Hash function

File `insecure_hash` implements the hash function `hash_string`.

- If a message `m` consists of two blocks `B1;B2` of 128 bits, the hash is computed as `EAS-decrypt(B1, B2)`: i.e. it decrypts the block `B1` using `B2` as key.
- If the message consists of three blocks `B1;B2;B3`, the hash is computed as `EAS-decrypt(EAS-decrypt(B1, B2), B3)`
- If the message consists of n blocks `B1;B2;B3;...Bn`, the hash is computed as `EAS-decrypt(... EAS-decrypt(EAS-decrypt(B1, B2), B3)..., Bn)`

The message is padded with =" "= to be block aligned, e.g. if the length of the message is 200 bits, then 56 spaces are appended to the message before computing the hash.

This hash function is not secure. In particular it is not weak collision resistant. Complete the stub in `collision.py`, implementing the function `find_collision(message)` that finds a collision. **Do not brute force**.

To test your solution execute `./test.py` or `py.test test.py`.