Code Challenge: MyHashCash

We are going to implement a simplified version of the hashcash proof-of-work algorithm, and serve it as an HTTP API. The hashcash algorithm is most commonly used as a proof-of-work system in Bitcoin mining, as well as spam detection, and denial-of-service attacks. You may implement in Python, NodeJS, Go, or Java. There are two endpoints: /find and /verify - the specification of each endpoint is up to you (method, path params, query params, headers, response codes, etc). The differences between this algorithm implementation and the hashcash version 1 specification is we will use SHA-256; allow an arbitrary number of "partial pre-image zero" bits (not just 20); and ignore the version, date, extension, and random fields.

The endpoints should do the following:

- `/find`: given an arbitrary challenge (c) string, and a number of bits (n) integer: we need to increment a proof of work counter (w) to find a SHA-256 digest (d) of the concatenation of c+w that has n leading 0 bits. For example given c="iBeat", n=16, we need to compute a hash digest token that starts with 16 leading 0-bits while also containing the concatenation of "iBeat" and the numeric counter w. The resulting match is `iBeat62073` which has a hash digest token of `0000d4ab4f89e8d1cd021a04151a280e4c76d487e04074a232bb0a8dec4a74cf`. This indicates proof of work counter incremented to 62073 before finding a match. Return the proof of work w, which can be passed into the subsequent /verify endpoint to verify a challenge string against a number of bits.
- `/verify`: given an arbitrary challenge (c) string, a number of bits (n) integer, and a proof of work counter (w) integer: verify the SHA-256 hash digest of the concatenation of c+w has n leading 0 bits. For example given c="iBeat", n=16, w=62073, we should return `True`. For the same c and n value but a different w value we should return `False`. Thus the caller has proof they did the work to compute this hash.

Notes:

- The leading zero bits n is number of bits, not bytes.
- Parameters n and w are base 10 integers.
- Please push the git repo to Github with any documentation or tests.
- Account for dependency management (language version, 3rd party packages).
- Provide a README that shows how to get up and running and verify the endpoints.