* Overall: Appropriately uses hashMap/unordered\_set/dictionaries
* Given an array, calculate the mode of the array i.e the item that shows up most often. If there are multiple modes, return any one of them.
* O(N) runtime, O(N) space
* Students may assume array has at least one element: no need to test
* Test where length of array == 1
* Test where length of array > 2 and two or more elements are same (normal case)
* Test where length of array > 2 and all elements are different
* Given two strings, write a method to decide if one is a permutation of the other.
* O(N+M) runtime, O(N+M) space
* Test true where each string does not have duplicate letters
* Test false where each string does not have duplicate letters
* Test false where each string has completely different letters from each other.
* Test true where each string has duplicate letters
* Test false where each string has the same set of letters but different numbers of duplicates.
* Check whether there exists a 1-1 mapping that converts one string to another.
  + O(N) runtime, O(N) space
  + Test false with different length strings
  + Test false when mapping many to 1.
  + Test false when mapping 1 to many
  + Test false when mapping many to many
  + Test true with equal strings
  + Test true with unique characters
  + Test true with duplicate characters.