

Đề bài

Câu 1 (3đ): Sắp xếp

Cho một danh sách sinh viên gồm các thông tin ID kiểu int và email kiểu char[21] không chứa dấu cách. Hãy sắp xếp danh sách sinh viên này theo thứ tự từ điển của email. Đầu vào có dòng đầu chứa n là số sinh viên ($n \leq 20000$). Ở n dòng tiếp theo mỗi dòng chứa ID và email cách nhau bởi dấu cách. Sau khi sắp xếp, chương trình in ra danh sách, mỗi sinh viên trên một dòng gồm ID và email cách nhau bởi dấu cách. Vì bị hạn chế thời gian, nên các bạn nên lựa chọn phương pháp sắp xếp nhanh nhất có thể.

Lưu ý: Các giải thuật sắp xếp cơ bản (vd sắp xếp chèn, sắp xếp nổi bọt) có thể không hoàn thành hết các testcase do thời gian chạy vượt quá thời gian cho phép.

Ví dụ:

INPUT:

3

2 b@b.c

3 a@b.c

1 m@b.c

OUTPUT:

3 a@b.c

2 b@b.c

1 m@b.c

Câu 2 (3đ): Danh sách liên kết

Cho cấu trúc có dạng:

```
typedef struct Node {  
    int value;  
    struct Node* next;  
} Node;
```

Hãy viết chương trình C để xây dựng danh sách liên kết đơn. Để xây dựng và hiển thị danh sách này, chương trình đọc và thực hiện các lệnh có dạng như sau:

- InsertHead v : thêm node có value bằng v vào đầu danh sách (Lưu ý: không thực hiện lệnh nếu v đã tồn tại trong danh sách hoặc v có giá trị âm)
- InsertTail v : thêm node có value bằng v vào cuối danh sách (Lưu ý: không thực hiện lệnh nếu v đã tồn tại trong danh sách hoặc v có giá trị âm)

- Print k : k có giá trị là 0 hoặc 1. Nếu k = 0 hiển thị danh sách. Nếu k = 1, trước khi hiển thị danh sách cần chuyển các giá trị chẵn lên đầu và các giá trị lẻ xuống dưới danh sách mà không làm thay đổi thứ tự giữa các giá trị chẵn với nhau cũng như thứ tự giữa các giá trị lẻ với nhau. Nếu danh sách rỗng, chương trình không in ra gì cả. Sau lệnh Print, chương trình giải phóng bộ nhớ đã xin cấp phát và kết thúc.

Ví dụ:

INPUT:

InsertTail 37
InsertHead -14
InsertHead 63
InsertHead 9
InsertTail -22
InsertTail 50
InsertHead -3
InsertTail 28
InsertHead -7
InsertHead 41
InsertHead -18
InsertTail 12
InsertTail -31
InsertTail 55
InsertTail 2
InsertHead -46
InsertHead 24
InsertHead -9
InsertTail 33
InsertTail -27
Print 1

OUTPUT:

24 50 28 12 2 41 9 63 37 55 33

Câu 3 (4đ): Cây nhị phân tìm kiếm

Lịch thi học kỳ của một cơ sở giáo dục bao gồm nhiều dòng, mỗi dòng gồm 2 thông tin, cách nhau bởi một **dấu cách**, với định dạng như sau:

Mã-lớp Phòng thi

Ví dụ: 20230001 TC-209

Định dạng dữ liệu: Mã lớp là xâu có 8 ký tự không chứa dấu cách; phòng thi là xâu có 6 ký tự không chứa dấu cách.

Yêu cầu: Sử dụng cây nhị phân tìm kiếm, dựa trên so sánh thông tin mã lớp để lưu trữ lịch thi. Các thao tác trên cây bao gồm:

- insert: Thêm lịch thi của một lớp. Mỗi lớp có một mã riêng, do đó chỉ lưu vào cây khi mã lớp chưa có trên cây.
- inOrder: In ra toàn bộ các mã lớp thi, ngăn cách nhau bởi dấu cách, bắt đầu từ nút gốc theo thứ tự: nút con trái => nút cha => nút con phải. Lưu ý, chỉ hiển thị một dấu cách giữa hai mã lớp, không hiển thị dấu cách ở vị trí đầu tiên, và kết thúc việc hiển thị với dấu !
- countStep: Nhập vào một mã lớp, hiển thị ra số lượng các nút trên cây đã được sử dụng để tìm kiếm mã lớp này khi sử dụng giải thuật tìm kiếm nhị phân. Lưu ý: đếm số nút cả trong trường hợp tìm thấy và không tìm thấy, không đếm nút NULL.
- Một ký tự đặc biệt # được sử dụng để kết thúc quá trình thao tác trên cây.

Hãy viết chương trình C cho phép đọc các thao tác trên cây và hiển thị kết quả ra màn hình.

Ví dụ:

INPUT1:

insert 20230020 TC-120

insert 20230090 TC-190

insert 20230010 TC-110

insert 20230015 TC-115

inOrder

#

OUTPUT1:

20230010 20230015 20230020 20230090!

INPUT2:

insert 20230020 TC-120

insert 20230090 TC-190

insert 20230010 TC-110

insert 20230015 TC-115

countStep 20230015

#

OUTPUT2:

3

INPUT3:

insert 20230020 TC-120

insert 20230090 TC-190

insert 20230010 TC-110

insert 20230015 TC-115

countStep 20230018


#

OUTPUT3:

3

- Hết -

Thay mặt nhóm chuyên môn



Nguyễn Kiêm Hiếu