Andy's Second Dictionary

Link submit:
https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=2003

Solution:

C++	https://ideone.com/6fGOKe
Java	https://ideone.com/1dfWe9
Python	https://ideone.com/In5TU0

Tóm tắt đề:

Andy bây giờ đã 9 tuổi và đang có rất nhiều tham vọng. Cậu ta muốn trở thành nhà biên tập từ điển lớn nhất thế giới. Bạn đã giúp cậu ta giải quyết một vấn đề bằng máy tính (bài UVA 10815 – Andy's First Dictionary) nên bây giờ cậu ta trở lại tìm bạn với một thử thách mới. Cậu ta có một chương trình sao chép tất cả các từ từ một văn bản và xuất chúng ta theo thứ tự alphabet. Tuy nhiên, chương trình này không xét dấu gạch nối ('-') và nó không thỏa mãn yêu cầu của Andy. Cậu ta muốn bạn viết một chương trình mới có thể sao chép được các từ và xử lý được dấu gạch nối.

Dữ liệu đầu vào là một văn bản không quá 500 từ có độ dài tùy ý. Kết thúc bằng kí hiệu kết thúc file EOF.

Một từ được định nghĩa là một dãy các kí tự, viết hoa thường tùy ý, chỉ chứa các kí tự alphabet. Các từ khác nhau về viết hoa hay viết thường thì vẫn được xem là 1, ví dụ "Apple", "apple" hay "APPLE" đều như nhau. Một từ có thể được nối giữa nhiều dòng, một từ là từ nối nếu như một phần tiền tố của nó nằm ở trên 1 dòng, sau đó là một dấu gạch nối ('-') và kết thúc dòng, phần còn lại của từ đó bắt đầu ở dòng tiếp theo. Phần còn lại đó cũng có thể là một phần nối. Một từ nối thì khi ghi lại sẽ bỏ qua dấu gạch nối để nối từ.

Input:

Dữ liệu đầu vào là một văn bản, kết thúc bằng EOF.

Output:

In ra các từ trong danh sách các theo alphabet từ danh sách các từ được rút trích từ đoạn văn bản đầu vào. Mỗi từ một dòng.

Ví dụ:

Adv- ent- ures in Dis- ney- land Two blondes were go- ing to Disney-land when they came to a fork in the road. The sign read: "Dis- neyland Left." So they went home.	a adventures blondes came disney-land disneyland fork going home in left read road sign so the they to two went were when

Hướng dẫn giải:

Ở bài này, nếu như ta có danh sách các từ thì việc xuất ra các từ theo thứ tự alphabet là tương đối dễ dàng. Tuy nhiên, vấn đề của bài này nằm ở việc xử lý dữ liệu đầu vào.

Để xử lý được dữ liệu đầu vào, ta sử dụng một chuỗi là text để chứa toàn bộ nội dung đoạn văn bản đã được chuẩn hóa và một chuỗi line để lưu chuỗi của dòng hiện tại vừa đọc.

Thực hiện việc đọc liên tục các dòng cho đến khi hết input. Với mỗi bước đọc, ta xét xem từ cuối cùng của nó có phải là từ nối hay không. Nếu cuối dòng là một dấu gạch nối thì có nghĩa rằng cuối từ đó vẫn còn một phần ở dòng sau, ta sẽ xóa dấu gạch nối đi (để khi nối với dòng tiếp theo sẽ được một từ hoàn chỉnh). Ngược lại thì có nghĩa là từ cuối cùng của dòng hiện tại và từ đầu tiên của dòng tiếp theo là 2 từ rời nhau, ta sẽ chèn thêm một dấu khoảng trắng (‘ ’) vào cuối dòng hiện tại (để khi nối 2 dòng thì sẽ có 1 khoảng trắng phân tách 2 từ).

Sau khi đã đọc hết dữ liệu đầu vào thì ta đang có một chuỗi text lớn chứa toàn bộ các từ trong văn bản, tuy nhiên các từ này chưa được phân tách và loại bỏ các kí tự như chấm (‘.’), phẩy (‘,’), hai chấm (‘:’), … và chưa chuẩn hóa (chuyển về in thường). Lúc này ta duyệt các kí tự trong text:

- Nếu nó là alphabet: chuyển nó thành kí tự thường.
- Nếu nó không là alphabet, cũng không là dấu gạch nối: chuyển nó thành khoảng trắng.

Sau khi thực hiện xong bước trên thì tất cả các từ trong văn bản đã được viết thường và phân tách nhau bằng các khoảng trắng. Nhiệm vụ bây giờ là chỉ việc tách các từ ra và thêm vào một cây nhị phân tìm kiếm để lưu lại các từ phân biệt.

- Với C++: có thể sử dụng stringstream để loại bỏ khoảng trắng sau này rất dễ dàng.
- Với Java/Python: có thể sử dụng hàm split trong chuỗi để phân tách các từ ra thành một mảng các từ.

Cuối cùng sẽ in toàn bộ chuỗi trong cây ra là đáp án đề bài.

Độ phức tạp: **O(N * logN)** với N là số lượng từ trong dữ liệu.