## **Bài Tập Lớn: Quản Lý Thông Tin Cư Dân Chung Cư**

### **Mục Tiêu**

### Phát triển một chương trình bằng C để quản lý thông tin cư dân trong chung cư, bao gồm các chức năng cơ bản: thêm, xóa, chỉnh sửa và tìm kiếm thông tin cư dân. Chương trình này sẽ giúp quản lý cư dân một cách hiệu quả và chính xác, đảm bảo dữ liệu được cập nhật và truy xuất một cách nhanh chóng.

### **Cấu Trúc Dữ Liệu**

* Member:

Mô tả: Cấu trúc để lưu trữ thông tin cơ bản của cư dân.

Trường thông tin: UID của RFID, số phòng, tên, biển số xe.

#### Ví dụ Code:

| typedef struct {  char uid[20]; // UID của RFID  char roomNumber[10]; // Số Phòng  char name[50]; // Tên  char licensePlate[20]; // Biển Số Xe } Member; |
| --- |

* MemberNode:

Mô tả: Node trong linked list chứa thông tin Member.

Trường thông tin: Dữ liệu Member, con trỏ đến MemberNode tiếp theo.

Ví dụ Code:

| // Định nghĩa cấu trúc MemberNode typedef struct MemberNode {  Member data;  struct MemberNode\* next; } MemberNode;  // Hàm tạo một MemberNode mới MemberNode\* createMemberNode(Member member) {  MemberNode\* newNode = (MemberNode\*)malloc(sizeof(MemberNode));  newNode->data = member;  newNode->next = NULL;  return newNode; } |
| --- |

### **Các Chức Năng Chính**

* Thêm Thành Viên (Add Member):
  + Mô tả: Ghi thông tin mới của thành viên vào cuối file CSV.
  + Đầu vào: Member mới.
  + Đầu ra: File CSV cập nhật.

Ví dụ:

| void addMember(const char\* filename, Member member) |
| --- |

* Xóa Thành Viên (Delete Member):
  + Mô tả: Xóa thông tin của thành viên dựa trên UID từ file CSV.
  + Đầu vào: UID của thành viên cần xóa.
  + Đầu ra: File CSV cập nhật.

Ví dụ:

| void deleteMember(const char\* filename, const char\* uid) |
| --- |

* Chỉnh Sửa Thông Tin Thành Viên (Edit Member):
  + Mô tả: Cập nhật thông tin của thành viên dựa trên UID trong file CSV.
  + Đầu vào: Member cập nhật và UID.
  + Đầu ra: File CSV cập nhật.

Ví dụ:

| void editMember(const char\* filename, Member updatedMember) |
| --- |

* Tìm Kiếm Thông Tin Thành Viên (Search Member):
  + Mô tả: Tìm kiếm thông tin thành viên dựa trên UID hoặc biển số xe.
  + Đầu vào: Giá trị tìm kiếm và tiêu chí (UID/Biển số xe).
  + Đầu ra: Thông tin tìm được hoặc thông báo không tìm thấy.

Ví dụ:

| int searchByUID(const Member\* member, const char\* uid)  int searchByLicensePlate(const Member\* member, const char\* licensePlate) Member searchMember(const char\* filename, const char\* searchValue, SearchFunction searchFunc) |
| --- |

### **Bổ Sung**

### Định Dạng File CSV: Đảm bảo file CSV tuân thủ định dạng đã quy định với các cột là UID của RFID, Số Phòng, Tên, và Biển Số Xe.

* UID của RFID: Chuỗi ký tự độc nhất định danh cho mỗi cư dân.
* Số Phòng: Mã số phòng của cư dân.
* Tên: Tên đầy đủ của cư dân.
* Biển Số Xe: Biển số xe (nếu có) của cư dân.

### Ví Dụ

Một ví dụ về dòng dữ liệu trong file CSV:

| uID,roomNumber,name,licensePlates  123456789,A-101,Nguyễn Văn A,51F-123.45 |
| --- |

Trong đó:

* 123456789 là UID của RFID.
* A-101 là số phòng.
* Nguyễn Văn A là tên của cư dân.
* 51F-123.45 là biển số xe.
* Tính Mở Rộng: Cấu trúc chương trình cần cho phép dễ dàng mở rộng thêm chức năng hoặc cấu trúc dữ liệu mới.
* Tính Mở Rộng: Cấu trúc chương trình cần cho phép dễ dàng mở rộng thêm chức năng hoặc cấu trúc dữ liệu mới.