

1

编译、测试 和调试

实验课程



廈門大學
XIAMEN UNIVERSITY



信息学院
(国家示范性软件学院)
School of Informatics

黃 煒
博士, 副教授
Dr. Wei Huang

知识框架

- 基础知识
- 程序的开发
- 程序的编译
- 程序的调试
- 程序的测试

内容纲要

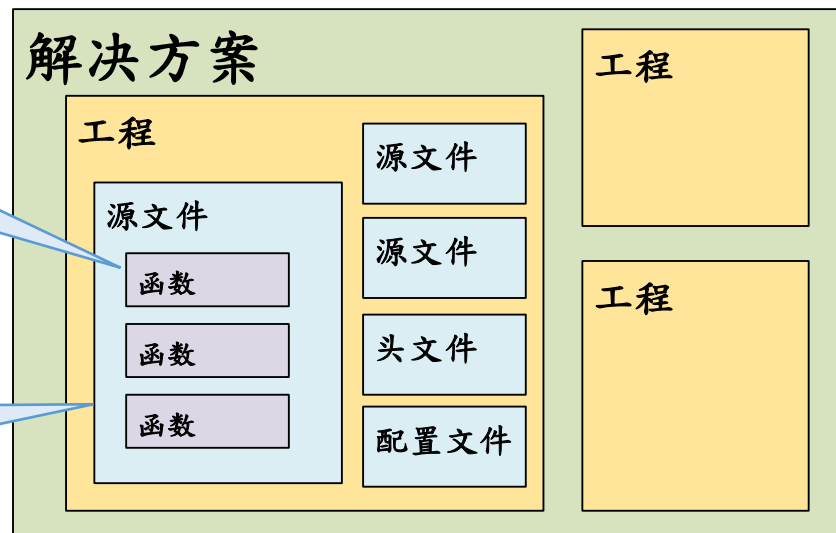
1	基础知识
2	程序的开发
3	程序的编译
4	程序的调试
5	程序的测试

基础知识

- 软件由模块组成，模块由源代码文件组成
 - 在一些开发环境里，称：解决方案、工程
- 一个源文件由多个函数组成

对初学者而言，先学会单个函数组成单个文件，组成单个工程，组成单个解决方案的开发过程。

对初学者应该认识到，一个复杂的软件通常由多位队员共同完成，书写程序应考虑队友的感受。



基础知识

- 程序员的工作日常

- 先设计算法，再将它实现出来
- 从代码服务器下载最新的代码到本地，再添加自己的代码
- 编译并运行程序，如果出现编译错误，应及时改正
- 设计测试用的输入输出，测试程序
- 如果出现结果错误，应调试程序，修正错误
- 重新编译程序，重复上述工作直到结果正确
- 将代码提交上传到远程代码服务器

基础知识

- 编译、测试和调试的重要性

- 程序员提交了存在编译错误的代码，将使得整个团队在更新代码后，无法顺利编译该解决方案
- 程序员提交了存在明显错误的代码，将使得依赖该模块的其它模块结果出错
- 程序员提交了存在错误的代码，队友将报告一个错误
- 但前两种情况，会激起公愤的

内容纲要

1	基础知识
2	程序的开发
3	程序的编译
4	程序的调试
5	程序的测试

集成开发环境

- 集成开发环境
 - IDE : Integrated Development Environment
 - IDE是用于提供程序开发环境的应用程序。
- IDE是一体化的开发软件服务套
 - 代码编辑器：代码编写功能、分析功能
 - 编译器：代码编译功能
 - 调试器：调试功能等
 - 图形用户界面等工具

常用的集成开发环境

- Windows操作系统：

-  Microsoft Visual Studio

-  DevC++

-  Eclipse

- MacOS操作系统

- Xcode

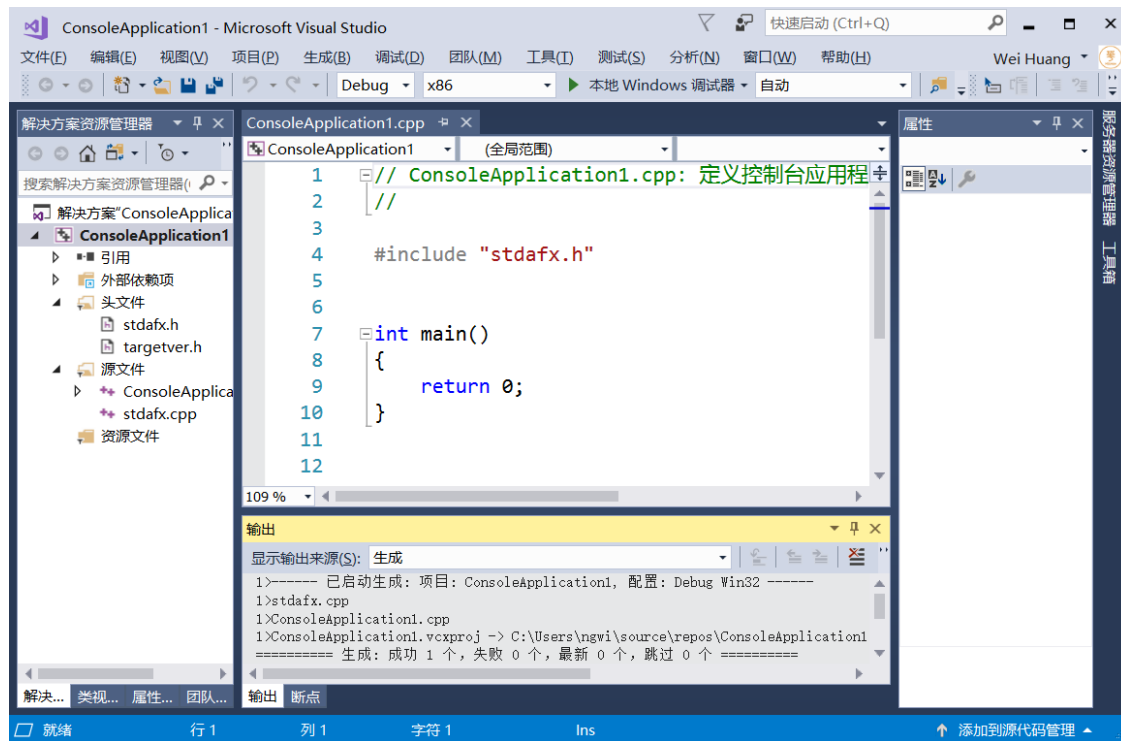
- Linux操作系统

- Qt ; Eclipse等



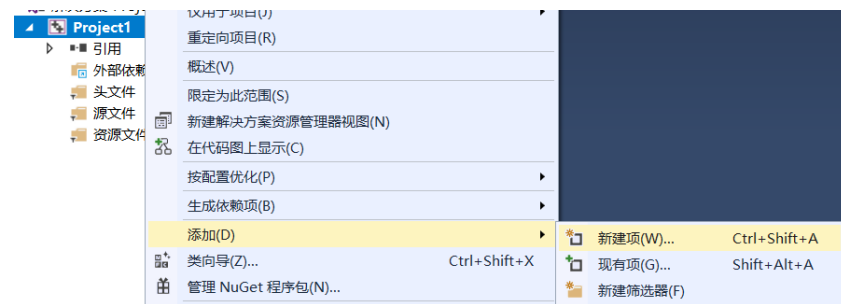
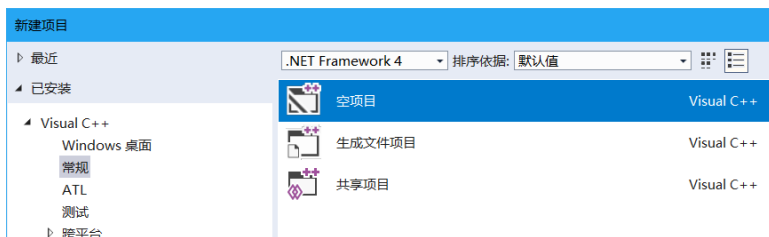
集成开发环境：Visual Studio

- 性质：免费（社区版）；付费（企业版）
- 优点：容易使用
- 缺点：占空间



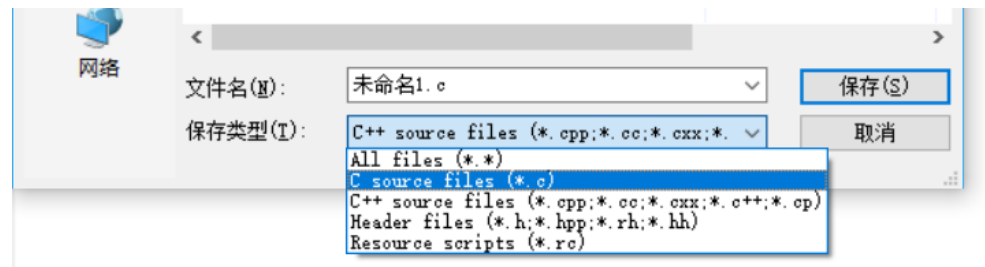
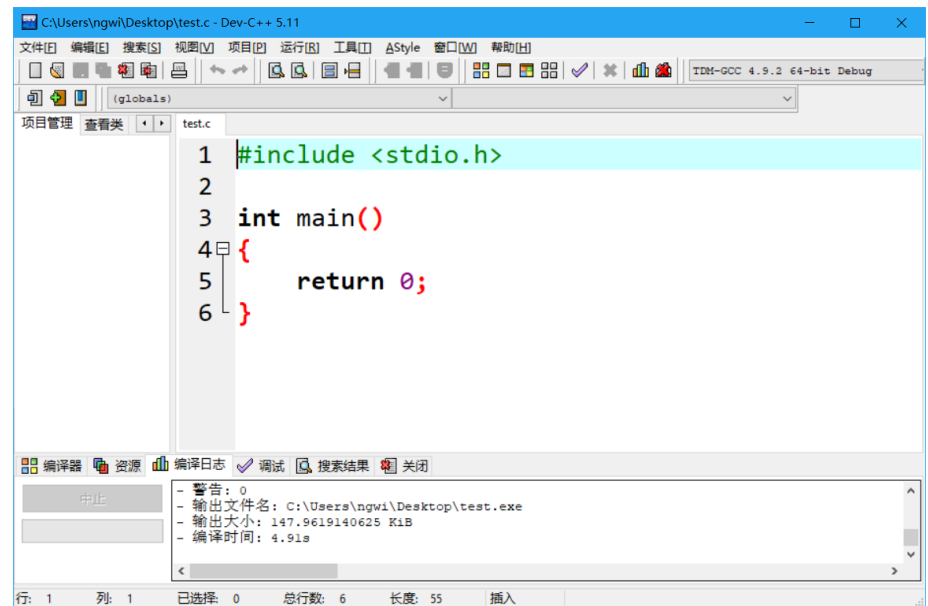
集成开发环境：Visual Studio

- 安装（请在网上搜索相应教程）
- 启动（Windows 10为例）
 - Windows键，输入vi，双击正确的图标
- 新建C语言项目
 - 文件/新建空项目
 - 添加，新建项，文件名改成c



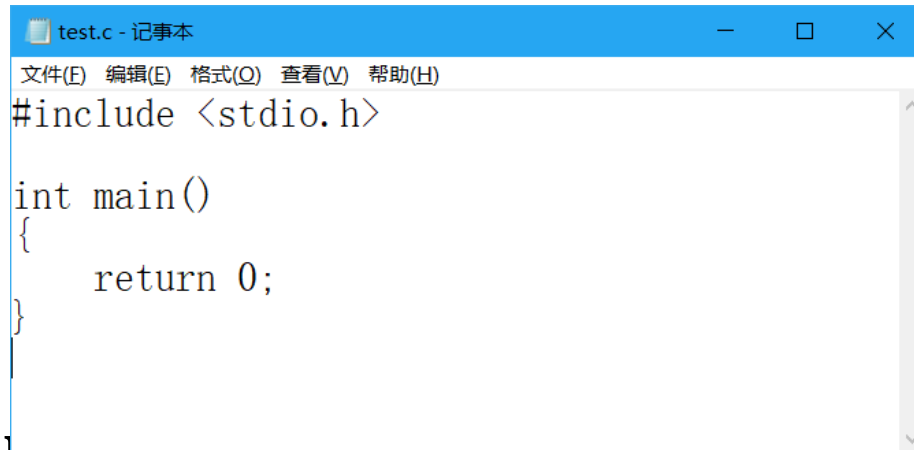
集成开发环境：DevC++

- 性质：免费（社区版）
- 优点：绿色版，支持GCC
- 缺点：不方便调试
- 操作
 - 新建（Ctrl+N）
 - 书写程序
 - 保存（Ctrl+S）
 - 文件类型选择C



非集成开发环境

- 记事本



```
test.c - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)
#include <stdio.h>

int main()
{
    return 0;
}
```

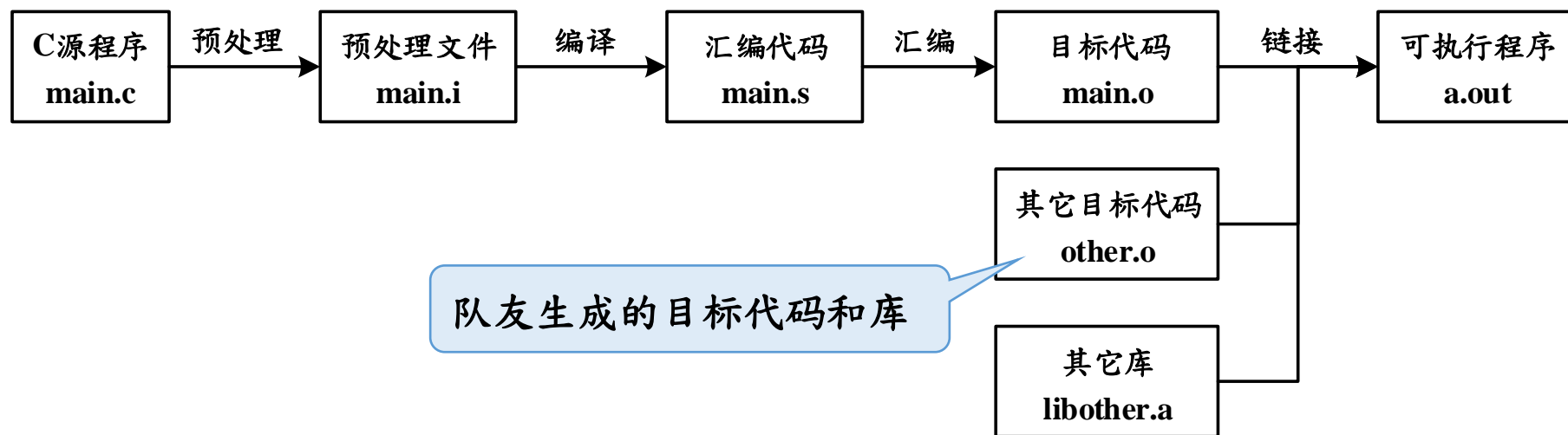
- GEdit (Linux版)

内容纲要

1	基础知识
2	程序的开发
3	程序的编译
4	程序的调试
5	程序的测试

编译程序

- 编译（ Compile ）：利用编译程序从源语言编写的源程序产生目标程序的过程。
- 编译器（ Compiler ）：执行编译功能的程序
- 生成一个C程序可以分为四阶段



常用的编译环境

- Visual Studio C 编译器：对C标准遵循不严格
- GCC编译器：遵循C标准
 - Windows系统：MinGW，Cygwin
 - Linux系统：GCC
- 不同的编译器对C标准未规定的部分处理不一样

编译程序

- 编译程序的方式

- 图形用户界面

- 图形用户界面程序调用命令行，得到编译错误信息后以图形用户界面形式显示出来。

- 命令行

- 直接查阅命令行输出

编译程序：Visual Studio

- 使用 Visual C++ 快捷键模式

- 菜单：【工具】/【导入导出设置】，单击【重置所有设置】，选择【Visual C++】，单击【完成】。

- 生成解决方案并运行

- 菜单：【调试】/【开始执行不调试】（快捷键：Shift+F5）

- 退出：按任意键退出

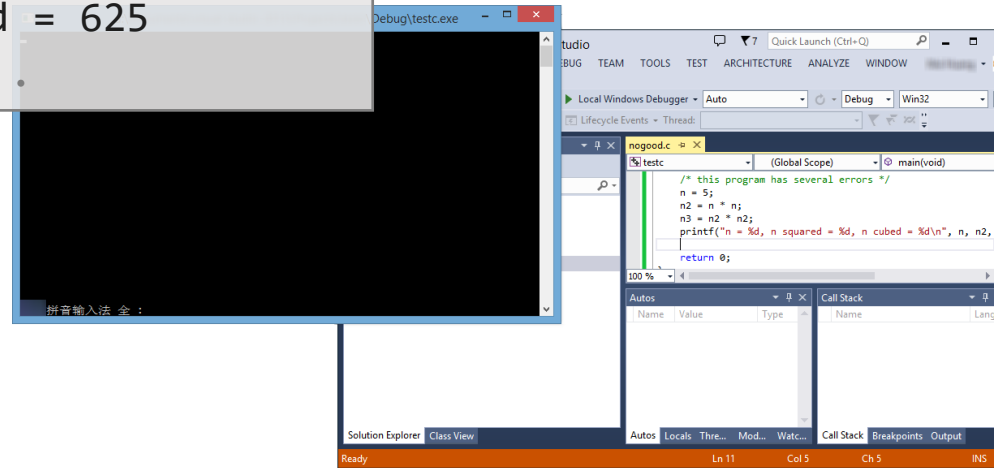


编译程序：Visual Studio

- 调试运行（Debug）：快捷键F5
 - 调试运行后看见控制台窗口一闪而过，因为运行太快了
 - 方法：在main()函数结束return 0;前，添加暂停命令

```
system("pause");
```

```
n = 5, n squared = 25, n cubed = 625  
Press any key to continue . . .
```



编译错误

- 编译错误

- 编译器层面发现的错误

- 编译错误往往是语法知识不扎实或不够细致引起的

- 括号不匹配，大小写错误

- 优秀的程序员能熟练书写无编译错误的程序

- 减少排查错误的时间浪费

- 程序员在提交程序前应确保无编译错误

- 应先自行编译再提交，避免提交了存在编译错误的程序导致整个团队的工程无法编译

编译错误：Visual Studio

• Visual Studio 2017 支持中文的错误信息

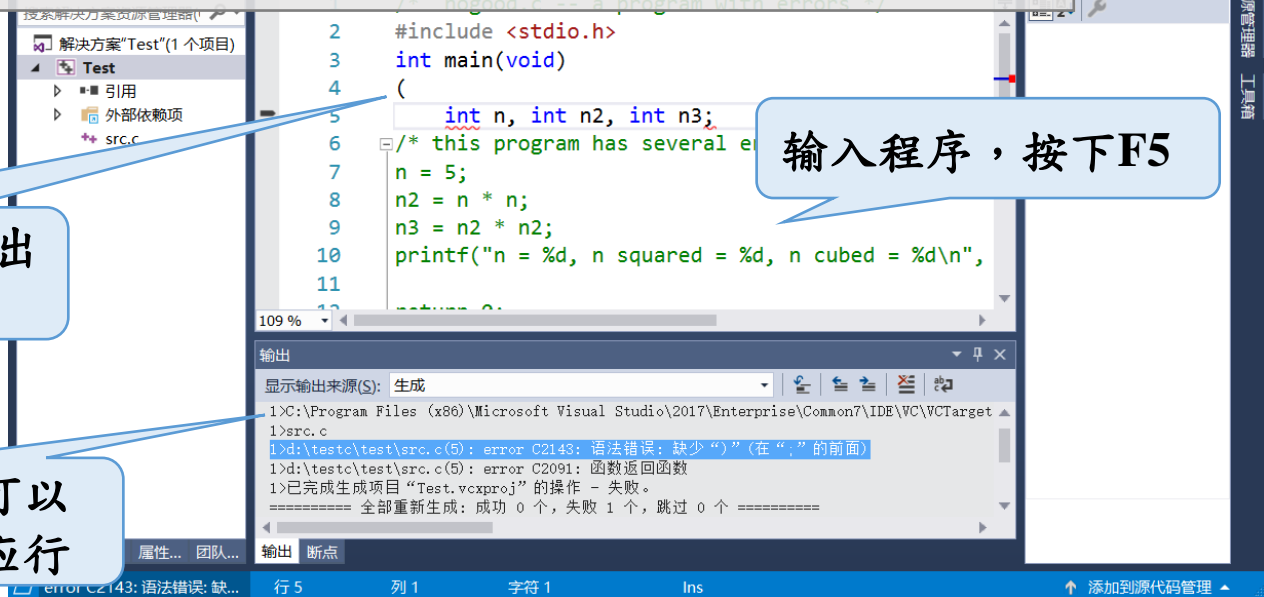
```
1>----- 已启动生成: 项目: Test, 配置: Debug Win32 -----
1>src.c
1>d:\testc\test\src.c(5): error C2143: 语法错误: 缺少 ")" (在 "/*" 的前面)
1>d:\testc\test\src.c(5): error C2091: 函数返回函数
1>已完成生成项目 "Test.vcxproj" 的操作 - 失败。
===== 生成: 成功 0 个, 失败 1 个, 最新 0 个, 跳过 0 个 =====
```

错误信息包括文件名，行号，
错误代码，错误提示。

前一行的错误导致后一行出
错（编译器无法回头）

双击错误信息可以
快速定位到相应行

输入程序，按下F5



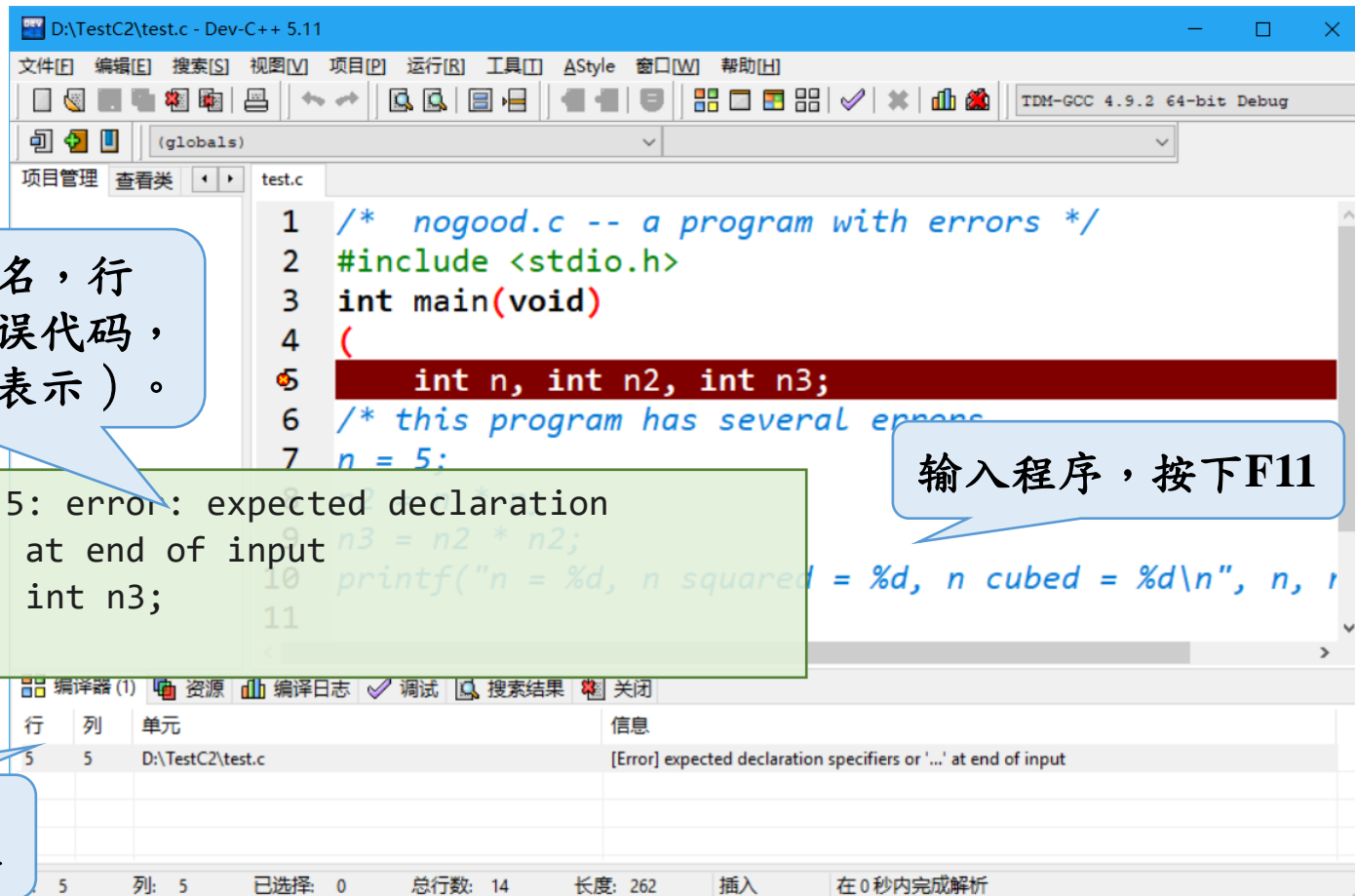
编译错误：DevC++

- DevC++支持不同的GCC编译器版本

错误信息包括文件名，行号，错误提示，错误代码，出错位置（用尖号表示）。

D:\TestC2\test.c:5:5: error: expected declaration specifiers or '...' at end of input
int n, int n2, int n3;
^

双击错误信息可以快速定位到相应行



输入程序，按下F11

内容纲要

1	基础知识
2	程序的开发
3	程序的编译
4	程序的调试
5	程序的测试

程序测试的方式

- 程序测试的目的

- 检查程序在不同的输入情况下是否得到预期输出

- 程序测试的方式

- 手工测试

- 重定向到文件测试

- 编程测试

- 程序测试的过程

- 设计测试实例，输入，比较输出

程序测试的设计

- 穷举测试：把所有可能的数据全部测试一遍
 - 例如：整数加法程序的定义域为 \mathbb{Z}^2 。
 - 穷举测试是最全面的测试，但往往数据量巨大，无法实施
- 理想的测试思想
 - 使用最少的测试数据，达到最好的测试质量（最高性价比）
 - 简化的测试毕竟不同于穷举测试，难免会遗漏缺陷

程序测试的设计

- 等价类划分法

- 把程序的输入域划分成若干部分，然后从每个部分中选取少数代表性数据作为测试用例。
- 例如整数除法：除数为零、可整除、不可整除

- 基本概念

- 有效等价类：对程序有意义的、合理的输入数据集合，用来测试功能是否正确实现
- 无效等价类：对程序无意义的、不合理的输入数据集合，用来测试程序是否有强大的异常处理能力（健壮性）。

程序测试的方式

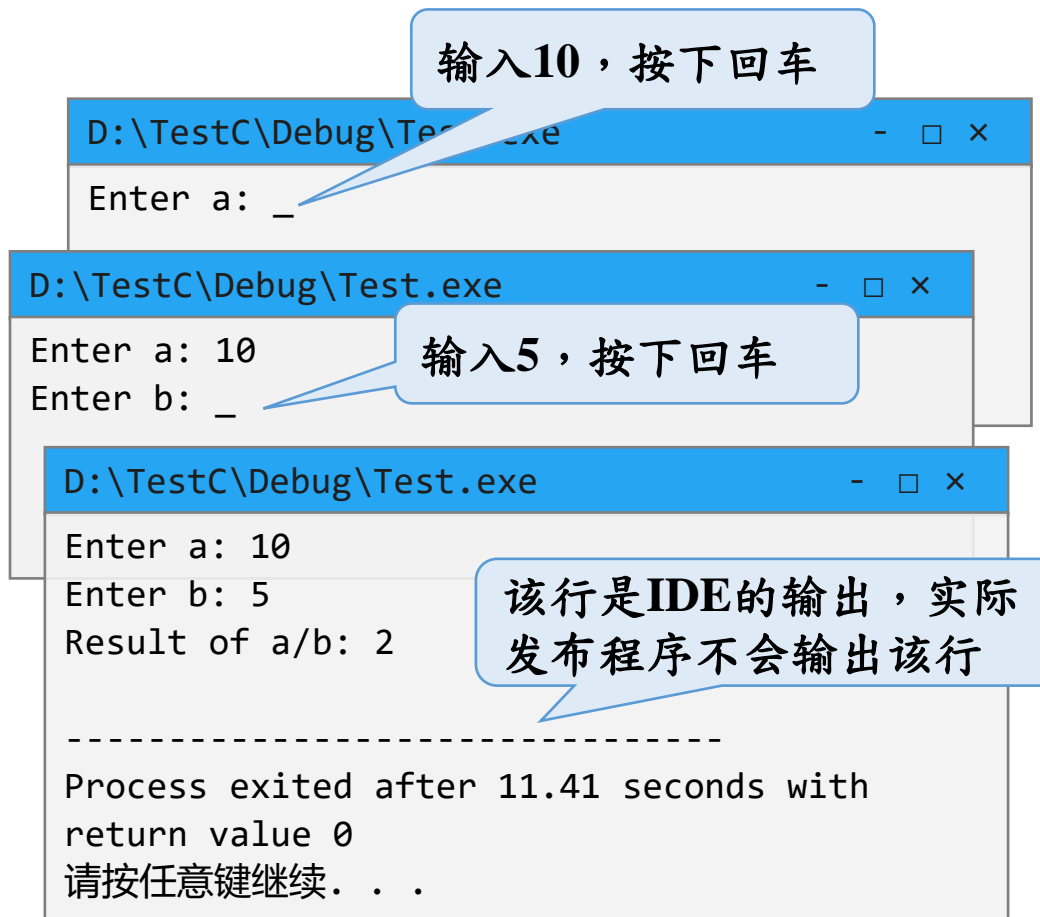
- 手工测试

- 键盘输入数据
- 观察输出结果
- 考虑可能存在的问题

- 优缺点


- 操作直观
- 效率低下

- 复杂程序从编程到完全排查错误需要测试成百上千次



程序测试的方式

- 重定向到文件测试
 - 运行前input文件输入数据



input.txt

10
5

- 优缺点

- 自动化程度高，特别输入或输出特别多的时候有用
- 提交时容易忘记删除相关文件，影响合作

```
#include <stdio.h>
int main(void)
{
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    int a, b;
    printf("Enter a: "); scanf("%d", &a);
    printf("Enter b: "); scanf("%d", &b);
    printf("Result of a/b: %d\n", a / b);
    return 0;
}
```

该2行将输入输出导向到同目录下的文件。

内容纲要

1	基础知识
2	程序的开发
3	程序的编译
4	程序的调试
5	程序的测试

程序调试的概念

- 程序错误(Program errors)经常称为bug
 - 查找和修复(fix)错误称之为调试(debugging)。
- 调试的必要性
 - 程序员编写的程序很难毫无瑕疵
 - 复杂程序从书写到基本无误，需要很多次调试
- 程序调试的目的
 - 程序投入实际运行前，用手工或编译程序等方法进行测试，修正语法错误和逻辑错误的过程。

程序调试的方式

- 程序调试的方式

- 断点调试

- 通过设置断点查看程序运行过程中的各变量值，从而发现错误。
 - 用于简单程序，或程序的细部环节

- 日志调试

- 通过在程序中输出日志查看程序运行过程中的各变量值，从而发现错误。
 - 用于复杂程序，或需要在复杂循环中观察变量值的变化

程序调试

- 除了语法错误之外，还有语义错误

- 答案显然错了

```
n = 5, n squared = 25, n cubed = 625  
Press any key to continue . . .
```

- 正确答案：n cubed = 125

- 编译器不能检测语义上的错误(semantic errors)

- 他们并不违反C规则。
 - 编译器无法发现你真正的意图。

- 需要程序员找到这些类型的错误。

- 一种方法是将程序与你的预期作比较

程序调试的术语

- 断点调试

- 将程序运行暂停在指定位置，方便观察或干预运行过程

- 断点




- 断点是调试器使程序中断的位置。

- 设置断点

- 程序员将在一些位置设置为断点，希望调试器在运行到这些位置时，暂停程序。

- 断点可以设置中断的附加条件

程序调试的术语

-  逐过程调试（单步调试）
 - 希望程序继续运行并再次暂停在下一条语句
 - 如果当前语句调用了函数，将进入所调用的函数中
-  逐语句调试（单步调试）
 - 希望程序继续运行并再次暂停在当前函数的下一条语句
-  跳出语句调试
 - 希望程序继续运行跳出该函数，并再次暂停在调用该函数的语句之后的语句

程序调试：Visual Studio

• 将程序的行为和预期输出做比较

程序员如果不清楚，
计算机更不清楚了

行数	程序	预期内存状态			程序预期行为
		n	n2	n3	
1	#include <stdio.h>				
2	int main(void)				程序从这里开始。
3	{				
4	int n, n2, n3;	*	*	*	开辟3个4字节空间。
5	n = 5;	5	*	*	赋值给n。
6	n2 = n * n;	5	25	*	赋值给n2。
7	n3 = n2 * n2;	5	25	125	赋值给n3。
8	printf("n = %d, n squared = %d, n cubed = %d\n", n, n2, n3);	5	25	125	此处打印n = 5, n squared = 25, n cubed = 125
9	return 0;	5	25	125	程序返回0并退出， 释放内存。
10	}				

程序调试：Visual Studio

• 将程序的行为和预期输出做比较

行数	程序	预期内存状态			实际内存状态		
		n	n2	n3	n	n2	n3
1	#include <stdio.h>						
2	int main(void)						
3	{						
4	int n, n2, n3;	*	*	*	*	*	*
5	n = 5;	5	*	*	5	*	*
6	n2 = n * n;	5	25	*	5	25	*
7	n3 = n2 * n2;	5	25	125	5	25	625
8	printf("n = %d, n squared = %d, n cubed = %d\n", n, n2, n3);	5	25	125			
9	return 0;	5	25	125			
10	}						

从这里开始，
一开始是对的

运行到这里，结果
是错的

程序调试：Visual Studio

- 查看每时每刻的变量名和对应的值

- 赋值前：0xCFFFFFFF

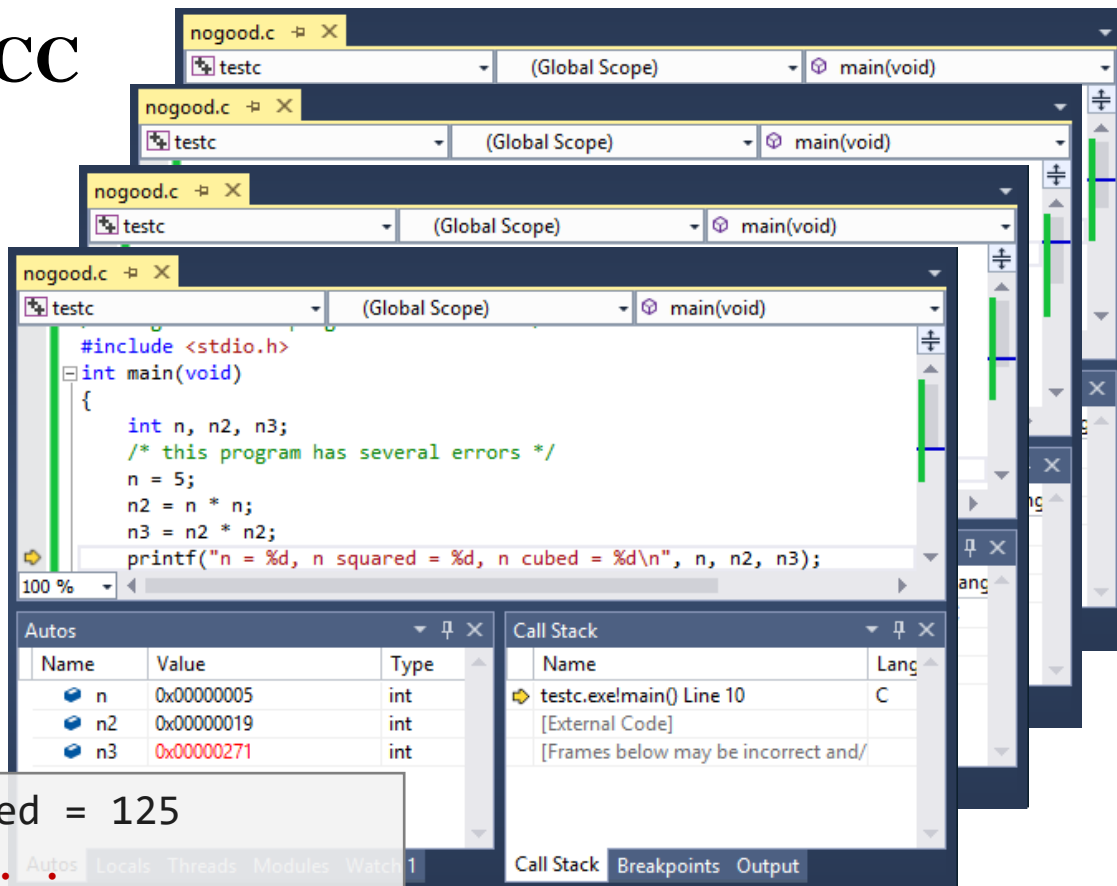
- n: 0x05 -- 5

- n2: 0x19 -- 25

- n3: 0x271 -- 625

- 发现 $n3 = n2 * n2$

- 改为 $n3 = n2 * n$



n = 5, n squared = 25, n cubed = 125

Press any key to continue . . .

程序调试：Visual Studio

- 题目：计算200位整数的加法。

- 从具体

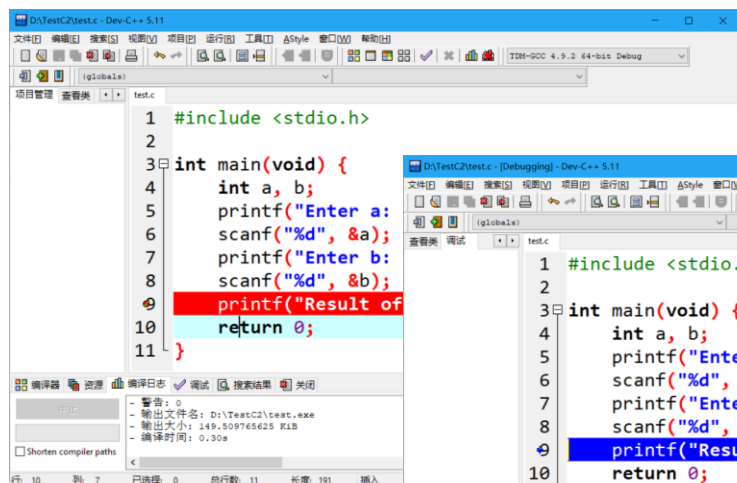
用普通整数加法，这道题很简单吧？

不可。长整型的最大表示范围为 $0 \sim 2^{64}-1$ ($2^{64} \approx 10^{19.3}$)

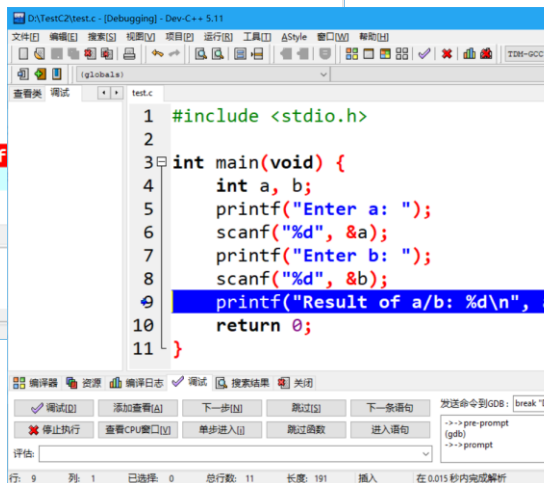
用普通整数加法，这道题很简单吧？

程序调试：DevC++

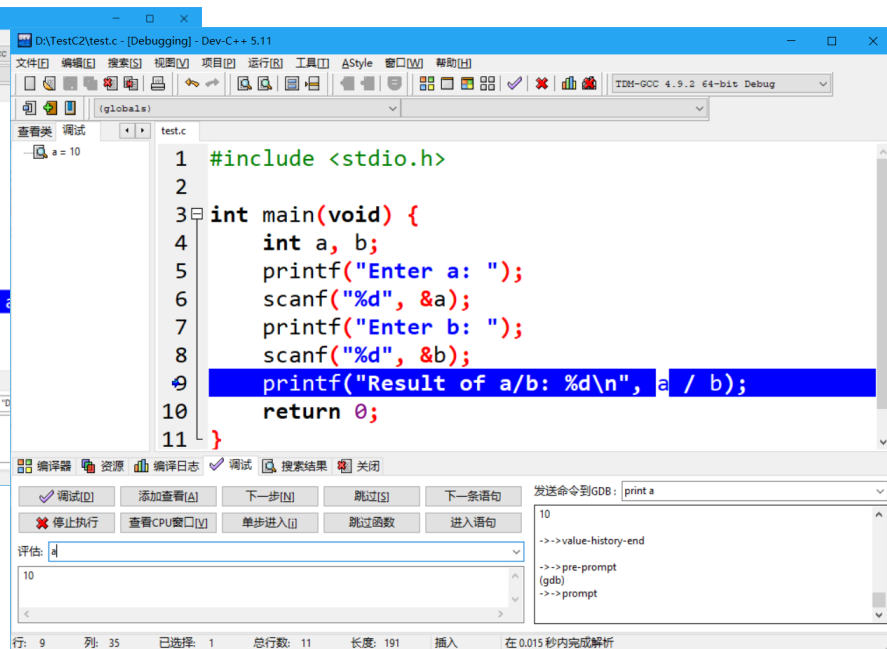
- DevC++的程序调试毕竟不如Visual Studio友好



设置断点



断点调试，添加查看



查看变量的值

程序调试：日志

- 日志输出

- 可以直接加上printf语句
- 如果输出太多，建议附加条件（if），或重定向到文件

- 缺点

- 在正式提交之前，应把多余的输出删去
 - 可以借助宏定义（#ifdef）

程序调试：日志

• 日志输出示例

D:\TestC\Debug\Test.exe

Enter a: 10
Enter b: 5
[LOG] a=10
[LOG] b=5
Result of a/b: 2

Process exited after 11.41 seconds with
return value 0
请按任意键继续. . .

```
#include <stdio.h>
int main(void)
{
    int a, b;
    printf("Enter a: "); scanf("%d", &a);
    printf("Enter b: "); scanf("%d", &b);
    printf("[LOG] a=%d\n", a);
    printf("[LOG] b=%d\n", b);
    printf("Result of a/b: %d\n", a / b);
    return 0;
}
```

这里输出了运行到此处的
各重要变量的值。

1

编译、测试 和调试

实验课程



廈門大學
XIAMEN UNIVERSITY



信息学院
(国家示范性软件学院)
School of Informatics

黃 煒
博士, 副教授
Dr. Wei Huang