CSCI 357 - Cryptography
Ryan Kulyassa
HW #9 - Comparing Different SHA-3 Variants

In this assignment, I computed hashes for my first name ("Ryan") using SHA3-224, SHA3-256, SHA3-384, and SHA3-512. Then, I used the same four algorithms to hash the concatenation of my first and last name ("RyanKulyassa") and compared this respective hash with the hash of just my first name to count how many matching characters there were, and test the concept of the avalanche effect. My results are in the table below:

SHA Variant	Number of matches
SHA3-244	3
SHA3-256	1
SHA3-384	8
SHA3-512	10

Which SHA-3 variant had the most matches? The fewest?
SHA3-512 had the most matches (10) while SHA3-256 had the fewest (1).

Did longer hashes (e.g., SHA3-512) have fewer matches than shorter ones (e.g., SHA3-224)? Why or why not?
Longer hashes generally had more matches, except for SHA3-384 which had fewer matches than the rest. I believe this is an outlier, however, because longer hashes introduce the potential for more matches between any two arbitrary hashes. This is provable, as there is a finite number of possibilities for each character, so as you increase the length of the hash, the probability of having more matches increases.

Does the avalanche effect (small input change → large output change) hold true for all SHA-3 versions?
Yes, as there was no significant number of matches between any two hashes. Further, I observed that the hashes themselves did not appear similar at all.

Which variant would you recommend for a security-critical application? Why?
Of these four SHA variants, the most secure is SHA3-512 simply due to it being the largest. While this may introduce a tradeoff with regard to data storage (larger hash = more bytes), for a security-critical application this may be a reasonable investment to ensure the application is using the most secure hashing algorithm of these four choices.