

# Attendance Test 02 (E)

Thời gian làm bài: 60 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 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.)*

## Nội dung

Đọc ghi file và hàm todo tương tự như Test 1. Các câu cách nhau =====. Barem điểm = [0,4,2,4]  
File data.txt (nhận vào biến data\_filename) như sau:

```
data.txt:
0.1.4... //level_list
t        //ch
```

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

**Câu 1** Từ file dữ liệu *g1.v2.jl*, với các struct được định nghĩa như trong file *func.cpp*. Hãy tìm các chuyến bay có **src** và **des** có cùng kí tự bắt đầu vd: ("Armenia,Austria"), xây dựng một cây BST bằng cách thêm lần lượt các chuyến bay này theo thứ tự từ trên xuống dưới trong file *g1.v2.jl*, với thứ tự ưu tiên khóa như sau:

- Ưu tiên 1: Theo khóa **src**
- Ưu tiên 2: Nếu có cùng khóa **src**, theo tổng thời lượng bay.
- Ưu tiên 3: Nếu có cùng thời gian bay, theo khóa **des**.

**Câu 2** Viết hàm in ra các node từ cây được tạo từ **Câu 1**, ở các mức *level\_list* theo thứ tự từ phải sang trái.

VD: *level\_list* = "0.1.4" → In ra các node ở mức 0, mức 1, mức 4

Yêu cầu prototype: void print\_levellist(Node\* root, string level\_list)

Output mẫu:

```
Vietnam,Sudan,Embraer 175,6,80,10,36
Vietnam,Tanzania,Antonov 24,9,43,10,41
...
```

**Câu 3** Xóa các nước có **src** bắt đầu bằng các ký tự đứng trước **char ch** trong bảng chữ cái, khỏi cây được tạo từ **Câu 1**, với **char ch** đọc từ file data.txt. Sau khi xóa, gọi lại hàm in cây (câu 2).

**Câu 4** Từ cây ở câu 3, hãy cho biết một đường đi từ Node gốc đến Node lá sao cho tổng số ghế từ các Node này là nhiều nhất.

Yêu cầu prototype: void ex4(Node\* root)

Output mẫu:

```
Vietnam,Sudan -> Sudan,Tanzania -> ...
Total seats = 12345
```