Scenario: Write code to find the first time in a string when a letter is duplicated.

1. Scenarios to consider:
   1. Words with a duplicate letter (eg: paper) -> p
   2. Words with more than 1 duplicate letter (eg: discuss) -> s
   3. Words with more than 1 set of duplicate letters (eg: banana) -> a
   4. Words with no duplicate letters (eg: card) -> none
2. A hashset is ideal for this case, as a hashset does not allow any duplicates in it. Thus, when adding the character into the hashset, the first occurrence of encountering a duplicate can be stopped by a condition. Moreover, the performance for adding elements and searching for elements are O(1).
3. Boundary conditions:
   1. Words with duplicate letters of different cases (eg: Bob)
   2. Words with different forms of letters (other Latin languages) (eg: nñ)
4. Outline:
   1. Create an empty set
   2. Create a counter variable (int) to keep track of which letter the code is on
   3. Insert each character of the word into the set
   4. Use “if” to look for duplicates before inserting
   5. counter + 1 if not a duplicate, break the loop and return counter if duplicate is encountered
   6. If the full loop is ran, then return no duplicates