🔁 U4L13 String Scramble A 🪢

Take your time to read through all the examples and go step by step. Even if you feel you understand what is happening, after you complete a section, make sure you reveal algorithms and hints where provided.
After you finish, check your answers against the College board rubric with your partner, or if one is not assigned, then check with the person beside you.

Part (a) Write the method scrambleWord

Write the method scrambleWord, which takes a given word and returns a string that contains a scrambled version of the word according to the following rules.

- The scrambling process begins at the first letter of the word and continues from left to right.
- If two consecutive letters consist of an “A” followed by a letter that is not an “A”, then the two letters are swapped in the resulting string.
- Once the letters in two adjacent positions have been swapped, neither of those two positions can be involved in a future swap.
- The following table shows several examples of words and their scrambled versions.

`word`	Result returned by `scrambleWord(word)`
"TAN"	"TNA"
"ABRACADABRA"	"BARCADABARA"
"WHOA"	"WHOA"
"AARDVARK"	"ARADVRAK"
"EGGS"	"EGGS"
"A"	"A"
""	""

```
public class ScrambledStrings
{
   /********************** Part (a) **********************/

   /** Scrambles a given word.
    *  @param word the word to be scrambled
    *  @return the scrambled word (possibly equal to word)
    *  Precondition: word is either an empty string or
    *                contains only uppercase letters.
    *  Postcondition: the string returned was created
    *                 from word as follows:
```

```
     *   - the word was scrambled, beginning at the
     *     first letter and continuing from left to right
     *   - two consecutive letters consisting of "A"
     *     followed by a letter that was not "A" were
     *     swapped
     *   - letters were swapped at most once
     */
    public static String scrambleWord(String word)
    {




    }
}
```

Rubric

5 points Intent: Return the number of leap years in a range

- +1 Accesses all letters in word, left to right (no bounds errors)
- +1 Identifies at least one letter pair consisting of "A" followed by non-"A" Reverses identified pair in constructing result string
- +1 Reverses identified pair in constructing result string
- +1 Constructs correct result string (Point lost if any letters swapped more than once, minor loop bounds errors ok)
- +1 Returns constructed string

Reflection

After you complete the problem sets, rate them and assess your ability to meet the goals of this lab.
Highlight the general category which matches your feeling about this exercise.

1 - easy	*2 - a little challenging*	*3 - challenging*	*4 - difficult!*

Did you meet the objectives?

- ... access all letters in word from left to right, identifying the pairs of letters to be swapped.

- reversing pairs of adjacent letters of word whenever the pair consisted of "A" followed by a letter that was not "A"
- All letters of word that were not involved in reversals were to be in their original positions in the result String, which was then returned by the method.

Copy a screensnip of the part of your code where you checked if the two letters were the same character and if not the character was added to the returned word. Did you make use of the equals method and the substring methods? Describe how the code works.

ANS: