Hughes, Joshua L CS362 Random Test Quiz 27 OCT 2018

In order for the random tester to fail, the testme() function must first receive eight ascii charactors (‘[‘, ‘(‘, ‘{‘, ‘ ‘, ‘a’, ‘x’, ‘}’, ‘)’ , ‘]’) from the inputChar() functions. Once each of these characters are received (in the necessary order), the state variable in testme() is incremented. Once the state variable reaches the value of 9, testme() will fail if and only if it receives the string ‘reset’ when it it calls inputString();

I first implemented inputChar() by creating a character array of the necessary ascii characters that are needed for testme() to increment the state variable. I used a random number generator to randomly proivide an integer between 0 and the (length of the array -1). This allowed for the inputChar() function to randomly select a character from the array.

When implementing the inputString() function, I realized I could use the inputChar() function to help randomly produce a string. I refactored the inputChar() function to include the ascii chartors (‘r’, ‘e’ ‘s’ ‘t’) and a null terminator (‘\0’). By adding these characters to the inputChar() function, this increased the possible output options of the function when providing a character to testme(). This also allowed for inputChar() to be used to build a random string when called from inputString().

Once testme() state variable is set to 9, it needs a string of ‘reset’ in order to fail. This string is six characters long – each of the letters and then the null terminator. In order to add more randomness, inputString() was implemented to build a string up to eight characters long. Since the null terminator is randomly returned from inputChar(), inputString() returns a string of random length: 9- 8 charactors long.

I found that the script quickly gets the required input to put the testme() state value to 9; however the requried string of ‘reset’ takes much longer to achieve. I actually had inputString implemented to be up to 10 charaters in length, but feared it may exceed five minutes before it reached a failure state.

Once inputchar() returns the following charactors in the following order: ‘[‘, ‘(‘, ‘{‘, ‘ ‘, ‘a’, ‘x’, ‘}’, ‘)’ , ‘]’ and then receives the string input of ‘reset’, it prints the message ‘error’ and exits with a status code of 200. The Makefile captures all of the output of testme and stores it in a file called test_results.out. The clean option in the Makefile will delete the gcov associated files, the test_result.out file and the executable file.