EX3\_task2说明文档及学习心得

这题思考了很久如何用递归去实现。在网上看到这样一段话：“递归是编程中一个相对难以理解但是却又很重要的概念. 对于从命令式语言开始学习编程的程序员天生对此有理解缺陷, 而对于从类似C++这种对函数式编程范式不友好的语言开始学习编程的程序员就更加如此了。”递归的定义说起来简单, 但是理解起来复杂, 因为递归并不直观, 平时习惯了去想如何一步步解决问题，就像做数学题的那个思维模式。相对于递归, 我觉得迭代更加容易理解。一开始一直自己想要怎么得到这种序列，研究了半天甚至还上网找到了这种序列叫做“挑剔数列”，只有4n和4n+1的整数才有这种序列。但是就算了解了这么多，也对如何让电脑按照一个指定的方法去实现没有什么大的帮助。后来才意识到，我没必要把这些细节了解得这么清楚，我应该是想如何拆分这个问题，里面哪些步骤是重复性的步骤，并且是一个函数调用自己的。

最后想到的思路是：固定一个数字放在数组首位和另一个对应位置，用一个循环实现每个数字都在首位的所有情况。然后开始调用递归函数 f( )，按照数字降序（如，n=4，则4-3-2-1或3-2-1-4，不是从最到大的数开始则，插入1结束后要调到最大的数）的顺序开始插入数字，每次判断从第2（output[1]）位到第2n位（实际可以缩短一点，到output[2\*n-(i+2)],i为当前操作的数字）能否插入该数字，能则成功插入，进入递归函数操作下一个数；不能则继续循环直到循环结束。循环结束后则要回退到上一个函数，则要先恢复现场，由于是由上一个数成功插入而来，故要将上一个数从数组中除去。递归最后成功的边界条件是数组是否已满，满了则打印整个数组，即为所求序列。打印后也要回退，与上面一样的恢复操作。当循环结束后就层层向上回退，最终结束递归函数。

通过这次实验对递归有了进一步的了解，感觉还是要多练习相关的递归题目才能更加熟练。