

Andrea Dias
5/6/18
CS 362-004
Randomstring.pdf

Random Testing Quiz

I was a little confused as to how to create a random test generator but doing this assignment has definitely cleared things up for me. I started off by looking at we were required to do and noticed that we needed to create a function that returned an array of chars and a function that returned an array of strings. I then looked at what the code was testing for.

This is when I decided that the array I needed to create in the inputChar function would contain all the characters that the program checks for. I set up a variable that would hold the value of a random number from 0 to 33 using the rand function. This random number would correspond to an element in the char array that I created within the inputChar function. I set up the inputChar to return a random element in the array that would be called by testme. Each iteration of the loop within the testme function would call inputChar which would then return a random character from that char array.

Understanding what was exactly needed for inputString was a little more difficult. Since inputString was supposed to return a complete string and not just a single char, it made me think a little harder about how to accomplish this. I noticed that it was checking to make sure the elements of the array created the string "reset". This prompted me to create a char array that held "reset" in positions 0, 1, 2, 3, 4 and the 5th element would be a null terminator. I then assigned two variables that would both spit out a random number between 0 and 5, to correspond with the 6 elements in the array. These two variables, that would correspond with an element of the array, would then be swapped and the whole string would be saved into a new array called stringArray. An example would be as follows:

The string we are manipulating is "reset".

One variable is randomly set to 1 and the other is randomly set to 2.

The two variables would correspond with element[1] and element[2].

These elements are then swapped, creating "rseet" instead of "reset".

Each time `inputString` was called in the `testme` function, it would return the letters 'r', 'e', 's', 'e' and 't' in a different position each time. This is how the function achieves randomness except I modified the code so that the null terminator was always at the end.

The `testme` function will call the `inputChar` function and will receive a random char from the array inside `inputChar`. The `inputString` function will return a random arrangement of the letters r, e, s, e and t. Once it receives the correct char from `inputChar` that is looking for, it will continue on its loop until checks for the string from `inputString`. The loop will continue until `inputString` returns the chars r,e,s,e,t and the null terminator in exactly that order.

My test got pretty good coverage. My lines executed are at 97.50%. my branches executed and calls executed are both 100%. The result for the taken at least once test was at 96.15%.