Problem 1

1. As discussed in the class on March 7, the Two-Factor Authentication (2FA) method has specific weakness. Please describe the weakness and suggest possible solutions to address them.

   - Weakness
     a. Phishing 網站偽裝登入頁竊取驗證碼（Social Engineering 攻擊）：攻擊者可假冒銀行或系統，誘導用戶提供 2FA 驗證碼。
     b. SIM Swap 攻擊：若使用 SMS 作為驗證手段，攻擊者可透過社交工程取得門號，攔截簡訊驗證碼。
     c. TOTP App 被入侵或竊取：若手機遭入侵，TOTP app 中的時間性密碼可能也被盜用。
     d. 中間人攻擊（Man-in-the-middle）：攻擊者可在用戶與網站之間竊聽或轉發驗證碼。
        ➔ 人為因素依然是 2FA 的最大弱點。

   - Suggest possible solutions
     a. 使用更安全的驗證方式：改用 App-based TOTP （如 Google Authenticator、Authy）取代 SMS，或者使用硬體 FIDO2 安全金鑰（如 YubiKey）。
     b. 防範釣魚攻擊：避免使用可被轉發的 One-Time Code，改使用 FIDO2 / WebAuthn 等無密碼登入，並對登入行為進行異常偵測
     c. 裝置綁定：將驗證流程與裝置綁定，例如僅允許綁定裝置產生 OTP。
     d. 風險控制機制：加入行為風控引擎（如連續輸入錯誤、異常裝置）阻擋風險登入。

2. How can the Yangming and Guangfu campuses design a VPN to securely share research resources?
   a. 固定 IP 與對等連線：兩校皆有固定對外 IP，可建構 Site-to-Site VPN
   b. 使用安全通訊協定與加密技術：如 WireGuard 或 OpenVPN，配合 TLS 1.3 與現代加密演算法（如 AES-GCM、ChaCha20-Poly1305）。

Ex. 兩端皆部署 WireGuard，使用 Curve25519 進行金鑰交換。

c. 身分認證與金鑰管理：每一端使用各自的金鑰進行雙向驗證且金鑰應每隔一段時間定期更新。

   Ex. 結合雙因素登入或憑證驗證確保用戶身份。

d. 區隔研究群組流量：可用 VLAN 或 Subnet 進行邏輯隔離並結合防火牆政策，限制特定 IP 或 port 的訪問權限。

   Ex. 每台研究室主機設置私鑰並註冊於 VPN Server。

Problem 2

1. How to run my program:

```
pip install -r problem2/requirements.txt
python3 problem2/main.py
```

2. What I do:

   利用 chatgpt 寫出的 code 第一次就可以成功解決 Easy hash 跟 Medium hash，但他看不懂 salt term，因此最後一個 Leet hacker hash 顯示為 not cracked.

```
Hash: db3ae03df555104cd021c6308d5d11cfa40aac41
Password: hotmom
Took 30568 attempts to crack message.

Hash: 884950a05fe822dddee8030304783e21cdc2b246
Password: scorpion
Took 302 attempts to crack message.

Hash: 9b467cbabe4b44ce7f34332acc1aa7305d4ac2ba
Password: wh00sh
Took 939438 attempts to crack message.

Hash: 9d6b628c1f81b4795c0266c0f12123c1e09a7ad3 not cracked.
```

   所以我更改成如果吃到有 salt，就跑雙層迴圈，如此整份答案就都出來了。

```
if salt:
    for i, a in enumerate(passwords, start=1):
        if sha1(a) == salt:
            for j, b in enumerate(passwords, start=1):
                if sha1(a + b) == target_hash:
```

```
Hash: db3ae03df555104cd021c6308d5d11cfa40aac41
Password: hotmom
Took 30568 attempts to crack message.

Hash: 884950a05fe822dddee8030304783e21cdc2b246
Password: scorpion
Took 302 attempts to crack message.

Hash: 9b467cbabe4b44ce7f34332acc1aa7305d4ac2ba
Password: wh00sh
Took 939438 attempts to crack message.

Hash: 9d6b628c1f81b4795c0266c0f12123c1e09a7ad3
Password: redbull + puppy
Took 2854 attempts to crack message.
```

3. What my program does:
   a. 下載密碼列表並存為 problem2/password.txt
   b. 讀入密碼列表
   c. 定義 SHA-1 雜湊函數
   d. 設定目標雜湊值（Hash）與 Salt
   e. 暴力破解（Brute-force search）
   f. 輸出結果

Problem 3

1. How to run my program: (logger.log 會被覆蓋)

```
pip install -r problem3/requirements.txt
python3 problem3/main.py
```

2. What I do:
   把檔案丟進 chatgpt 後，他給出了幾乎正確的版本，我修改了
   a. [EROR] 的寫入
   b. 保存檔案的處理
   c. 在找到該輪的 hash 後，檢查是否也符合下一輪的 prefix，並紀錄 without nonce（chatgpt 給的檔案不管有沒有符合都會進入下一輪並加上 nonce）
   d. 因為後面的數字要找比較久，所以加入 Progress Bar 來確認進度

3. What my program does:
   a. Log 檔處理
      - 若 logger.log 已存在，程式會自動刪除舊檔案後重新建立，否則自動建立新檔案
      - 設定 Log 檔的結構
   b. 初始化
      - 將學號以 SHA-256 雜湊後產生 preImage
      - 根據 preImage 與學號的前幾位是否相符，決定從第幾區塊（Round）開始
   c. 挖礦過程
      - 每一輪會先判斷上一輪產生的雜湊值是否已符合本輪的 prefix
      - 若已符合，則直接記錄 [Round X without nonce] 並進入下一輪
      - 若未符合，則開始從 00000000 到 ffffffff 的 nonce 值進行暴力測試
      - 將前一輪 hash 加上 nonce，再做 SHA-256 雜湊，檢查是否符合 prefix
      - 一旦成功找到符合條件的 hash，就記錄 [Round X with nonce xxxxxxxx]

d. 錯誤處理

- 若一整輪所有 nonce 都無法找到符合條件的雜湊，記錄 [EROR] 並結束