**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo

Description automatically generated-----o0o----

**HỌC PHẦN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

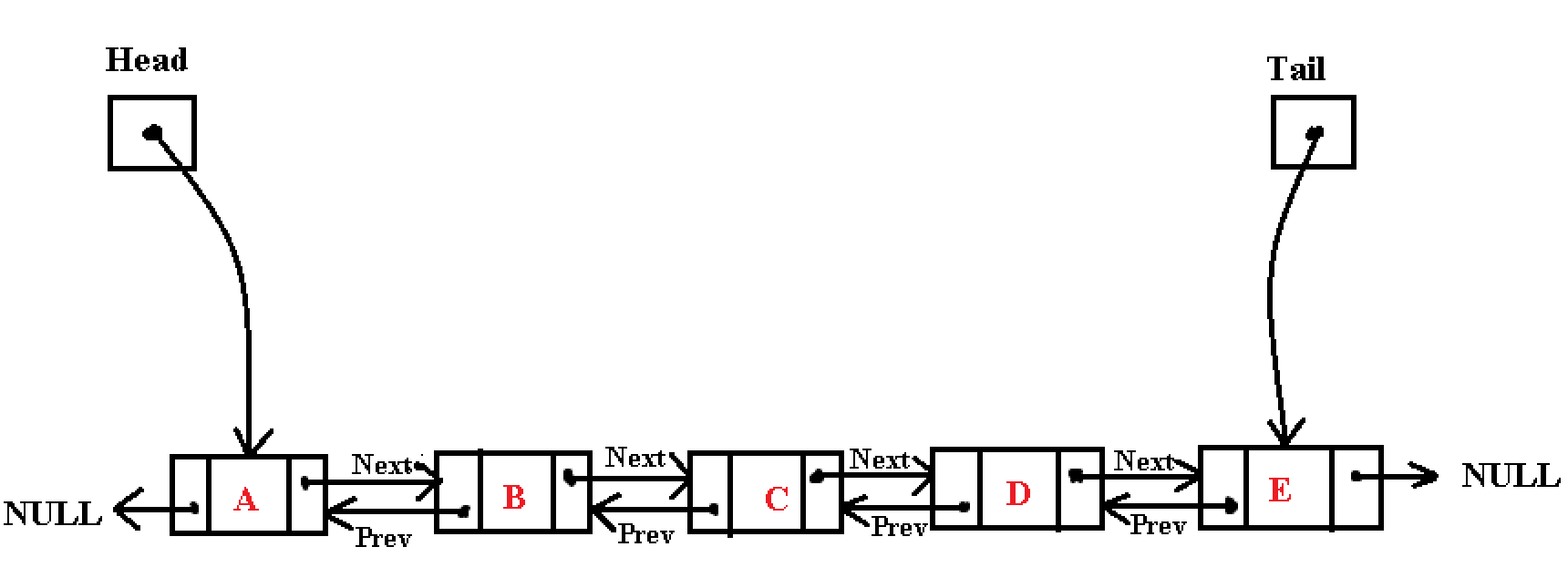
**ĐỀ TÀI:**

**TÌM HIỂU CẤU TRÚC DỮ LIỆU DOUBLE LINKED LIST: CẤU TRÚC DƯ LIỆU, DIỄN GIẢI CÁC THAO TÁC CƠ BẢN CỦA CẤU TRÚC DỮ LIỆU TƯƠNG ỨNG TRONG PYTHON. CÀI ĐẶT VÀ VÍ DỤ MINH HOẠ CHO CÁC THAO TÁC ĐÓ TRÊN PYTHON. ỨNG DỤNG VIẾT CHƯƠNG TRÌNH C/C++ THỰC HIỆN BÀI TOÁN QUẢN LÝ SÁCH TRONG THƯ VIỆN.**

|  |  |
| --- | --- |
| **Thành viên nhóm:**  **1. Nguyễn Duy Ân**  **2. Thái Ngọc Thạch**  **3. Nguyễn Huyện Anh Khang**  **4. Ngô Gia Huy**  **5. Nguyễn Nhật Tường** | **Giảng viên hướng dẫn:**  **Ngô Thanh Hùng** |

1. **Khái niệm Danh sách liên kết đôi(Double Linked List).**

Danh sách liên kết đôi (double linked list) là một cấu trúc dữ liệu trong lập trình, được sử dụng để lưu trữ các phần tử dưới dạng các nút, mỗi nút chứa một giá trị và hai con trỏ, một con trỏ trỏ tới nút trước đó và một con trỏ trỏ tới nút tiếp theo.



Mỗi nút trong danh sách liên kết đôi có thể được truy cập trực tiếp thông qua các con trỏ và do đó cho phép truy cập ngược lại từ cuối danh sách đến đầu danh sách. Điều này khác với danh sách liên kết đơn (single linked list) chỉ cho phép truy cập theo hướng một chiều từ đầu đến cuối.

Một danh sách liên kết đôi có thể được sử dụng để giải quyết nhiều vấn đề, chẳng hạn như tìm kiếm, sắp xếp và chèn các phần tử vào giữa danh sách một cách hiệu quả. Tuy nhiên, danh sách liên kết đôi sử dụng nhiều bộ nhớ hơn so với danh sách liên kết đơn và phức tạp hơn trong việc triển khai.

1. **Các ưu nhược điểm, các vấn đề và cách giải quyết trong danh sách liên kết đôi.**

Ưu điểm:

* Cho phép truy cập nút tiếp theo và nút trước đó một cách hiệu quả.
* Cho phép thêm, xoá phần tử vào vị trí bất kỳ trong danh sách một cách dễ dàng.
* Cho phép duyệt danh sách liên kết đôi cả theo chiều thuận và chiều ngược.
* Cho phép tối ưu hóa các thao tác chèn, xoá, sắp xếp, tìm kiếm.

Nhược điểm:

* Sử dụng nhiều bộ nhớ hơn so với danh sách liên kết đơn.
* Cần phải cập nhật các con trỏ của nút trước, nút sau khi thêm hoặc xoá một phần tử.
* Phức tạp hơn trong việc triển khai so với danh sách liên kết đơn.

Các vấn đề phổ biến với danh sách liên kết đôi bao gồm:

* Xử lý các trường hợp đặc biệt như nút đầu tiên và nút cuối cùng.
* Xử lý trường hợp danh sách trống và chỉ chứa một phần tử.
* Xử lý việc giải phóng bộ nhớ khi xoá danh sách.

Để giải quyết các vấn đề trên, cần sử dụng các giải pháp như:

* Sử dụng các biến trỏ và điều kiện kiểm tra để xử lý các trường hợp đặc biệt.
* Sử dụng cấu trúc điều kiện để xử lý trường hợp danh sách trống hoặc chỉ chứa một phần tử.
* Sử dụng hàm giải phóng bộ nhớ để giải phóng các nút khỏi danh sách khi xoá danh sách.

1. **Các thao tác trên danh sách liên kết đôi trong lập trình Python từ thư viện có sẵn.**
2. Khởi tạo một danh sách liên kết đôi.

from collections import deque

double\_linked\_list = deque()

Đoạn code “**from collections import deque**” sử dụng để import lớp “**deque**” từ module “**collections**” vào mã nguồn Python. Khi đã import được “**deque**”, chúng ta có thể sử dụng nó để tạo ra một danh sách liên kết kép (double linked list) trong Python.

Đoạn code “**double\_linked\_list = deque()**” tạo ra một đối tượng danh sách liên kết đôi rỗng (deque) và gán cho biến “**double\_linked\_list**”.

1. Thêm phần tử vào đầu/cuối.

Để thêm một phần tử vào đầu danh sách liên kết đôi, ta có thể sử dụng phương thức **appendleft():**

double\_linked\_list.appendleft(1)

double\_linked\_list.appendleft(2)

double\_linked\_list.appendleft(3)

Kết quả: 

Tương tự, để thêm một phần tử vào cuối danh sách liên kết đôi, ta có thể sử dụng phương thức **append()**:

double\_linked\_list.append(1)

double\_linked\_list.append(2)

double\_linked\_list.append(3)

Kết quả: 

1. Thêm vào vi trị bất kỳ

Để thêm một phần tử vào một vị trí bất kỳ trong danh sách liên kết kép, ta có thể sử dụng phương thức **insert():**

double\_linked\_list.insert(vị trí, giá trị cần thêm)

Ví dụ:

double\_linked\_list = deque([1, 2, 3])

double\_linked\_list.insert(1, 4)

Chỉ mục trong Python bắt đầu từ 0, do đó chỉ mục 1 tương ứng với vị trí thứ 2 trong danh sách liên kết đôi.

Kết quả: 

1. Xoá phần tử ở đầu/cuối.

Đê xoá phần tử ở đầu danh sách, ta sử dụng phương pháp **popleft()**:

double\_linked\_list = deque([1,2,3])

double\_linked\_list.popleft()

Kết quả: 

Để xoá phần tử ở danh sách, ta sử dụng phương pháp **pop()**:

double\_linked\_list = deque([1,2,3])

double\_linked\_list.pop()

Kết quả: 

1. Xoá phần tử bất kỳ.

Để xoá phần tử bất kỳ, ta sử dụng phương pháp **remove()**:

double\_linked\_list.remove(giá trị cần xoá)

Ví dụ:

double\_linked\_list = deque([1,2,3])

double\_linked\_list.remove(2)

Kết quả: 

1. Tìm kiếm phần tử.

Để tìm kiếm phần tử trong danh sách, ta sử dụng phương pháp **index()**:

double\_linked\_list = deque([1,2,3])

index = double\_linked\_list.index(2)

print(index)

Kết quả sẽ là **1**, tức là phần tử có giá trị **2** nằm ở vị trí thứ 1 trong danh sách liên kết kép. Nếu không tìm thấy phần tử trong danh sách liên kết kép, phương thức “**index()**” sẽ raise ra một ngoại lệ “**ValueError**”.



1. Sắp xếp phần tử.

Để sắp xếp một danh sách liên kết kép theo thứ tự tăng dần, ta có thể sử dụng phương thức sort():

-Sắp xếp tăng dần:

double\_linked\_list = deque([5, 2, 8, 1, 6])

double\_linked\_list.sort()

Kết quả: 

-Sắp xếp giảm dần:

double\_linked\_list = deque([5, 2, 8, 1, 6])

double\_linked\_list.sort(reverse=True)

Kết quả: 

1. Duyệt danh sách.

Để duyệt danh sách, ta sử dụng phương pháp **printf()**:

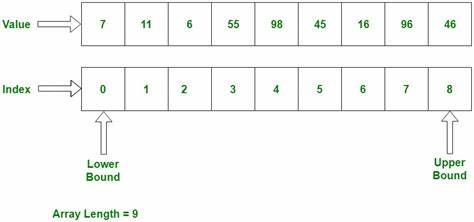
print(double\_linked\_list)

1. **Bài toán thực tế Quản lý sách trong thư viện.**

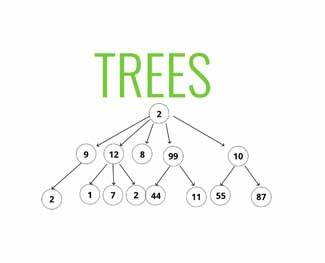
Ý nghĩa: Quản lý sách trong thư viện là bài toán rất quan trọng trong lĩnh vực thư viện. Việc quản lý sách đảm bảo sự an toàn và bảo quản tài liệu, đồng thời cung cấp cho người đọc dễ dàng tìm kiếm và mượn sách.

**Các giải pháp khác nhau:**

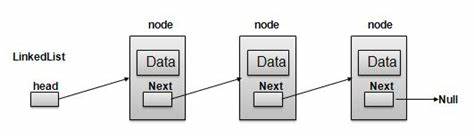
* Dùng mảng: Sử dụng mảng 1 chiều hoặc 2 chiều để lưu trữ thông tin về sách và độc giả. Tuy nhiên, việc thêm, xoá, sửa sách sẽ khó khăn và tốn thời gian.



* Dùng cây: Sử dụng cây nhị phân tìm kiếm hoặc cây đỏ-đen để lưu trữ thông tin về sách và độc giả. Giải pháp này giúp tối ưu hóa thời gian tìm kiếm, tuy nhiên vẫn khó khăn trong việc thêm, xoá, sửa sách.



* Dùng danh sách liên kết: Sử dụng danh sách liên kết đơn hoặc đôi để lưu trữ thông tin về sách và độc giả. Giải pháp này linh hoạt hơn trong việc thêm, xoá, sửa sách, nhưng có thể dẫn đến tình trạng lãng phí không gian bộ nhớ.



**Giải pháp đề xuất:**

Sử dụng danh sách liên kết đôi để lưu trữ thông tin về sách và độc giả, vì giải pháp này linh hoạt hơn trong việc thêm, xoá, sửa sách. Các thao tác cần hiện thực:

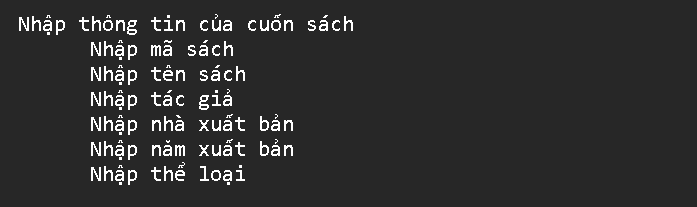
* Thêm sách: Thêm một sách mới vào danh sách liên kết. Nếu sách đã tồn tại, thông báo cho người dùng. Nếu danh sách đã đầy, thông báo cho người dùng.
* Xoá sách: Xoá một sách khỏi danh sách liên kết. Nếu sách không tồn tại, thông báo cho người dùng.
* Tìm sách: Tìm kiếm thông tin của một sách trong danh sách liên kết.
* Sắp xếp sách: sắp xếp thứ tự các cuốn sách theo tên, mã số, thể loại,…
* Mượn sách: Đánh dấu sách đã được mượn bởi một độc giả.
* Trả sách: Đánh dấu sách đã được trả bởi một độc giả.

**Mô tả chi tiết thuật toán:**

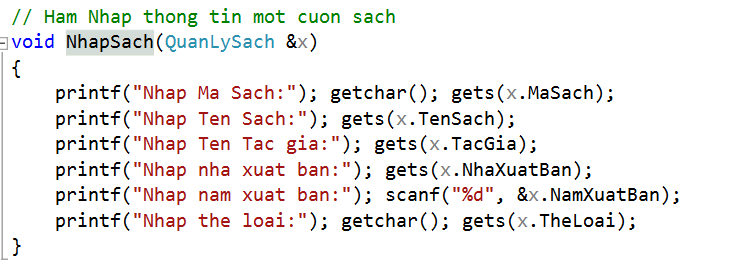
a. Thêm sách mới:

* Bước 1: Nhập thông tin về sách mới (tên, tác giả, nhà xuất bản, năm xuất bản, thể loại, mã sách).
* Bước 2: Cập nhật lại thông tin số lượng sách trong thư viện.

Mã Giả:



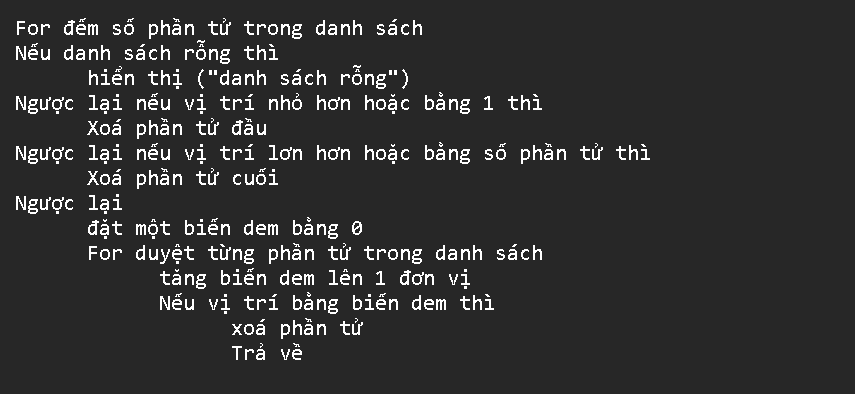
Mã code:



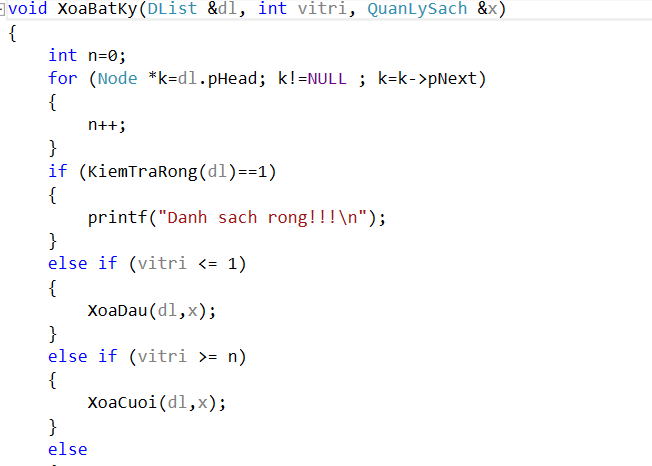
b. Xoá một cuốn sách:

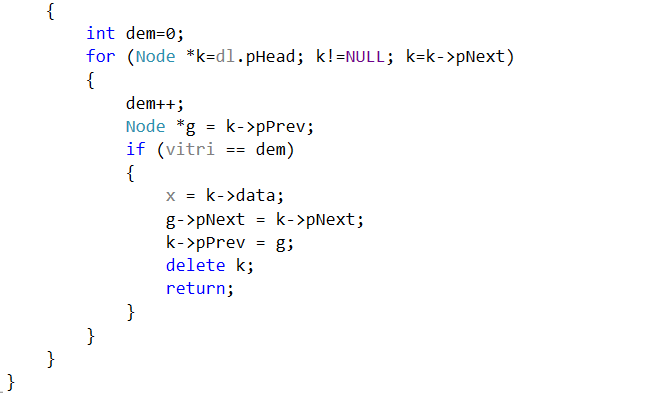
* Bước 1: Nhập vị trí cuốn sách cần xoá.
* Bước 2: Nếu danh sách rỗng thì thực hiện không thành công. Ngược lại nếu danh sách không rỗng thì tìm vị trí phần tử cần xoá.
* Bước 3: Cập nhật lại thông tin số lượng sách trong thư viện.

Mã Giả:



Mã Code:

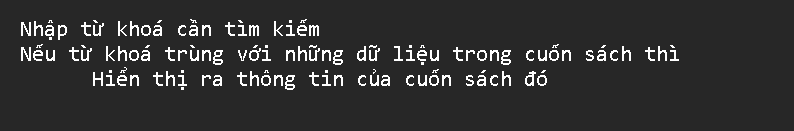




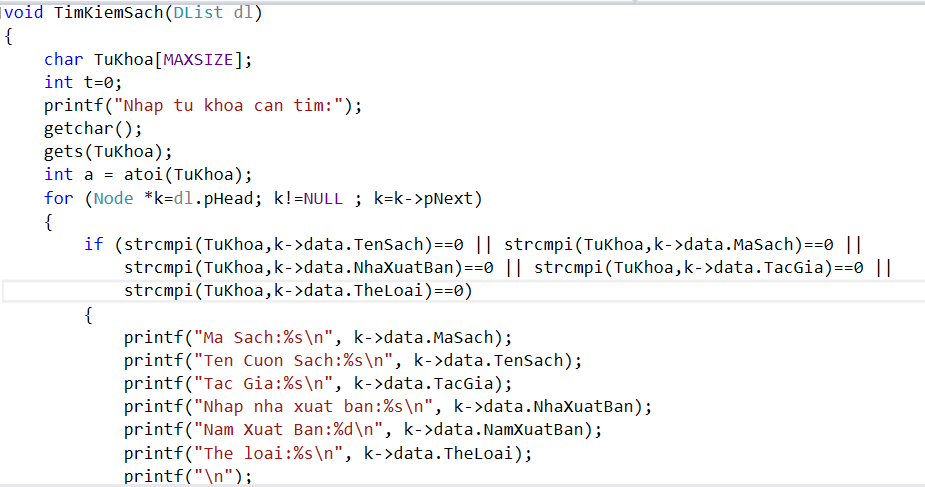
c. Tìm kiếm sách:

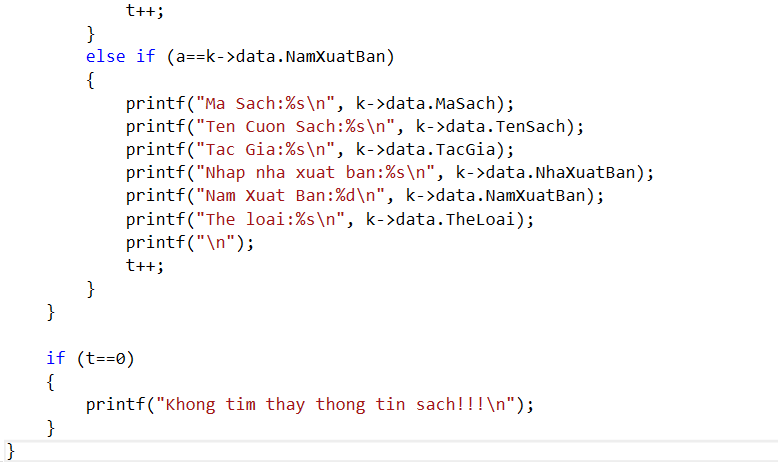
* Bước 1: Nhập từ khóa tìm kiếm (tên sách, tên tác giả, thể loại sách).
* Bước 2: Tìm kiếm sách theo từ khóa và hiển thị kết quả tìm kiếm.

Mã Giả:



Mã Code:

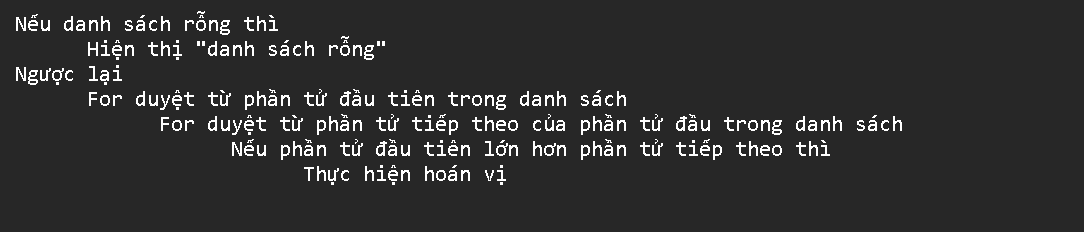




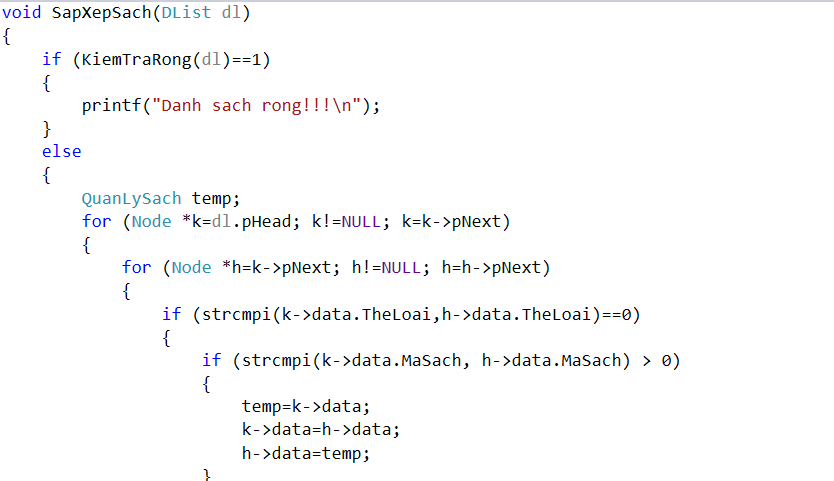
d. Sắp xếp danh sách:

* Bước 1: Sắp xếp danh sách theo tên, mã sách, thể loại,…
* Bước 2: Nếu danh sách rỗng thì không xuất ra gì cả, thực hiện không thành công. Ngược lại xuất ra danh sách sau khi hoàn tất sắp xếp.

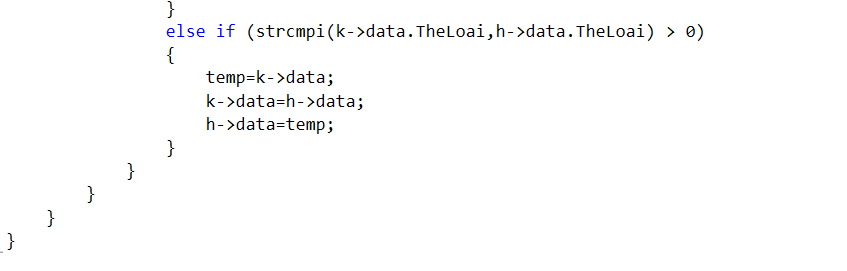
Mã Giả:



Mã Code:



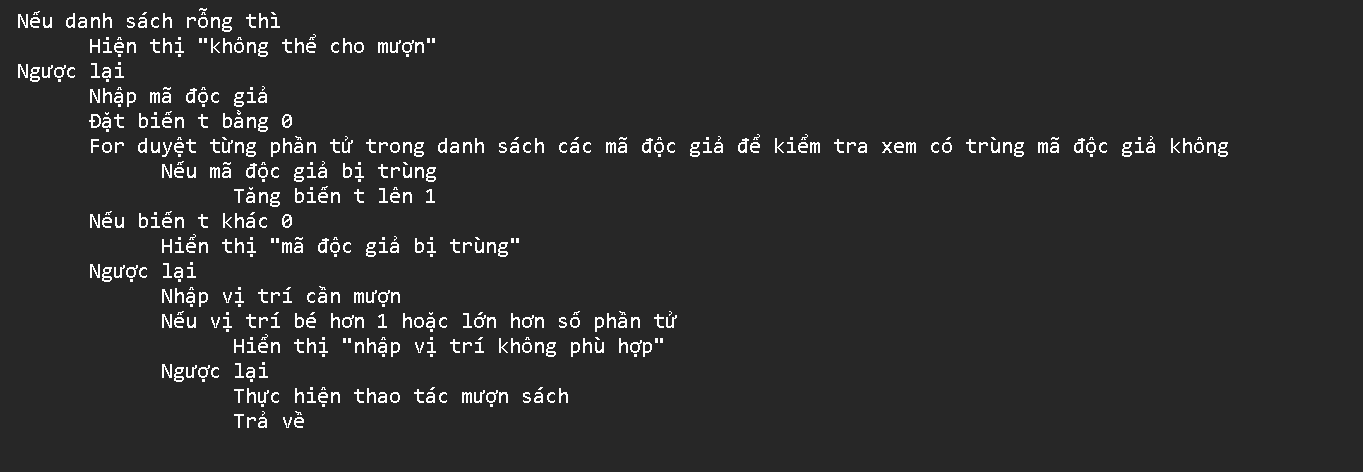




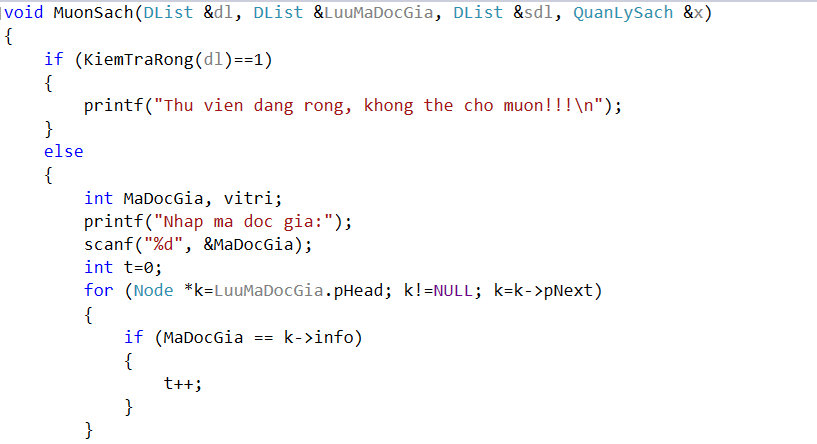
e. Mượn sách:

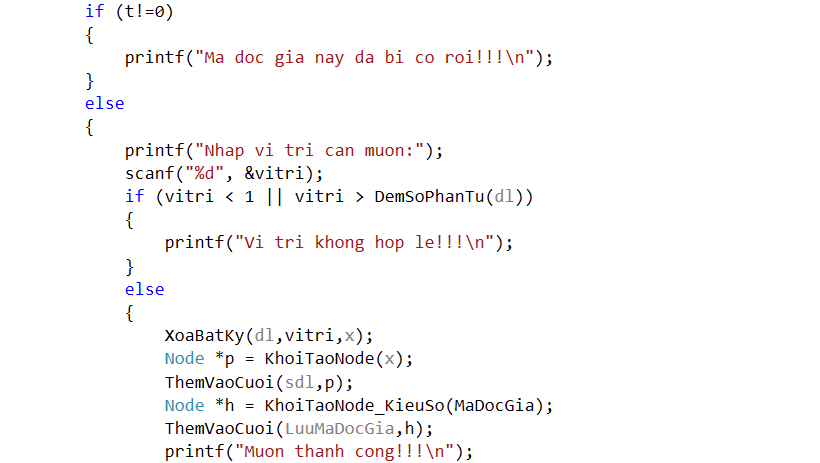
* Bước 1: Nhập mã số của độc giả để tiến hành mượn sách.
* Bước 2: Nếu danh sách rỗng thì không thực hiện thao tác. Ngược lại thực hiện thao tác mượn sách.
* Bước 3: Lưu mã số của độc giả để sau này còn trả lại sách.

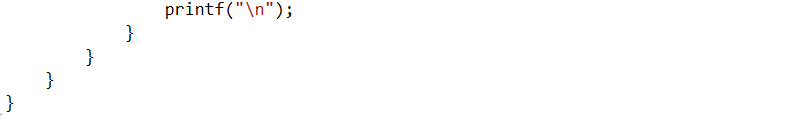
Mã Giả:



Mã Code:



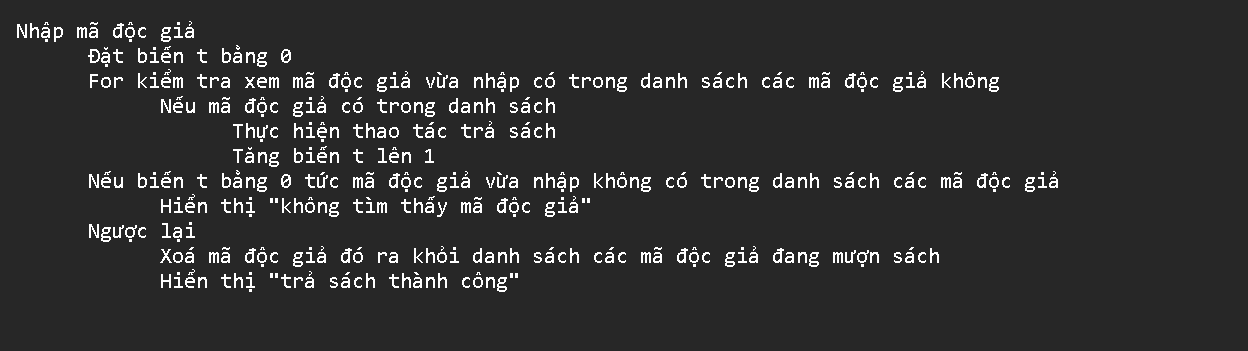




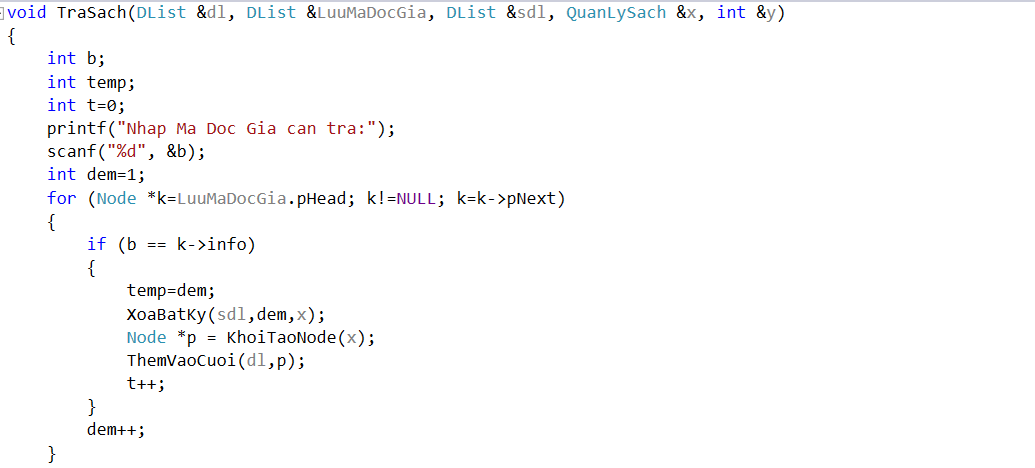
f. Trả sách:

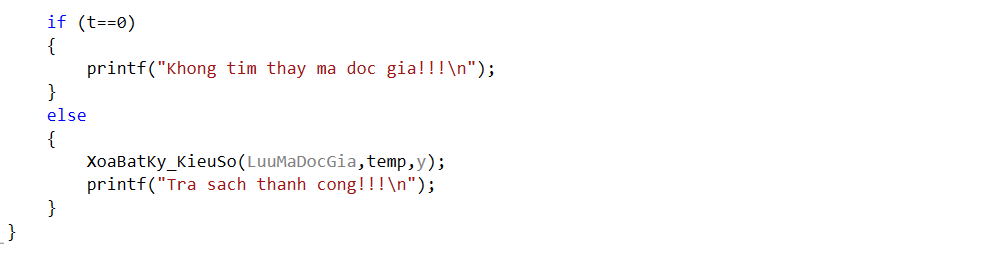
* Bước 1: Nhập mã số của độc giả đã mượn sách để tiến hành trả sách.
* Bước 2: Tiến hành trả sách về thư viện.

Mã Giả:

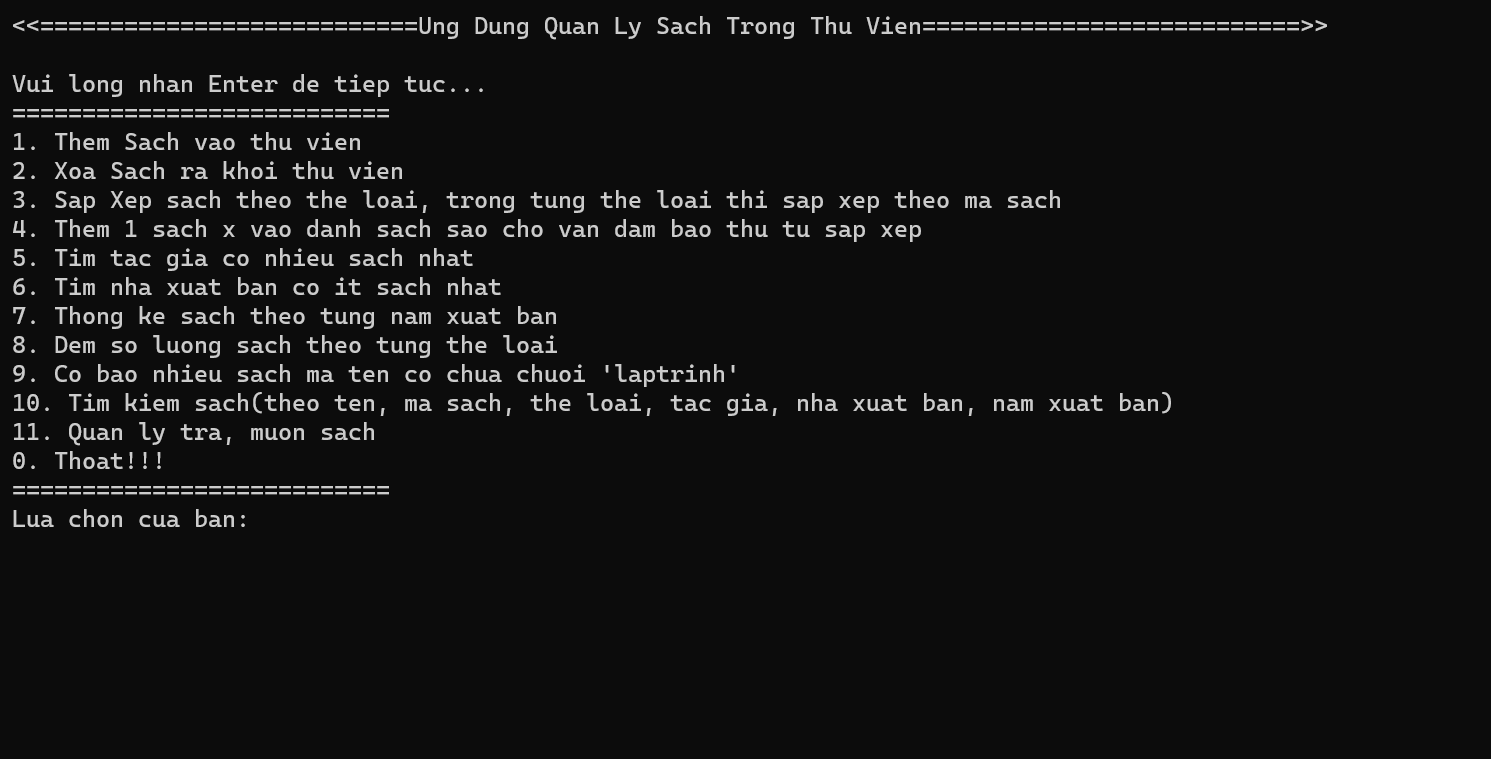


Mã Code:

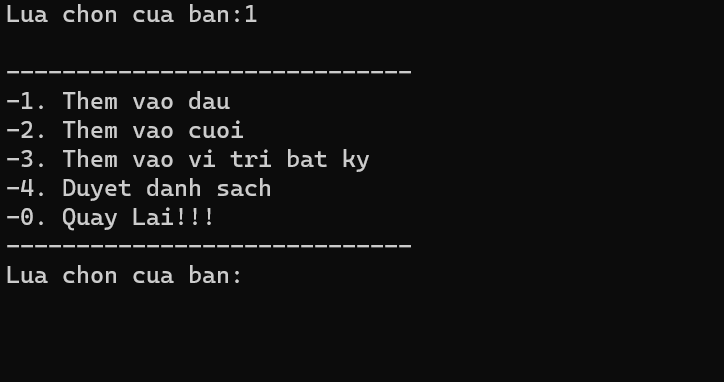




**Kiểm thử - mô tả chạy ứng dụng:**

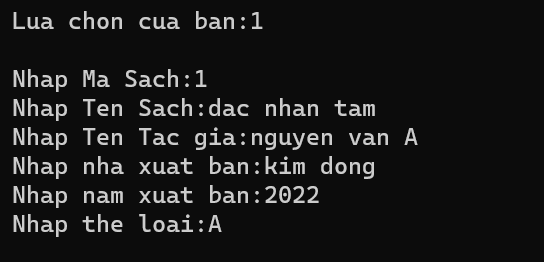


1. Thêm sách mới:

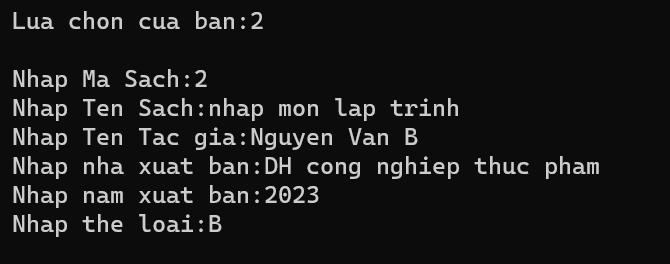


Người dung thực hiện các chức năng như:

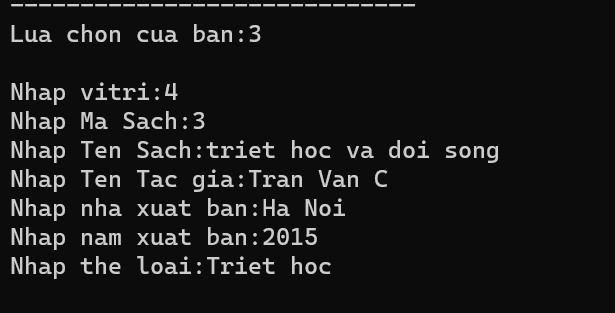
* Thêm vào đầu.
* Hệ thống sẽ thực hiện thao tác thêm vào đầu danh sách sau khi người dung nhập đầy đủ thông tin yêu cầu.



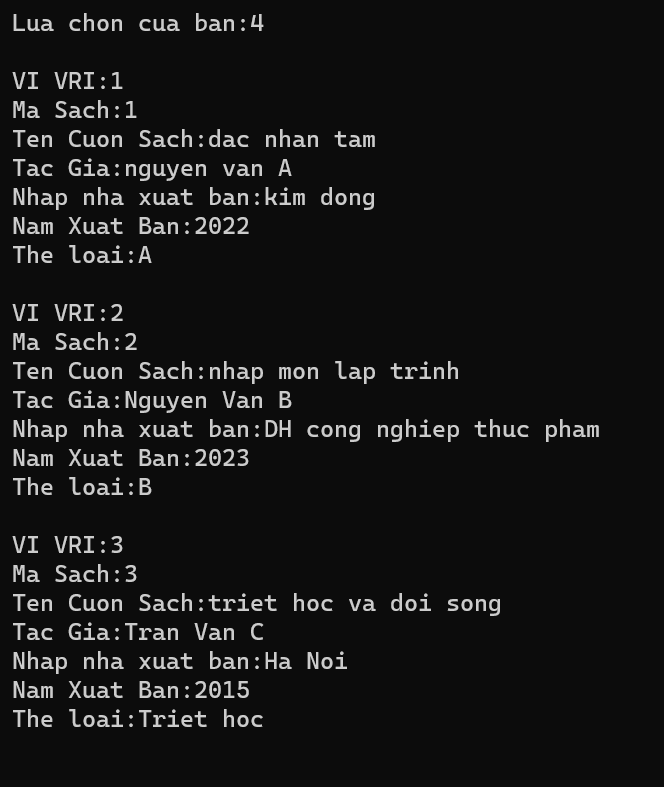
* Thêm vào cuối.
* Hệ thống sẽ thực hiện thao tác thêm vào cuối danh sách sau khi người dung nhập đầy đủ thông tin yêu cầu.



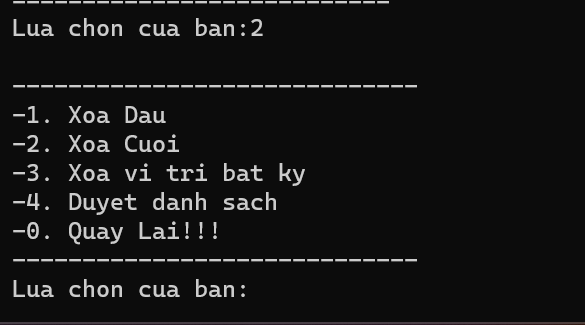
* Thêm vào vị trí bất kỳ.
* Hệ thống sẽ thực hiện thao tác thêm bất kỳ vào danh sách.



* Sau khi thêm có thể duyệt danh sách để xem.
* Người dung có thể xem qua danh sách sau khi thực hiện các thao tác thêm phần tử.

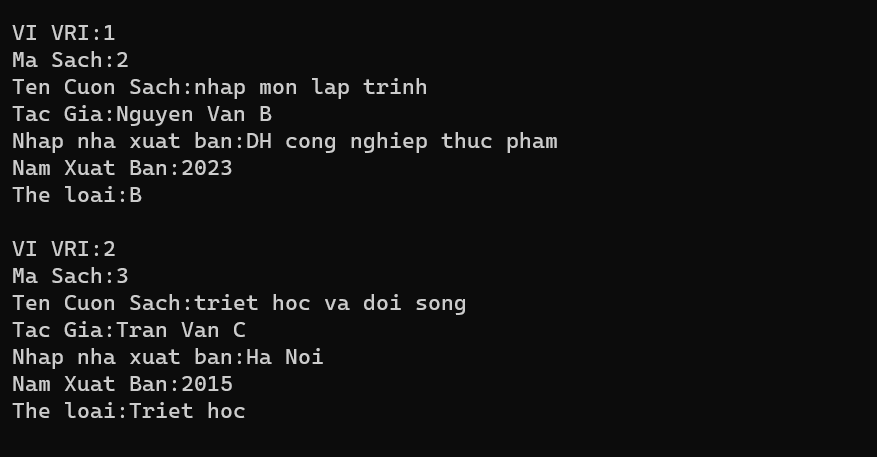


1. Xoá sách:

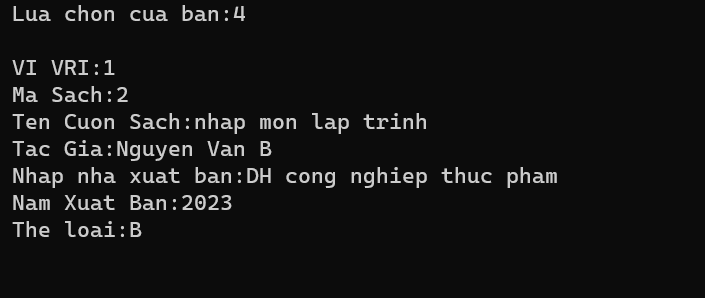


Người dung thực hiện các chức năng như:

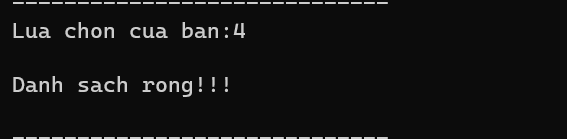
* Xoá đầu.
* Hệ thống sẽ thực hiện thao tác xoá phần tử đầu danh sách.



* Xoá cuối.
* Hệ thống sẽ thực hiện thao tác xoá phần tử cuối.



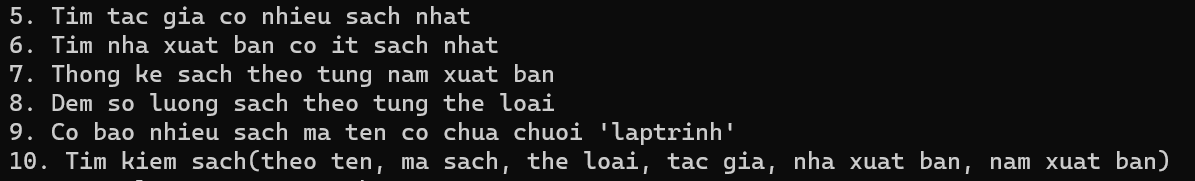
* Xoá vị trí bất kỳ.
* Hệ thống thực hiện thao tác xoá phần tử bất kỳ.

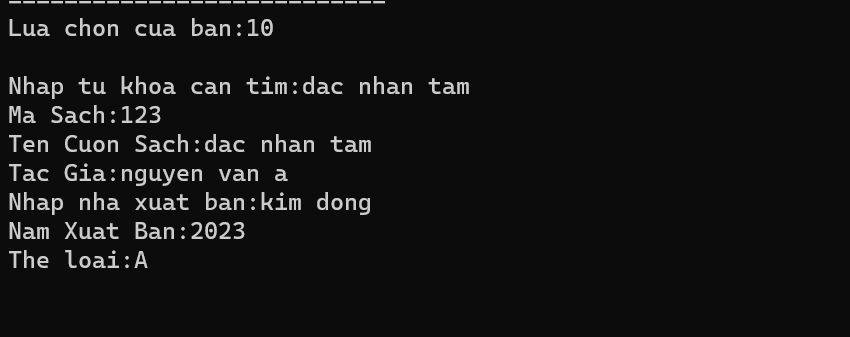


* Sau khi xoá có thể duyệt danh sách để xem.

1. Tìm kiếm sách:

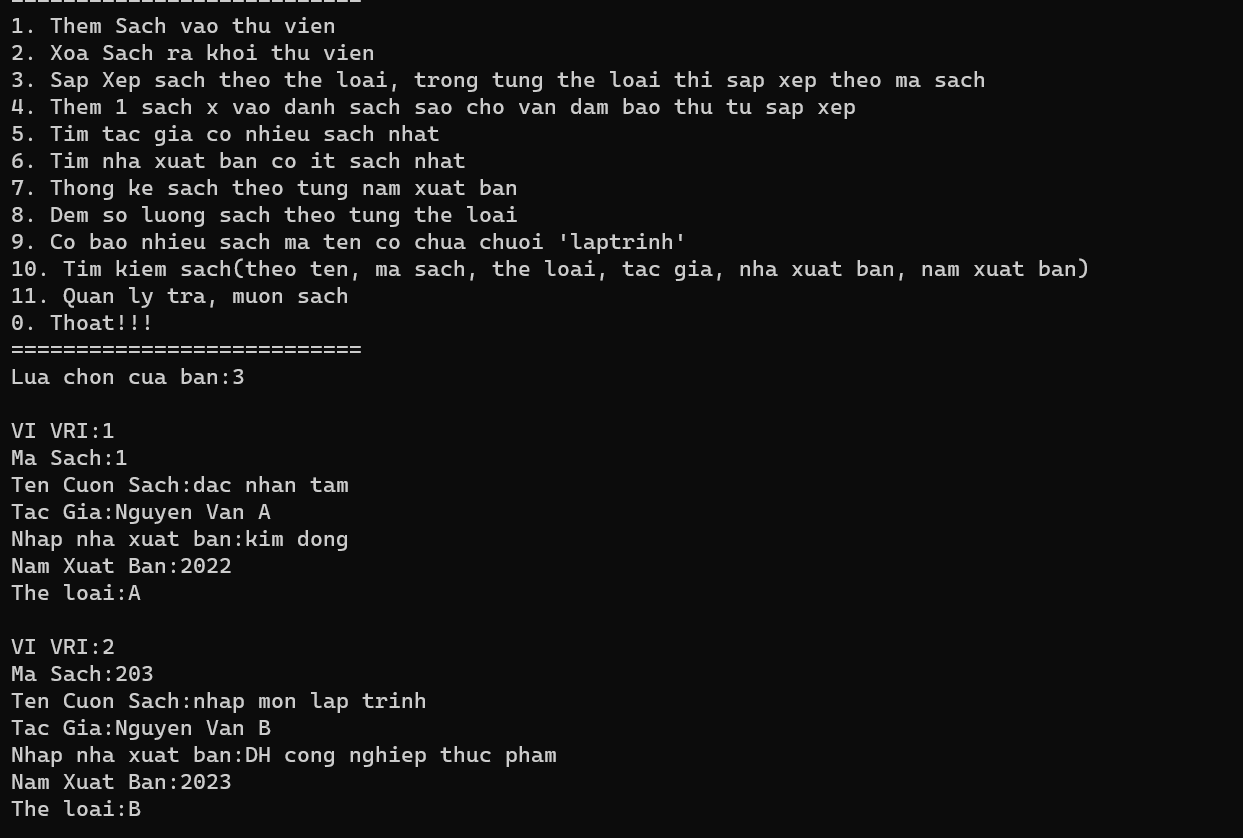
Người dung thực hiện các chức năng như tìm kiếm sách theo tên, mã, thể loại, tác giả,… .Hoặc tìm kiếm bằng những mục có sẵn.



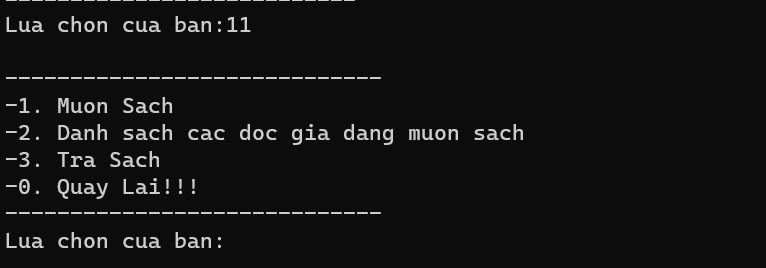


1. Sắp xếp sách:

* Hệ thống sẽ tự động sắp sếp danh sách theo thể loại, trong từng thể loại thì sắp xếp theo mã sách.

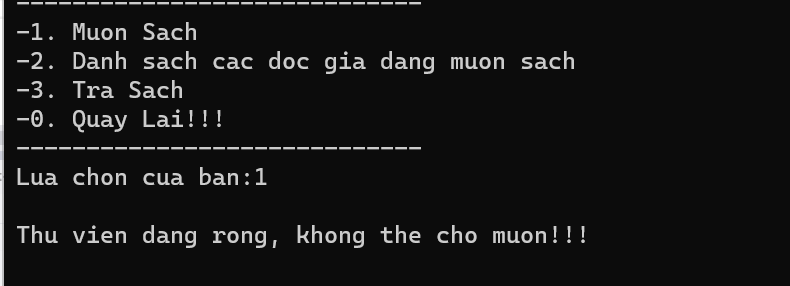


1. Quản lý mượn, trả sách:

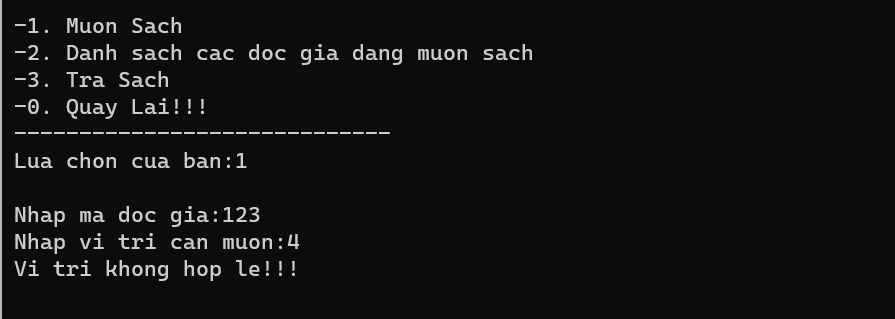


+Người dung sẽ thực hiện các chức năng như:

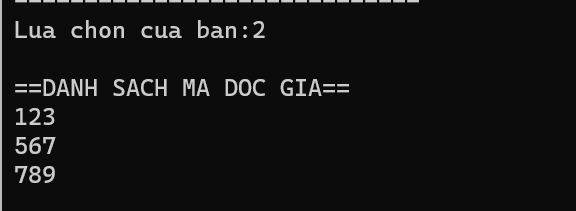
* Mượn sách.
* Hệ thống sẽ in ra dòng “Thư viện đang rỗng, không thể cho mượn!!!” nếu trong danh sách đang rỗng.



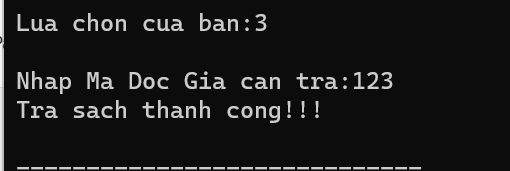
* Hệ thống sẽ in ra dòng “vị trí sách không hợp lệ” khi nhập vi trí không đúng hoặc vị trí đó không có cuốn sách.

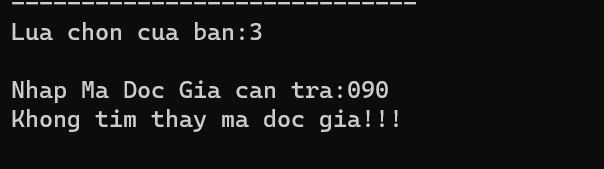


* Xem danh sách các độc giả đang mượn sách(xem mã độc giả).
* Hệ thống sẽ in ra danh sách mã số các độc giả đang mượn sách.



* Trả sách:
* Hệ thống sẽ thực hiện việc trả sách nếu nhập đúng mã độc giả hoặc ngược lại.





---HẾT---