

# Attendance Test 01 (A)

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)
- 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.
  - LẤP VÔ TẬN.
  - Sử dụng **vector** hoặc **string** cho phần bài làm con trỏ.

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

## Nội dung

**Câu 1 :** Cho ma trận A có kiểu char có kích thước  $m \times n$ . Thực hiện các yêu cầu sau:

- Viết hàm in ra một ma trận 2 chiều. (1 điểm)  
Prototype: void print2D(char\*\* A, int m, int n)
- Viết hàm in ra một ma trận 3 chiều. (xem ví dụ bên dưới) (1 điểm)  
Prototype: void print3D(char\*\*\* B, int k, int p, int q)
- Viết hàm tìm và trả về tập hợp các ma trận con của A với kích thước  $p \times q$  sao cho tổng số lượng phần tử viết hoa và số lượng phần tử viết thường trong ma trận này bằng nhau. (Tham số k là giá trị trả về thể hiện số ma trận thỏa yêu cầu) (2 điểm)  
Prototype: char\*\*\* findBalanceMatrices(char\*\* A, int m, int n, int &k, int p, int q)

Ví dụ: A = {{A, b, c}, {d, E, f}, {g, h, i}} có m = 3, n = 3

→ findBalanceMatrices(A, 3, 3, k, 2, 2) = {{A, b}, {d, E}} với k = 1

**Câu 2 :** File <Players.txt> chứa danh sách các cầu thủ, có cấu trúc mỗi dòng: |Vị trí/Tên/Năm sinh/Quốc tịch/Câu lạc bộ|. Thực hiện các yêu cầu sau:

- Khai báo cấu trúc Player phù hợp với các thông tin cầu thủ nêu trên. Tạo một danh sách liên kết ĐÔI từ file đã cho. (1 điểm)  
Prototype: PList createPlayerList()
- Viết hàm in danh sách liên kết đôi theo chiều ngược và xuôi.
  - Đối với chiều xuôi: Tên cầu thủ - Vị trí - Năm sinh (1 điểm)  
Prototype: void printPlayerList1(PList Players)
  - Đối với chiều ngược: Tên cầu thủ - Quốc tịch - Câu lạc bộ (1 điểm)  
Prototype: void printPlayerList2(PList Players)
- Viết hàm tạo một danh sách liên kết ĐÔI mới chứa tất cả các cầu thủ ở các quốc gia cho trước. Đồng thời, xóa các cầu thủ này khỏi danh sách liên kết ban đầu.  
Prototype: PList findPosition(PList &Players, vector <string> countries)