

Final (A)

Thời gian làm bài: 90 phút

1 Quy định nộp bài

- Sinh viên phải thực hiện bài kiểm tra tại phòng máy.
- Sinh viên thực hiện các hàm với Prototype có sẵn, trong một file `MSSV.cpp` duy nhất. (Chỉ cần viết hàm, không cần viết hàm `main()`). Nộp `MSSV.cpp`

Lưu ý: Thay cụm MSSV thành mã số sinh viên của bạn.

2 Quy định chấm bài

- Chấm bài bằng trình biên dịch g++ (thư viện chuẩn)
- Đối với những hàm sinh viên không thực hiện được → giữ nguyên trạng thái ban đầu.
- KHÔNG chấm ý tưởng, chỉ có đúng hoặc sai.
- Những trường hợp sau đây sẽ bị 0 điểm bài thi:
 - Nộp sai quy định.
 - BÀI LÀM GIỐNG NHAU.
 - KHÔNG BIÊN DỊCH ĐƯỢC.
 - LẬP VÔ TẬN.

3 Xử lý dữ liệu đầu vào

Cho file `g1.v2.jl` chứa dữ liệu thời gian các chuyến bay của hãng hàng không quốc gia.

```
{from,to,[name of airliner,the number of business seats,the number of economy seats,hours,minutes]}
{"Myanmar,South Sudan": ["Comac ARJ-900/C909-200 90", "11 business, 84 economy", "9 hours, 25 minutes"]}
{"Faeroe Islands,Ethiopia": ["Comac C919", "92 business, 98 economy", "9 hours, 16 minutes"]}
{"Italy,Greece": ["Bombardier C-110 100", "8 business, 117 economy", "1 hour, 36 minutes"]}
{"Netherlands,Tunisia": ["Antonov 24", "21 business, 31 economy", "3 hours, 4 minutes"]}
...
```

Đọc file và chuyển dữ liệu về dạng cấu trúc theo yêu cầu đề bài.

(Sinh viên sang trang kế tiếp để xem đề bài.)

4 Quy ước

Đọc ghi file và hàm todo tương tự như Test 1 & 2. Các câu cách nhau =====.

Sai hàm insert ở (câu đầu của mỗi phần) cây/đồ thị => KHÔNG CHẤM những câu còn lại

Phúc khảo cuối kì: -3 điểm cho MỖI lỗi mã nguồn sai mà giáo viên phải sửa

Quy ước 1 số thông số của cây: $h_{leaf} = 1$, $level_{root} = 0$

data.tx có cấu trúc 2 phần, dòng đầu tiên là dữ liệu dùng để thực thi phần *Đồ thị*, những dòng còn lại là dữ liệu cho *Cây AVL*:

```
Vietnam,Myanmar //2 nước country1, country2
a a a a a //AVLNode
e f g g h i //AVLNode
a a b b c //AVLNode
b a a a a c //AVLNode
x y z //AVLNode
a b c d e //AVLNode
```

Nội dung

1 Đồ thị

Xét đồ thị $G(V, E)$ với tập đỉnh V là tập hợp các quốc gia(nước), E là đường bay giữa 2 quốc gia tương ứng với các đỉnh trong V .

Câu 1 (1 điểm) Từ file dữ liệu *g1.v2.jl*, với các struct được định nghĩa như trong file *func.cpp* và *27127000.cpp*, hãy tìm các chuyến bay có thời gian bay ít hơn 10 tiếng, và xây dựng 1 đồ thị G vô hướng biểu diễn các chuyến bay vừa tìm được. In ra 10 nước bất kì có đường bay trực tiếp đến nước **country1** được cho trước.

Output mẫu:

Vietnam: Nepal, Reunion, Lebanon, Saudi Arabia, Brunei, Hong Kong, Cyprus, UAE, Afghanistan, Laos,

Câu 2 (2 điểm) Viết hàm đếm số nước có bậc chẵn và in ra nước có bậc chẵn cao nhất. Nếu có nhiều nước có cùng bậc chẵn cao nhất, in ra tất cả các nước này.

Output mẫu:

97 even vertice, the highest: Mauritania

Câu 3 (3 điểm) Một khu vực phát triển là 1 tập hợp 4 nước, sao cho luôn tồn tại chuyến bay trực tiếp giữa hai nước bất kì trong 4 nước này. Cho 2 nước **country1** và **country2**, hãy tìm tất cả các khu vực phát triển có cả **country1** và **country2**. Yêu cầu xuất: in ra tổng số khu vực phát triển và 3 khu vực phát triển bất kì trong tổng số các khu vực nói trên.

Output mẫu:

```
212 quad
Myanmar, Qatar, Saudi Arabia, Vietnam
Myanmar, Qatar, Taiwan, Vietnam
Macao, Myanmar, Pakistan, Vietnam
```

2 Cây AVL

Cho cấu trúc Node của một cây AVL như sau:

```
struct Node{  
    vector<char> A;  
    Node* left; Node* right;  
};
```

Câu 4 (2 điểm) Hãy xây dựng một cây AVL bằng cách thêm lần lượt từng Node vào cây AVL rỗng ban đầu, với khóa của mỗi Node là mảng các kí tự `vector <char>` được đọc theo dòng từ trên xuống dưới trong file `data.txt`. Xét 2 khóa A và B , định nghĩa của $A > B$ khi thỏa các ưu tiên sau đây:

- Ưu tiên 1: xét số lượng kí tự là nguyên âm (A E O U I).
- Ưu tiên 2: Nếu có cùng số lượng nguyên âm, xét số lần xuất hiện của kí tự xuất hiện nhiều nhất.
- Nếu 2 khóa có cùng cả ưu tiên 1 và 2 thì không thêm khóa mới vào cây.

Duyệt LNR cho cây AVL vừa tạo.

Output mẫu:

```
{x, y, z} {a, b, c, d, e} {e, f, g, g, h, i} {b, a, a, a, a, c} {a, a, a, a, a, a}
```

Câu 5 (2 điểm) Xóa tất cả các Node có số lượng kí tự là nguyên âm bằng 2. Duyệt LNR cho cây vừa xóa. Khi xóa, chọn Node thay thế là Node con lớn nhất bên trái.

Output mẫu:

```
{x, y, z} {b, a, a, a, a, c} {a, a, a, a, a, a}
```