装 订 线

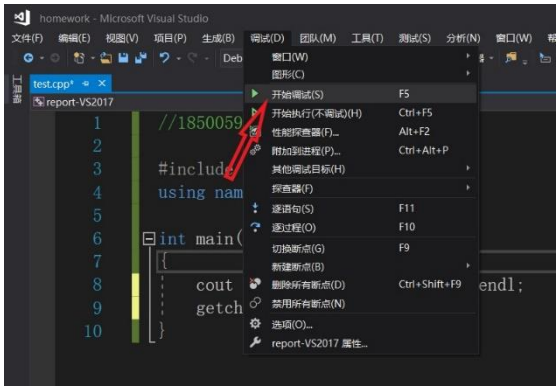
高级语言程序设计实验报告
——VS2017调试工具的使用

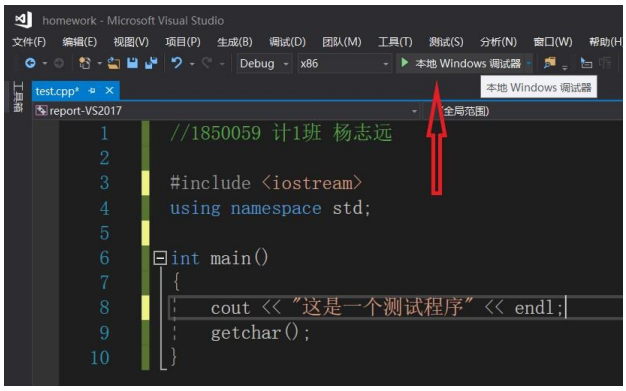
计算机一班
1850059
杨志远
2018年12月27日

1 VS2017 调试工具基本使用方法

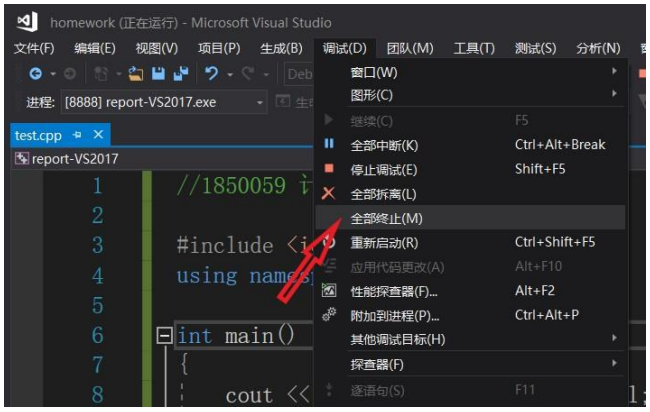
1.1 开始调试和结束调试

在已经编写好的代码文件下，选择菜单-调试-开始调试（如下左图所示），或者按下工具栏中的“本地 Windows 调试器”按钮（如下右图所示），或者按下快捷键 F5，就可以开始调试模式。



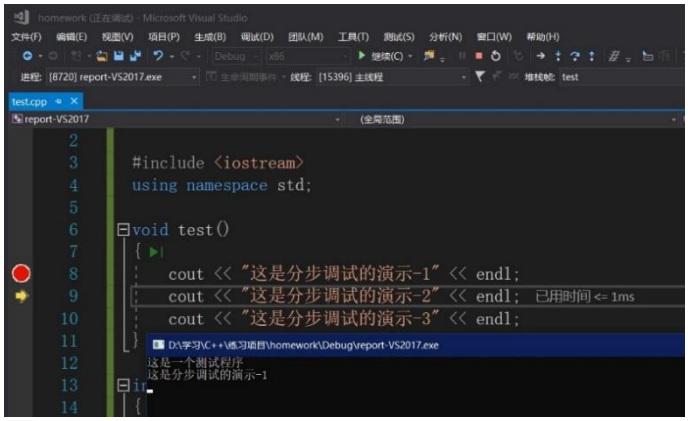


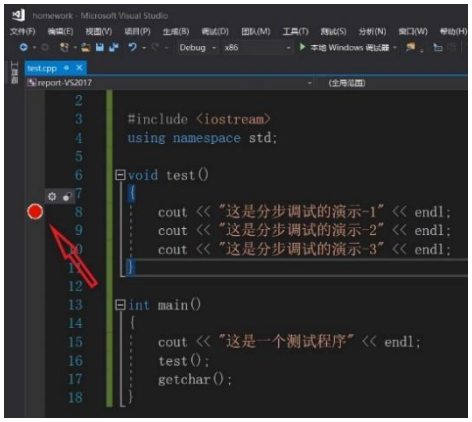
在已经进入调试模式后，选择菜单-调试-全部终止（如下左图所示），或者按下快捷键 Shift+F5,，或者执行完成并关闭程序，就可以结束调试模式。



1.2 函数中单步执行每个语句

在想要开始单步执行的语句上按下左侧代码行左边的边缘，或者按下快捷键 F9（如下左图所示），可以在开始调试模式后，跳转到该语句。每次按下 F11 可以执行当前语句并跳转到下一个语句（如下右图所示）。





1.3 关于系统类或系统函数的调用

在当前的 VS 版本(15.9.3)中，除非调用该函数时出现错误，否则不会进入系统类或系统函数的内部。如果出现了这种现象，可以按下 Shift+F11 快捷键跳出当前函数，并返回到调用该函数的语句。这样做不仅可以跳出系统类和函数，也可以跳出用户自定义的函数。

1.4 关于自定义函数的调用

在单步调试的过程中，如果想要跳过当前执行的函数中某个语句调用的子函数，可以按下快捷键 F10 进入下一步（如下左图所示，按下了 3 次 F10）,这样可以一步完成自定义函数的执行。如果想要进入当前执行的函数中某个语句调用的子函数内部，可以按下快捷键 F11 进入下一步（如下右图所示，按下了 1 次 F11），这样可以转到被调用函数中单步执行。

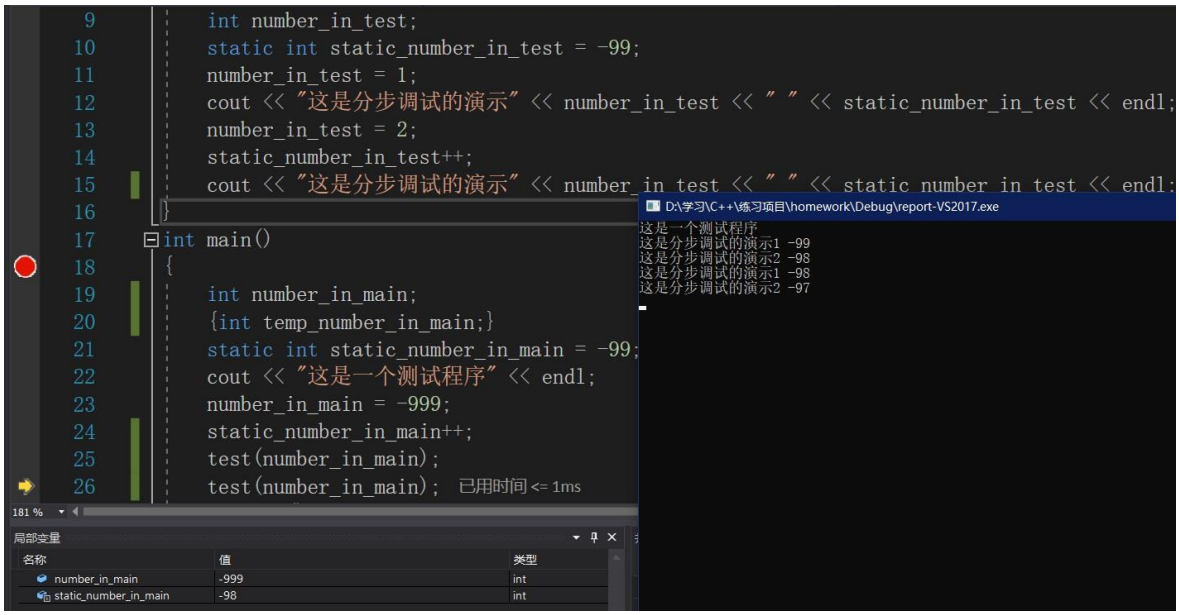
2 查看生存期/作用域变量

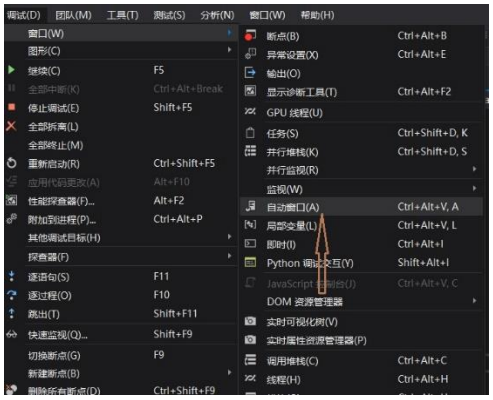
2.1 查看形参/自动变量/静态局部变量的变化情况

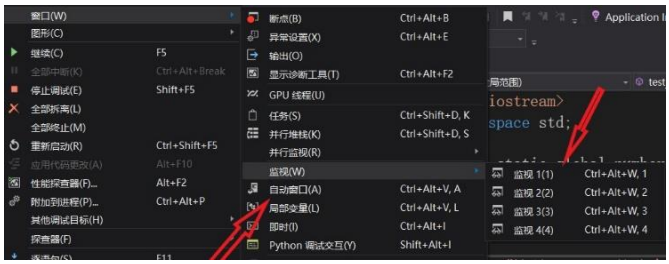
在调试模式中，选择菜单-调试-窗口-局部变量（如下左小图所示），或者按下快捷键 Ctrl+Alt+V 接 L，可以打开局部变量窗口（可以显示形参、自动变量和静态局部变量），接下来

就可以按下 F10 或 F11 单步执行代码并观察自动变量和静态局部变量的数值、类型、变化情况（如下大图所示）。

可以看到，形参和自动变量的作用域仅为当前函数体或代码块，离开作用域即消失。静态局部变量的作用域同样为当前函数体或代码块，离开作用域即消失，但在作用域外依旧能保持它的值不变。





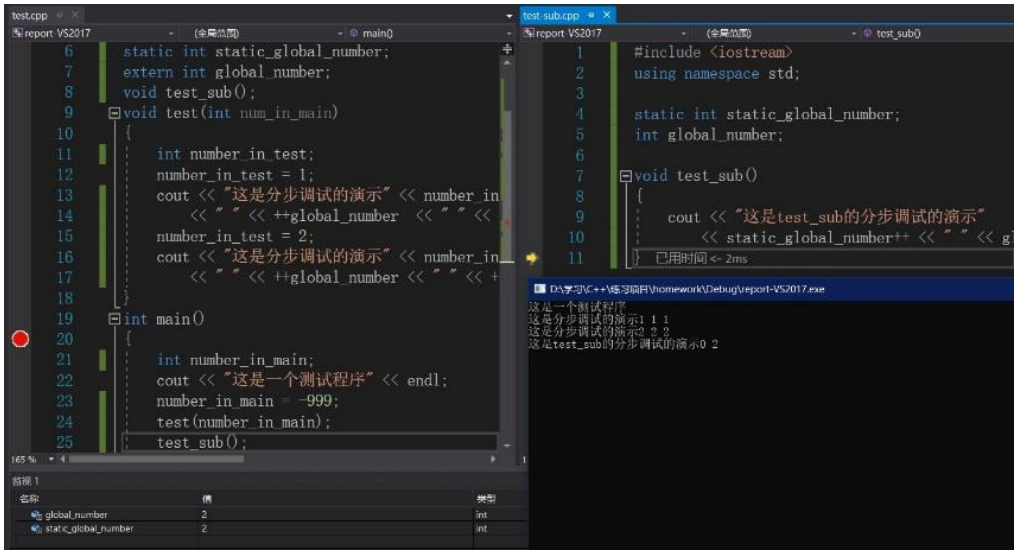


2.2 查看静态全局变量/外部全局变量的变化情况

在调试模式中，选择菜单-调试-窗口-自动变量（如上左图所示），或者按下快捷键 Ctrl+Alt+V 接 A，可以打开自动变量窗口（可以显示与当前代码行有关的变量的数据），接下来就可以按下 F10 或 F11 单步执行代码并观察各个变量的变化情况（如下大图所示）。

也可以通过菜单-调试-窗口-监视-监视 1/2/3/4（如上右小图所示），或者按下快捷键 Ctrl+Alt+W 接 1/2/3/4，打开监视窗口，手动添加需要监视的变量。可以通过在变量名上点击鼠标右键-添加监视，也可以通过在监视窗口输入变量名甚至表达式来监视各类变量。

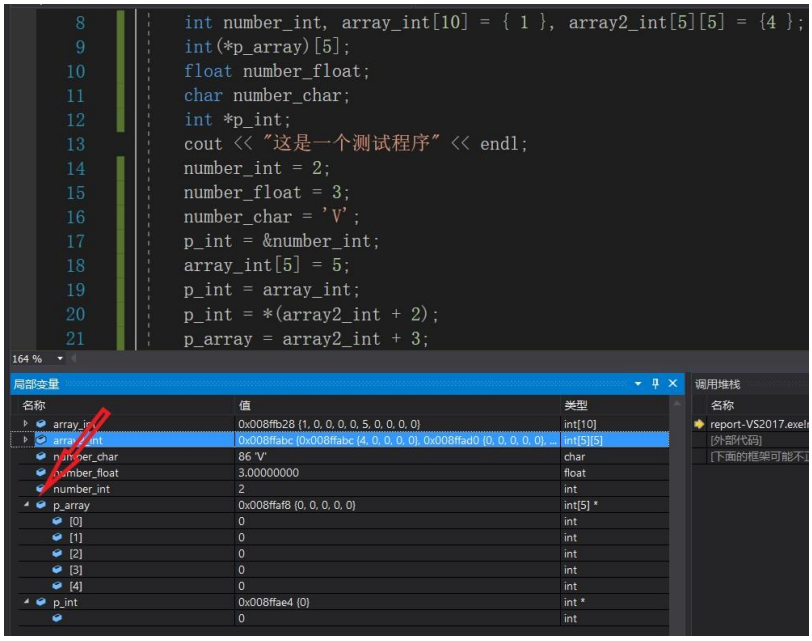
可以看到，外部全局变量在整个文件中都可以被读取到，作用域为所有文件，但静态全局变量只能在单个文件中被读到，作用域为其所在的单个文件。另外，通过以上两种方法都无法查看非 main 函数所在的文件中的静态全局变量（无论变量名是否相同）,但外部全局变量、静态局部变量、自动变量都不受影响。



3 查看不同类型变量

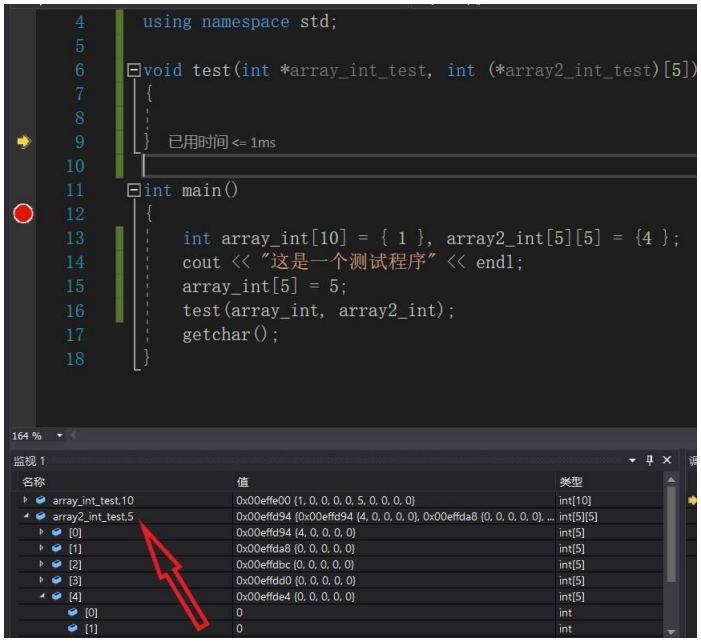
3.1 简单变量/指向简单变量的指针变量/一维数组/指向一维数组的指针变量/二维数组/指向 m 个元素组成的一维数组的指针变量

查看变量的方法和 2 中演示的大致相同。指针变量可以看到其指向的地址，在窗口中展开指针变量左边的小三角后可以看到地址中的数据（如下图所示）。



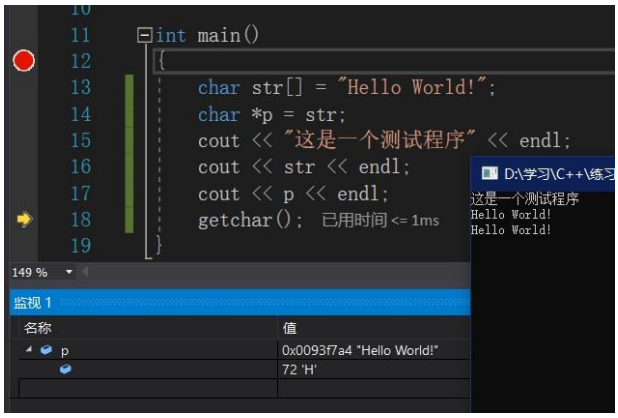
3.2 实参是一维数组/二维数组，形参是指针/行指针

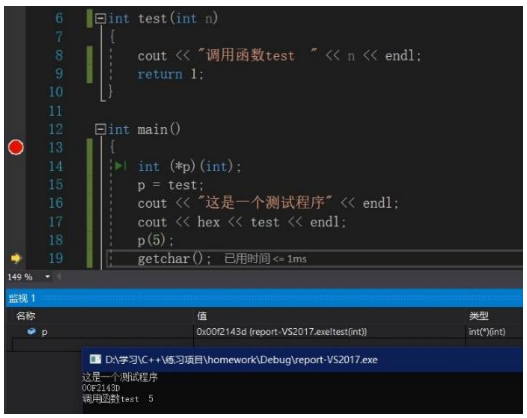
函数传入数组的指针作为的形参情况与之前有所不同，直接在局部变量或自动变量窗口中只能看到指针所对应的单个地址或单个行的地址的数据。因此，需要在监视窗口中手动输入信息，无论一维数组或二维数组，格式都为“变量名,数组大小”（如下图所示），前者可以直接看到一维数组中的数据，后者则是通过访问大小为 N（此处 N=5）的一位数组的首地址来间接看到数据。



3.3 指向字符串常量的指针变量

指向字符串常量的指针变量无法通过监视一维数组的方式来查看内容，只能直接通过指针间接读取字符串（如下左图所示）。



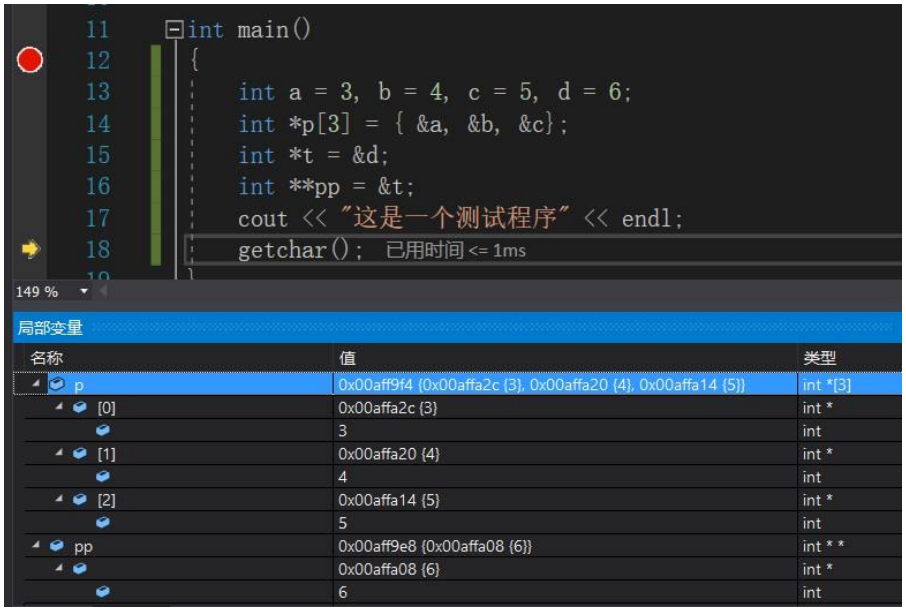


3.4 指向函数的指针变量

指向函数的指针可以在窗口中看到其所对应的地址、文件名和函数名称、形参类型（如上右图所示）。

3.5 指针数组及指向指针的指针

查看指针数组与指向指针的指针的方式与二位数组类似，都是在窗口展开两层（如下图所示）。

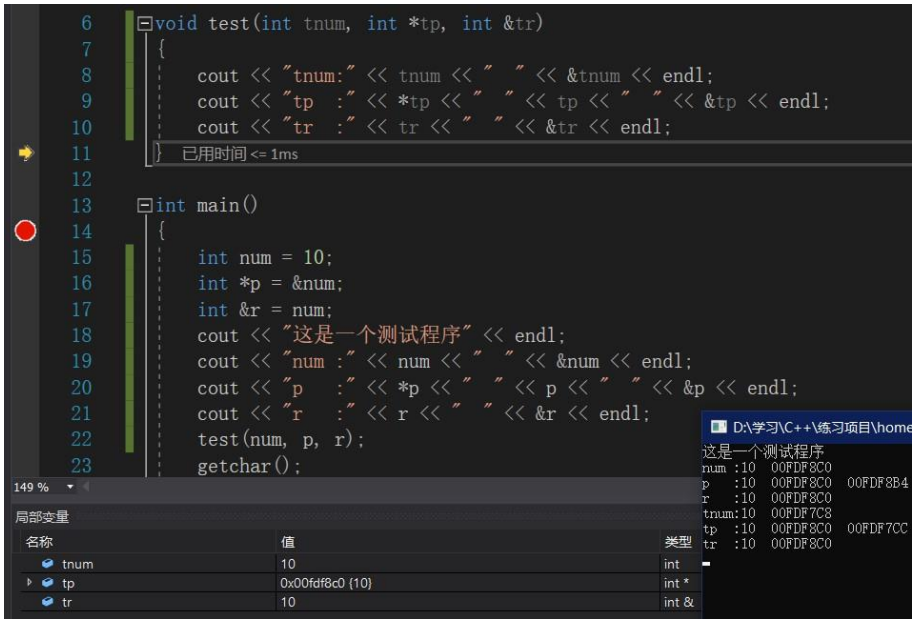


3.6 引用以及函数的形参为数组的指针/引用

在任意一个窗口中，都可以看到引用变量的数值和类型（如下图所示）。

可以看到，引用变量的地址和它所对应的变量的地址相同，也就是说它没有被分配独立的内存。但指针变量的地址和它所对应的变量的地址不同，被分配了新的内存。因此引用和指针有很大的区别。

当引用作为实参传入函数时，它的地址依旧不变，依然指向被调用函数中的那个变量，也就是说它依然没有被分配独立的内存。而指针作为实参传入函数时，它作为变量所存储的值不变，所指向的变量也不变，但它本身的地址发生了变化，被分配了新的内存，这段内存不同于被调用函数中那个指针的内存。因此，当作为实参传入函数时，引用和函数也有很大的区别。



3.7 使用指针时出现了越界访问

在自动变量或局部变量窗口中，只能看到合法范围的数组内容。要想看到越界访问的内容，必须要在监视窗口中手动添加数组并指定长度（如下图所示）。程序成功执行，返回值为0，没有报错。