Lab wk1-3: Recursive generation of Combinatorial Objects

Generating the Power Set of a set recursively

The bitstring representation of sets. Given a set A containing n-elements. Associate each elements with a number from 1 to n=|A|, then any subset can be represented by a bitstring of length n. E.g.
A = {a,b,c,d} then the subset {a,c} is represented by 1010 if we have associate a with position 1, b with position 2 etc. This is a bijection between P(A) the set of all subsets of A with the set of all the bitstrings of length |A| Subsets are examples of combinatorial objects. **You will use this representation later but not in this lab!**

The goal of this lab is to practice your recursive programming skills and prepare you for the next assignment.
Given a set A, the set of all its subsets is called its Power set, and is usually denoted $\mathcal{P}$(A). The **number** of subsets of a finite set = |P(A)| is = $2^{|A|}$.
For example: if A = {a, b, c} then $\mathcal{P}$(A) has 8 elements (written |$\mathcal{P}$(A)| = 8) since |A| = 3 and the number of subsets is 2^3. The subsets are: {}, {a}, {b}, {c}, {a,b}, {a,c}, {b,c}, {a,b,c}

Write a recursive Java method that will generate all the subsets of the letters in a string (which is passed as an explicit parameter) and return the subsets as an ArrayList of strings. You should follow the high-level pseudo-code given below. (Your program will not necessarily generate the subsets in the order shown above!)

```
subsetGen(setString : a string with the characters that make up the set)
     let A and temp be empty ArrayLists
     if len(setString)>0
          temp = subsetGen(set without last character)
          //  now loop over temp and create the subsets with and without
          //  the last character of the original string
          for (int i = 0; i < temp.size(); i++)
               A.add(temp.get(i))  // adds subsets without last character
               A.add(temp.get(i) + " " + setString(length-1)  subsets with
          return A
     else
          A.add("")   // array list with only the empty string
          return A
```

Before you implement this make sure you can draw the call tree if it is called on "abc" so you are sure you understand what is going on.

A template for your program is provided on PolyLearn.
Deliverable:
Source code for a single class, **SubsetGen.java** with the method described below. Submit on PolyLearn.

public class SubsetGen
contains the method
public ArrayList<String> getSubsets (String word) {}