
1. Hãy xây dựng cấu trúc dữ liệu cho danh sách liên kết đơn để lưu trữ các số nguyên và viết các thao tác cơ bản như mô tả sau: (3.5 điểm)

- Khởi tạo: *public Node(int value); public LinkedList();*
- Thêm một phần tử vào đầu danh sách: *public void AddFirst(int value)*
- Thêm một phần tử vào sau một phần tử khác trong danh sách: *public void AddAfter(Node pre, int value);*
- Xóa một phần tử sau một phần tử khác trong danh sách: *public void RemoveAfter(Node pre);*
- Tìm kiếm một phần tử có dữ liệu là value trong danh sách: *public Node Find(int value);*
- Sắp xếp danh sách tăng dần dùng giải thuật Interchange Sort: *public void InterchangeSort();*

2. Trong chương trình chính, sử dụng class LinkedList ở câu 1, tạo ra 2 danh sách liên kết l1 và l2. Viết và gọi thực thi các chức năng sau: (5.5 điểm).

- a. Viết hàm nhập danh sách N số nguyên dương (có thể Random) bằng cách thêm từng phần tử vào đầu danh sách. Gọi thực thi cho 2 danh sách l1, l2.
- b. Viết hàm xuất danh sách. Gọi thực thi cho 2 danh sách l1, l2.
- c. Viết hàm thêm một phần tử (có dữ liệu là tổng giá trị các số lẻ trong danh sách l2) vào sau phần tử có dữ liệu bằng x đầu tiên trong danh sách l1 (với x được nhập từ bàn phím). *Lưu ý: Nếu l2 không có số lẻ hoặc x không tìm thấy trong l1 thì không thêm.*

Ví dụ:

Input	Output
L1: 2 <u>5</u> 4 7 5 3 L2: 1 2 3 4 5 //TongLe = 9 X: 5	L1: 2 <u>5</u> 9 4 7 5 3

- d. Viết hàm xóa một phần tử sau phần tử có dữ liệu là số nhỏ nhất trong danh sách. Gọi thực thi cho danh sách l2. (Giả sử danh sách chỉ có một phần tử có dữ liệu nhỏ nhất).

Ví dụ:

Input	Output
-------	--------

L2: 2 <u>1</u> 4 7 5 3	L2: 2 <u>1</u> 7 5 3
-------------------------------	----------------------

- e. Viết hàm tạo ra danh sách l3 bao gồm các phần tử vừa có trong danh sách l1 vừa có trong danh sách l2. (Giả sử các phần tử trong l1, l2 đều có giá trị phân biệt)

Ví dụ:

Input	Output
L1: 3 4 <u>5</u> <u>2</u> 7 <u>6</u> 1 L2: <u>5</u> <u>6</u> <u>2</u> 8 9 0	L3: 5 2 6

- f. Sắp xếp danh sách l2 tăng dần.

-Hết-