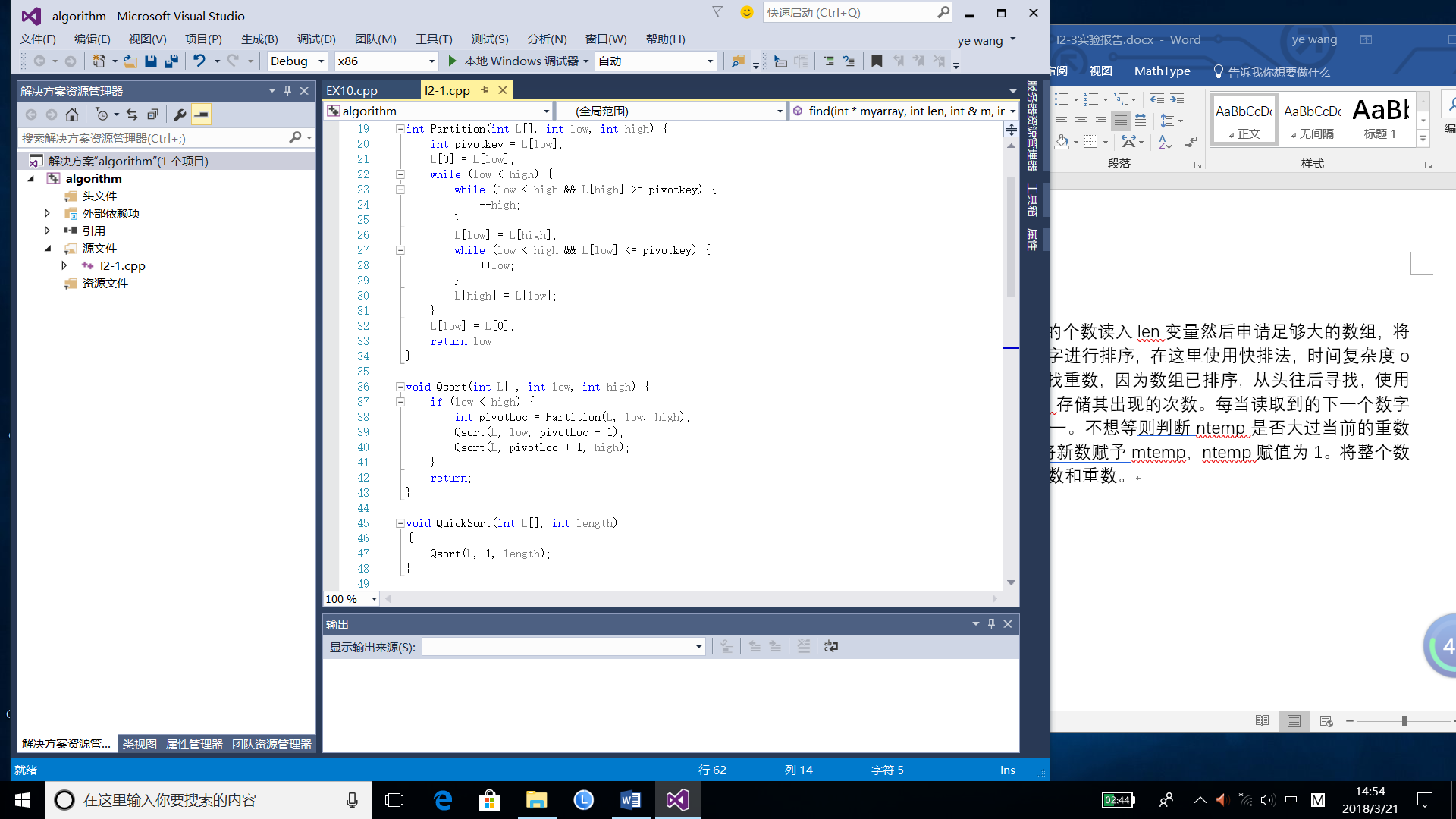
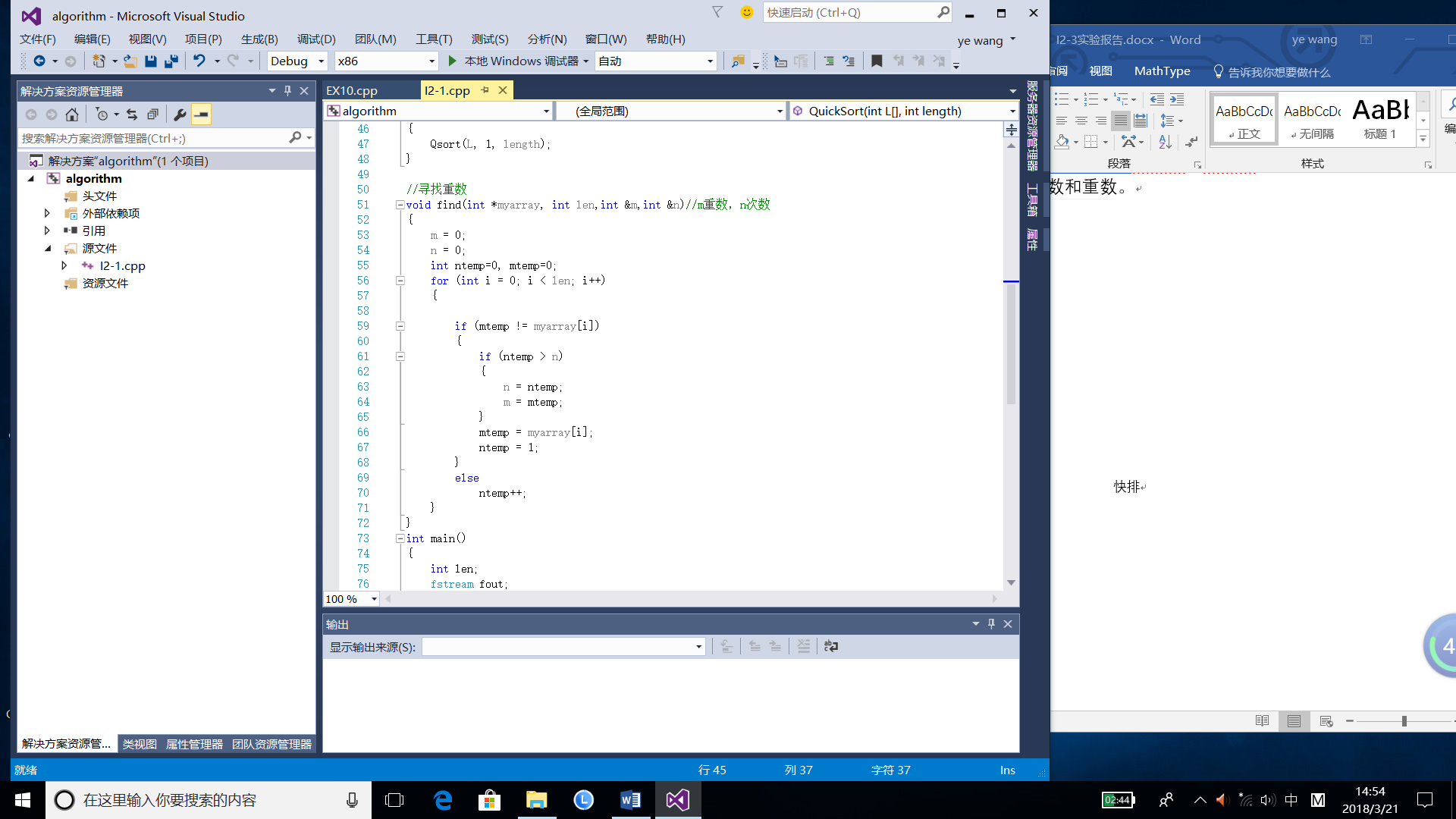
I2-1

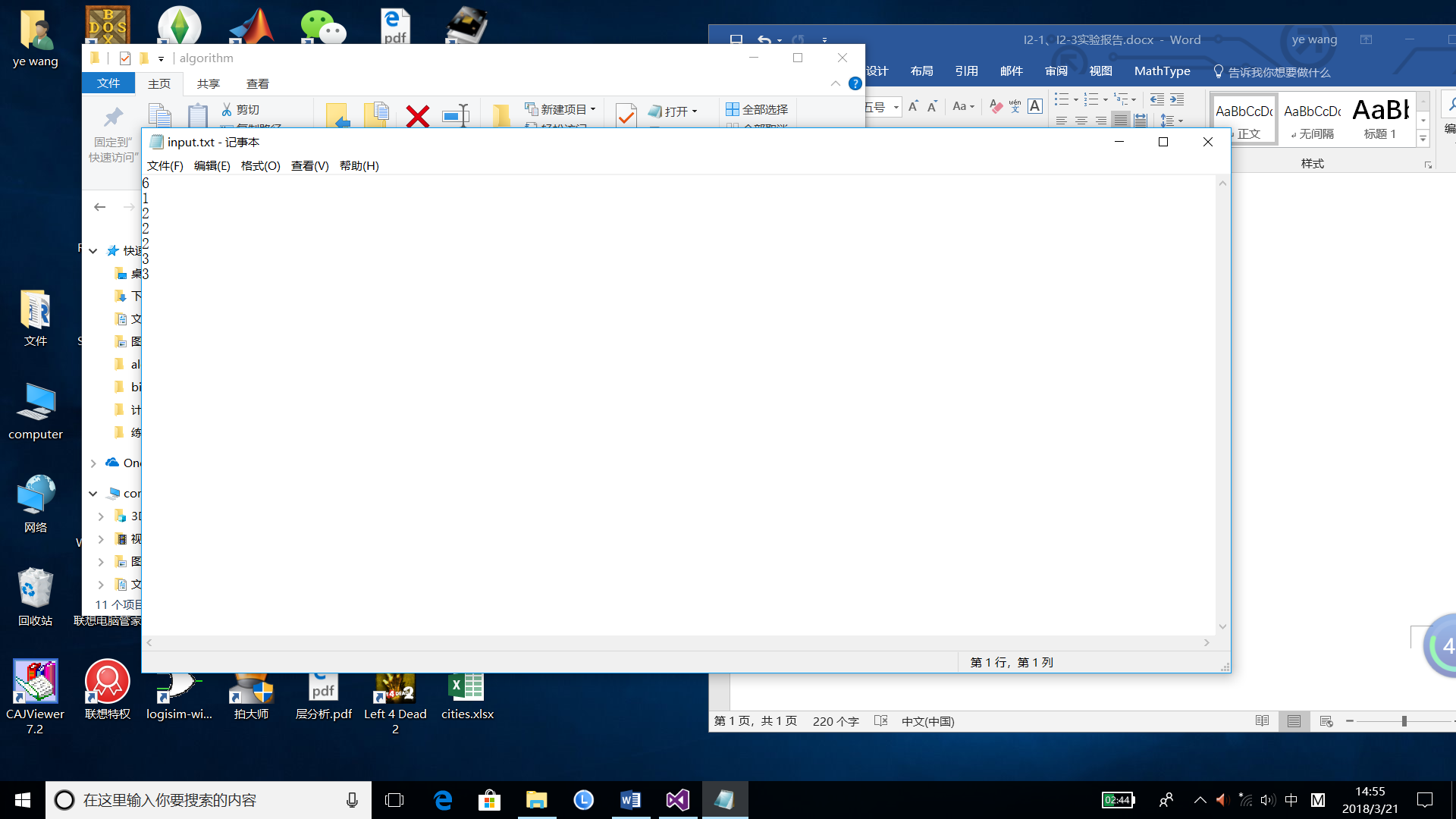
1.**题目** 略

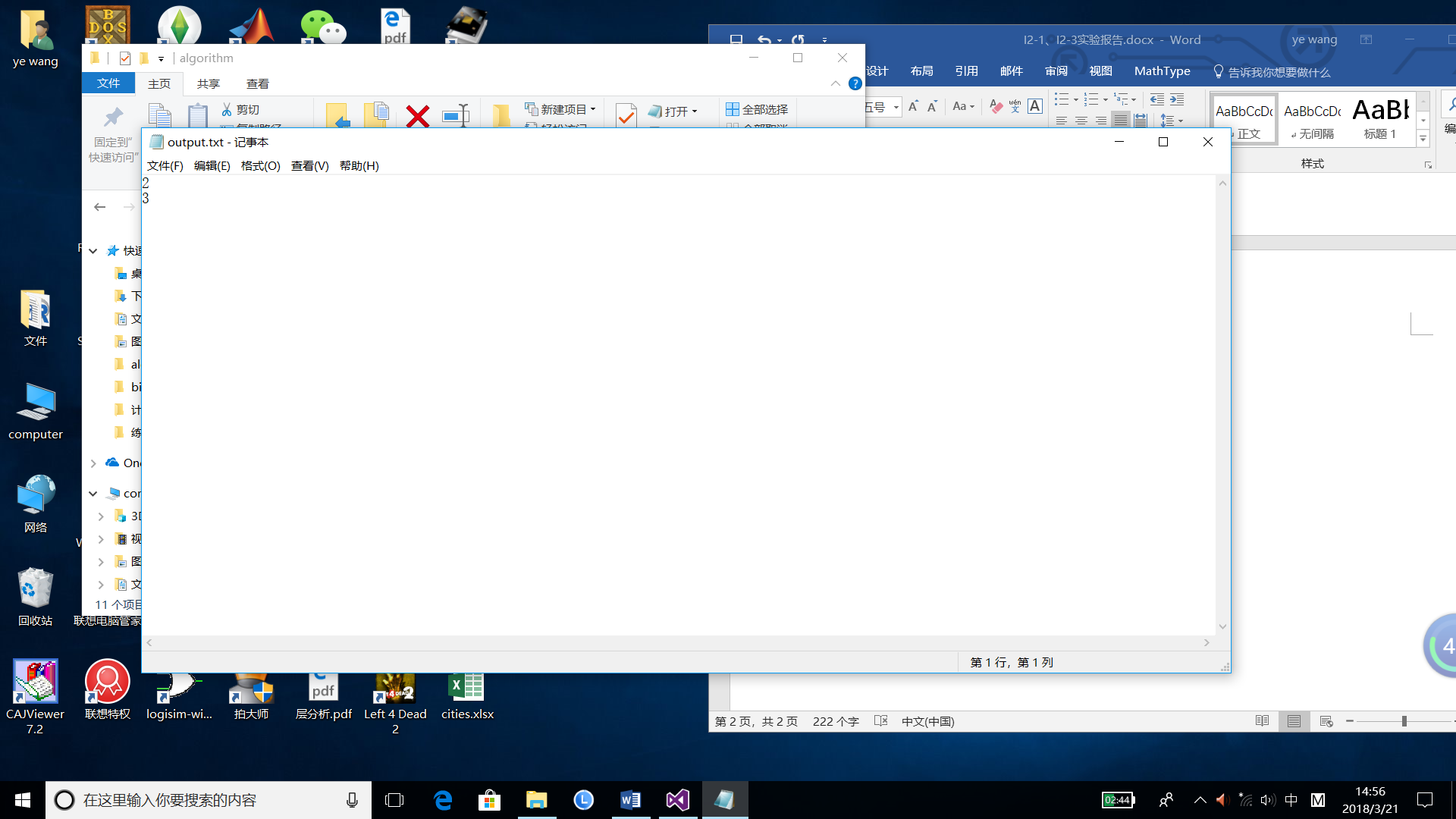
2.**算法设计：**首先将文件中的数的个数读入len变量然后申请足够大的数组，将文件中的数全部读入。对这些数字进行排序，在这里使用快排法，时间复杂度o（nlogn），然后开始在数组中寻找重数，因为数组已排序，从头往后寻找，使用mtemp存储暂时的重数，ntemp存储其出现的次数。每当读取到的下一个数字与mtemp相等，则给ntemp加一。不想等则判断ntemp是否大过当前的重数出现次数n，大则替换n。然后将新数赋予mtemp，ntemp赋值为1。将整个数组读完n，m即为重数出现的次数和重数。找到的时间复杂度为O（n）,则总体时间复杂度为O(nlogn)，空间复杂度为O(n)，总结来看，省时间，不省空间

快排

找重数

3.**实验结果**

输入

输出

**4.时间空间复杂度分析**

**算法采用递归法的思想，通过快速排序算法来完成对数组的排序，通过计算可知时间复杂度为O(nlogn)，空间复杂度为O(n)**

**I2-3**

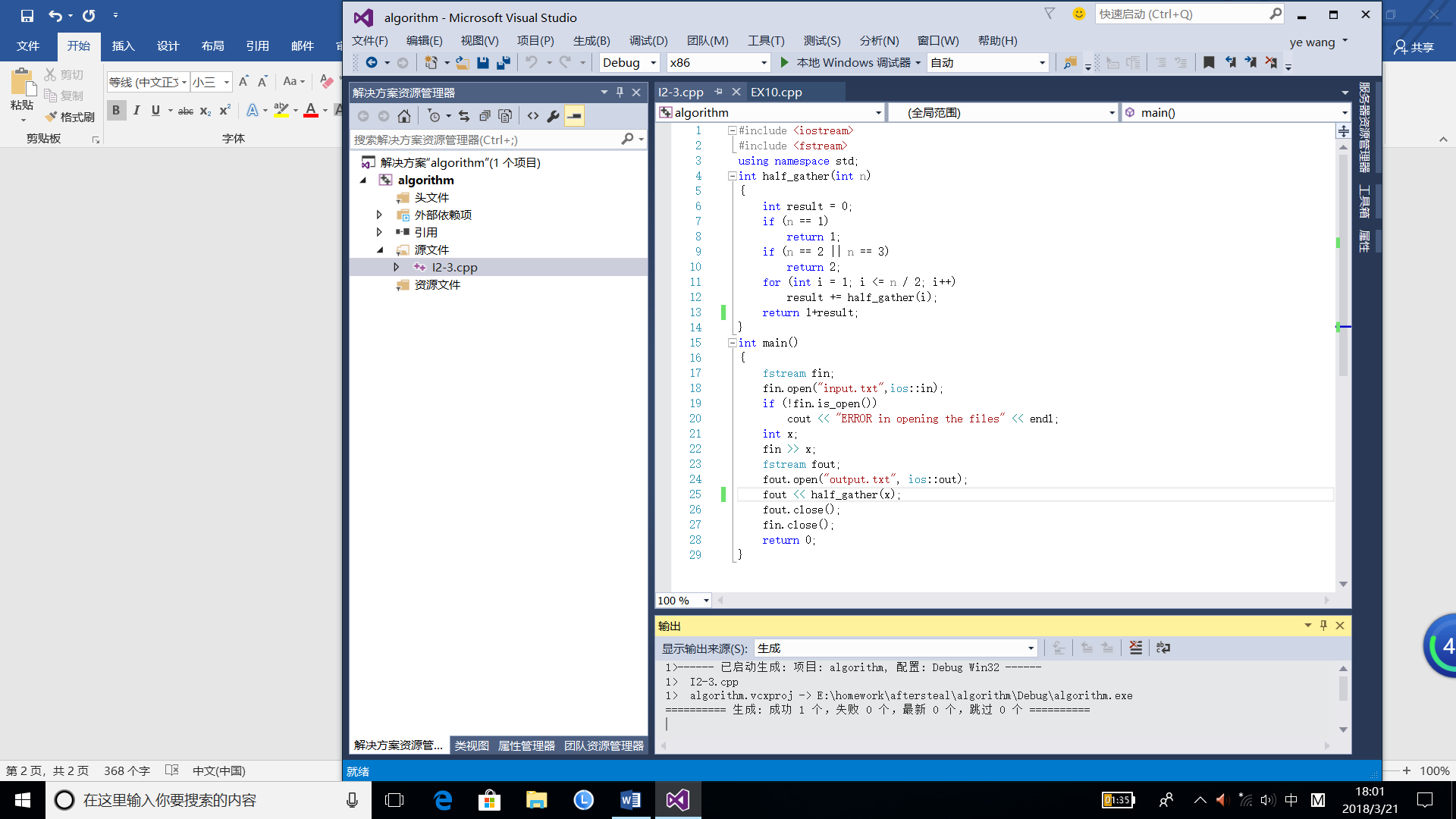
1. **题目：略**
2. **算法设计**

在这道题目里，我采用了递归的方法，设递归函数为f,观察可发现，若初始给出的数字为6，则

F(6)=1+f(3)+f(2)+f(1)

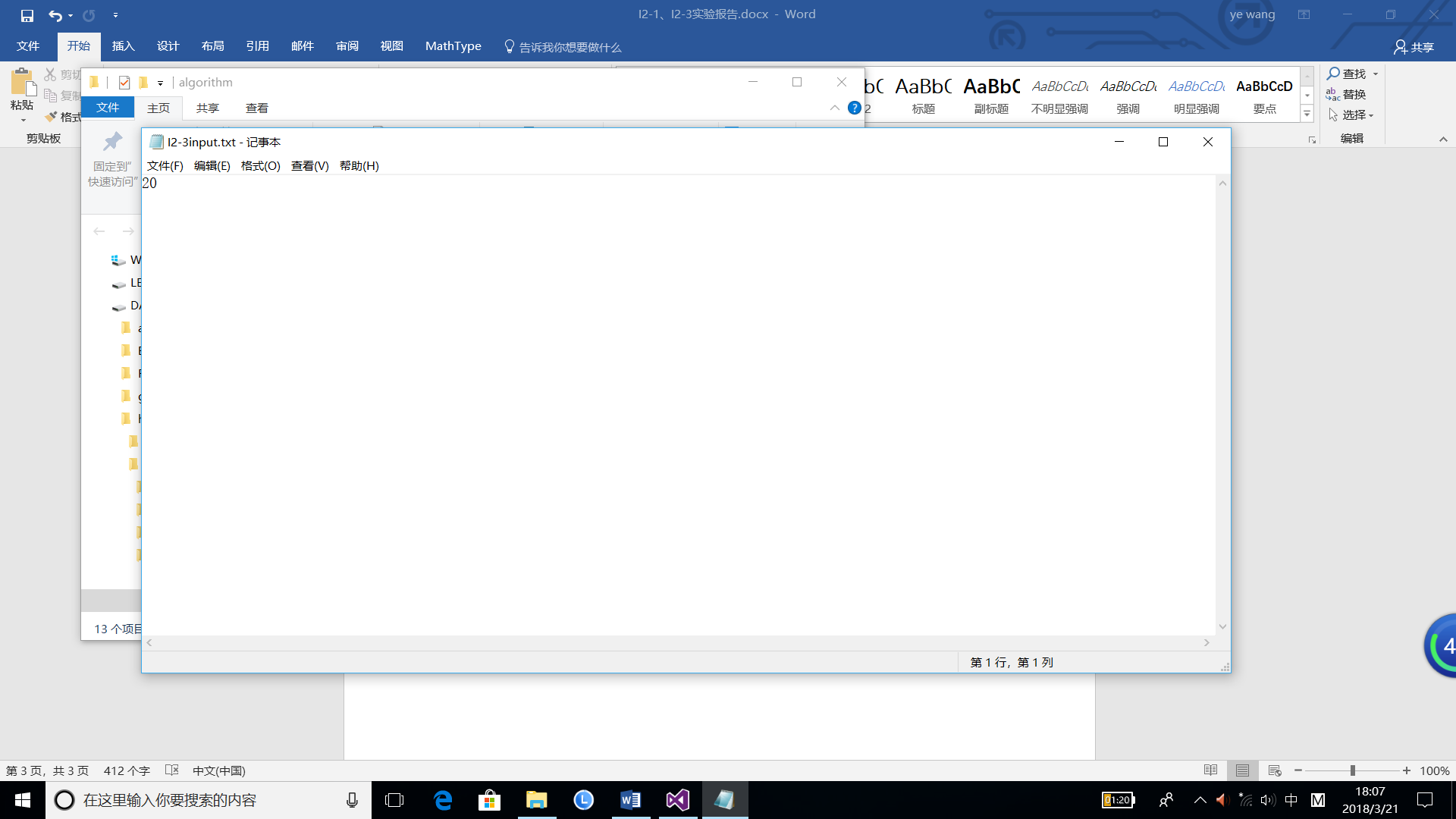
其中1指它本身，其余指由左边衍生数字所得到的半数集的数字个数。

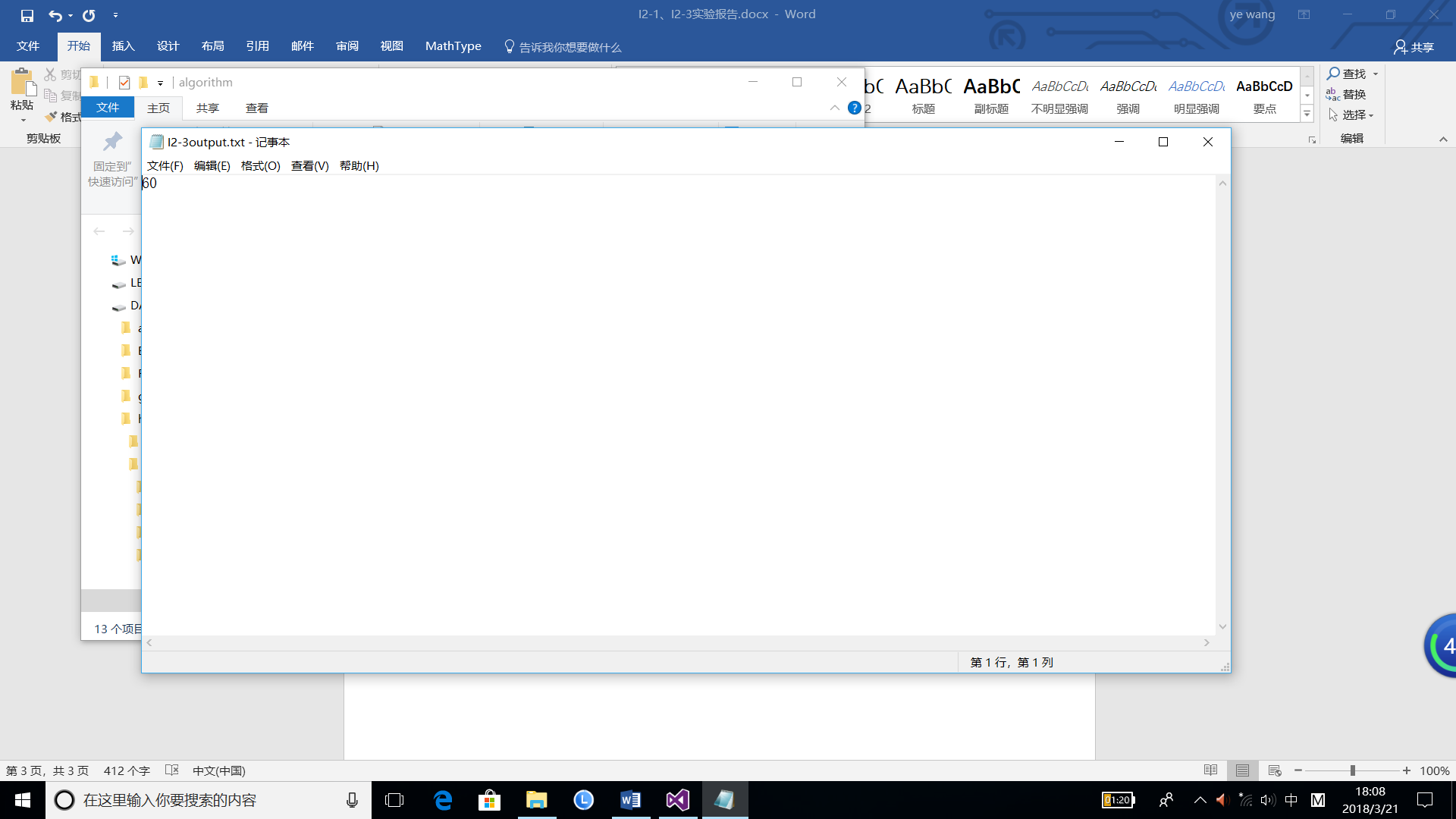
递归的终点为,f(1)=1,f(2)=2,f(3)=2。



将某数小于其一半的数代入到递归函数然后全部相加，最后结果即为某数的半数集的个数

1. **实现结果**





经与同学比对，运行结果正确。

以上程序经vs2015编译顺利通过

**4.时间空间复杂度分析**

**算法采用递归法的思想，给出递归方程解递归方程可知**

****

**时间复杂度为O(（n/2)！)**