Please aim to complete this code test within 48 hours. If you are unable to answer a question, be sure to show as much of your work and thought process as possible. While we want to be able to run your solution, we are much more interested in how you arrived there. You can use online resources to help with your solutions, but please make a note of them in your answer.

Feel free to use any programming language you like on the coding questions.

Question 1

On shapeways.com, you can create a new user account by registering with an email address, username, and password. There is a register form on https://www.shapeways.com/register, https://www.shapeways.com/education, and also on the "Join" popup modal displayed on the top navigation bar of most pages. The form itself and the back-end are all implemented with the same shared code.

Come up with functional test cases which test the various flows for user registration, account creation, and account verification. Don't worry about testing logging in with existing accounts: we're looking for you to focus on new account creation and verification. You do not need to actually write tests, just provide the cases you would like to test for this feature.

Question 2

Write a function that accepts a string with some number of consecutive repeating characters and returns a copy of the string with consecutive repeating characters replaced by a count of the repetitions.

"sssssTTTTTToNNps" ⇒ "s5T6oN2ps"

Question 3

When an item is ordered on Shapeways, we create a production order ("ProductionOrder") for that item. If an ordered product has 2 items, a production order can be "split" into two. This results in two new child production orders.

So if a customer orders a product for a pair of earrings, we will "split" it. So we now have 3 total production orders:

- **Parent Earrings (PE)**
- **Left Earring (LE)**
- **Right Earring (RE)**

```
  PE
 /   \
LE   RE
```

For our purposes, let's represent a ProductionOrder as a class with an ID and a parent ID:

```
# Python
class ProductionOrder(object):
        id = None
        parent_id = None

// Java
public class ProductionOrder {
        public int id;
        public int parentId;
}
```

Write a program which can accept a list of production orders:

[ProductionOrder(), ProductionOrder(), ……. ProductionOrder()]

that will find and return all of the leaf production orders -- i.e. any production orders that do not have any children.

For instance, a list has production orders [A, B, C, D, E, F, G]

```
  A            D        G
 / \           /
F   C         B
     \
      E
```

F, E, B, and G are leaf production orders because they do not have any child production orders.

Question 4

Write unit tests for each of your programs above.