**VS2010调试技巧基础**

**核心功能**

F5: 启动调试程序

F9：设置断点，再按一次，删除该断点

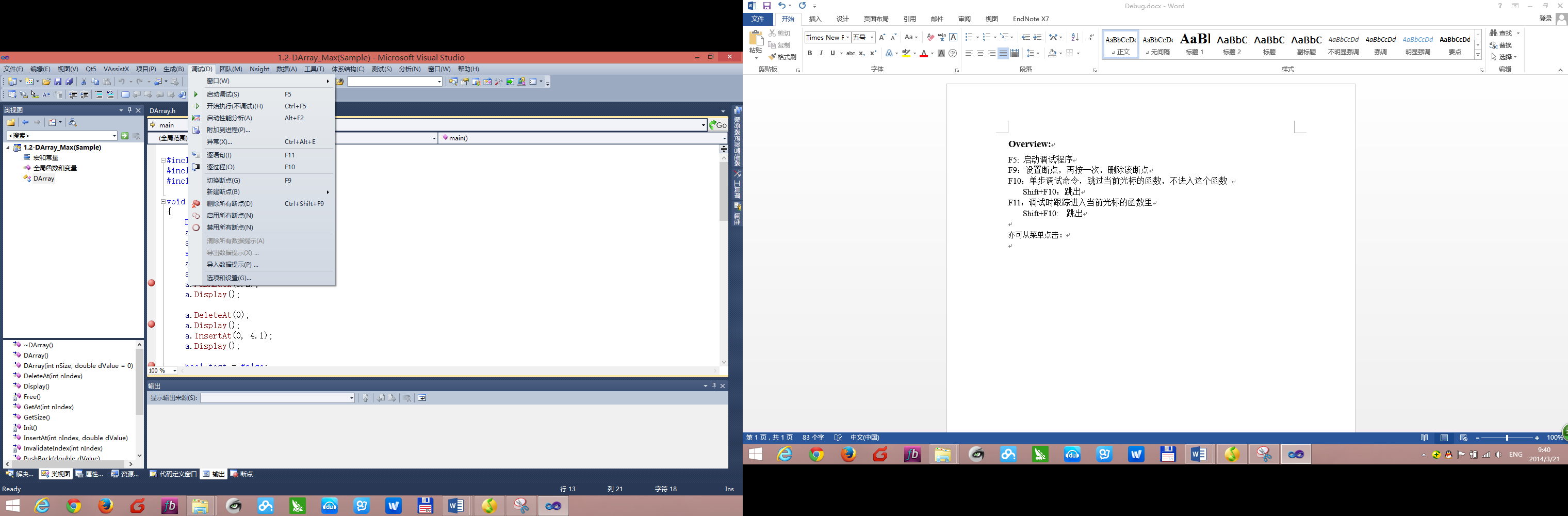
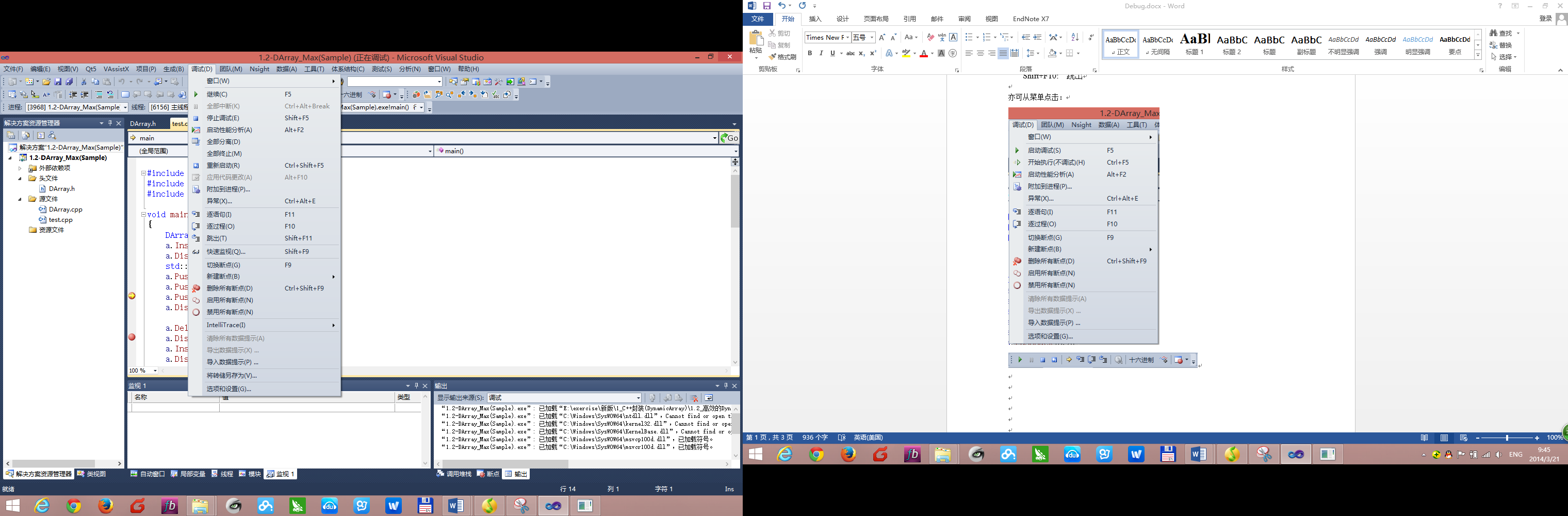
F10：单步调试命令，跳过当前光标的函数，不进入这个函数

Shift+F10：跳出

F11：调试时跟踪进入当前光标的函数里

Shift+F10: 跳出

亦可从菜单点击：

左一为未进入调试状态时的菜单选项。右一为进入调试状态后的菜单选项（有更多功能）。



在进入调试状态后，工具栏会出现额外的工具，如上图。

**各项功能讲解**

 **Restart**(Ctrl+shift+F5):此debugger功能将从程序的开始(第一有效行)处全速执行,而不是从当前所跟踪的位置开始调

试,这时所有变量的当前值都将被丢弃,debugger会自动停在程序的main()开始处.这时如果选择**Step Over**(F10)就可以逐步执行main()函数了.

 **Stop Debugging**(Shift+F5):此debugger功能将终止(所有)调试,并返回到常规编辑状态.

 **Break**(此功能常常在遇到调用函数的语句时可见.):此功能将在调试过程中的debugger当前位置挂起程序的执行,然后就可以在调

试状态一修改程序的代码,接着可以用**Apply Code Changes**(Alt+F10)来应用修改的代码到正在调试的程序当中.如果,当前(需要,待)可以(从DOS

等窗口)输入值,挂起后将不能再输入.

 **Apply Code Changes**(Alt+F10):此功能可以在程序正在调试程序过程中应用(挂起)修改后的源代码.如,选择**Break**功能并修改代

码后,只要选择**Apply Code Changes**(Alt+F10)就能将修改后的代码应用到正在调试的程序当中.

 **Show Next Statement**(Alt+Num\*):此功能将显示程序代码的下一条语句,如果源代码中找不到,则在Disassembly窗口中显示语句.



当在Disassembly窗口中显示时,可以单击Disassembly 返回到源代码窗口.

 **Step Into**(F11):此功能可以单步进入到在调试过程中所跟踪的调用函数的语句的函数内部.如,当前语句是"d.Display()",

选择**Step Into**(F11)后,Debugger将进入Display()函数内部并停在Display()函数内部的第一条语句上.(此时,

就可以**Step Over**(F10)对Display()函数进行单步调试了.)

 **Step Over**(F10):此功能可以单步对所在函数单步调试,如果调试的语句是一个调用函数的语句时,

Debugger将全速执行所调用的函数,单步(一步)通过所调用的函数,Debugger停该调用语句的下一条语句上.

 **Step Out**(Shift+F11):此功能将使Debugger切换回全速执行到被调用函数结束,并停在该函数调用语句的下一条语句上.

当确定所调用的函数没有问题时可以用这个功能全速执行被调用函数.

 **Run to Cursor**(Ctrl+F10):此功能将全速执行到包含插入点光标所在的行,可以作为在插入点光标处设置常规断点的一种选择.

注意,当光标处不是一个有效的执行语句时此功能将不起作用.

 **Go**(F5):此功能将全速执行程序直到遇到一个断点或程序结束,或直到程序暂停等待用户输入.

注意,此功能最能有效的调试循环,常将断点设置在循环体内,重复的按F5全速执行循环体可以测试循环过程中的产生的变化.

 Step Into Specific Function:此功能可以可以单步通过程序中的指令,并进入指定的函数调用,此功能对于函数的嵌套层不限.

**总结**

调试最重要的还是你要思考，要猜测你的程序可能出错的地方，然后运用你的调试器来证实你的猜测。而熟练使用上面这些技巧无疑会加快这个过程。最后，大家如果有关于调试方面的问题，我乐意参与探讨。