

Attendance Test 01 (B)

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,hours,minutes]
{"Myanmar,South Sudan": "9 hours, 25 minutes"}
{"Faeroe Islands,Ethiopia": "9 hours, 16 minutes"}
{"Italy,Greece": "1 hour, 36 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 file `data.txt`, trong đó $m, n, p, q, matrix$ là dữ liệu cho câu 1 + 2, `positions_list` câu 3, $alpha, beta$ (có thể có hoặc không) + `line(s)` câu 4.

```
m n
matrix[0][0] matrix[0][1] ... matrix[0][n-1]
...
matrix[m-1][0] matrix[m-1][1] ... matrix[m-1][n-1]
p q

positions_list

alpha beta
line1
line2
...
```

Sinh viên hoàn thiện hàm `todo`. Hàm này in ra kết quả tất cả các câu trong đề, kết quả các câu cách nhau bởi `=====`. Khi được yêu cầu, sinh viên bắt buộc dùng các hàm có sẵn trong `func.cpp`

```
void todo(string flightdata_filename, string
          data_filename)
```

Một ví dụ output in ra màn hình:

```
a b d d e f
f e c c b a
a b d d e f
=====
d d
c c

c c
d d

=====
Myanmar,South Sudan,9,25
Faeroe Islands,Ethiopia,9,16
Estonia,Andorra,3,33
=====
Vietnam,Congo
Congo,Falkland Islands
...
```

Câu 1 In ra ma trận kí tự $m \times n$ đọc từ `data.txt`.

Yêu cầu: dùng hàm `printmatrix2d`

Câu 2 Cho ma trận kí tự có chiều $m \times n$. Tìm tất cả các ma trận con vuông $p \times p$, sao cho với mỗi kí tự viết thường/hoa trên đường chéo chính, tồn tại trên đường chéo phụ 1 kí tự viết hoa/thường tương ứng của nó và ngược lại. Viết hàm trả về ma trận 3 chiều là kết quả của tất cả các mảng 2 chiều con thoả mãn yêu cầu trên.

VD: `p = 3`

```
matrix2d = {{a, e, A, e, a},{f, b, e, F, h},{B, E, B, E, b},{
=> matrix3d = { {{a, e, A},{f, b, e},{B, E, B}}, {{e, A, e},{b, e, F},{E, B, E}}, }
```

Yêu cầu: dùng hàm `printmatrix3d` để in ma trận kết quả

Câu 3 Dựng 1 danh sách liên kết từ dữ liệu file `g1.jl`, với các struct được định nghĩa như trong file `func.cpp`. Hãy duyệt danh sách liên kết. Viết hàm in ra các node ở vị trí `position_list`. VD: `positions_list = "0.1.10"` → in node đầu tiên, node 1, node 10.

Yêu cầu prototype hàm: `void printlistpos(List list, string positions_list)`

Câu 4 Cho danh sách các nước mà bạn Khái bị cấm nhập cảnh. Hãy xóa tất cả các đường bay đến nước này. Đếm số đường bay đã xóa và tổng thời gian của chúng.

Yêu cầu: dùng hàm `printlistpos(list, position_list)` in lại danh sách sau khi xóa. Sau đó in tổng số đường bay đã xóa và tổng thời gian của chúng

```
Myanmar,South Sudan,9,25
Faeroe Islands,South Sudan,9,16
Estonia,South Sudan,3,33
...
delete 35 lines, 65 hours 14 minutes
```