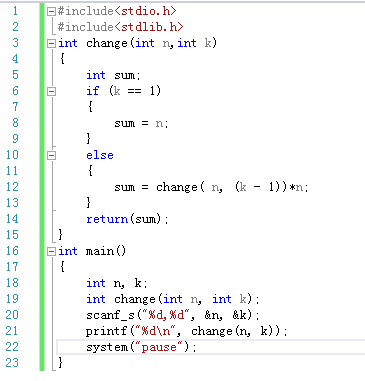
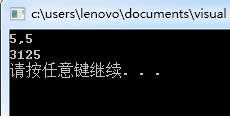
## 实验九（补充）递归函数

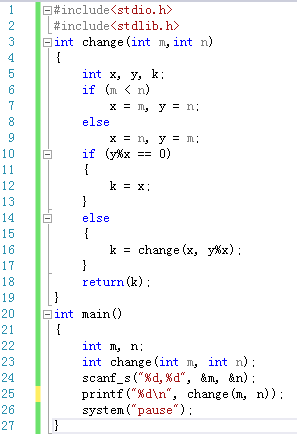
1. 使用C编写一个指数函数，通过递归调用实现n^k（n的k次方）

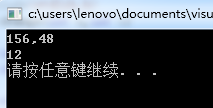




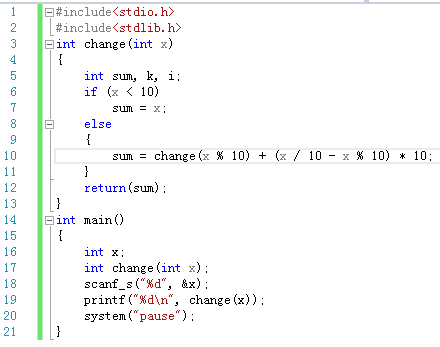
（二） 使用欧几里得公式写一个递归函数gcd（m,n)，用来计算m与n的最大公约数.

注：欧几里德算法又称[辗转相除法](https://baike.baidu.com/item/%E8%BE%97%E8%BD%AC%E7%9B%B8%E9%99%A4%E6%B3%95)，用于计算两个[正整数](https://baike.baidu.com/item/%E6%AD%A3%E6%95%B4%E6%95%B0)a，b的[最大公约数](https://baike.baidu.com/item/%E6%9C%80%E5%A4%A7%E5%85%AC%E7%BA%A6%E6%95%B0)。计算公式gcd(a,b) = gcd(b,a mod b)。





（三）写一个递归函数DigitSum(n)，输入一个非负整数，返回组成它的数字之和，例如，调用DigitSum(1729)，则应该返回1+7+2+9，它的和是19



（四）写一个递归函数DigitalRoot（n），计算整数n的数字根

注：整数n的数字根是如下定义的：它是一个整数的所有数字的和，反复相加，直到只剩下一位数字为止。

例如：1729的digital root按照如下的步骤计算：

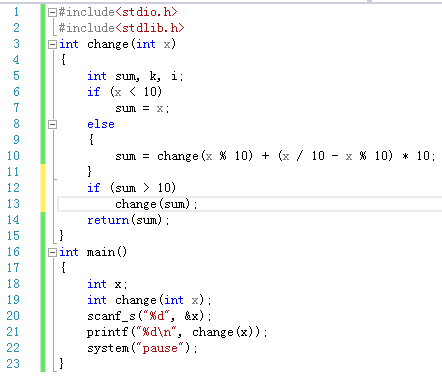
step 1:    1+7+2+9   ----->  19

step 2:    1+9    ----->  10

step 3:    1+0     ----->  1

因为第三步的结果是1，所以1就是数字根的值。

   注意：写一个纯粹的、不使用任何循环结构的递归函数。



（五）写一个递归版本的reverse（s）函数，以将字符串s转置