ALLANGRAY

Introduction – Coding Assignment

The assignment is more than delivering a working solution, we are looking for qualitative things too (e.g. SOLID principles).

In the end we would like to see what you consider to be a good solution and want to see if it is aligned with our understanding.

You may use any language you prefer (although we recommend using your strongest language) as long as you provide us with the source code and notes on how to run it.

Your source code must be provided via GitHub or BitBucket or some other public code repository store.

**Please also ensure that you do not include Allan Gray's name in your repo or assignment Readme files.*

The following assessment guide will be used:

- Does the code work?
- Does the code read well?
- Does the code handle errors?
- Is the code unit tested?

Please see below for details.
Good luck!

Twitter Feed Coding Assignment

Please write a program to simulate a twitter-like feed. Your program will receive two seven-bit ASCII files. The first file contains a list of users and their followers. The second file contains tweets. Given the users, followers and tweets, the objective is to display a simulated twitter feed for each user to the console. You will receive sample text files to use as input.

The program should be well designed, handle errors, have unit tests and should be of sufficient quality to run on a production system. Indicate all assumptions made in completing the assignment.

Each line of a well-formed user file contains a user, followed by the word 'follows' and then a comma separated list of users they follow. Where there is more than one entry for a user, consider the union of all these entries to determine the users they follow.

Lines of the tweet file contain a user, followed by greater than, space and then a tweet that may be at most 140 characters in length. The tweets are considered to be posted by each user in the order they are found in this file.

Your program needs to write console output as follows. For each user / follower (in alphabetical order) output their name on a line. Then for each tweet, emit a line with the following format: *<tab>@user: <space>message*

Here is an example. Given user file named user.txt:

```
Ward follows Alan
Alan follows Martin
Ward follows Martin, Alan
```

And given tweet file named tweet.txt:

```
Alan> If you have a procedure with 10 parameters, you probably missed some.
Ward> There are only two hard things in Computer Science: cache invalidation, naming things and off-by-1 errors.
Alan> Random numbers should not be generated with a method chosen at random.
```

So invoking your program with user.txt and tweet.txt as arguments should produce the following console output:

```
Alan
	@Alan: If you have a procedure with 10 parameters, you probably missed some.
	@Alan: Random numbers should not be generated with a method chosen at random.
Martin
Ward
	@Alan: If you have a procedure with 10 parameters, you probably missed some.
	@Ward: There are only two hard things in Computer Science: cache invalidation, naming things and off-by-1 errors.
	@Alan: Random numbers should not be generated with a method chosen at random.
```