Doseon Kim
TCSS 487
Part 1

For part 1, I previously had a working SHA-3 java program that was based off of the Keccak Machinery Summary Specs on their website at https://keccak.team/keccak_specs_summary.html

From my understanding it doesn't really tell us to re-create SHA-3, but rather a hash function using the KMACXOF256 within the pseudo-code details which can be found in the project description.

The description states…

Computing a cryptographic hash digest *h* of a given byte array *m*:

- *h* <- KMACXOF256("", *m*, 512, "D")

Therefore I took the given cSHAKE256 and KMACXOF256 code provided by professor Barreto and implemented a GUI to run the given pseudo-code above.

We use "" Key value, message the user wants to hash, 512 bit length and "D" as the secret customization string.

Only issues I have are that I cannot check the hash value to anything on the internet because this isn't SHA-3. I attempted to look up KMAC256 hash but I wasn't able to find working tool to allow me to check the correct outputs. But the cSHAKE and KMACXOF256 initially provided passed the NIST vector, therefore we should be able to assume that the hashing works since we properly followed the pseudo-code provided.

With the provided part 1, a user can hash a message directly into the console or they may insert a location address to a text file, they would like to hash. One issue with reading files is that it gets an extra byte from the end of the file if there are new line characters created by the text file; therefore if the user inputs a message "abc" and the text file contains an extra new line character, the results would differ.