

Chido Nguyen

CS362

### Random Input Quiz

The first thing I did was look at the condition statements under the while loop. What characters and trends are set to trigger an increase in the state variable. The if conditions checked various ascii symbols and letters so there was not a short-cut way to isolate the specific instances, we'll have to consider them all during our random input testing. Ascii decimal values from 32 all the way to 126 inclusive. In the "inputChar()" function we generate a random character to be returned. We've already seeded the random function in the main method, no need to redeclare a seed again we'll just call it. Like I stated earlier we only want the characters with decimal values (ascii) from 32 and 126 inclusive. Normally `rand() % 100` gets us 0-99, to get 100 we'll have to add 1 to the post random value or do modulus of 127 (126+1). I chose to go with the post random plus 1 process, this will get us all values from 1 – 126. Now to eliminate everything below 32, I wrapped this under a do and repeat while the random integer is less than 32. Then I recast the integer back to a char and return it. This hopefully satisfies the randomization requirement.

For `inputString()` it required a little more tinkering. Looking at the if statement conditions it spells the word "reset" which is 5 letters, but we want an array of 6 characters to account for the null terminal. I declare a string pointer and malloc it to fit 6 characters. Then I memset the whole string to be filled with null-terminals, I won't have to worry about post-processing the string later. Since we know the letters it looking for to trigger the error I created a 4-character array of R E S T which is required to spell reset. Saves us time on having to randomize through 26 different alphabet characters. We're looking for a 5-letter word (reset), I run a for-loop five times to randomize 5 characters to input into the string. I call `rand` and modulus it by 4 this will get me an integer value from 0 to 3 which lines up with our array index for R E S T. After randomizing an index position, I pull the character from my array of 4-characters and insert it into the string. After finishing the for loop, I return the string. I achieve the randomization aspect by randomizing which characters are picked from the 4 character array, while still giving it a chance to hit our error trigger at a faster rate.

Here is an example of a run. We can see that character is randomized to achieve state 9, which then starts examining the string for our target phrase. It randomizes and when reset is hit exits with Error 200.

```
Iteration 1230: c = 8, s = etrrr, state = 9
Iteration 1231: c = *, s = stsst, state = 9
Iteration 1232: c = ;, s = etsse, state = 9
Iteration 1233: c = z, s = eerrt, state = 9
Iteration 1234: c = V, s = tserr, state = 9
Iteration 1235: c = 5, s = tesrs, state = 9
Iteration 1236: c = k, s = ttsse, state = 9
Iteration 1237: c = k, s = eetre, state = 9
Iteration 1238: c = H, s = errrt, state = 9
Iteration 1239: c = z, s = tstse, state = 9
Iteration 1240: c = :, s = tteee, state = 9
Iteration 1241: c = e, s = etsee, state = 9
Iteration 1242: c = 8, s = esetr, state = 9
Iteration 1243: c = D, s = tssss, state = 9
Iteration 1244: c = L, s = rsrte, state = 9
Iteration 1245: c = ;, s = errte, state = 9
Iteration 1246: c = c, s = sesst, state = 9
Iteration 1247: c = J, s = setse, state = 9
Iteration 1248: c = A, s = ssest, state = 9
Iteration 1249: c = i, s = rstst, state = 9
Iteration 1250: c = /, s = ssese, state = 9
Iteration 1251: c = l, s = srtrt, state = 9
Iteration 1252: c = , s = erstt, state = 9
Iteration 1253: c = ), s = rrrsr, state = 9
Iteration 1254: c = y, s = ertsr, state = 9
Iteration 1255: c = ", s = rtree, state = 9
Iteration 1256: c = B, s = sstet, state = 9
Iteration 1257: c = d, s = rsrtt, state = 9
Iteration 1258: c = U, s = etees, state = 9
Iteration 1259: c = I, s = tstse, state = 9
Iteration 1260: c = G, s = rests, state = 9
Iteration 1261: c = j, s = eerrs, state = 9
Iteration 1262: c = W, s = ersss, state = 9
Iteration 1263: c = i, s = retee, state = 9
Iteration 1264: c = 7, s = rssst, state = 9
Iteration 1265: c = 7, s = srrte, state = 9
Iteration 1266: c = K, s = reset, state = 9
error make: *** [makefile:2: run] Error 200
```