

THI CUỐI KỲ

Thời gian làm bài: 75 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

- Câu lệnh dùng để chấm

```
>> g++ mssv.cpp --std=c++11 -o mssv(.exe)
>> mssv(.exe)
```

- Đối với những hàm sinh viên không thực hiện được → thân hàm để rỗng.
- Sinh viên KHÔNG thêm/xóa/sửa **prototype** hàm và **struct** mà đề bài cho
- 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.

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

Nội dung

Sinh viên chỉ được dùng 3 thư viện trong bài làm của mình, gồm `iostream`, `string`, `cstring`

Câu 1 (5) : Một chữ gồm 2 phần:

- phần phụ âm: từ đầu cho đến trước khi gặp nguyên âm đầu tiên, trong đó nguyên âm gồm A E I O U Y a e i o u y
- phần vần: từ nguyên âm đầu tiên đến hết chữ

Chơi chữ đảo vần, là cách đảo phần vần của 2 chữ cho nhau, ví dụ: `ca doi`, `coi da`. Cho chuỗi gồm 2 chữ (cách nhau bởi khoảng trắng), hãy viết hàm đảo vần cho 2 chữ này.

Yêu cầu prototype:

```
string makeSpoonermism(string twowords);
```

Ví dụ: Với `string twowords = "Ca doi"`

→ `makeSpoonermism(twowords) = "Coi da"`

Câu 2 (5) : Ở dòng nhạc rap, một trong những phương pháp sáng tác cơ bản nhất là gieo vần. Nếu chữ cuối cùng của 2 câu liên tiếp có vần giống nhau hoàn toàn thì gọi là **gieo vần đơn hoàn hảo**. Nếu 2 chữ cuối cùng của 2 câu liên tiếp có vần giống nhau hoàn toàn thì gọi là **gieo vần đôi hoàn hảo**.

VD: 2 câu "Nếu mà mệt quá giữa thành phố sống chồng lên nhau. Cùng lắm thì mình về quê mình nuôi cá và trồng thêm **rau**." gieo vần đơn (nhau - rau)

VD: 2 câu "Mình sướng hơn những người **giàu nhĩ**. Vâng trán mình chưa hề **nhàu nhĩ**." gieo vần đôi (au i)

Cho 1 đoạn lời rap được lưu bằng mảng các chuỗi, mỗi câu là 1 chuỗi. Hãy viết hàm cho biết:

- Số lượng câu liên tiếp dài nhất được gieo vần đơn hoàn hảo (3 điểm)

Yêu cầu prototype:

```
int findLongestSingleRhyme(string sentences[], int n); //n = so luong cau, 0 <= n <= 100
```

- Số lượng câu liên tiếp dài nhất được gieo vần đôi hoàn hảo (2 điểm)

Yêu cầu prototype:

```
int findLongestDoubleRhyme(string sentences[], int n); //n = so luong cau, 0 <= n <= 100
```

Ví dụ: Với `n = 6`; `string sentences[] = {"Minh suong hon nhung nguoi giao nhi",`

`"Vang tran minh chua he nhau nhi", "Du chenh venh nhu la cau khi", "Doi song gio minh la tau thuy",`

`"Vi mot ngay con song la mot ngay dam say", "Ngay dep troi nhat la ngay minh con nam tay"}`

→ `findLongestSingleRhyme(sentences, n) = 3`; (câu 0, 1, 2)

→ `findLongestDoubleRhyme(sentences) = 3`; (câu 0, 1, 2)

(Trong trường hợp sinh viên muốn dùng mảng ký tự)

Cách chuyển đổi từ mảng chuỗi kiểu `std::string` sang mảng chuỗi kiểu mảng ký tự (`char str[], cstring`):

```
// Mang cac chuoi kieu std::string
string ls[10] = {"This", "is", "CLC5", "class"};
int n = 4;

// Chuyen doi mang cac chuoi std::string sang mang cac chuoi cstring
char lc[10][121]; // Mang co toi da 10 chuoi, moi chuoi co toi da 120 ky tu
for (int i = 0; i < n; i++)
    strcpy(lc[i], ls[i].c_str());
```

Câu 3 (5) : Cho định nghĩa struct như sau:

```
struct Point {
    int x; int y; //x and y coordinates
};
struct Polygon {
    Point A; Point B; Point C; Point D; //4 vertices of the Polygon
};
```

Hãy viết hàm kiểm tra xem đa giác 4 đỉnh **Polygon P** có phải hình chữ nhật hay không?

Gợi ý:

- 4 điểm phân biệt được xem là hình chữ nhật khi với mọi bộ 3 điểm bất kỳ từ 4 điểm đã cho luôn tạo thành 1 tam giác vuông.
- Định lý Pytago đảo: Một tam giác là tam giác vuông khi tồn tại một cạnh có bình phương bằng tổng bình phương 2 cạnh còn lại.
- Cho 2 điểm $(x_1, y_1), (x_2, y_2) \rightarrow$ khoảng cách 2 điểm: $D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Yêu cầu prototype:

```
bool isRectangle(Polygon P);
```

Ví dụ: Polygon P = {{4, 2}, {7, 5}, {2, 4}, {5, 7}} \rightarrow isRectangle(P) = true