```
#Problem
#Multi-Threaded Password Validator


import threading
import re

class Password:
    def __init__(self, passwordd=None):
        self.passwordd = passwordd

    def validate_password(self, passwordd):
        if len(passwordd) < 8:
            raise ValueError("Password must be at least 8 characters long.")
        if not re.search(r"[A-Z]", passwordd):
            raise ValueError("Password must contain at least one uppercase
letter.")
        if not re.search(r"[a-z]", passwordd):
            raise ValueError("Password must contain at least one lowercase
letter.")
        if not re.search(r"[0-9]", passwordd):
            raise ValueError("Password must contain at least one digit.")
        if not re.search(r"[@$!%*?&]", passwordd):
            raise ValueError("Password must contain at least one special
character.")
        return True

    def save(self, passwordd):
        with open("valid_password.txt", "a") as f:
            f.write(passwordd + "\n")

    def saveinvalid(self, passwordd):
        with open("invalid_password.txt", "a") as f:
            f.write(passwordd + "\n")

    def process_password(self, passwordd):
        try:
            if self.validate_password(passwordd):
                self.save(passwordd)
                print("Password saved successfully!")
        except ValueError as e:
            self.saveinvalid(passwordd)
            print(f"Password validation failed: {e}")

if __name__ == '__main__':
    a = input("Enter your password: ")
    pwd = Password()
    thread1 = threading.Thread(target=pwd.process_password, args=(a,))
    try:
        thread1.start()
```

```
    thread1.join()
except Exception as e:
    print(e)
```