NOTE: Numbered problems for **Hints** are located in Solutions to String-1 Regular Section.

01 **conCat**, Hint: 06 **firstTwo**, 11 **left2,** 10 **nonStart**

02 **first3**, Hint: Like 06 **firstTwo**, 28 **extraFront**

03 **conCat2**, Hint: Like 09 **comboString**

04 **isWeekend**, Hint: String Basics 3: **isSummer** (look at both versions)

05 **bothEnds7**, Hint: 16 **endsLY**, 20 **hasBad**

06 **secondHalfTwice**, Hint: 07 **firstHalf** 05 **ExtraEnd**

07 **reverse**, Hint: Create 3 substrings, one for each of the 1st, 2nd and 3rd characters.
 Then concatenate them together in reverse order!
 Look at: 11 **left2,** 12 **right2**

08 **dropMiddleChar**, Hint: Create 2 substrings.
 The **1st** will be the 1st half of the string up to,
  but not including, the middle character.
 The **2nd** will be the 2nd half, starting with
  the character **AFTER** the middle character.
 Then concatenate them together.
 Look at: 07 **firstHalf,** 19 **middleThree,** 15 **middleTwo**

09 **goodBad**, Hint: 32 **withoutX,** 33 **withoutX2**
   OR: you can do it the LONG way (described below).
   There are four different cases you need to consider in the following order:
   (1) The single word "good". If the string equals "good", just change it to "bad".
   (2) The string **BOTH** starts <u>AND</u> ends with "good", e.g. **good**AndgoodAnd**good**.
       Use substring to get the middle of the string, e.g. "AndgoodAnd".
       Change the string to "bad" + middle + "bad".
   (3) The string starts with "good" (but does not end with "good"), e.g. "goodEnough".
       Use substring to get all but the first 4 characters, e.g. "Enough".
       Prepend "bad" to this substring, i.e. change the string to "bad" + **lastPart**.
   (4) The string ends with "good" (but does not start with "good"), e.g. "reallygood".
       Use substring to get all but the last 4 characters, e.g. "really".
       Append "bad" to this substring, i.e. change the string to **firstPart** + "bad".

10 **nearEnd**, Hint: 08 **withoutEnd** 16 **endsLY** String Basics 3: **almostEndsFUL**
   2 ways to think about this:
   (a) create a substring without the first and last letters,
        then use **startsWith**() and **endsWith**()
   (b) Use the version of **startsWith**() with 2 parameters.
        See: the version of **endsLY** that uses **startsWith**() or
             String Basics 3: **almostEndsFUL**