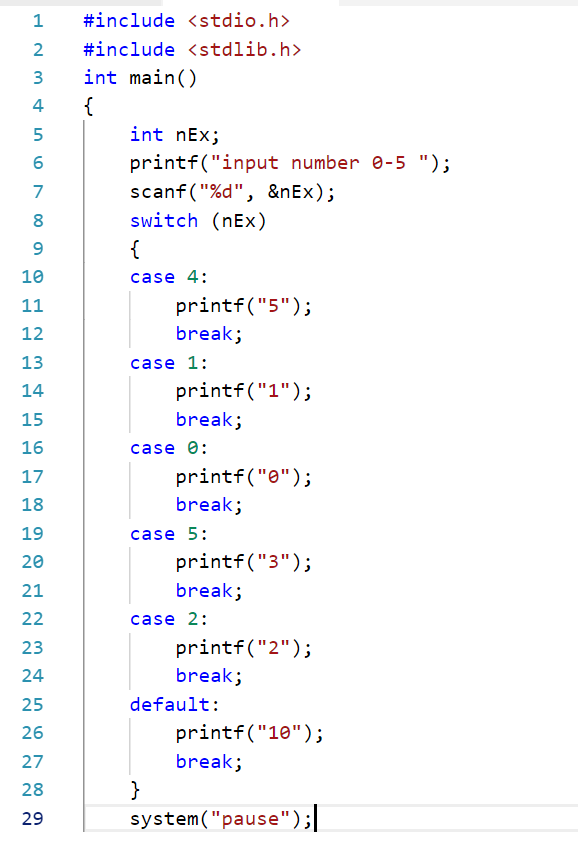
# Switch

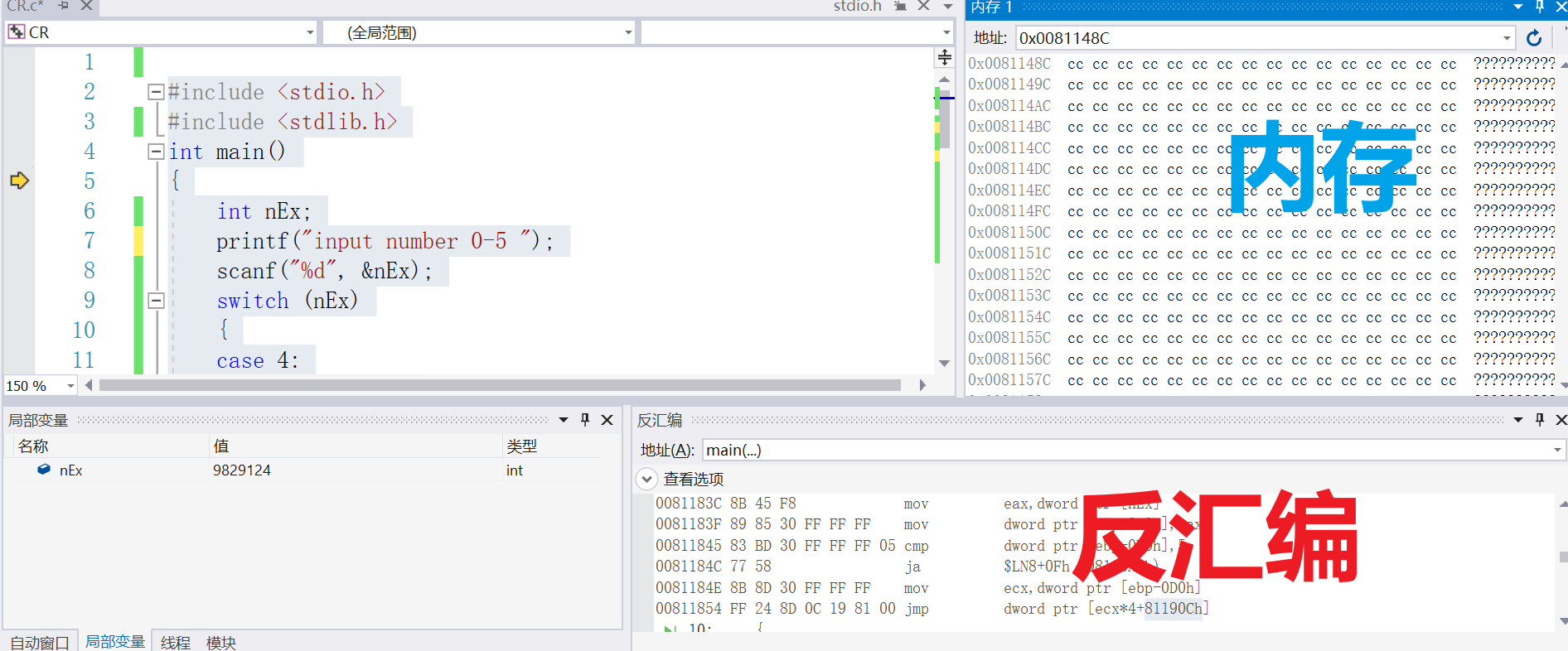
**Switch 原理简单分析：**



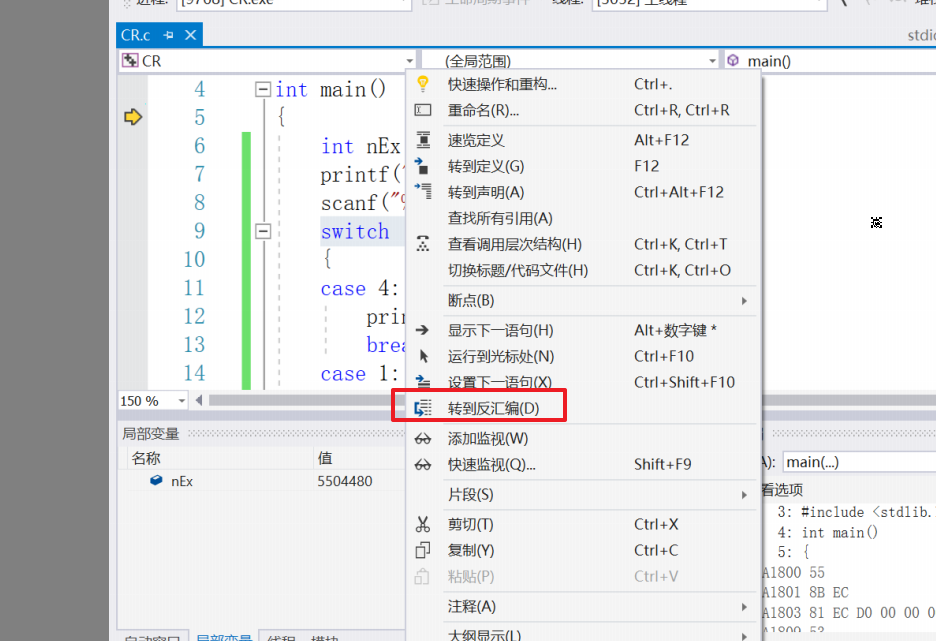
一般的，编译器对switch语句的编译最终结果可分为4种情况，在这里只分析其中一种情况———case值小不连续的情况。

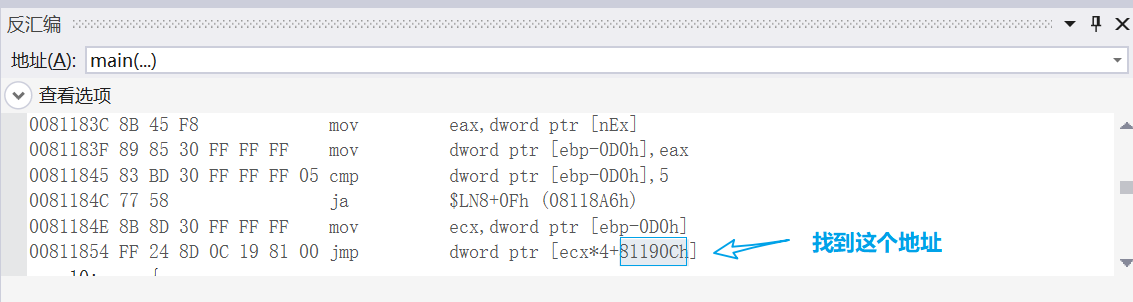
首先打开VS调试界面（IDE环境：vs2017）

如图，打开标注的十六进制反汇编和内存窗口。

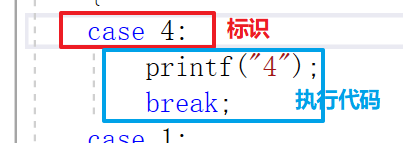


如图，左键选择switch（nEx）处右键然后选择转到反汇编

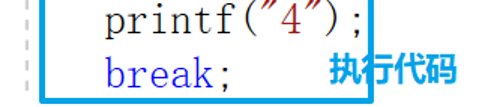
如图然后找到地址 8A190ch ，



意思是编译器将case各个对应的值排序做成了一个表，而8A190ch是这个表的首地址，在这个表中各个地址存放的内容是各case标识后的 存放执行代码的地址，举个例子：

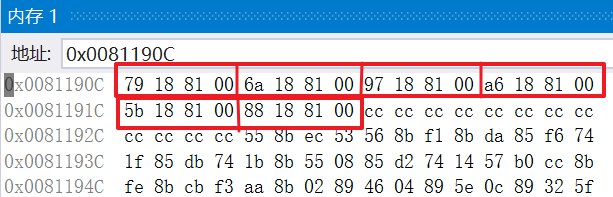


我们尝试在内存单元中找到 case 4 对应的执行代码：

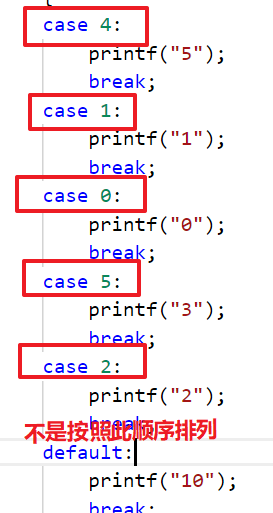


步骤如下：

1. 将switch表的地址 8A190ch在内存框查看值



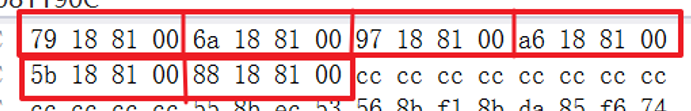
在表中编译器按照case n 的值大小顺序排列，而不是以源代码中case n的顺序排列



因此case在内存中的排列的顺序是 0 - 1 - 2 - 4 -5，

Case对应的有效值有5种可能，

但是在表中有6个值，如图：



这是因为在2到4之间缺少3，只缺少1个数的差值，编译器将这种情况视为小不连续排序，则会将 default 存放的地址放入其中 所以上图中

各标识和所存放的地址对应关系为

0 - 00811879h

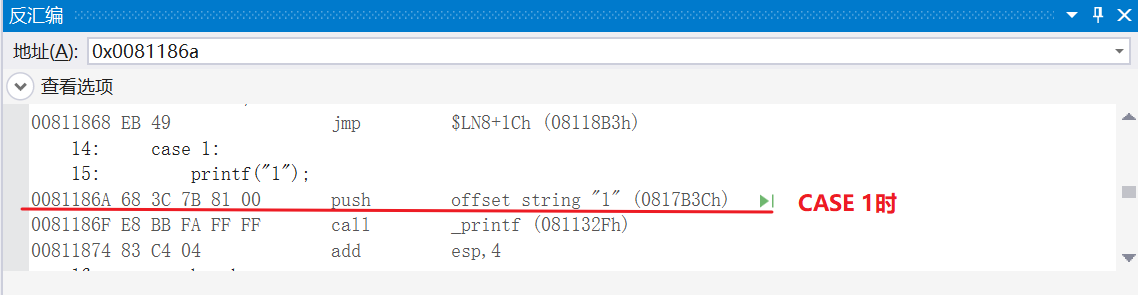
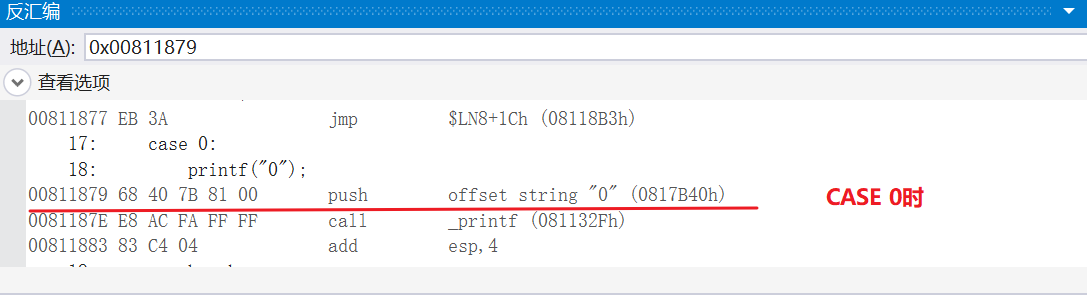
1. 0081186ah
2. 00811897h

3（default）- 008118a6h

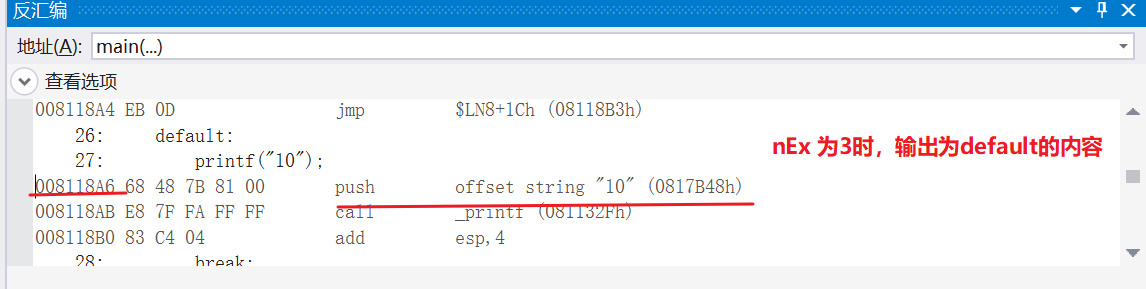
4 – 0081185bh

5- 00811888h

现在我们去验证一下



再来验证下 “case 3” 是否和default对应



可以看到，验证都通过了。

至此，简单说下switch的工作流程：

1. switch 对参数n取值，参数可以是整数变量，整数常量，也可以是表达式（最终值要为整数值）
2. n-最小CASE值然后判断是否大于最大case值减最小CASE值之后的值，如果大于则走default
3. 不大于则在case范围内，查表执行相应代码，如果是小不连续值则走default。
4. CASE>2 switch才会做表。