Derek Yang
CS 362
Random String Quiz

The following assignment required me to look at a test function called testme() and run a random test using two functions inputChar() and inputString(). I had to keep the following things in mind when testing the code: to keep the code concise and run under 5 minutes; to make sure to use all ASCII characters; and finally to make sure that my inputs were really random.

I chose to start by looking at what would cause an error with exit 200 to occur in the code. Luckily, the requirements were very straightforward in the last if statement of tesme(). First, I needed to make sure that inputChar() would go through all ASCII characters so that all of the states were triggered. This meant that the inputChar() function would have to go through all the ASCII characters randomly. At first, I was a bit stumped by how many ASCII characters I needed to use, I asked myself was it 256? After a bit more thinking, I looked at the following resource:
https://stackoverflow.com/questions/27679137/what-does-256-means-for-128-unique-characters-in-ascii-table

This post basically explained that there are realy just 128 total characters which are used in the program. 0 to 31 ASCII chars are control, 32 to 127 are printable characters, and 128-255 are extended ASCII codes. Therefore, I just really need to use range of 32 to 127. This will really cut down our testing (and extra errors) if I just use the range of 32 to 127. Honestly, this was the longest part of the assignment. Finally I fed this in using inputChar(). I had to make sure to stop the program immediately to view my results as it quickly went to stage 9.

The next step was to submit inputString(). I first looked at which position the terminating character needed to be in. Therefore, I made an array[5] and pointed to that array so that I could change each character's values. It took me a while, but I figured out that you need to have the word reset in order to finally pass the error. It took me a while because I ran it at first and realized how slow it was when I was using the full character values. It decreased in time when I excluded ranges of characters which are not used in the word "reset". Finally after a couple tries, my random testing code was producing results in under 5 minutes.