Jason Lim

CS 362

Quiz 3

Random Testing Quiz

I developed the random tester for inputChar and inputString first by examining all the functions so far in testme.c file. The testme document seems to compare all of the characters and strings using the inputString and inputChar function and checks to make sure that what is in the file such as the characters ike in the first test case '[' are available as per a random test. We will see what the char or strings are and set the state's value based on this. If there is no value per the testme function then we will print error and exit out of the function. The main function drives the testme and also seeds random so that we are using random values for the two functions above: inputString and inputChar.

I used resources online to check to see how we would randomize these input functions. For example, the first function, inputChar I used random value between 32 and 125 for ascii conversion because these are the relevant char values that we want to test. For inputString we create an allocation for the memory for the string and then have a randomizer for 5 values for the 5 states in our testme function so that we can have values between 97-122 in ascii which are a-z in string values for all lowercase values in the alphabet which we are testing in testme. ** I had originally used inputStr for 97-122 ascii but now instead used "reset" char values only for testing purposes.

In lines 8-9 I had initialized a char called charRandomize which would take a random ascii value between 32 and 125 and then return the char. The rand which was seeded in main would take care of randomization. In lines 16-17 I defined a random str and then used calloc to define memory for the string and also set the memory to zero. We need the size of items and the size of the array for calloc. Additionally in the for loop in lines 18-21 I had a for lop for 5 items to put into the string and a random ascii value between a-z. Once the value hits reset the testme function will throw an error and the iterations will end. The line 22 returns the final null termination to null terminate the string which we use for string programming in C. Then we return the ranomdstr. In test me we see that if we hit "reset" then we will exit the program and will finish our coverage of our random test. Until the string is 'reset" and null terminated we will not exit out our program.

This was the way I devised and designed my random tester.

P.S. I had originally put all characters a-z for the string utilization for randomization but changed this to only use the values in 'reset' to terminate the program sooner.