```python
# LCYS_2022 / L. M. Saxton

# https://github.com/pyauth/pyotp
# !pip install pyotp
import pyotp

# coding_project/create_user_classes.py
from create_user_classes import CreateUsername, CreatePassword, EncryptPass
# coding project/create_user_classes.py
from login_classes import VerifyPass, VerifyUser

# Create username and password

print('~~User Registration~~')

# Dictionary for username and password
user_1 = {}

# Create a username within certain parameters
while(True):
    successes = []
    success_count = 0

    username = input("\nPlease enter a username: ")

    # Cross-check user input
    username = CreateUsername(username)
    username.userLength()
    username.userSpaces()
    username.userSpecSym()

    # Assign output value to successful input
    username.successUserLength()
    username.successUserSpaces()
    username.successUserSpecSym()

    # Assign variable names to function output
    success_length = username.successUserLength()
    success_spaces = username.successUserSpaces()
    success_spec_sym = username.successUserSpecSym()

    # Append function variables to list 'successes'
    successes.append(success_length)
    successes.append(success_spaces)
    successes.append(success_spec_sym)

    # Add the value of each output variable in 'successes' to 'success_count'
    for successes in successes:
        if successes == 1:
            # If the variable == 1, success_count +=1
            success_count = success_count+1

    # If all variables == 1, success_count = 3: username accepted
    if success_count == 3:
        # Append username to dictionary
        user_1['username'] = username.username
        break

# Create a password within certain parameters
while(True):
    successes = []
    success_count = 0

    password = input("\nPlease enter a password: ")

    # Cross-check user input
    password = CreatePassword(password)
    password.passLength()
    password.passUpper()
    password.passDigit()
    password.passSpaces()
    password.passSpecSym()

   # Assign output value to successful input
    password.successPassLength()
```

```
    password.successPassUpper()
    password.successPassDigit()
    password.successPassSpaces()
    password.successPassSpecSym()

    # Assign variable names to function output
    success_length = password.successPassLength()
    success_upper = password.successPassUpper()
    success_digit = password.successPassDigit()
    success_spaces = password.successPassDigit()
    success_spec_sym = password.successPassSpecSym()

    # Append function variables to list 'successes'
    successes.append(success_length)
    successes.append(success_upper)
    successes.append(success_digit)
    successes.append(success_spaces)
    successes.append(success_spec_sym)

     # Add the value of each output variable in 'successes' to 'success_count'
    for successes in successes:
        if successes == 1:
            # If the variable == 1, success_count +=1
            success_count = success_count+1

    # If all variables == 1, success_count = 5: password accepted
    if success_count == 5:
        print("\nPassword accepted!")

        # password.password = variable + function
        password = password.password

        # Encrypt password with a salted hash function
        password = EncryptPass(password)
        password.encryptPass()

        hashed = password.encryptPass()

        # Included in test code only to make sure hash is working
        print(f"\nPassword hash: {hashed}")

        # Append successful password to dictionary
        user_1['password'] = hashed

        # Get saved username
        username = user_1.get('username')
        # Keep hash private
        password = password.password

        print("\nThank you for registering!")

        print(f"\tYour username: {username}")
        print(f"\tYour password: {password}")
        break


# Login using username, password, and Two Factor Authorization

print('\n~~User Login~~')

# Verify an entered username
while(True):
    successes = []
    success_count = 0

    # User input
    username = input("\nPlease enter your username: ")
    username = username
    # Cross-check variable
    crosscheck_name = user_1.get('username')

    # Verify username with verify functions
    verify_username = VerifyUser(username, crosscheck_name)
    verify_username.verifyUser()
    verify_username.verifyUserSuccess()
```

```
    verified_username = verify_username.verifyUserSuccess()
    successes.append(verified_username)

    # Add the value of each output variable in 'successes' to 'success_count'
    for successes in successes:
        # If the variable == 1, success_count +=1
        if successes == 1:
            success_count = success_count+1
    # If success_count = 1: username accepted
    if success_count == 1:
        break

# Verify an entered password
while(True):
    successes = []
    success_count = 0

    # User input
    password = input("\nPlease enter your password: ")

    #Change string to bytes for comparison
    password = password.encode('ASCII')
    hashed = user_1.get('password')

    # Verify password with verify function
    verify_password = VerifyPass(password, hashed)
    verify_password.verifyPass()
    verify_password.verifyPassSuccess()

    verified_password = verify_password.verifyPassSuccess()
    successes.append(verified_password)

    # Add the value of each output variable in 'successes' to 'success_count'
    for successes in successes:
        # If the variable == 1, success_count +=1
        if successes == 1:
            success_count = success_count+1
    # If success_count = 1: password accepted
    if success_count == 1:
        break

# Two Factor Authorization
while(True):
    #Activates only if password is verified
    if success_count == 1:
        # OTP code generator
        totp = pyotp.TOTP('base32secret3232')
        print("\nYour OTP is: ", totp.now())

        # User input
        otp_code = input("What is your OTP code?\t")

        # verify the input code
        if totp.verify(otp_code)==True:
            # Simulated login
            print("\nSuccessful login")
            break
        else:
            print("\tUnsuccessful. Resending code.")
```