- 参数两部分
 - “.....”: 格式
 - , 之后: 数据
- 占位符%d、%f指示变量的显示位置
 - **d**: 输出变量是整数
 - **f**: 输出变量是浮点数

计算箱子空间重量

dweight.c

```
/* Computes the dimensional weight of a 12" x 10" x 8" box
 */
#include <stdio.h>
int main(void){
    int height, length, width, volume, weight;
    height = 8;
    length = 12;
    width = 10;
    volume = height * length * width;
    weight = (volume + 165) / 166;

    printf("volume: %d\n", length * width * height);
    显示表达式的值

    return 0;
}
```

初始化

- 声明变量的同时赋值
 - 初始值，eg，初生体重，身长
 - `int height = 8;`
 - `int height = 8, length= 12, width=30;`
- 未初始化化变量，值不可预期
 - eg. 黑板（上堂课未擦除）
 - `int weight;`
 - `printf(“my weight is %d\n”, weight);`

2.5 读入输入

- scanf () 格式化输入函数，与printf对应
- 用格式串指定输入或输出数据的格式。
 - **scanf("%d", &i);**
 - **/* reads an integer; stores into i */**
- &取址符，表示取变量i的地址，scanf函数读入数据必须在变量前加&。
- 读入浮点数
 - **scanf("%f", &x);**
- 尽量使用默认格式串（只有占位符）
 - **“%f”，“%d%d%f”**

计算箱子空间重量（改进版）

dweight2.c

```
/* Computes the dimensional weight of a box from input
   provided by the user */
#include <stdio.h>
int main(void)
{
    int height, length, width, volume, weight;
    printf("Enter height of box: ");
    scanf("%d", &height);
    printf("Enter length of box: ");
    printf("Enter height,length,and width:\n");
    scanf("%d%d%d", &height, &length, &width);

    volume = height * length * width;
    weight = (volume + 165) / 166;
    printf("Volume (cubic inches): %d\n", volume);
    printf("Dimensional weight (pounds): %d\n", weight);
    return 0;
}
```

2.6 定义常量

- 常量：程序执行过程中不变的量，比如圆周率，货币汇率
- 常量定义通常通过宏定义来完成
 - 一种规则或模式，或称语法替换，指定一个串替换另一个串
 - 格式：**#define A B**
 - **#define PI 3.1415926**
 - 宏名（常量名）通常大写
- 宏定义是预处理功能的一种，预处理时宏将替换为其表示的值（常量）

```
area=r*r*PI;  
perimeter=2*r*PI;
```

预处理

```
area=r*r*3.1415926;  
perimeter=2*r*3.1415926;
```

思考宏定义的意义？

程序练习2-输入半径计算圆面积

```
/*calculate the area and perimeter of a circle*/  
#include <stdio.h>  
int main(void)  
{  
    变量申明 (.....) ;  
    提示;  
    读入半径;  
    计算周长;  
    计算面积;  
    输出结果;  
    return 0;  
}
```


错误解析-输入半径计算圆面积

计算圆面积.c

```
#include <stdio.h>
#define PI3.14;
int mian(void)
{
    int radius, area;
    printf("input the radius: ")
    scanf("%d", radius);
    area = PI * radius * radius;
    perimeter = 2*PI*radius;
    printf("the area is %f\n the perimeter is %f", area,
    perimeter);
}

return 0;
```


错误解析-输入半径计算圆面积

计算圆面积.c

文件名

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

分隔，分号

```
#define PI3.14;
```

```
int mian(void)
```

函数名

```
{
```

缺分号

```
    int radius, area;
```

```
    printf("input the radius: ")
```

取址&

```
    scanf("%d", radius);
```

```
    area = PI * radius * radius;
```

先声明，后使用

```
    perimeter = 2*PI*radius;
```

```
    printf("the area is %f\n the perimeter is %f", area,  
    perimeter);
```

类型匹配

```
}
```

```
    return 0;
```

返回语句；

2.7 标识符

- 变量、函数、宏和其他实体的名字统称标识符
- 由字母、数字和下划线构成，必须以字母或下划线开头，大小写敏感
- eg:
 - **int h1, h2, H1, H2, _w;**
 - **int max(int num1, num2);**
 - **PI**
 - **92h, tom-kate; //???**

关键字

- 对C编译器有特殊意义的字符串
 - **如int, float, enum, if, while等**
- 一般小写
- 不能用作标识符

C关键字

- The following **keywords** can't be used as identifiers:

auto	enum	restrict*	unsigned
break	extern	return	void
case	float	short	volatile
char	for	signed	while
const	goto	sizeof	_Bool*
continue	if	static	_Complex*
default	inline*	struct	_Imaginary*
do	int	switch	
double	long	typedef	
else	register	union	

*C99 only

2.8 C程序书写规范

- C程序书写格式自由
 - **C程序没有行号**
 - 一行内可以写几个语句，除预指令外全部程序可一行
 - 一个语句可以分写在多行上

C程序布局

- C程序由一连串记号构成：
 - **Identifiers:** 标识符
 - **Keywords:** 关键字
 - **Operators:** 运算符
 - **Punctuation:** 标点
 - **Constants:** 常量
 - **String literals:** 字符串字面量
 - “hello world!”

C程序布局

- 语句: `printf("Height: %d\n", height);`
 - 包含七个记号:
 - `printf` Identifier
 - `(` Punctuation
 - `"Height: %d\n"` String literal
 - `,` Punctuation
 - `height` Identifier
 - `)` Punctuation
 - `;` Punctuation

记号间不一定需要空白（空格、空行、制表符）间隔

不讲布局的程序

```
/* Converts a Fahrenheit temperature to  
Celsius */  
#include <stdio.h>  
#define FREEZING_PT 32.0f  
#define SCALE_FACTOR (5.0f/9.0f)  
int main(void){float  
    fahrenheit,celsius;printf(  
"Enter Fahrenheit temperature:  
");scanf("%f", &fahrenheit);  
celsius=(fahrenheit-  
    FREEZING_PT)*SCALE_FACTOR;  
printf("Celsius equivalent: %.1f\n",  
    celsius);return 0;}
```

节约空间，牺牲阅读性

如何布局

- C允许在记号间插入任意数量的空白
 - 空格、制表符、空行
- 添加空格和空行使程序便于阅读和理解
 - 记号间空格使其容易区分；
 - 空行把程序划分成逻辑单元；
- 缩进助于识别程序的嵌套层次，如书目录
- 一条语句一行，长语句可分开放在任意多行内

布局好的程序

```
/* Converts a Fahrenheit temperature to Celsius */
```

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

```
#define FREEZING_PT 32.0f
```

```
#define SCALE_FACTOR (5.0f/9.0f)
```

```
int main(void)
```

```
{
```

```
    float fahrenheit, celsius;
```

```
    printf("Enter Fahrenheit temperature: ");
```

```
    scanf("%f", &fahrenheit);
```

```
    celsius=(fahrenheit-FREEZING_PT)*SCALE_FACTOR;
```

```
    printf("Celsius equivalent: %.1f\n", celsius);
```

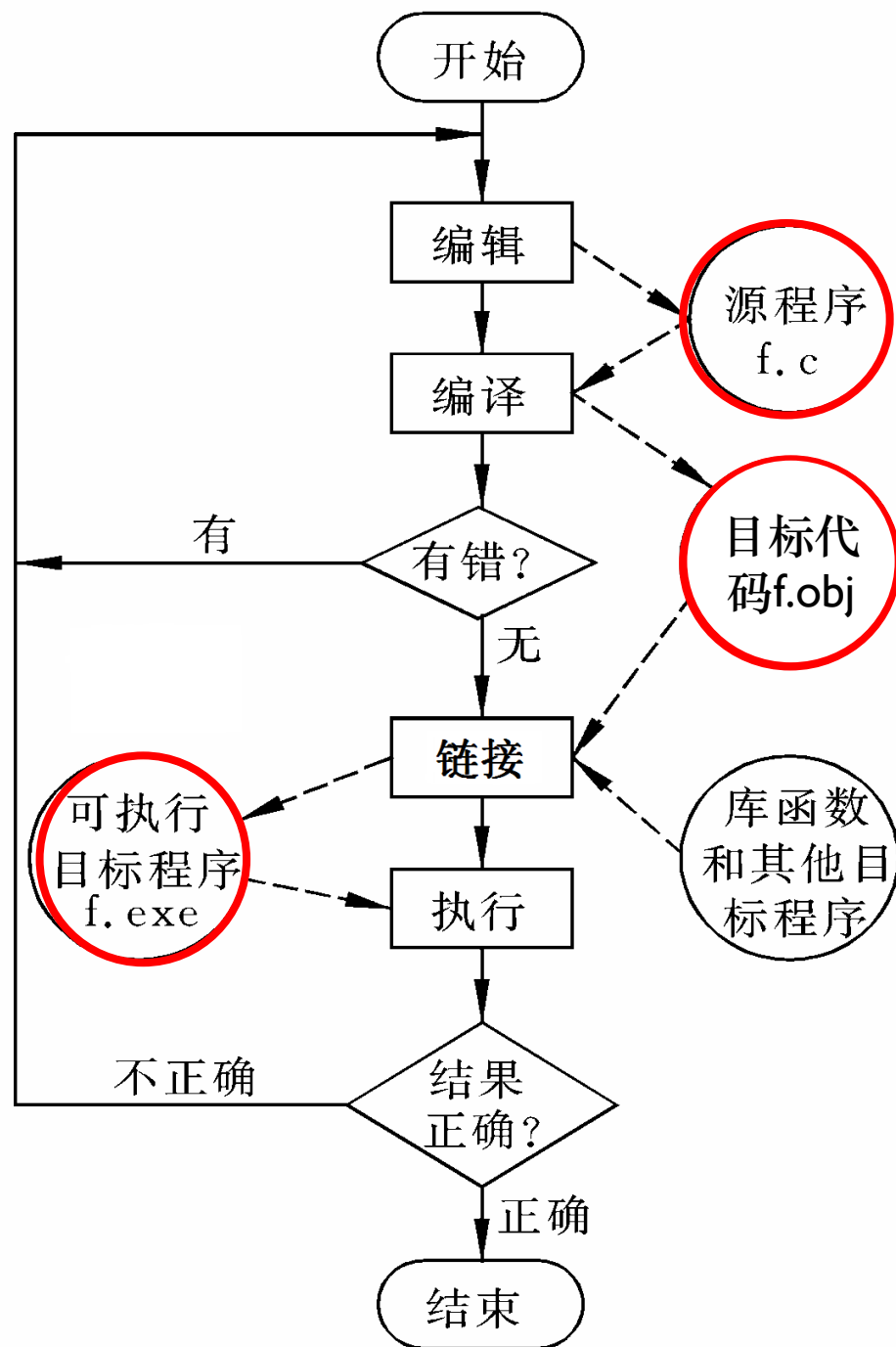
```
    return 0;
```

```
}
```

2.9 运行 C 程序的步骤和方法

一、运行 C 程序的步骤

- 上机输入与编辑源程序
- 预处理：预处理器执行以#开头的指令
- 对源程序进行编译
- 与库函数链接
- 运行目标程序



```
/*hello.c*/
```

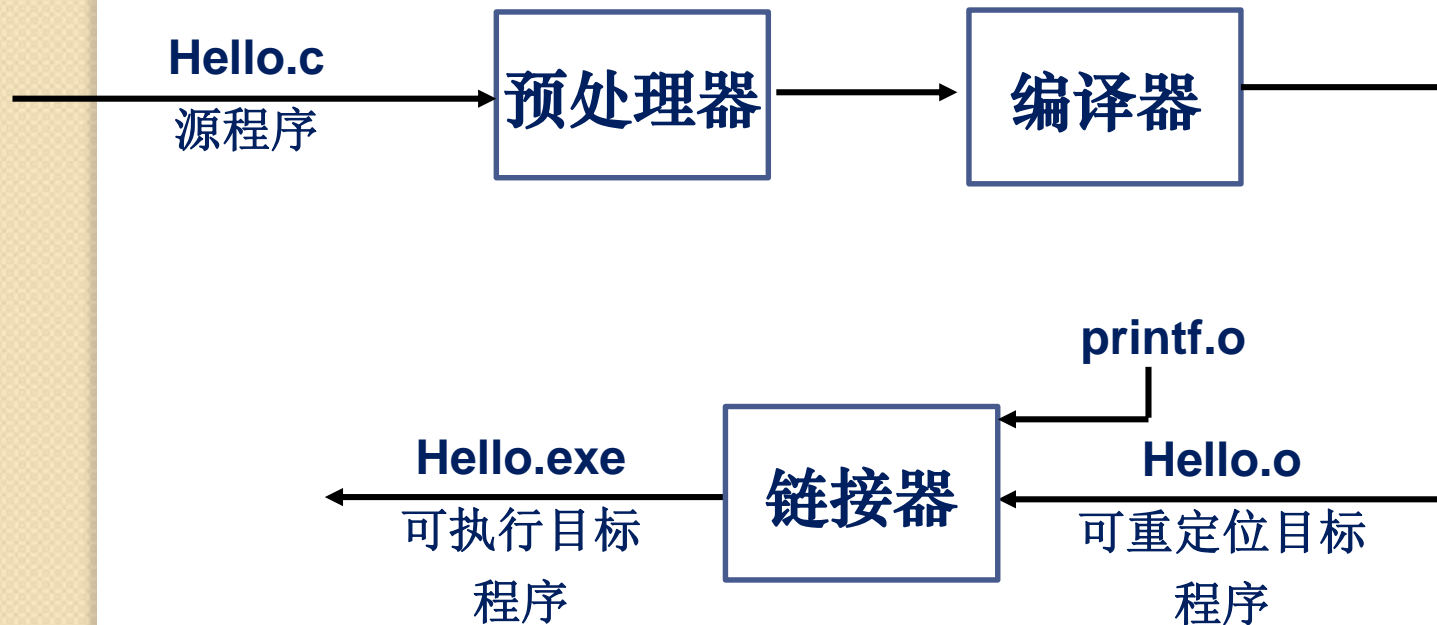
```
#include <stdio.h>
```

```
void main( )
```

```
{
```

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

```
}
```



程序练习2——试卷统分

信息安全理论与技术试卷_陈伟....

文件 开始 插入 页面布 引用 邮件 审阅 视图 设计 布局

电子科技大学在职攻读硕士学位研究生试卷

..... (考试时间:至....., 共2小时) ..

课程名称:信息安全理论与技术..... 学时: 40..... 学分: 2.....

考试方式:闭卷..... 教师:陈伟..... 考核日期: 2012年.....月.....日 ..

大题	一	二	三	四	五	六	七	八	九	十	总成绩
成绩	24	18	31	19							??

一、选择 (30分, 每题2分) ..

1. 数据加密标准DES和.....的加密算法是.....

页面: 1/3 60%

程序练习2——试卷统分

/*提示用户输入四道大题分数，并求和*/

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

```
int main(void)
```

```
{
```

```
    int score1, score2, score3, score4, sum; /*变量声明*/
```

```
    printf("Enter four scores.\n");
```

```
scanf ("%d%d%d%d", &score1, &score2, &score3, &score4) ;
```

```
    sum = score1 + score2 + score3 + score4;
```

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

```
    /*输出结果*/
```

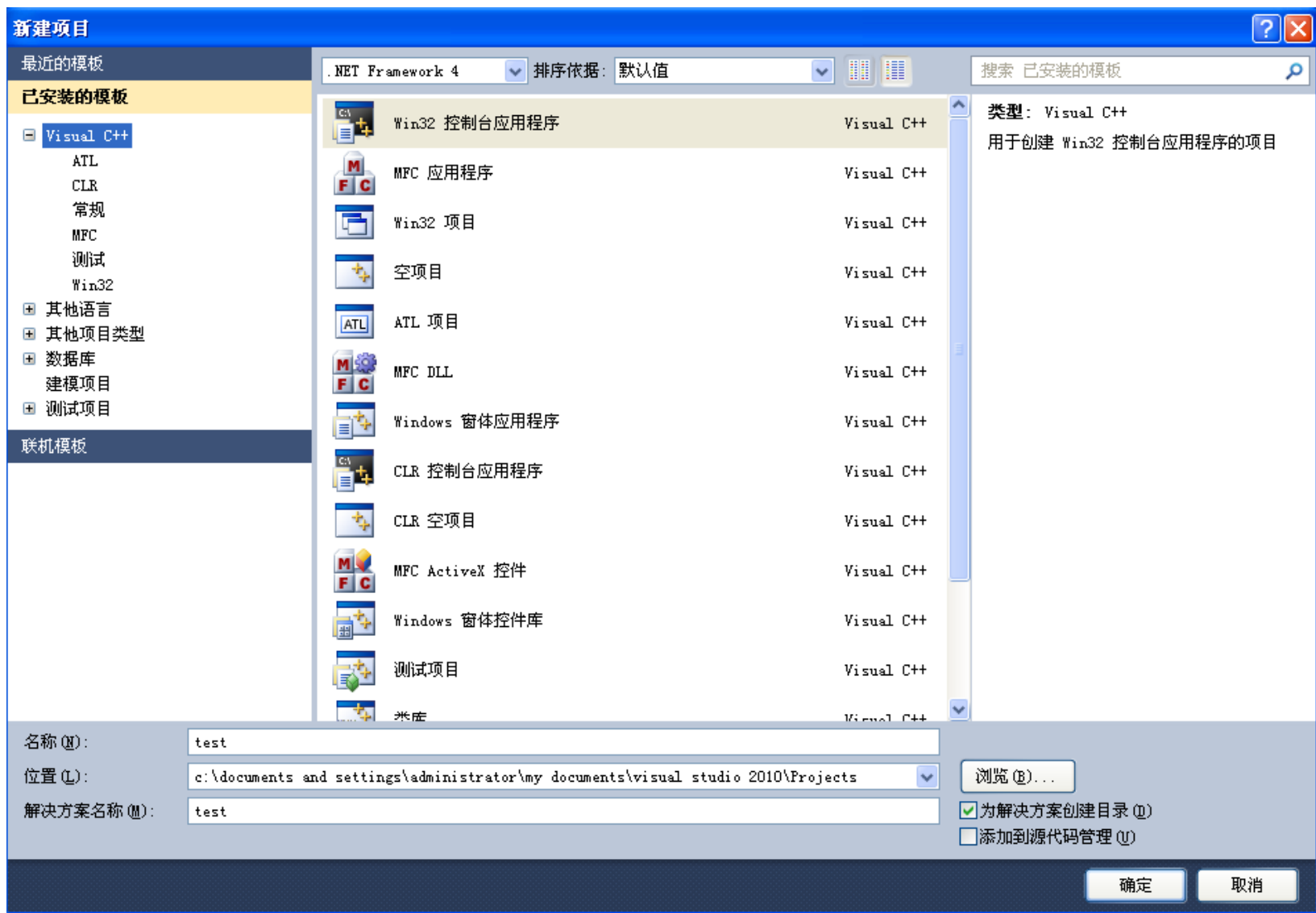
```
}
```

程序问题？

例： Visual Studio 2010

1、文件-新建-项目。

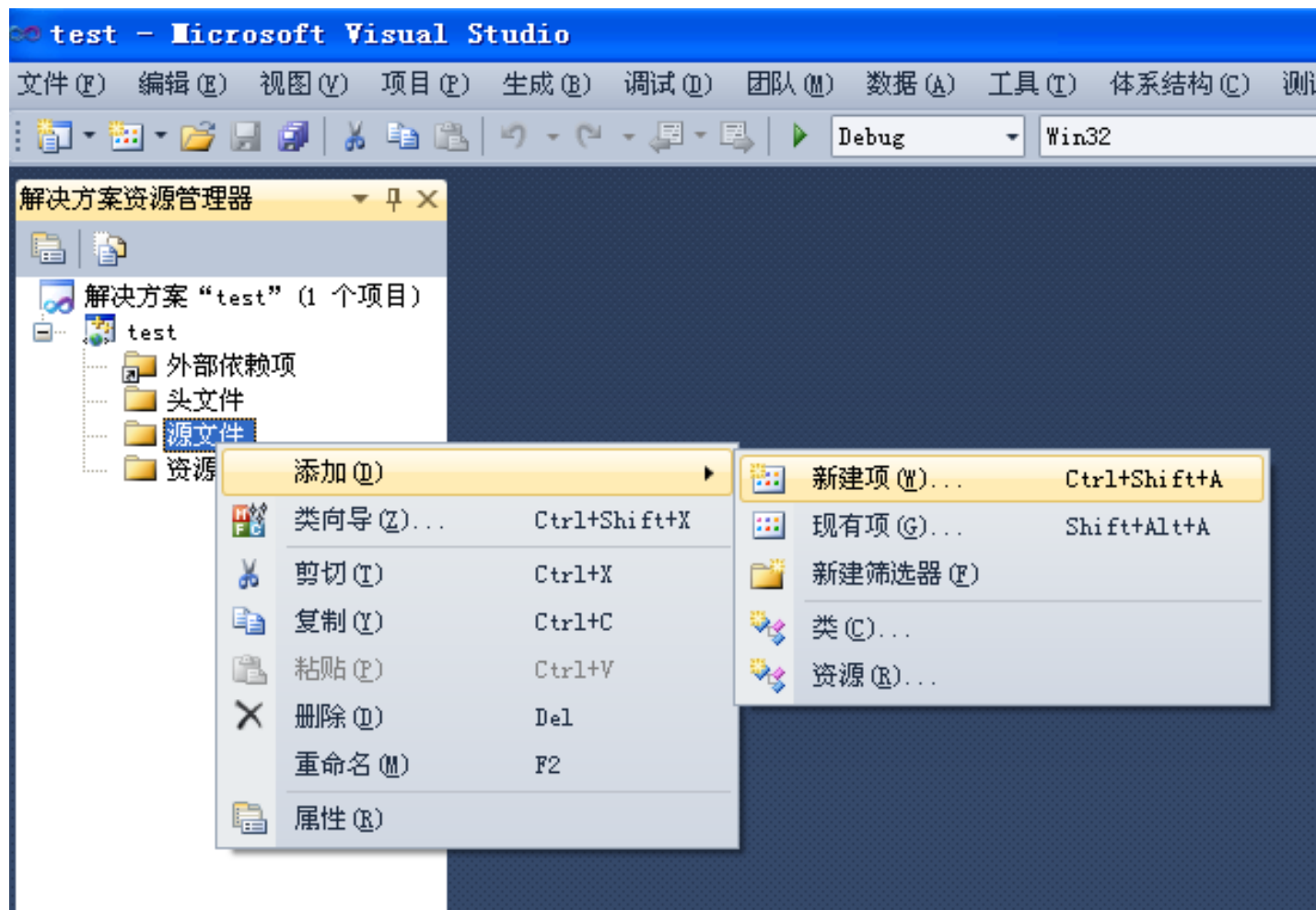
选择“Win32 控制台应用程序”，输入“名称”，点“确定”。



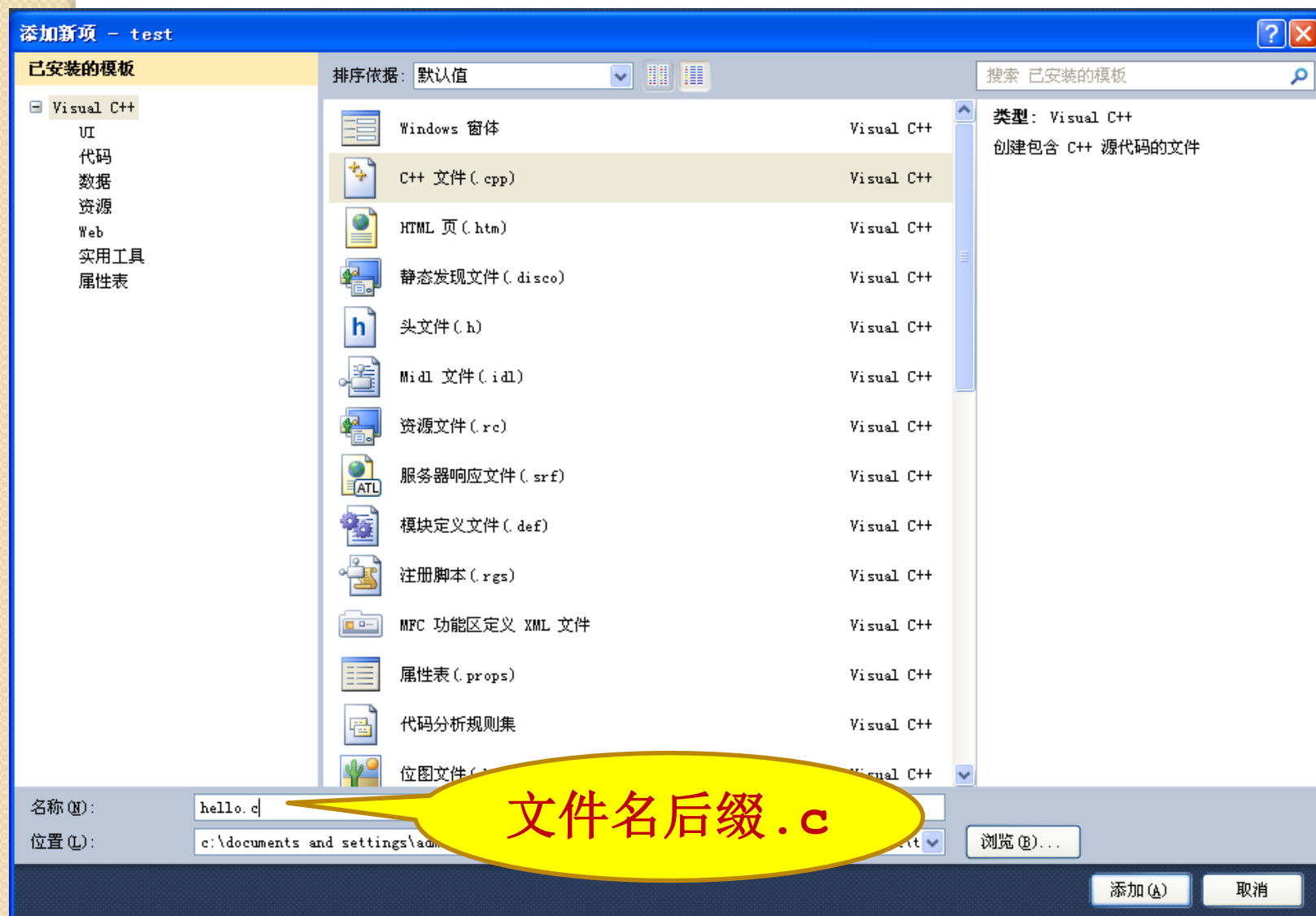
2、点“下一步”，在下面窗口选择“空项目”，点“完成”



3、点鼠标右键， 添加-新建项



4、在下面窗口选择“C++文件(.cpp)”，名称输入“hello.c”，点“添加”。



5、在hello.c中输入源程序，点“保存”，

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

```
void main( )
```

```
{
```

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

```
    system("pause"); //增加该语句使字符界面可以暂时停留
```

```
}
```

6、点菜单：调试-启动非调试（或ctrl+F5），运行程序。

作业

- lcoding实验 I