**CS362: Random Testing Quiz**

Jessica Aten

The very first step I did was rather unproductive. I wrote inputString to generate a char array of random length with any characters, and I wrote inputChar to generate any random character. While this was running, I had time to look at the testme() function and realize that this was not an efficient way to test the function. I decided that I should limit the input pool.

First, I studied the while loop inside the testme() function to determine whether I needed to include all of the ASCII characters or just a subset of them. I realized that I could limit the length of inputString() to 6 characters, with the final character being '\0'. I also determined that the first 5 characters of inputString needed to be "reset," which allowed me to also reduce the input domain to the characters between 'e' and 't', inclusive.

At this point, I wrote the char *inputString() function. A char array called randString with a length of 6 was initialized. Then, a for loop was used to generate random numbers in the range from the char values for 'e' to 't' for the first 5 characters in the array randString. In order to produce these random numbers, I used the rand() function and did the following calculation.

(rand() % (upperLimit – lowerLimit + 1)) + lowerLimit

'e' or lowerLimit= 101; 't' or upperLimit= 116.

(rand() % 16) + 101

Then, the 6$^{th}$ character was set to '\0' and the address was stored in inString, which the function returns.

Next, I wrote the function inputChar() to return a random character in the range from 32 to 126 to encompass the alphabet and special characters. The following calculation is used

(rand() % (126 – 32 + 1)) + 32

(rand() % 95) + 32

Next, I ran the program and was relieved to see that it ran relatively quickly compared to my first attempt, and that the branch coverage was within the specified range for the assignment.

```
flip3 ~/randomquiz 164% make clean
rm -f testme *.gcov *.gcda *.gcno *.o
flip3 ~/randomquiz 165% make testme
gcc -o testme -g testme.c -Wall -fpic -coverage -lm -std=c99
testme >> testresults.out
make: *** [testme] Error 200
flip3 ~/randomquiz 166% gcov testme.c -b
File 'testme.c'
Lines executed:97.30% of 37
Branches executed:100.00% of 52
Taken at least once:96.15% of 52
Calls executed:100.00% of 10
Creating 'testme.c.gcov'
```