# Debug in VS2015

概述：

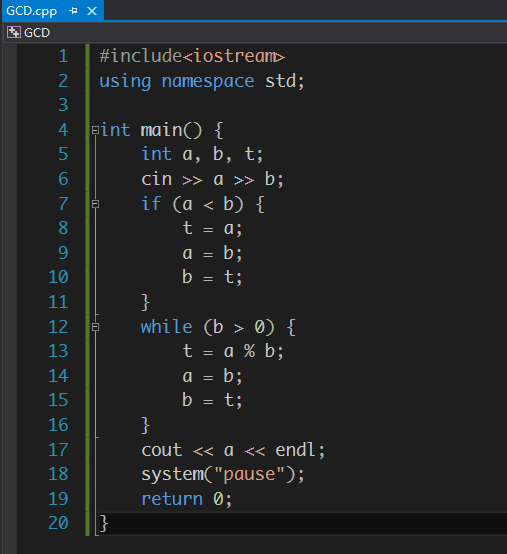
在程序运行过程当中，因各种原因时常会发生错误。通常情况下，直接阅读代码难以寻找到错误的原因，这时便需要程序调试工具的帮助，在此，我们简要介绍使用Visual Studio 2015调试C++程序的方法。

1. **例子演示**

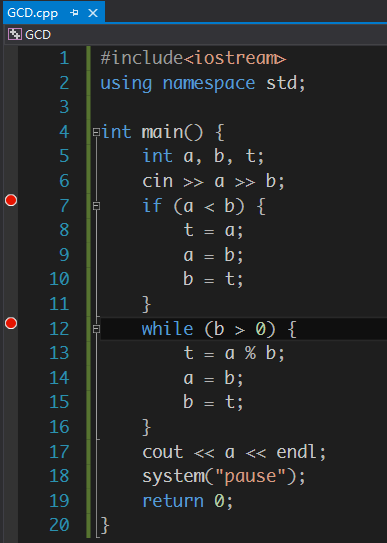
首先我们准备一份利用辗转相除法(欧几里得算法，简称gcd)求解最大公因数的代码，代码只有寥寥数行，其中却体现着深刻的数学原理，我们通过调试工具，观察程序运行过程中各个变量的变化。

辗转相除法：整数a和b的最大公约数 = b和a%b的最大公约数。

更多关于辗转相除法的数学原理，可以参考链接中的博客：  
<https://xuanwo.io/2015/03/11/number-theory-gcd/> 数论——欧几里得算法  
我们首先以循环的方式写出gcd的代码：

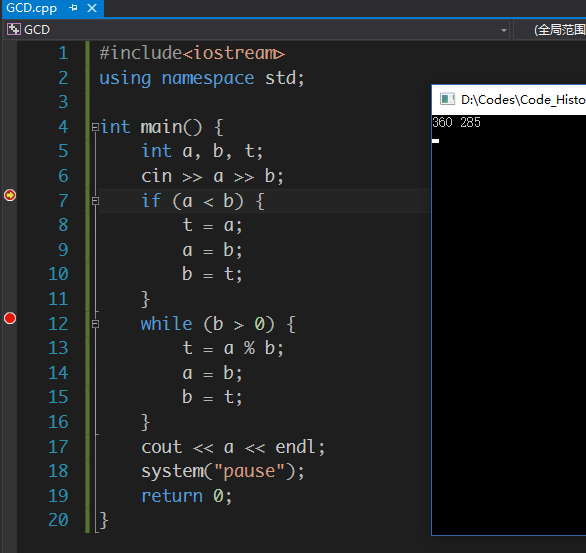


接着，我们点击图中行号左侧灰色边栏添加**断点**，或使用快捷键F9在光标所在行添加断点。

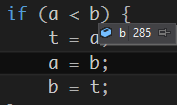
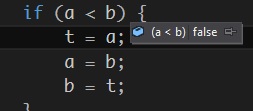


**断点：在调试模式下，程序运行到断点处就会暂停。**

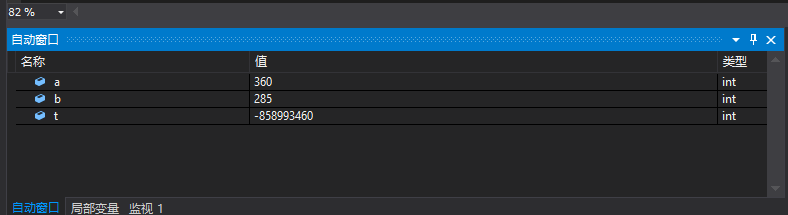
在弹出的控制台中输入测试数据360和285，按下回车，程序即开始运行，并按照预想暂停在第一个if处。



在暂停状态下，鼠标移动到某个变量或表达式上，即可获得他们当前的值。

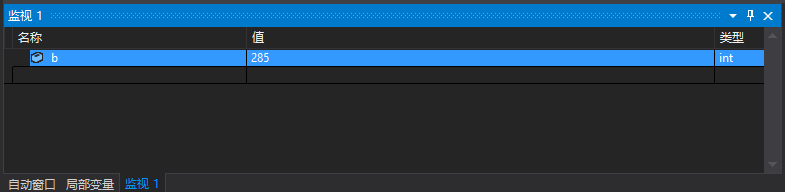


或者，可以在底部的自动窗口中监视到部分变量。



如果你只想监视某些变量的值，可以右键这个变量->添加监视，这样你就能在下面的监视1窗口中看到它了。





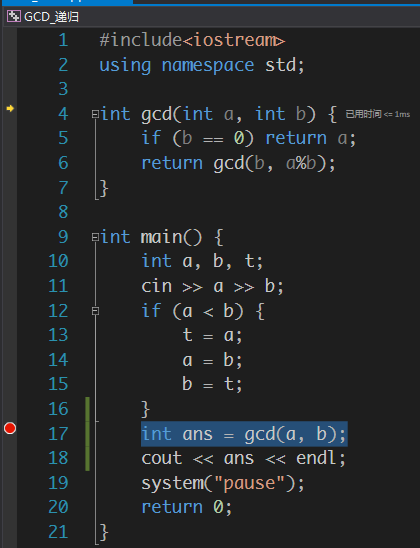
接下来，我们继续调试，打开顶部的**调试**菜单。



我们介绍其中几个常用的功能：

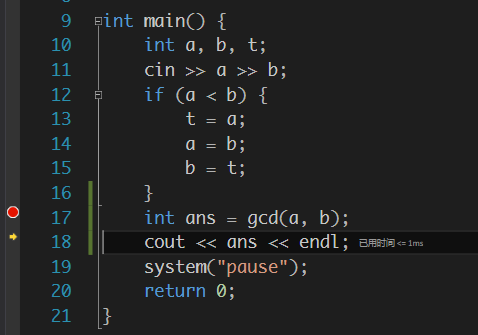
1. **继续（F5）：继续运行程序，直到下一个断点。**
2. **逐语句（F11）：逐条语句地运行程序，若遇到函数调用，则会进入函数内部。**
3. **逐过程（F10）：同样逐条语句地运行程序，但不会进入函数内部。**
4. **跳出（Shift+F11）：跳出当前调用的函数。**
5. **结束调试（Shift+F5）：结束调试。**

我们通过实例演示F10和F11的区别



在递归版gcd中，让程序暂停在调用gcd函数的这一行，按下F11，发现程序会进入gcd函数内部；

如果按下F10，程序则会直接运行下一行，不进入gcd函数内部。如下图所示：



请你自行尝试其它功能，观察程序的运行过程，监视变量的变化，熟练地运用调试工具。

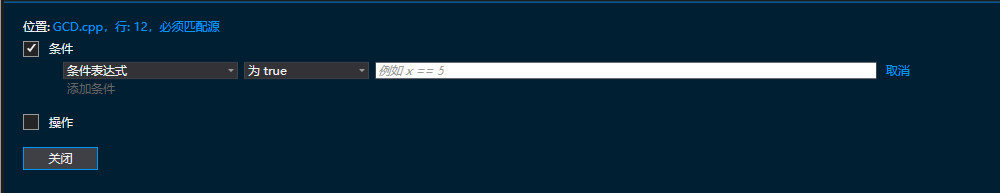
1. **进阶技巧——条件断点、断点操作**

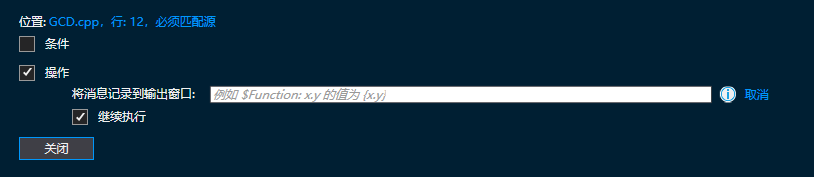
右键断点，你可以看到下面的菜单：



点击**条件**，进行条件断点的条件设置。

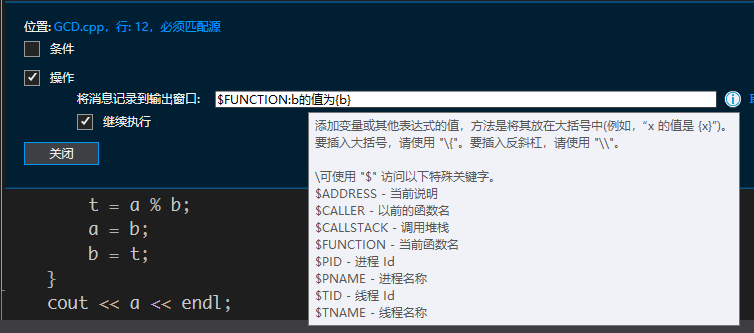
* **条件断点：在满足某个条件时，程序运行到断点处暂停；若条件不满足，则不会暂停。**
* **操作：在到达断点时，可以将消息记录到输出窗口。**

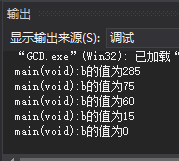




你可以在这里按照需要进行断点的设置。

我们输入 $FUNCTION:b的值为{b}，即可在**输出**窗口中看到消息记录。

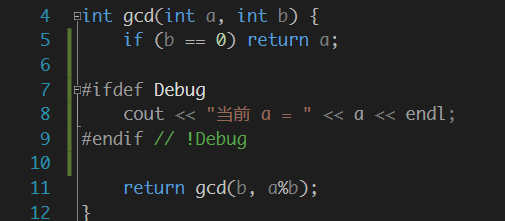




1. **条件编译**

在程序中加入输出语句，监测中间变量，这是一种简单有效的程序调试方法。然而调试完毕之后，需要将调试语句不厌其烦地删除或注释掉，一旦程序再次出错，则又要重新再来一遍，有没有什么简单的方法呢？

答案是——条件编译语句，请先看图：



#ifdef和#endif的功能是：如果**定义了Debug标识符（#define Debug）**，则编译#ifdef和#endif之间的语句，否则不进行编译。

这样的话，我们把每个调试代码块都用#ifdef和#endif包围，在需要调试时，在程序开头加上#define Debug就行了。

关于条件编译的更多说明，参见下面的博客：

<https://www.cnblogs.com/wengzilin/archive/2012/04/26/2471018.html>

以上便是Visual Studio调试程序的基本方法，希望你能有所收获。

附件里包含gcd的递归实现，你可以用上述技巧调试另一份代码，加深印象。

参考资料：

1. <https://blogs.msdn.microsoft.com/c/2015/11/29/visual-c-2015/> Visual C++ 2015 中的可视化调试工具
2. <http://blog.jobbole.com/33865/> 11个强大的Visual Studio调试小技巧
3. <https://xuanwo.io/2015/03/11/number-theory-gcd/> 数论——欧几里得算法
4. <https://www.cnblogs.com/wengzilin/archive/2012/04/26/2471018.html> 条件编译#ifdef的妙用