

[DSLK - Tìm] Tìm danh sách điểm 3D.

Hãy viết chương trình cài đặt danh sách liên kết đơn để nhập vào một danh sách A gồm tọa độ n điểm trong không gian 3 chiều. Sau đó, nhập hai giá trị b và e, với $b \leq e$, để chọn tập R chứa các điểm $A[i]$ sao cho khoảng cách từ gốc tọa độ $O(0,0,0)$ đến điểm $A[i]$ là $d(O, A[i]) \in [b, e]$. Thứ tự dò tìm từ đầu danh sách A đến cuối danh sách A.

Đầu vào:

- Dòng đầu là số n ($1 \leq n \leq 10000$).
- n dòng kế tiếp, mỗi dòng 3 số có phần thập phân x, y và z là giá trị các chiều của một điểm trong danh sách A.
- 2 số b,e

Đầu ra: Nếu tập R rỗng thì xuất ra "KHONG". Nếu R không rỗng thì liệt kê các giá trị x, y và z của tọa độ các điểm trong R theo từng dòng, mỗi dòng là một điểm.

Ví dụ 1:

Input	Output
6	636.94 176.5 496.06
636.94 176.5 496.06	9.06 499.59 658.4
48.96 14.87 21.83	398.16 399.52 604.37
9.06 499.59 658.4	
0.38 74.03 10.76	
1.64 53.26 59.72	
398.16 399.52 604.37	
826.32 974.31	

Ví dụ 2:

Input	Output
5	KHONG
87.32 63.79 34.01	
23.4 101.38 47.42	
75.29 92.08 48.96	
0.38 74.03 10.76	
1.64 53.26 59.72	
826.32 974.31	

[DSLK - Tìm] Tìm điểm 3D.

Hãy viết chương trình cài đặt danh sách liên kết đơn và nhập vào một danh sách A gồm tọa độ n điểm trong không gian 3 chiều. Sau đó, nhập m điểm p_i và kiểm tra xem các điểm p_i này có xuất hiện trong danh sách A hay không.

Yêu cầu:

- Không sử dụng thêm thư viện nào khác ngoài **iostream** và **cmath** đã được include sẵn.
- Không sử dụng **mảng**.
- Cài đặt hàm **void run()** để hoàn thành các yêu cầu của đề bài. Hàm **run()** này được hàm **main** gọi để thực hiện yêu cầu. Nếu không cài đặt hàm **run()**, chương trình sẽ không chạy được.

Đầu vào:

- Dòng đầu là số n ($1 \leq n \leq 10000$)
- n dòng kế tiếp, mỗi dòng 3 số có phần thập phân lần lượt là giá trị x, y và z của điểm trong danh sách.
- Dòng tiếp theo là số m ($1 \leq m \leq 3500$).
- m dòng tiếp theo, mỗi dòng có 3 số có phần thập phân lần lượt là giá trị x, y và z của điểm p_i.

Đầu ra: Gồm m dòng, dòng thứ i là kết quả tìm điểm p_i trong A. Kết quả tìm là

- "KHONG" nếu không tìm thấy,
- Chỉ số của điểm đầu tiên được tìm thấy trong A theo hướng tìm từ đầu danh sách đến cuối danh sách.

Ví dụ 1:

Input	Output
5	0
48.96 14.87 21.83	KHONG
0.38 74.03 10.76	3
1.64 53.26 59.72	
15.29 90.53 43.64	
87.32 63.79 34.01	
3	
48.96 14.87 21.83	
38.16 41.07 12.23	
15.29 90.53 43.64	

[DSLK] Tổng hai số nguyên.

Hãy hoàn thành hàm **void Sum(List, List, List &)** để cộng hai số nguyên được biểu diễn bằng hai danh sách liên kết đơn và trả kết quả là một danh sách liên kết đơn thứ ba. Mỗi phần tử trong các danh sách liên kết đơn là một chữ số trong số nguyên.

Lưu ý:

- 1) Chỉ cài đặt hàm Sum.
- 2) Kiểu dữ liệu và một số hàm liên quan đã được cài đặt sẵn. Sinh viên có thể đọc để sử dụng.
- 3) Đầu vào và đầu ra đã được xử lý sẵn và phù hợp với định dạng nhập/xuất.

Input	Output
3 1 4 9 5 3 2 9 9 1	3 3 1 4 0
61 2 2 0 7 0 6 5 2 3 0 0 9 5 9 9 6 4 7 8 1 4 7 3 1 1 8 1 2 8 0 7 2 2 0 8 8 0 6 4 5 5 9 7 2 9 6 8 5 7 4 5 6 1 7 7 3 8 6 2 4 1	2 2 0 7 0 6 5 2 3 0 0 9 5 9 9 6 4 7 8 1 4 7 3 1 1 8 1 2 8 7 3 2 4 0 5 3 1 0 4 3 6 7 7 0 5 1 9 6 2 6 9 2 0 7 2 1 3 5 8 4 7
32 6 6 0 1 9 6 5 0 3 9 8 0 7 9 7 5 5 1 0 5 2 3 5 8 9 4 7 4 9 6 0 6	

[DSLK] In ngược thứ tự nhập 10 phần tử đầu tiên thỏa điều kiện.

Hãy viết một chương trình nhập vào n số nguyên sau đó in ra không quá 10 phần tử được nhập vào đầu tiên có giá trị lớn hơn hoặc bằng 90. Các phần tử này được in ngược với thứ tự nhập. Biết rằng, trong bài này, không được sử dụng mảng và không được include thêm thư viện.

Đầu vào của bài toán gồm:

- Dòng đầu tiên là số n.
- n số kế tiếp là các số nguyên từ 0 đến 99.

Đầu ra của bài toán là các số nguyên đầu tiên có giá trị lớn hơn hoặc bằng 90. Số lượng số nguyên được in ra không vượt quá 10 và liền sau kết quả là một khoảng trắng và chuỗi "END".

Input	Output
25 11 97 29 39 44 14 66 76 16 5 59 21 93 46 54 97 72 80 49 63 93 49 17 95 77	95 93 97 93 97 END

[DSLK - Sắp xếp] Sắp xếp học sinh theo thứ tự tăng dần.

Hãy hoàn thành hàm **Sort(List &)** để sắp xếp danh sách học sinh theo thứ tự tăng dần của tên, nếu tên trùng nhau thì xét thứ tự Họ đệm tăng dần. Nếu tên và họ đệm trùng nhau, thì xét tuổi tăng dần. Một người có tuổi nhỏ hơn nếu có ngày sinh (bao gồm ngày, tháng, năm) lớn hơn. Thứ tự của tên, họ đệm cũng chính là thứ tự chuỗi theo bảng mã ASCII.

Lưu ý:

- 1) Chỉ cài đặt hàm Sort và những hàm liên quan (nếu cần) để sắp xếp sao cho thời gian sắp xếp nhỏ.
- 2) Kiểu dữ liệu và một số hàm liên quan đã được cài đặt sẵn. Sinh viên có thể đọc để sử dụng.
- 3) Đầu vào và đầu ra đã được xử lý sẵn và phù hợp với định dạng nhập/xuất.

Input	Output
3 Le Tuan Anh 1 4 11 2005 8.5 8.5 10 6.5 7.5 7 Le Tuan Anh 1 29 11 2006 10 9 9 8.5 7 6 Le Tuan Anh 1 10 12 2006 6 7 8.5 7.5 9 10	Le Tuan Anh 1 10/12/2006 6 7 8.5 7.5 9 10 Le Tuan Anh 1 29/11/2006 10 9 9 8.5 7 6 Le Tuan Anh 1 4/11/2005 8.5 8.5 10 6.5 7.5 7

[DSLK] Nhập và tìm danh sách học sinh.

Dùng danh sách liên kết thay cho mảng, vector, hãy viết chương trình chọn một danh sách học sinh có điểm trung bình các môn lớn hơn hoặc bằng 9.0 và in ra danh sách đó theo thứ tự được nhập vào. Trong đó, điểm trung bình bằng tổng của 6 điểm gồm; Toán, Văn, Lý, Hóa, Anh, Sinh; chia cho 6.

Đầu vào:

- Số nguyên dương n, cho biết có n thông tin học sinh.

- n khối kế tiếp, mỗi khối là toàn bộ thông tin của một học sinh, được đọc bằng hàm InputElement(Hocsinh &) đã được cài đặt sẵn.

Đầu ra: Dùng hàm Output(List) đã được viết sẵn.

Lưu ý:

- 1) Cần cài đặt các hàm phù hợp để sử dụng kiểu DSLK đơn List.
- 2) Cần cài đặt hàm main để thực hiện chương trình.
- 3) Đầu vào (hàm InputElement) và đầu ra (hàm Output) đã được xử lý sẵn và phù hợp với định dạng nhập/xuất.

Input	Output
3	Bui Minh Quan 1 13/9/2006 10 9.5 8 10 10 9
Le Phu	//Chỉ có Bui Minh Quan có điểm trung bình là 9.42 lớn hơn 9.0 nên
Quy	được chọn
1 22 10 2006 10 10 9.5 6.5	
7.5 8	
Bui Minh	
Quan	
1 13 9 2006 10 9.5 8 10 10	
9	
Bui Duy	
Quy	
1 29 9 2005 10 8.5 10 8 10	
0.5	

[DSLK - Tìm] Tìm học sinh xuất hiện sau cùng trong danh sách.

Bài này phải dùng danh sách liên kết thay cho mảng, vector. Hãy cài đặt hàm **FindLast(List, Hocsinh)** để tìm học sinh cuối cùng có họ đệm và tên trùng với học sinh cần tìm. Thứ tự tìm từ đầu danh sách đến cuối danh sách. Hàm trả về Node chứa thông tin của học sinh tìm được, hoặc là NULL nếu không tìm được.

Đầu vào:

- Số nguyên dương n, cho biết có n thông tin học sinh.
- n khối kế tiếp, mỗi khối là toàn bộ thông tin của một học sinh, được đọc bằng hàm InputElement(Hocsinh &) đã được cài đặt sẵn.

Đầu ra: Đã được cài đặt sẵn trong hàm main.

Lưu ý: Chỉ cài đặt hàm **FindLast(List, Hocsinh)**.

Input	Output
6 Dinh Van Phuong 1 23 10 2006 10 7 6.5 8.5 10 9 Le Hoang Quan 1 25 8 2006 6.5 9 10 8.5 10 8 Le Tuan Anh 1 29 11 2006 10 9 9 8.5 7 6 Le Tuan Anh 1 10 12 2006 6 7 8.5 7.5 9 10 Nguyen The Nhu Quynh 0 20 7 2006 8 6.5 7 8.5 10 8 Le Tuan Anh 1 4 11 2005 8.5 8.5 10 6.5 7.5 7 Le Tuan Anh 0 0 0 0 0 0 0 0 0	Le Tuan Anh 1 4 11 2005 8.5 8.5 10 6.5 7.5 7

[DSLK] Vị trí chính giữa của dãy số chẵn.

Hãy viết một chương trình nhập vào n số nguyên sau đó in ra số ở vị trí chính giữa của dãy số chẵn theo đúng thứ tự nhập vào. Nếu không vị trí chính giữa, thì lấy số ở vị trí lệch bên trái gần vị trí chính giữa nhất. Biết rằng, trong bài này, không được sử dụng mảng và không được include thêm thư viện.

Đầu vào của bài toán gồm:

- Dòng đầu tiên là số n.
- n số kế tiếp là các số nguyên dương.

Đầu ra của bài toán là một số duy nhất, chính là số ở vị trí chính giữa

Input	Output
10 1 2 3 4 5 6 7 8 9 11	4
10 1 2 3 4 5 6 7 8 9 10	6

[DSLK] In danh sách số chẵn ngược thứ tự nhập.

Dùng danh sách liên kết thay cho mảng, vector, hãy viết chương trình in ra các số chẵn từ các số đã nhập vào sao cho thứ tự in ra ngược với thứ tự nhập.

Đầu vào:

- Số nguyên dương n, cho biết có n số nguyên dương.
- n nguyên dương kế tiếp là các số được nhập vào.

Đầu ra: Một dòng duy nhất chứa các số nguyên dương chẵn theo thứ tự ngược với thứ tự nhập. Mỗi số cách nhau một khoảng trắng.

Lưu ý:

- 1) Cấu trúc List và một số hàm CreateList, CreateNode, AddHead, AddTail đã được cài đặt sẵn. Sinh viên có thể tham khảo để sử dụng.
- 2) Cần cài đặt hàm main để thực hiện chương trình.

Input	Output
8 1 8 2 7 9 3 6 6	6 6 2 8