I can take no credit for working out the details for properly sending data to the second viewController (please don't award full marks for that). I initially wanted to try to access my second view from a TabController, and had success at touching the second from the first, but could not manage to access any of its properties. Using the segues built in to the NavigationController was easier, but I still couldn't figure out how to push data into my second view without help from prof Liss.

As for my approach to the heart of the assignment, it was obvious from the start that I would need to iterate over the input string. After a minor hiccup with comparing Strings and Characters, I decided to simply be specific in my dictionary declaration so that I wouldn't have to do any casting elsewhere in the program.

I realized that I needed to find a way to sort my dictionary in order to meet the display requirements (listed in frequency order). I settled on something I did playing with Python a few semesters ago: stripping the dictionary into separate, index-corresponding arrays. Then JUST before I started to code that out, I was hit with the idea of using a single array of tuples. My first such try listed the tuples as (Character, Int), but I was running into comparison issues during sorting and when I reversed the order, my sorting worked fine. This only required a small change in how I was building my output string to be implemented smoothly.

During construction of my output string, I quickly realized that Characters like " ", "\t", and "\n" would need special handling in order to be escaped properly during display. I decided to implement a switch and it worked out rather nicely.

I did all of my work directly in xcode, so i have no pseudocode, but I will pseudocode an outline of my work here:

```
Navigation Controller loads the first view
First view:
  link variable to textView object
  overload viewController's prepare method to send data from variable
into the resultsView on segue (activated by the button)

Results view:
  define variable to hold raw input
  link variable to output textView object

  define func to loop over the raw string & build a dictionary
    check if char exists in dictionary
      if yes: count += 1
      if no: add to dictionary, count = 1
    return dictionary
```

```
define func to strip dictionary into an array of tuples
  loop over dictionary
    add element to array
  return sorted array

define func to build & output a single string
  for tuple in sortedArray {
    format tuple elements via switch
    append values to string
  }
  correct output string for no data
  update textView

upon initialization:
  run nested functions using raw input as primary argument
```