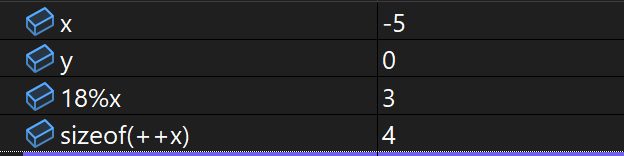
实验1

首先分析原程序，希望输出y=40，分析40这个答案，合理的流程应该为:

18%x得到x=3；sizeof(++x)使x变为4，x\*=x+1使x=4\*(4+1)=20;x+=x\*=x+1使x=20+20=40

而实际输出为24，合理推测x\*=x+1这一步传入的数据为x=3才导致与预期不符

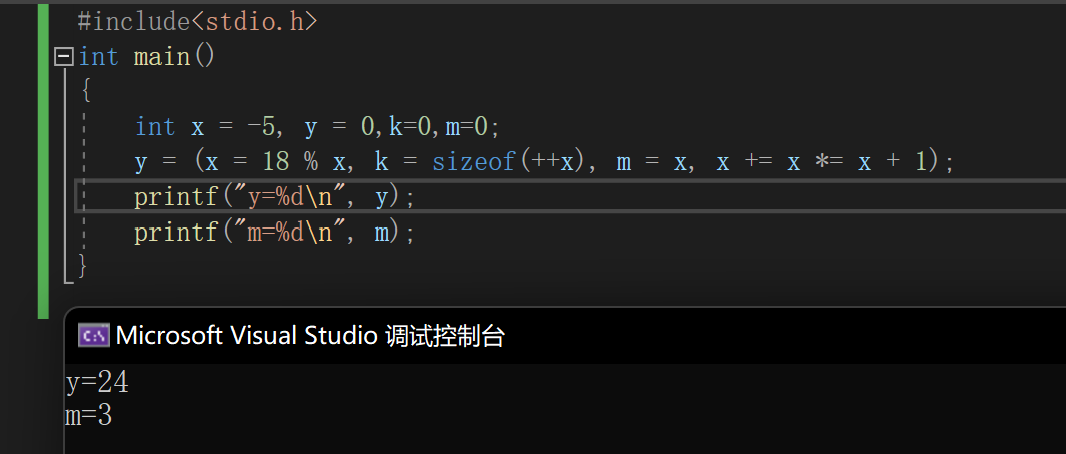
检查18%x，结果为3，没有问题：



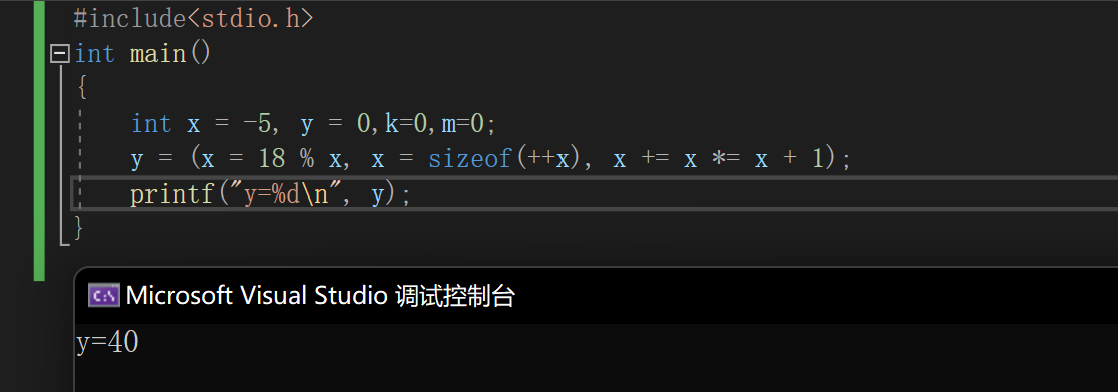
检查sizeof(++x)，为4；检查sizeof(++x)步骤后的x，仍为3

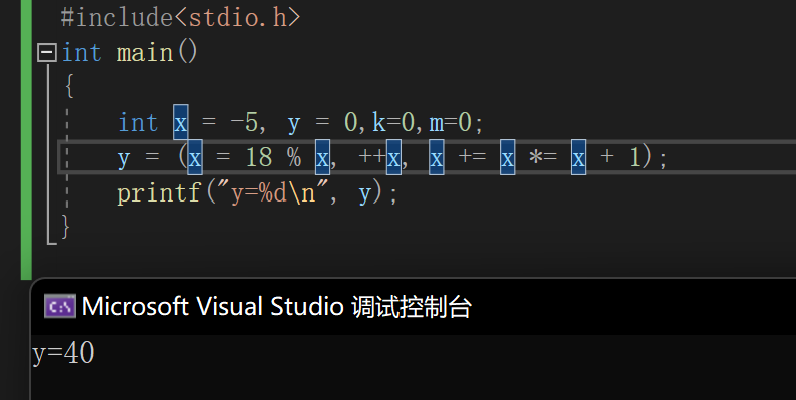
所以问题就在于，这一步并没有改变x的值

Sizeof是一个单目预算符，不是一个函数，这个语句既没有作赋值处理，也没有使x独立自增，所以x的值没有改变：



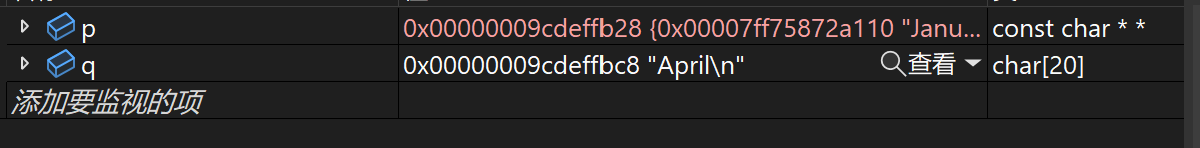
修改后可以输出期望的结果：





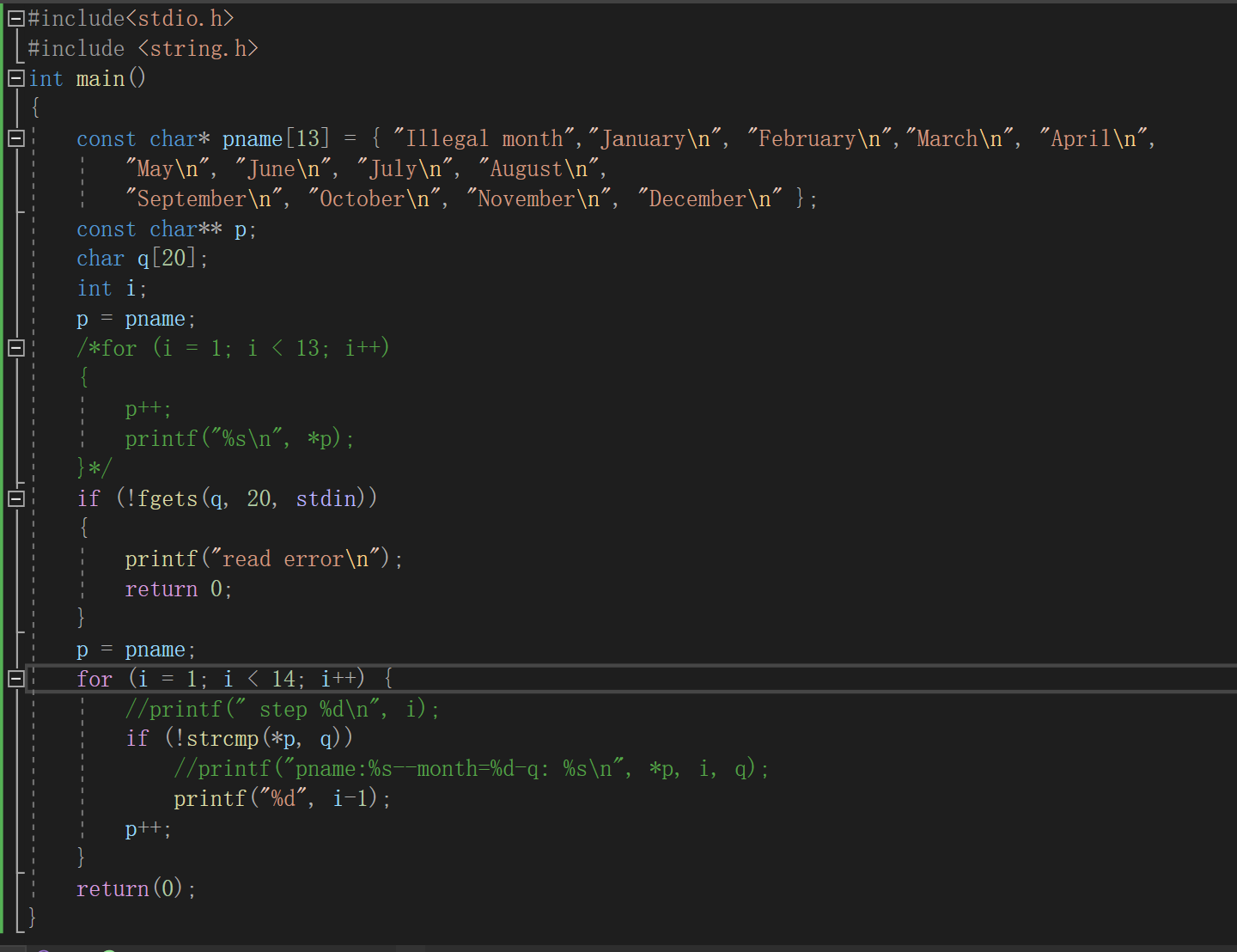
实验2

输入April后，查看此时对应的字符串p和q的内容



可以看到，根据fgets函数的性质，q读入控制台输入的内容时也存入了换行符’\n’，而根据初始化，p中的内容只包含了月份的正常拼写，所以使用strcmp函数比较字符串是否相同时，p和q中内容必然不同。而且p和q的内容第一次比较时是“Illegal month”和q中内容比较，i的计数比实际月份数多了1，所以输出时要输出i-1，for循环范围也是从i=1到i=13.

最粗暴的方法就是在定义pname数组时，就在每个月份英文名后加上“\n”，使程序能够正常识别（注:我把一些原程序里不需要的输出注释掉了）



除了这种方法以外，也可以依次比较\*p中每个字符和数组q[i]（0<=i<strlen(q)-2），如果全都相等，则可输出对应的月份数。

结果示例如下：

