**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO THỰC HÀNH CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**Sinh viên thực hiện:**

**Huỳnh công tính**

**Giảng viên hướng dẫn: Đỗ Như Tài**

**Thành phố Hồ Chí Minh, ngày 14 tháng 3 năm 2025**

**CÀI ĐẶT CÁC TÁC VỤ DANH SÁCH LIÊN KẾT ĐƠN**

### Mô tả bài toán:

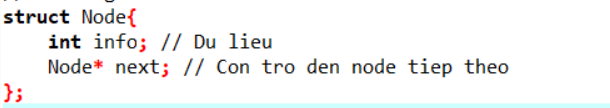
Bài toán yêu cầu cài đặt các tác vụ cơ bản của danh sách liên kết đơn (DSLKĐ), bao gồm:

1. Khởi tạo danh sách rỗng
2. Kiểm tra danh sách rỗng
3. Tìm kiếm phần tử trong danh sách
4. Xóa phần tử có giá trị cho trước
5. Truy xuất phần tử theo vị trí
6. Duyệt danh sách
7. Hủy danh sách
8. Sắp xếp danh sách theo thứ tự tăng dần
9. **Cấu trúc dữ liệu sử dụng:**

Chương trình sử dụng danh sách liên kết đơn, trong đó mỗi phần tử là một Node với hai thành phần:

**Info:** Lưu giá trị của phần tử

**Next:** Con trỏ trỏ đến phần tử tiếp theo



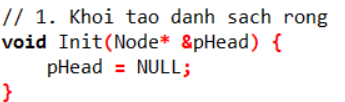
### Cài đặt các chức năng chính:

### 3.1. Khởi tạo danh sách rỗng

**Main idea:** Tạo một danh sách liên kết chưa có phần tử nào.

**Giải thuật:**

Gán con trỏ đầu danh sách pHead bằng NULL để biểu thị danh sách rỗng.

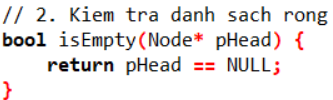


#### **3.2. Kiểm tra danh sách rỗng**

**Main idea:** Xác định danh sách có phần tử hay không.

**Giải thuật:**

* Nếu con trỏ đầu danh sách pHead là NULL, danh sách rỗng, trả về true.
* Ngược lại, trả về false.

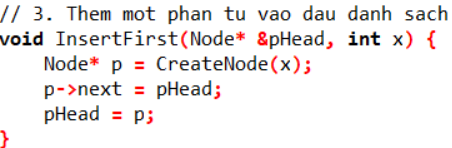


#### 3.3. Thêm phần tử vào đầu danh sách

**Main idea:** Chèn một phần tử mới vào đầu danh sách liên kết đơn.

**Giải thuật:**

* Cấp phát một node mới, gán giá trị cho info.
* Trỏ next của node mới về pHead.
* Cập nhật pHead trỏ đến node mới.



#### 3.4. Tìm kiếm phần tử trong danh sách

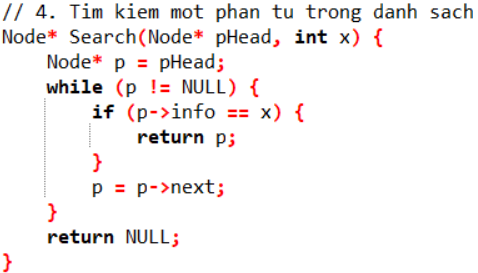
**Main idea:** Kiểm tra xem danh sách có chứa phần tử cần tìm hay không.

**Giải thuật:**

Duyệt từng node trong danh sách.

Nếu tìm thấy giá trị x, trả về con trỏ đến node đó.

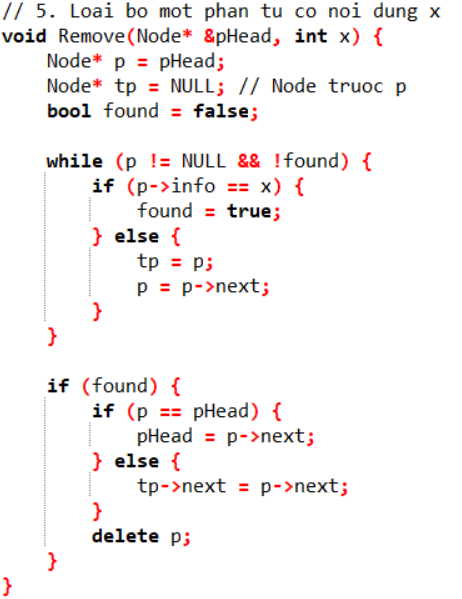
Nếu duyệt hết danh sách mà không tìm thấy, trả về NULL.



#### 3.5. Xóa phần tử có giá trị cho trước

**Main idea:** Loại bỏ một phần tử khỏi danh sách nếu có.

**Giải thuật:**

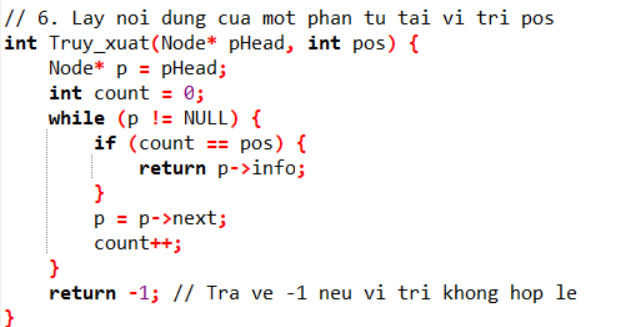
* Duyệt danh sách tìm phần tử có giá trị x.
* Nếu là phần tử đầu tiên, cập nhật pHead.
* Nếu ở giữa hoặc cuối, cập nhật con trỏ next của node trước nó.
* Giải phóng bộ nhớ của node bị xóa.

#### 3.6. Truy xuất phần tử theo vị trí

**Main idea:** Lấy giá trị của phần tử tại vị trí xác định.

**Giải thuật:**

* Duyệt danh sách và đếm số node.
* Khi đếm đến vị trí mong muốn, trả về giá trị info của node đó.

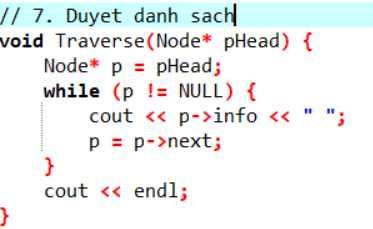


#### 3.7. Duyệt danh sách

**Main idea:** Hiển thị tất cả phần tử trong danh sách liên kết.

**Giải thuật:**

* Bắt đầu từ pHead, duyệt từng phần tử của danh sách.
* In ra giá trị của mỗi phần tử.
* Lặp cho đến khi NULL (danh sách kết thúc).

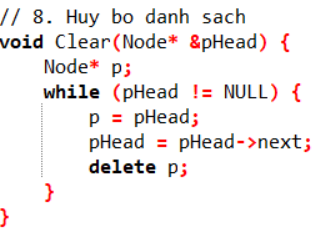


#### 3.8. Hủy danh sách

**Main idea:** Xóa tất cả các phần tử để giải phóng bộ nhớ.

**Giải thuật:**

* Lặp qua danh sách, mỗi lần xóa phần tử đầu tiên.
* Cập nhật pHead trỏ đến phần tử tiếp theo.
* Tiếp tục cho đến khi danh sách rỗng.

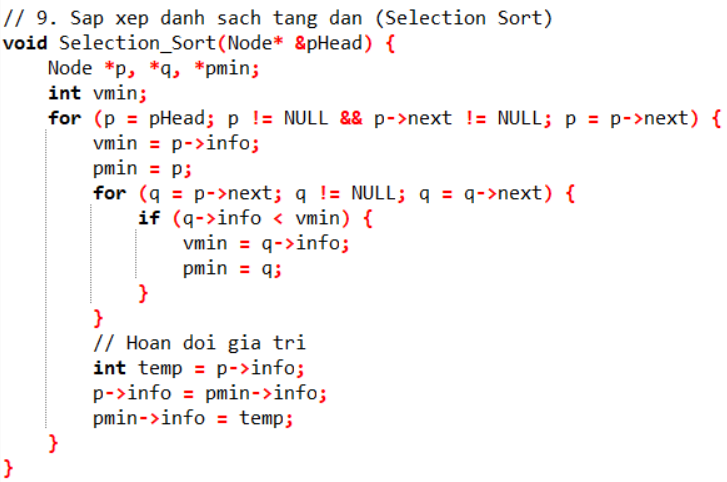


#### 3.9. Sắp xếp danh sách tăng dần (Selection Sort)

**Main idea:** Sắp xếp danh sách liên kết theo thứ tự tăng dần bằng thuật toán Selection Sort.

**Giải thuật:**

* Duyệt từng phần tử p trong danh sách.
* Tìm phần tử nhỏ nhất pmin trong các phần tử còn lại.
* Hoán đổi giá trị giữa p và pmin.
* Tiếp tục cho đến khi danh sách được sắp xếp.



INPUT VÀ OUTPUT SAU KHI CODE XONG CHƯƠNG TRÌNH:

|  |  |
| --- | --- |
| Input | Output |
| Nhap so luong phan tu: 5  Nhap 5 gia tri: 7 3 9 2 5  Nhap gia tri can tim: 3  Nhap vi tri can truy xuat: 2  Nhap gia tri can xoa: 9 | Danh sach vua nhap: 5 2 9 3 7  Tim thay 3 trong danh sach!  Gia tri tai vi tri 2: 9  Danh sach sau khi xoa phan tu 9: 5 2 3 7  Danh sach sau khi sap xep tang dan: 2 3 5 7  Danh sach sau khi xoa toan bo: Danh sach rong! |

