1. ravishingly
2.
    a.
        - Yes. The hash function accepts messages *M* of varying length *t*. *t* can be any positive integer, which means the function can process messages of any size.
        - Yes. Regardless of the input size, the output of the function is an integer in the range [0, n-1]. Thus, it always has a fixed size represented by log2(n) bits.
        - If *t* is the length of the input, then the time complexity is $O(t)$.
        - For both first and second pre-image resistance, the simple structure of the function does not guarantee this property.
        - Given the simple nature of the function, it's plausible that collisions can be found, especially when *t* is large and *n* is relatively small.
        - The output's unpredictability is not guaranteed since the squaring operation might introduce some non-linearity.
    b.
        - Yes. The function can accept sequences *M* of varying lengths *t*, so it can process messages of any size.
        - Yes. Regardless of the size of the input, the output of the function will always be an integer in the range [0, n-1], ensuring a fixed output size.
        - The primary computation here involves squaring integers and summing them up, followed by a

modulo operation. Hence O(t) time in which t is the length of the input.

- For both first and second pre-image resistance, the simple structure of the function does not guarantee this property.
- Tt's conceivable that collisions could be identified. This is especially true when *t* is large, and *n* is small, as many different sequences of squared numbers could sum up to values that differ by multiples of *n*, resulting in the same hash value after the modulo operation.
- The output's unpredictability isn't inherently guaranteed. The function is deterministic and fairly simple.

c. $h2(M) = (1892 + 6322 + 9002 + 7222 + 3492) \bmod 989 =$

$h2(M) = (2205230) \bmod 989 =$

$h2(M) = 2205230 \bmod 989 =$

$h2(M) = 535$

3. Key: 89, Word: INTERNATIONALIZATION