Project 1 Algorithms

Note: this is pseudocode, not real code. While your program execution might follow the flow as this example, it will look much different. Your responsibility is to write the code to perform the same steps including those hidden in the three functions, `GET_PASSWORD_CHARACTER`, `HEX`, and `HEX_DIGITS_TO_VALUE`. You are not expected to write functions, just write the code that performs the same tasks.

Encrypt

```
encrypt (message, password) -> ciphertext
      parameters
      message: the message to be encrypted
      password: the password with which to encrypt the message
      returns: the encrypted ciphertext in the form of a hexadecimal string

      Local variables
      ciphertext: the string to store the hexadecimal digits of the encrypted chars.
      message_char: a variable to hold each character of the message
      password_char: a variable to hold each character of the password
      cipher_char: a variable to hold each character of the ciphertext

      ciphertext = "0x"

      begin:
            for each message_char in the message
            do:
                  // GET_PASSWORD_CHARACTER returns the corresponding character from the password
                  password_char = GET_PASSWORD_CHARACTER(password)

                  cipher_char = message_char XOR password_char

                  // HEX produces two hexadecimal digits as a string without "0x" prefix
                  // these two digits represent the value of the inputted char
                  ciphertext = ciphertext + HEX(chipher_char)
            loop:

            return ciphertext
      end:
```

Decrypt

```
decrypt (ciphertext, password)
      parameters
      ciphertext: the string that holds the hex digits that represent the encrypted chars.
      password: the password with which to encrypt the message
      returns: the encrypted ciphertext in the form of a hexadecimal string

      Local variables
      message: the string to store the plaintext message chars.
      two_cipher_digits: a variable to hold the string containing each pair of hex digits
      cipher_char: a variable that holds each cipher character
      password_char: a variable to hold each character of the password
      plaintext_char: a variable to hold each character of the plaintext

      message = ""

      begin:
            for each two_cipher_digits in the ciphertext
            do:
                  // HEX_DIGITS_TO_VALUE converts two hex digits to their decimal value
                  cipher_char = HEX_DIGITS_TO_VALUE(two_cipher_digits)

                  // GET_PASSWORD_CHARACTER returns the corresponding character from the password
                  password_char = GET_PASSWORD_CHARACTER(password)

                  plaintext_char = cipher_char XOR password_char

                  // HEX produces hexadecimal digits as a string without "0x" prefix
                  message = message + plaintext_char
            loop:

            return message
      end:
```