**Conclusion**

**1. Để Quản lý các thiết bị trong mạng chọn kiểu dữ liệu nào? giải thích vì sao chọn nó?**

- Danh sách các thiết bị trong mạng

**Yêu cầu:** Cần một cấu trúc để lưu danh sách các thiết bị trong mạng, vì chúng ta sẽ có nhiều thiết bị và muốn quản lý chúng một cách thuận tiện.

**Giải pháp:** Sử dụng kiểu *list* trong Python.

* **Lý do:** *list* cho phép lưu trữ nhiều đối tượng và dễ dàng truy xuất từng thiết bị bằng chỉ số. Thêm, xóa, và duyệt qua các thiết bị trong list cũng rất thuận tiện, giúp chúng ta quản lý danh sách thiết bị dễ dàng và hiệu quả.

**2. Quản lý địa chỉ IP của các thiết bị trong mạng chọn kiểu dữ liệu nào? giải thích vì sao chọn nó?**

- Địa chỉ IP của các thiết bị

**Yêu cầu:** Địa chỉ IP là thông tin cần thiết để xác định vị trí của từng thiết bị trong mạng. Địa chỉ IP phải hợp lệ và dễ tra cứu.

**Giải pháp:** Chúng ta có thể sử dụng kiểu dữ liệu *set* trong Python để lưu trữ các địa chỉ IP.

* *set* là một cấu trúc dữ liệu lưu trữ các phần tử không trùng lặp, đảm bảo rằng mỗi địa chỉ IP chỉ tồn tại một lần trong danh sách.
* *set* có hiệu suất tốt khi thực hiện các thao tác thêm, xóa, và kiểm tra sự tồn tại của phần tử, điều này rất phù hợp cho việc quản lý các địa chỉ IP.

**3. Quản lý số lượng cổng kết nối của thiết bị trong mạng chọn kiểu dữ liệu nào? giải thích vì sao chọn nó?**

- Số lượng cổng kết nối của thiết bị

**Yêu cầu:** Một số thiết bị mạng có nhiều cổng kết nối (ví dụ: router có nhiều cổng mạng). Chúng ta cần lưu trữ thông tin về số lượng cổng kết nối của từng thiết bị.

**Giải pháp:** Sử dụng kiểu int để lưu trữ số lượng cổng kết nối.

* **Lý do:** *int* là kiểu dữ liệu tối ưu cho các giá trị số nguyên. Số lượng cổng kết nối là một số nguyên dương, và *int* giúp đảm bảo lưu trữ dữ liệu dạng này một cách gọn gàng, không có phần thập phân. Ngoài ra, int trong Python hỗ trợ các phép tính số học, nên chúng ta có thể dễ dàng thực hiện các phép tính nếu cần.

**4. Quản lý thông tin chi tiết của từng thiết bị trong mạng chọn kiểu dữ liệu nào? giải thích vì sao chọn nó?​**

- Thông tin chi tiết của từng thiết bị

**Yêu cầu:** Cần lưu trữ nhiều thông tin về từng thiết bị, bao gồm địa chỉ IP, địa chỉ MAC, loại thiết bị, và số lượng cổng kết nối. Các thông tin này cần được lưu dưới dạng từng thuộc tính riêng để dễ quản lý và truy xuất.

**Giải pháp:** Sử dụng dict (từ điển) hoặc tạo một class để tổ chức thông tin của từng thiết bị.

* **Lý do:**
  + **Với** *dict***:** Từ điển cho phép lưu trữ các cặp khóa-giá trị, nên mỗi thông tin của thiết bị có thể lưu dưới dạng cặp khóa-giá trị như "IP": *"192.168.1.1", "MAC": "00:1A:2B:3C:4D:5E", v.v.* Điều này giúp chúng ta truy xuất dữ liệu theo tên thuộc tính một cách dễ dàng.
  + **Với** *class***:** Sử dụng class cung cấp một cách tổ chức tốt hơn cho thông tin của từng thiết bị, đặc biệt khi có nhiều thuộc tính và các hành vi riêng biệt. Ví dụ, chúng ta có thể tạo lớp ThietBiMang với các thuộc tính như IP, MAC, loai\_thiet\_bi, so\_cong, và có thể dễ dàng thêm các phương thức (như phương thức hiển thị thông tin thiết bị).
* **Cài đặt bằng Python:**

Dưới đây là cách triển khai mã Python cho bài toán này:

import ipaddress

# Định nghĩa lớp ThietBiMang để lưu trữ thông tin chi tiết của từng thiết bị

class ThietBiMang:

def \_\_init\_\_(self, ten, ip, mac, loai\_thiet\_bi, so\_cong):

self.ten = ten # Tên thiết bị

self.ip = ipaddress.ip\_address(ip) # Kiểm tra và lưu địa chỉ IP

self.mac = mac # Địa chỉ MAC

self.loai\_thiet\_bi = loai\_thiet\_bi # Loại thiết bị (server, router, etc.)

self.so\_cong = so\_cong # Số lượng cổng kết nối

def hien\_thi\_thong\_tin(self):

print(f"Tên thiết bị: {self.ten}")

print(f"Địa chỉ IP: {self.ip}")

print(f"Địa chỉ MAC: {self.mac}")

print(f"Loại thiết bị: {self.loai\_thiet\_bi}")

print(f"Số cổng kết nối: {self.so\_cong}")

# Tạo danh sách các thiết bị sử dụng list

ds\_thiet\_bi = [

ThietBiMang("Server 1", "192.168.1.1", "00:1A:2B:3C:4D:5E", "Server", 4),

ThietBiMang("Router A", "192.168.1.2", "00:1A:2B:3C:4D:5F", "Router", 8)

]

# Hiển thị thông tin từng thiết bị

for thiet\_bi in ds\_thiet\_bi:

thiet\_bi.hien\_thi\_thong\_tin()

print("----------")