Deloitte Social

In this exam, you are asked to develop some features for the new Deloitte Social Media application. Here are some instructions:

- You are going to use VSCode and implement the given requirements below.
- Your code must be within the method body, changing the signature or return type is unaccepted.
- You can import any standard Java 8 library.
- You can run the code pressing F5 and see the results on the debug console. Feel free to change the input on the main method as you like.

1) Likes

People at Deloitte  each other on Facebook pretty frequently. Given an array of people that liked a post, you are instructed to return a small caption with information about these people. Here are the requirements:

a) Implement the method: public static String whoLikedMe(String... names)
b) Sample example:

INPUT	OUTPUT
names {}	// "no one likes this"
names {"John"}	// " John likes this"
names {"John", " Alicia"}	// " John and Alicia like this"
names {"John", "Alicia", "Mark"}	// " John, Alicia and Mark like this"
names {"John", " Alicia", "Mark", "Alex"}	// " John, Alicia and 2 others like this"

For 4 or more names, increase the “and 2 others” accordingly.

2) External ID

Our posts will be synchronized with external systems, so we need to create a more complex unique ID. Given the ID of a post as integer, you are going to create another integer as external ID. Our System Architect decided the following:

a) Implement the method: public static int complexId(int postId)
b) Input: The input is going to be the post ID, which is an integer.
c) Output: An integer, which is the concatenation of the square of each digit in the ID.
   For Example:
   a. Input: 15, Transformation: 1^2=1 - 5^2=25, Output: 125
   b. Input: 9118, Transformation: 9^2=81 - 1^2=1 - 1^2=1 - 8^2=64, Output: 811164
   c. Input: 752, Transformation: 7^2=49 - 5^2=25 - 2^2=4, Output: 49254

3) Coffee Time!

In Deloitte we value coffee, that's why our app is going to suggest the best coffee store for our 10 minute breaks. The directions are going to be an array of characters (you don't need to check for the validity of the array) with a random assortment of direction letters 'n'(for north), 'e'(for east), 'w'(for west) and 's'(for south). Each step requires 1 minute, the whole path must bring you back to Deloitte and the total time must be exactly 10 minutes as no one wants to be late(or early right?). Here are the requirements:

a) Implement the method: public static boolean isValid(char[] steps)
b) Input: An array of directions. E.g.
   new char[] {'n','n','s','s','n','s','n','s','n','s'}
   new char[] {'w','e','w','e','w','e','w','e','w','e','w','e'}
   new char[] {'w'}
   new char[] {'n','n','n','s','n','s','n','s','n','s'}
c) Output: Boolean.
   True if: the path **takes 10 minutes** and **ends where it started**,
   False otherwise.

   It does not matter if the path is "logical". You need to check if starting from a point A and following the path(the character array) you get back to point A.