

KIỂM TRA GIỮA KỲ

Quy định

1 Quy định nộp bài

- Sinh viên thực hiện cài đặt theo đúng tên hàm và struct được khai báo trong tập tin function.h đính kèm.
- Sinh viên KHÔNG ĐƯỢC PHÉP THAY ĐỔI định nghĩa struct trong tập tin function.h đính kèm.
- Đặt các tập tin function.cpp và function.h vào thư mục MSSV sau đó nén thành MSSV.rar(.zip).
- Sinh viên nộp bài dưới dạng MSSV.rar(.zip).

2 Quy định chấm bài

- 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.

Nội dung

1 Mô tả dữ liệu

Dữ liệu được dùng trong bài tập thực hành là dữ liệu về điểm thi của tỉnh Hà Tĩnh (thông tin thật của thí sinh đã được thay đổi).

Các tập tin được cung cấp có một phần nội dung như sau:

```
Số Báo Danh, Họ và Tên, Toán, Ngữ Văn, Vật Lý, Hóa Học, Sinh Học,  
HT1200001,,9.0,7.75,,,,6.0,8.5,9.0,,,9.4,N1,Ha Tĩnh  
HT1200002,,5.4,7.5,,,,2.75,5.5,7.5,,,2.0,N1,Ha Tĩnh  
HT1200003,,3.6,6.25,,,,5.0,5.0,7.0,,,3.2,N1,Ha Tĩnh  
HT1200004,,4.4,5.0,,,,1.75,3.0,5.0,,,2.6,N1,Ha Tĩnh  
HT1200005,,7.4,6.5,6.75,5.0,4.75,,,,,7.2,N1,Ha Tĩnh  
HT1200006,,5.2,7.5,,,,6.5,7.25,8.75,,,3.8,N1,Ha Tĩnh  
HT1200007,,8.6,6.5,5.25,3.75,3.5,,,,,2.8,N1,Ha Tĩnh
```

Trong đó:

- Dòng đầu tiên thể hiện tên các trường thông tin có trong tập tin.
- Những dòng tiếp theo thể hiện thông tin thí sinh, mỗi trường thông tin cách nhau bởi 1 dấu phẩy (,).
- Trường thông tin về Họ và tên thí sinh đã được bỏ.
- Những trường thông tin về điểm được bỏ trống nghĩa là thí sinh không thi môn đó. Để cho đơn giản, khi đọc thông tin thí sinh gặp những trường thông tin về điểm được bỏ trống sinh viên sẽ lưu trữ điểm đó trong struct mặc định là 0. Đối với điểm KHTN và KHXH sinh viên thực hiện tính như sau:
 - $KHTN = Lý + Hóa + Sinh$
 - $KHXH = Sử + Địa + Giáo dục công dân$

2 Yêu cầu

Cho struct `Examinee` được định nghĩa như sau:

```
#include <iostream>  
  
using namespace std;  
  
struct Examinee  
{  
    string ID;  
    float math, literature, physic, chemistry, biology, history, geography, civic_education, natural_science,  
          social_science, foreign_language;  
};
```

Câu 1: (4 điểm)

Viết hàm tìm kiếm thí sinh có điểm **Văn** = **v** trong danh sách thí sinh được sắp xếp giảm dần theo điểm Văn:

- `string FindExamineeByLiterature(vector<Examinee> sorted_list, float v);`
- Input: - `sorted_list` - danh sách thí sinh được sắp xếp giảm dần theo điểm Văn đọc từ tập tin `descending_Literature.txt`.
- `v` - điểm Văn muốn tìm kiếm.
- Output: Kiểu dữ liệu `string` lưu trữ định danh (ID) của thí sinh đầu tiên tìm thấy có điểm **Văn** = **v**. Trả về "Not Found" nếu không tìm thấy.

Lưu ý: Có nhiều cách giải khác nhau cho bài này, chỉ có cách giải tối ưu mới được điểm tối đa.

Câu 2: (6 điểm)

1. (3 điểm) Viết hàm trả về top **k** ($1 \leq k \leq 200$) thí sinh có điểm **Văn** cao nhất:

- `vector<Examinee> GetTopHighestLiterature(vector<Examinee> list_examinee, int k);`
- Input: `list_examinee` - danh sách tất cả thí sinh đọc từ tập tin `data.txt`.
- Output: Kiểu dữ liệu `vector<Examinee>` lưu trữ top **k** thí sinh có điểm **Văn** cao nhất.

Lưu ý: Có nhiều cách giải khác nhau cho bài này, chỉ có cách giải tối ưu mới được điểm tối đa.

2. (3 điểm) Gộp 2 danh sách thí sinh được sắp xếp giảm dần theo điểm Văn sao cho danh sách vẫn giảm

- `vector<Examinee> MergeTwoList(vector<Examinee> list1, vector<Examinee> list2);`
- Input: - `list1` - danh sách các thí sinh được đọc từ tập tin `descending_Literature_1.txt` chứa các thí sinh được sắp xếp giảm dần theo điểm Văn.
- `list2` - danh sách các thí sinh được đọc từ tập tin `descending_Literature_2.txt` chứa các thí sinh được sắp xếp giảm dần theo điểm Văn.
- Output: Kiểu dữ liệu `vector<Examinee>` lưu trữ danh sách được gộp bởi 2 danh sách đầu vào và vẫn bảo toàn tính chất giảm dần theo điểm Văn.

Lưu ý: Có nhiều cách giải khác nhau cho bài này, chỉ có cách giải tối ưu mới được điểm tối đa.