Exercise 6

- Task 1 - Password cracking
  - You have found the key (hash) and the salt for Ola's password on a server:
    - key: "ab29d7b5c589e18b52261ecba1d3a7e7cbf212c6"
    - salt: "Saltet til Ola"
    - You know that PBKDF2 with SHA1 is used by the server (the pbkdf2 function in OpenSSL should be used, take ntnu-tdat3020/openssl-example as a starting point) with 2048 iterations
    - You know that Ola does not bother to use many letters in his password
  - What is Ola's password?
- Task 2
  - Create a (web) client and server where the client authenticates against the server using PBKDF2
    - The password must be hashed on both the client and server side
    - For those who would like to use JavaScript on both the client and server, crypto-js is an ok alternative
      - Optionally use Node.js Crypto (openssl bindings) on the server side
    - Voluntary: when a client is authenticated, send an access token to the client that can be used, without re-authenticating, in subsequent requests
      - the design of the token is up to you
      - assume HTTPS is used (so that you can use simple access/bearer tokens)