

Để tiện truy vết Covid19, một ứng dụng trên điện thoại đã lưu lại lịch sử di chuyển trong ngày của một người dùng dưới dạng một cấu trúc với các trường gồm

- Tên địa điểm (xâu ký tự độ dài tối đa 40 không chứa ký tự trắng)
- Giờ và Phút người dùng xuất hiện tại địa điểm trên (giờ trong ngày từ 0h00 tới 23h59)

Lịch sử di chuyển này sẽ được lưu trữ trên máy dưới dạng file văn bản - text với cấu trúc như sau

=====

```
Nha_rieng 0 0
C1_BKHN 8 15
Highland_coffee_BK 8 47
B1_BKHN 9 12
Nha_an_1-5 11 57
Hoi_truong_C2 15 25
```

=====

Mỗi dòng trong file là tên địa điểm (tối đa 40 ký tự), theo sau là giờ và phút. Các trường dữ liệu cách nhau bởi một ký tự space. Trình tự xuất hiện tại các địa điểm được lưu trữ theo thứ tự tăng dần về thời gian. Để lưu trữ cho thuận lợi và tối ưu về bộ nhớ, lịch sử di chuyển này sẽ được lưu trữ dùng danh sách liên kết đơn (hoặc đôi) khi nạp vào bộ nhớ trong.

Bạn hãy viết chương trình với các chức năng sau

1. Tạo giao diện menu để lựa chọn các tính năng (2 điểm)

=====

CHUONG TRINH TRUY VET COVID19

1. Nạp file log lịch sử di chuyển
2. Tìm kiếm lịch sử di chuyển theo địa điểm
3. Tìm kiếm lịch sử di chuyển theo thời gian
4. Kiểm tra truy vết mỗi ngày
5. Thoát

=====

2. Nạp dữ liệu log lịch sử di chuyển (3 điểm) : Đọc dữ liệu từ file văn bản, hiển thị ra màn hình theo 3 cột, căn thẳng lề:

Địa điểm (độ rộng 50)	Giờ (độ rộng 10)	Phút (độ rộng 10)
-----------------------	------------------	-------------------

sau đó lưu vào danh sách liên kết.

3. Tìm kiếm lịch sử di chuyển theo địa điểm (2 điểm) : Chương trình hỏi người dùng nhập vào một địa điểm và trả về thông tin về thời gian người dùng xuất hiện tại địa điểm đó nếu có. Để cho đơn giản thì tìm kiếm này sẽ là tìm kiếm chính xác tên (dùng hàm strcmp). Nếu người dùng có mặt tại địa điểm đó nhiều lần thì in ra thông tin tất cả các lần xuất hiện (theo định dạng địa điểm - giờ - phút), ngược lại thì in ra thông báo là bạn chưa tới địa điểm đó trong ngày.
4. Tìm kiếm thông tin theo thời gian (1.5 điểm) : Chương trình yêu cầu người dùng nhập vào giờ và phút là 2 số nguyên (phải thỏa mãn điều kiện $0 \leq \text{giờ} < 24$ và $0 \leq \text{phút} < 60$, nếu không thỏa mãn

yêu cầu nhập lại cho đến khi đạt yêu cầu). Chương trình in ra địa điểm của người dùng trong thời gian đó, hoặc thông báo “KHÔNG tìm thấy lịch sử di chuyển!”. VD. với file lịch sử minh họa ở trên thì người dùng nhập vào giờ là 8 phút là 15 sẽ thu được C1_BKHN

5. Kiểm tra truy vết mới nhất. Chương trình lần lượt tên địa điểm và thời gian (giờ và phút) trong ngày, nếu người dùng có xuất hiện tại địa điểm đó sau khoảng thời gian trên thì sẽ in ra thông báo “Bạn có khả năng bị lây Covid, cần phải khai báo y tế ngay lập tức!”, còn ngược lại thì thông báo “Lịch sử di chuyển của bạn OK”.
- VD. Với file lịch sử di chuyển ở trên, khi đầu vào là : Hoi trung C2 14 30 thì in ra là “Bạn có khả năng bị lây Covid, cần phải khai báo y tế ngay lập tức!”

Barem điểm

- + Khai báo cấu trúc NODE của danh sách liên kết đơn: 0.5 điểm
- + Tạo giao diện menu: 1 điểm
- + Chức năng thoát: 0.5 điểm
- + Đọc file lịch sử di chuyển vào danh sách liên kết đơn (3 điểm): Sinh viên in ra được nội dung file (có thể chưa nạp được vào danh sách) 2 điểm, nạp vào danh sách (1 điểm)
- + Chức năng tìm kiếm theo tên địa điểm: 2 điểm
- + Chức năng tìm kiếm theo thời gian: 1.5 điểm
- + Chức năng kiểm tra truy vết : 1.5 điểm

Khi Thoát nên kiểm tra việc có giải phóng bộ nhớ hay không (thiếu -0.5 điểm)

Có thể tăng độ dễ bằng cách nâng barem câu 2 lên 3.5 điểm và câu cuối giảm còn 1 điểm.

Remark:

- Sinh viên chỉ cần xong menu và đọc file in ra màn hình (ko cần nạp vào danh sách) là đạt 4 điểm.
- Hoàn thành tìm kiếm theo địa điểm đầu tiên: Đạt 7 điểm
- Nên bỏ các từ hàm trong đề để không mang tính chất định hướng bắt buộc sinh viên, sinh viên có thể không dùng hàm cũng không sao.
- Đề này chỉ có điểm hơi khó duy nhất ở câu truy vết là sinh viên sau khi so khớp địa điểm thì cần so thời gian (để xác định xuất hiện sau -- (so sánh giờ lớn hơn hoặc giờ bằng phút lớn hơn). Nhìn chung đề này cũng không có gì đòi hỏi sinh viên phải quá thông minh, giải lập trình để đạt 10