**SHA-512 Hashing Algorithm Documentation**

This documentation provides an explanation of the code for computing the SHA-512 hash of a message. SHA-512 is a widely used cryptographic hash function that produces a fixed-size (512-bit) hash value for a given input.

**Code Overview**

The provided code consists of several functions to perform SHA-512 hashing:

**sha512\_compression(w, round\_constants, a, b, c, d, e, f, g, h): This** function performs the core compression step of the SHA-512 algorithm. It takes the message block w, round constants, and the current state variables a to h as input. It iterates through 80 rounds and updates the state variables based on the message block and round constants.

**binary\_padding(message\_binary):** This function calculates and applies the binary padding to the input message. It ensures that the message length is a multiple of 1024 bits and appends the original message length as a 128-bit binary representation.

**divide\_padded\_binary(block):** This function divides the padded binary message into 1024-bit blocks and converts them into a list of 64 16-bit words.

**process\_message(message):** This is the main function for processing a message and calculating its SHA-512 hash. It initializes the initial state variables, converts the input message to binary format, applies padding, and then processes the message in 1024-bit blocks using the sha512\_compression function.

**Sample Round Constants**

The code uses a predefined set of 80 round constants in decimal format to be used during the compression step. These round constants are essential for the SHA-512 algorithm and are defined as hexadecimal values.

**Example:**

To use the code to hash a message, you can call the process\_message function with the message as an input.

Here's an example of how to use it:

hashed\_message = process\_message("Sakar")

print("SHA-512 Hash:", hashed\_message)

**Output:**

SHA-512Hash: a89539726a28f7a4d8127aad45680f693da3f04dc33c467563cbc62e27b976a8936507e5006ae808671fde5742be68fea6751c334db37ebacc96b0da2dc0f57a