

CS362

Quiz-2

Tyler Forrester

The random tester uses two functions `inputChar()` and `inputString()` to generate random input for the `testme` function. The random data are generated from a char array of state mutating chars. Since not all chars change the state of the `testme` function, only those chars which do change the state were added to that array. These chars in `inputChar()` were `{ '[', ']', '(', ')', '{', '}', 'a', 'x', ' ' }`. The chars were selected randomly using the `rand()` function initializing by the computer time. For `inputString()`, there was slight modification in `testme` to populate the string as a side-effect of `inputString` per Kernigan and Kerrie's recommendation in the "C-Programming Language" and the instructor's piazza post stating this was acceptable. The `inputString` function therefore takes `char *`s and the size of `char *`s as an input and then again using `rand()` populates the string with chars chosen for the list of chars which mutate state, `{ 'r', 'e', 's', 't' }`. The length of the string is only the length that mutates state, 6. The string is allocated and deallocated in each loop of `testme()` to allow for repopulation. The length of the string is kept in a constant char called `STRING_INPUT_LENGTH`. The length of the char array of mutable values is `STRING_STATE_LENGTH`.

The random tester generates random input from the mutable states running in a loop until the error state is hit. Since we are only pulling chars that mutate state this speeds up the process substantially.