**Instruction**

程序先读取整个excel文档里的所有人的需要的数据，即所有researcher的name，university，department，interest（topic和target），然后建立三种容器来存储数据，分别是HashMap，ArrayList，HashSet

HashMap：搜索最快的数据结构O（1），存储的时候把name作为key，其他信息作为value，每次查询根据name就能获得其对应的所有个人信息

ArrayList：所有researcher的信息都保存在list里面，因为arraylist可以动态增加减少存储的信息

HashSet：只用来存储有多少个topic，因为很多researcher的topic可能有重复，T3-3需要我们输出一共有多少种不同的topic，因为HashSet不允许存储重复的数据，所以当加入的topic发现已经在前面存储过了，那么就会直接忽略这个新加的信息，最后使用set.size()就能获得set的大小，也就是不同的topic的个数。

另一个数据结构则是linkedlist，用来存储interest node的每个topic内容，linkedlist包含两个属性，一个是topic的content，另一个是下一个interest node

所有的功能都是基于这三个数据容器（container）

1. T3-1: the number of distinct researcher也就是上面的ArrayList的size（）
2. T3-2: the number of distinct topics也就是上面的HashSet的size（）
3. T3-3: Given researcher’s name, show detail, 也就是利用hashmap的map.get(name)此处name就是key，就会返回这个name对应的detail
4. T3-4: given an interest, calculate the number of researchers who have that interest

先设置一个count=0，然后两个for循环，遍历整个上面提到的包含researcher的arraylist，然后for循环查看每一个人的interest的list里面是否包含这个topic，如果包含，就count+1，如果不包含就一直遍历下去直到整个list里面所有researcher的所有interest topic都被遍历了一遍

1. T3-4: given two interests, show the number of times they co-occur.

想法和前面一道题一样，循环包含所有researcher的arraylist，检查每个researcher的interest list里面是否同时包含两个interest，如果包含就count+1

1. 这道题要求是根据interest来判断两个researcher多有相似，其实也就是判断两个researcher的interest内容有多少相同的，程序是获得用户输入的researcher的姓名，然后先用map找到这个researcher的detail，获得他的interest然后转换成string字符串，然后和arraylist里面所有其他的researcher的interest来做对比，计算两个string的相似度。这里的问题就是如何计算两个string的相似度。

用到的的算法就是用到了stackoverflow，网址是: https://stackoverflow.com/questions/955110/similarity-string-comparison-in-java的代码和解释，里面用到了 [Levenshtein distance algorithm](http://en.wikipedia.org/wiki/Levenshtein_distance)维基百科上就有，一个很好理解的算法：

https://zh.wikipedia.org/zh-cn/%E8%90%8A%E6%96%87%E6%96%AF%E5%9D%A6%E8%B7%9D%E9%9B%A2