**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**

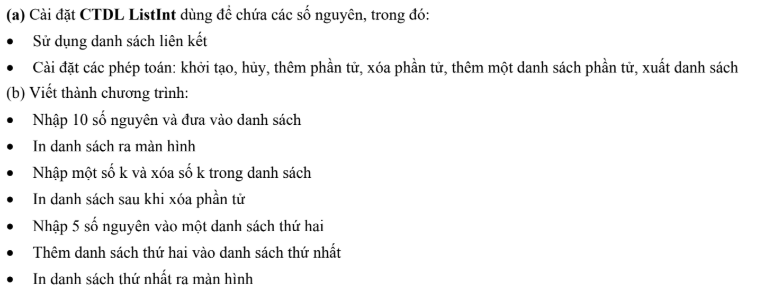
**Mssv:3124411315**

**Giảng viên hướng dẫn: Đỗ Như Tài**

**Thành phố Hồ Chí Minh, ngày 15 tháng 3 năm 2025**

**BÀI 3: DANH SÁCH LIÊN KẾT**

**Câu 1:**



## 1. Main Idea :

Chương trình này triển khai một danh sách liên kết đơn (Singly Linked List) để quản lý danh sách các số nguyên. Các thao tác chính bao gồm:

* **Thêm phần tử vào đầu danh sách** bằng phương pháp chèn đầu
* **Xóa một phần tử có giá trị cụ thể** nếu nó tồn tại trong danh sách.
* **In danh sách liên kết** để hiển thị dữ liệu.
* **Gộp hai danh sách liên kết** bằng cách nối danh sách thứ hai vào cuối danh sách thứ nhất.
* **Giải phóng bộ nhớ** khi kết thúc chương trình để tránh rò rỉ bộ nhớ.

Chương trình hoạt động dựa trên việc nhập hai danh sách số nguyên từ bàn phím, thực hiện các thao tác thêm, xóa, gộp, in danh sách và cuối cùng là giải phóng bộ nhớ trước khi kết thúc.

## 2. Giải Thuật:

### 2.1. Cấu trúc dữ liệu

Chương trình sử dụng cấu trúc Node để biểu diễn một nút trong danh sách liên kết đơn, trong đó:

* data: lưu trữ giá trị số nguyên.
* next: là con trỏ trỏ đến nút kế tiếp trong danh sách.

Danh sách liên kết được quản lý thông qua lớp ListInt với con trỏ head trỏ đến phần tử đầu tiên.

### 2.2. Các thao tác trên danh sách

#### a) ****Thêm phần tử vào đầu danh sách****

* Tạo một nút mới với giá trị được nhập vào.
* Trỏ next của nút mới đến head hiện tại.
* Cập nhật head để trỏ đến nút mới.

#### b) ****Xóa phần tử có giá trị cụ thể****

* Duyệt danh sách từ đầu đến cuối để tìm phần tử có giá trị cần xóa.
* Nếu tìm thấy, cập nhật con trỏ next của nút trước để bỏ qua nút cần xóa.
* Giải phóng bộ nhớ của nút bị xóa.

#### c) ****In danh sách liên kết****

* Bắt đầu từ head và duyệt từng nút.
* In ra giá trị của từng nút.

#### d) ****Gộp hai danh sách liên kết****

* Nếu danh sách thứ nhất rỗng, gán head của danh sách thứ hai cho danh sách thứ nhất.
* Nếu không, duyệt đến nút cuối cùng của danh sách thứ nhất và trỏ next của nó đến head của danh sách thứ hai.
* Đặt head của danh sách thứ hai thành NULL để tránh trùng lặp dữ liệu.

#### e) ****Xóa toàn bộ danh sách****

* Duyệt danh sách, giải phóng từng nút một.
* Cập nhật head về NULL sau khi xóa hết phần tử.

### 2.3. Luồng thực thi trong main()

1. Nhập vào 10 số nguyên và chèn vào danh sách thứ nhất.
2. In danh sách thứ nhất ra màn hình.
3. Nhập một số nguyên cần xóa và xóa nó khỏi danh sách thứ nhất.
4. In danh sách sau khi xóa.
5. Nhập vào 5 số nguyên và chèn vào danh sách thứ hai.
6. In danh sách thứ hai.
7. Gộp danh sách thứ hai vào danh sách thứ nhất.
8. In danh sách sau khi gộp.
9. Giải phóng bộ nhớ và kết thúc chương trình.

### 2.4. Độ phức tạp thuật toán

* **Chèn phần tử**: O(1) (do chỉ cần cập nhật con trỏ head).
* **Xóa phần tử**: O(n) (trong trường hợp xấu nhất, phải duyệt toàn bộ danh sách để tìm phần tử cần xóa).
* **In danh sách**: O(n) (duyệt qua toàn bộ danh sách để in).
* **Gộp danh sách**: O(n) (tìm nút cuối danh sách thứ nhất, sau đó nối danh sách thứ hai vào).
* **Xóa toàn bộ danh sách**: O(n) (duyệt từng phần tử và giải phóng bộ nhớ).

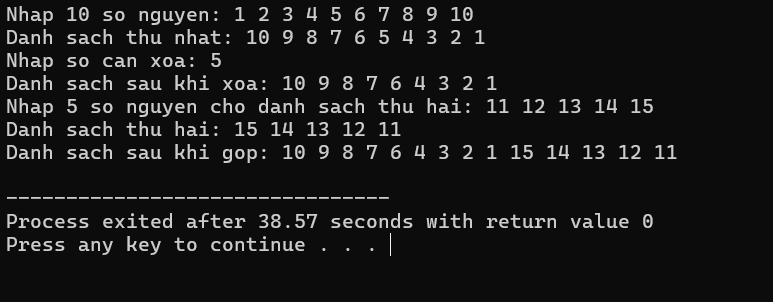
Chương trình này sử dụng danh sách liên kết đơn để tiết kiệm bộ nhớ hơn so với mảng động, đồng thời dễ dàng thao tác thêm, xóa phần tử một cách linh hoạt.

1. **INPUT VÀ OUTPUT:**

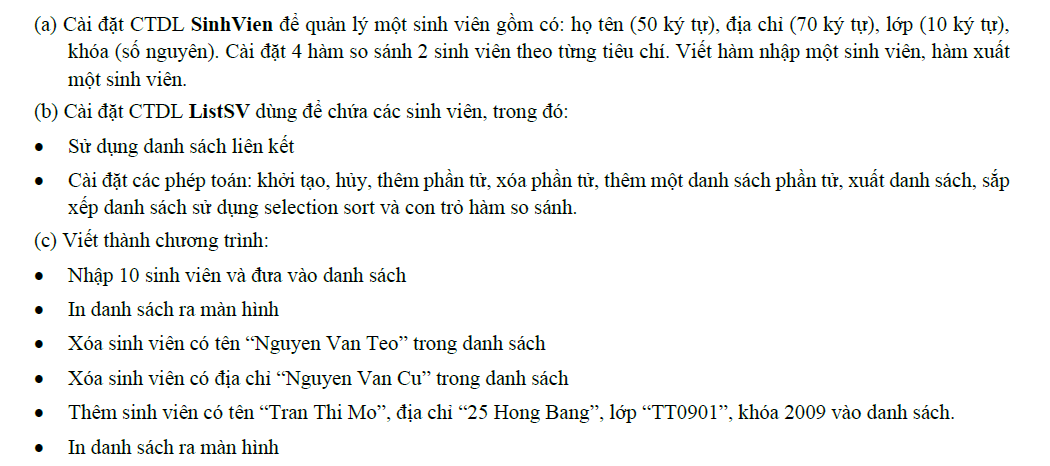
**Input: Nhap 10 so nguyen: 1 2 3 4 5 6 7 8 9 10**

**Nhap so can xoa: 5**

**Nhap 5 so nguyen cho danh sach thu hai: 11 12 13 14 15**

**Output:** 

**Câu 2:**



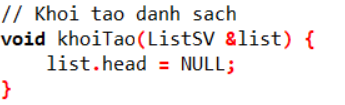
1. **Main Idea (Ý tưởng chính)**

Chương trình này quản lý danh sách sinh viên bằng cách sử dụng danh sách liên kết đơn. Các chức năng chính bao gồm:

* ****Thêm sinh viên** vào danh sách.**
* ****Xuất danh sách sinh viên** để hiển thị thông tin.**
* ****Xóa sinh viên theo tên** khỏi danh sách.**
* ****Thêm sinh viên mới** vào danh sách sau khi xóa.**

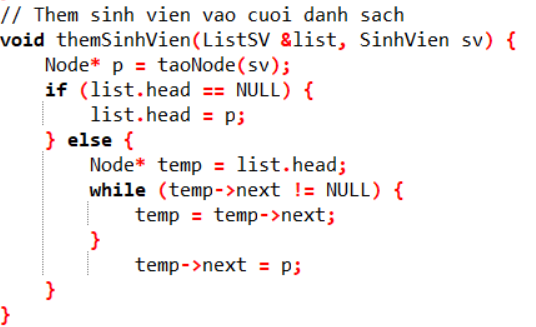
1. **Giải thuật:**
   1. **Cấu trúc dữ liệu:**

* **struct SinhVien**: Lưu trữ thông tin của một sinh viên, bao gồm họ tên, địa chỉ, lớp và khóa.
* **struct Node**: Đại diện cho một nút trong danh sách liên kết, chứa dữ liệu của một sinh viên và con trỏ next trỏ đến nút tiếp theo.
* **struct ListSV**: Quản lý danh sách sinh viên, chứa con trỏ head trỏ đến sinh viên đầu tiên.
  1. ****Khởi tạo danh sách:****



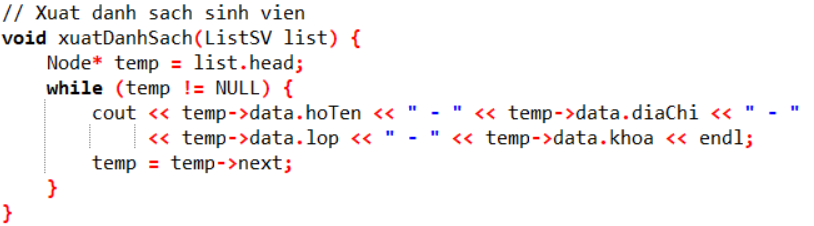
**Giải thuật**:

* Gán head bằng NULL, tức là danh sách rỗng.
  1. **Thêm sinh viên vào danh sách:**



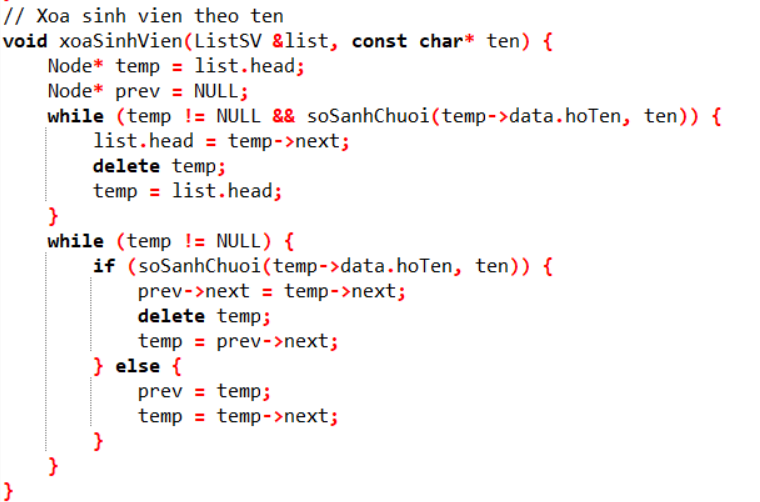
**Giải thuật**:

* Tạo một node mới chứa thông tin sinh viên.
* Nếu danh sách rỗng (head == NULL), gán head = p.
* Nếu không, duyệt đến cuối danh sách (temp->next != NULL) rồi gán temp->next = p để thêm sinh viên vào cuối.
  1. **Xuất danh sách sinh viên:**



**Giải thuật**:

* Khởi tạo temp = head.
* Lặp cho đến khi temp == NULL, mỗi lần in ra thông tin sinh viên hiện tại rồi di chuyển đến sinh viên tiếp theo (temp = temp->next).
  1. **Xóa sinh viên theo tên:**



**Giải thuật:**

* Kiểm tra xem node đầu tiên có phải sinh viên cần xóa không. Nếu có, cập nhật head trỏ đến node kế tiếp và giải phóng node.
* Duyệt danh sách, tìm node có tên trùng.
* Nếu tìm thấy node cần xóa, cập nhật prev->next = temp->next để bỏ qua node đó.
* Giải phóng bộ nhớ của node bị xóa.
  1. **Chương trình chính:**

**Giải thuật:**

1. Khởi tạo danh sách sinh viên.
2. Nhập danh sách sinh viên từ bàn phím.
3. Hiển thị danh sách sinh viên.
4. Xóa một số sinh viên theo tên.
5. Hiển thị danh sách sau khi xóa.
6. Thêm sinh viên mới.
7. Hiển thị danh sách sau khi thêm.
8. Tìm một sinh viên theo tên.
9. Giải phóng bộ nhớ trước khi kết thúc chương trình.
10. **INPUT VÀ OUTPUT:**

**Input: Nhập số lượng sinh viên: 3**

**Nhập thông tin sinh viên thứ 1:**

**Họ tên: Nguyen Van Teo**

**Địa chỉ: 123 Nguyen Trai**

**Lớp: IT01**

**Khóa: 2023**

**Nhập thông tin sinh viên thứ 2:**

**Họ tên: Nguyen Van Cu**

**Địa chỉ: 456 Le Loi**

**Lớp: IT02**

**Khóa: 2022**

**Nhập thông tin sinh viên thứ 3:**

**Họ tên: Tran Van B**

**Địa chỉ: 789 Tran Hung Dao**

**Lớp: IT03**

**Khóa: 2021**

**Output:**

