**BÀI TẬP TUẦN 1**

*Lê Hồng Hải*

**Bài 1**: Viết chương trình quản lý điểm thi đại học năm 2016 của trường bạn yêu cầu những thông tin sau.

- Thông tin thí sinh: Họ tên, số báo danh, địa chỉ, giới tính.

- Mỗi thí sinh sẽ có điểm thi của 3 môn: Toán, Lý, Hóa.

Biết rằng, chương trình có khả năng thực hiện những chức năng sau:

0. Thêm, chỉnh sửa, xóa thông tin của 1 sinh viên.

1. Nhập dữ liệu sinh viên theo 2 cách:

- Nhập dữ liệu cho n sinh viên từ bàn phìm, với n > 0.

- Nhập dữ liệu cho n sinh viên từ file với đường dẫn nhập từ bàn phím.

2. Tìm kiếm sinh viên theo:

- Tổng số điểm.

- Số báo danh

- Họ và tên

- Điểm một trong 3 môn.

3. Và sắp xếp danh sách sinh viên tìm kiếm được theo một trong các tiêu chí sau:

- SBD (mặc định)

- Thứ tự tăng dần của bảng chữ cái Anphabe.

- Thứ tự tăng dần của tổng điểm.

4. Trích xuất thông tin tìm kiếm được ra file.

5. (Bonus) Thống kê, tổng hợp kết quả thí sinh theo các tiêu chí sau:

- Tổng số thí sinh.

- Tổng số thí sinh có tổng điểm: nhỏ hơn 15, từ 15 đến 20, từ 20 đến 25 và 25 trở lên.

**I. Cấu trúc chương trình “QLDT DH 2016”**

1. Thêm, Sửa xóa

1.1. Thêm

1.2. Sửa

1.3. Xóa

1.4. Thoát

2. Tìm kiếm

2.1. Theo tổng điểm

2.2. Theo số báo danh

2.3. Theo tên

2.4. Theo điểm 1 trong 3 môn

2.4.1. Theo điểm toán

2.4.2. Theo điểm lý

2.4.3. Theo điểm hóa

2.4.4. Thoát

2.5. Thoát

3. Sắp xếp

3.1. Theo số báo danh

3.2. Theo Anphabe

3.3. Theo tổng điểm

3.4. Thoát

4. Xuất file

Xuất ra file “output.txt” (vị trí lưu file cùng với vị trí của file chạy .exe)

5. Thống kê

Thống kê tất cả các sinh viên, thống kê theo các tiêu chí điểm.

6. Thoát

**II. Chương trình**

1. **Lớp SinhVien**

- Các thuộc tính: ***HoTen; SBD; DiaChi; GioiTinh; Toan; Ly; Hoa;***

+ Các thuộc tính lưu các giá trị người dùng nhập

+ Các thuộc tính này như là biến toàn cục

- ***Các phương thức:***

+ Các phương thức ***SET***: Set giá trị mỗi thuộc tính tương ứng với tham số truyền vào khi gọi hàm

+ Các phương thức ***GET***: Trả về các giá trị thuộc tính khi có gọi hàm (phương thức)

+ ***Nhap();*** Nhập thông tin sinh viên

+ ***Xuat();*** Xuất thông tin sinh viên

+ ***TongDiem();*** Tổng: toán + lý + hóa

1. **Các Hàm kiểm tra**

***- bool CheckSV(string sbd, int &vitri);***

+ Hai tham số truyền vào

+ Dùng vòng lặp FOR để duyệt mảng kết hợp IF kiểm tra xem số báo danh truyền vào đã tồn tại chưa

+ Trường hợp SBD này có tồn tại trong mảng thì gán &vitri bằng với phần tử ở lần duyệt mảng đó, thao tác này quan trọng vì sau này gọi hàm Sua(); thì ta căn cứ vào vị trí của SBD có tồn tại đó để sửa thông tin sinh viên.

+ Nếu SBD có tồn tại trong mảng thì hàm này trả về TRUE ngược lại trả về FALSE

***- bool CheckSbd(string sobaodanh);***

+ Hàm Kiểm tra chuỗi truyền vào (số báo danh) có tồn tại trong mảng hay không

+ Dùng vòng lặp FOR duyệt các phần tử của mảng, Mỗi lần duyệt dùng IF kiểm tra xem chuỗi truyền vào có trùng với số báo danh trong mảng hay không

+ Số báo danh trong mảng là kết quả khi gọi phương thức ***MangSV[i].getSBD()***

+ Nếu có tồn tại chuỗi này trong mảng thì hàm ***CheckSbd(string sobaodanh)*** trả về ***TRUE,*** ngược lại trả về ***FALSE***

***\* Các hàm sau tương tự***

***- bool CheckHoTen(string hoten);***

***- bool CheckTongDiem(double tong);***

***- bool CheckDiemHoa(double hoa);***

***- bool CheckDiemToan(double toan);***

***- bool CheckDiemLy(double ly);***

1. **Các hàm Tìm kiếm**

***- void TimKiemTheoSbd(string sbd);***

+ Hàm có tham số truyền vào là một chuỗi

+ Thực hiện gọi hàm ***CheckSbd(sbd)*** với tham số tryền vào chính là tham số truyền vào của hàm ***TimKiemTheoSbd()*** kết quả trả về lưu vào biến kiểu bool

+ Nếu kết quả (biến bool vừa lưu vào) thu được là **TRUE** Dùng IF-FOR lồng nhau để kiểm tra xem số báo danh truyền vao có tồn tại trong mảng hay không, thao tác này cũng tương tự như hàm kiểm tra bên trên

+ Với mỗi lần lặp, kiểm tra bằng IF nếu thỏa mãn thì gọi hàm **Xuat()**

+ Nếu kết quả (biến bool vừa lưu vào) thu được là **FALSE** thì thông báo số báo danh không tồn tại

\* ***Các hàm sau tương tự***

***- void TimKiemTheoTongDiem(double tong);***

***- void TimKiemTheoTen(string hoten);***

***- void TimKiemTheoToan(double diemToan);***

***- void TimKiemTheoLy(double diemLy);***

***- void TimKiemTheoHoa(double diemHoa);***

1. **Các hàm sắp xếp**

***- void SapXepTheoTongDiem(SinhVien mangsv[], int n);***

+ Tham số truyền vào là đối tượng kiểu mảng của lớp SinhVien đã định nghĩa và biến kiểu nguyên hiểu như đây chính là kích thước mảng

+ Dùng giải thuật sắp xếp bằng so sánh trực tiếp: So sánh 2 phần tử kề nhau, nếu phần tử đứng trước (j) lớn hơn phần tử đứng sau (j+1) thì:

\* gán phần tử đứng trước vào biến tạm

\* gán phần tử đứng sau bằng phần tử đứng trước

\* gán phần tử đứng sau bằng biến tạm

+ Mảng sau khi được sắp xếp sẽ theo thứ tự tăng dần

+ Dùng vòng lặp FOR để Xuất mảng

\* ***Các hàm sau tương tự***

***- void SapXepTheoSbd(SinhVien mangsv[], int n);***

***- void SapXepTheoAnphabe(SinhVien mangsv[], int n);***

1. **Các hàm khác**

***- void Them(int n);***

+ Gọi hàm Nhap();

+ Sau mỗi lần nhập tăng chỉ số mảng lên 1

***- void Sua(string sbd);***

+ Sửa theo số báo danh người dùng nhập

+ Kiểm tra xem số báo danh có tồn tại hay không

+ Nếu tồn tại gọi hàm Nhap() để sửa

+ Nếu không tồn tại thì thông báo đến người dùng

***- void Xoa(string sbd);***

+ Xóa theo số báo danh người dùng nhập

+ Dùng kỹ thuật dồn mảng để xóa

+ Sau mỗi lần xóa giảm chỉ số mảng đi 1

***- void ThongKeAll();***

+ Thống kê toàn bộ

+ Thống kê theo điểm

+ Thống kê số lượng sinh viên đạt mức điểm nào đó: Dùng biến đếm kết hợp IF và FOR để Xuất mảng

***- void GhiFile();***

+ Xuất danh sánh sinh viên đang có trong mảng ra file: ” output.txt” (lưu tại cùng thư mục với file chạy exe)