**Thiết kế kiểu Complex (Số phức) và xây dựng các thao tác khởi tạo giá trị, nhập, xuất dữ liệu, các phép toán cộng, trừ, nhân, chia, . . . .**

#include <iostream>

using namespace std;

struct PhanSo {

int tuSo;

int mauSo;

};

int ucln(int a, int b);

void rutGonPhanSo(PhanSo &p);

void nhapPhanSo(PhanSo &p);

void inPhanSo(PhanSo p);

PhanSo congPhanSo(PhanSo a, PhanSo b);

PhanSo truPhanSo(PhanSo a, PhanSo b);

PhanSo nhanPhanSo(PhanSo a, PhanSo b);

PhanSo chiaPhanSo(PhanSo a, PhanSo b);

int main() {

PhanSo a, b, c;

int choice;

while (true) {

system("cls");

cout << "Chon phep tinh:" << endl;

cout << "1. Cong 2 phan so" << endl;

cout << "2. Tru 2 phan so" << endl;

cout << "3. Nhan 2 phan so" << endl;

cout << "4. Chia 2 phan so" << endl;

cout << "0. Thoat" << endl;

cout << "Nhap lua chon cua ban: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Nhap phan so thu nhat:" << endl;

nhapPhanSo(a);

cout << "Nhap phan so thu hai:" << endl;

nhapPhanSo(b);

c = congPhanSo(a, b);

inPhanSo(a);

cout << " + ";

inPhanSo(b);

cout << " = ";

inPhanSo(c);

cout << endl;

system("pause");

break;

case 2:

cout << "Nhap phan so thu nhat:" << endl;

nhapPhanSo(a);

cout << "Nhap phan so thu hai:" << endl;

nhapPhanSo(b);

c = truPhanSo(a, b);

inPhanSo(a);

cout << " - ";

inPhanSo(b);

cout << " = ";

inPhanSo(c);

cout << endl;

system("pause");

break;

case 3:

cout << "Nhap phan so thu nhat:" << endl;

nhapPhanSo(a);

cout << "Nhap phan so thu hai:" << endl;

nhapPhanSo(b);

c = nhanPhanSo(a, b);

inPhanSo(a);

cout << " \* ";

inPhanSo(b);

cout << " = ";

inPhanSo(c);

cout << endl;

system("pause");

break;

case 4:

cout << "Nhap phan so thu nhat:" << endl;

nhapPhanSo(a);

cout << "Nhap phan so thu hai:" << endl;

nhapPhanSo(b);

while (b.tuSo == 0) {

cout << "Nhap lai phan so thu hai (mau so khac 0):" << endl;

nhapPhanSo(b);

}

c = chiaPhanSo(a, b);

inPhanSo(a);

cout << " / ";

inPhanSo(b);

cout << " = ";

inPhanSo(c);

cout << endl;

system("pause");

break;

case 0:

cout << "Cam on da su dung chuong trinh" << endl;

return 0;

default:

cout << "Lua chon khong hop le. Vui long nhap lai." << endl;

}

}

return 0;

}

int ucln(int a, int b)

{

a = abs(a);

b = abs(b);

if (b == 0) return a;

return ucln(b, a % b);

}

void rutGonPhanSo(PhanSo &p)

{

int gcd = ucln(p.tuSo, p.mauSo);

p.tuSo /= gcd;

p.mauSo /= gcd;

}

void nhapPhanSo(PhanSo &p)

{

cout << "Nhap tu so: ";

cin >> p.tuSo;

cout << "Nhap mau so: ";

cin >> p.mauSo;

while (p.mauSo == 0)

{

cout << "Nhap lai mau so (khac 0): ";

cin >> p.mauSo;

}

}

void inPhanSo(PhanSo p)

{

rutGonPhanSo(p);

if (p.tuSo == 0) cout << "0";

else if (p.mauSo == 1) cout << p.tuSo;

else cout << p.tuSo << "/" << p.mauSo;

}

PhanSo congPhanSo(PhanSo a, PhanSo b)

{

PhanSo c;

c.tuSo = a.tuSo \* b.mauSo + b.tuSo \* a.mauSo;

c.mauSo = a.mauSo \* b.mauSo;

rutGonPhanSo(c);

return c;

}

PhanSo truPhanSo(PhanSo a, PhanSo b)

{

PhanSo c;

c.tuSo = a.tuSo \* b.mauSo - b.tuSo \* a.mauSo;

c.mauSo = a.mauSo \* b.mauSo;

rutGonPhanSo(c);

return c;

}

PhanSo nhanPhanSo(PhanSo a, PhanSo b)

{

PhanSo c;

c.tuSo = a.tuSo \* b.tuSo;

c.mauSo = a.mauSo \* b.mauSo;

rutGonPhanSo(c);

return c;

}

PhanSo chiaPhanSo(PhanSo a, PhanSo b)

{

PhanSo c;

c.tuSo = a.tuSo \* b.mauSo;

c.mauSo = a.mauSo \* b.tuSo;

rutGonPhanSo(c);

return c;

}

//Bài 2. Định nghĩa kiểu dữ liệu NhanVien gồm các thông tin sau :

//-Mã nhân viên kiểu string

//- Họ tên kiểu string

//- Năm vào làm kiểu int

//- Năm sinh kiểu int

//Viết các hàm :

//-Nhập vào đầy đủ thông tin của một nhân viên.

//- Xuất lại thông tin của nhân viên vừa nhập.

//- Tính tuổi và thâm niên làm việc của nhân viên này.Biết rằng :

//tuổi = năm hiện tại – năm sinh

//thâm niên = năm hiện tại – năm vào làm

//- Hàm main kiểm chứng chương trình

#include <iostream>

#include <string>

#include <ctime>

using namespace std;

struct nhanvien

{

string manv;

string hoten;

int nam;

int namvaolam;

};

void nhapnv(nhanvien &nv);

void thongtinnv(nhanvien nv);

int tinhtuoinv(nhanvien nv);

int tinhthamnien(nhanvien nv);

#pragma warning(disable : 4996)

int main()

{

nhanvien nv;

cout << "Nhap thong tin nhan vien: " << endl;

nhapnv(nv);

thongtinnv(nv);

cout << "Tuoi cua nhan vien: " << tinhtuoinv(nv) << endl;

cout << "Tham nien cua nhan vien: " << tinhthamnien(nv) << endl;

return 0;

}

void nhapnv(nhanvien &nv)

{

cout << "Ma nhan vien: ";

cin >> nv.manv; // do ma nhan vien ko co khoang tranh

cin.ignore();

cout << "Ho ten: ";

getline(cin, nv.hoten);

cout << "Nam sinh: ";

cin >> nv.nam;

cout << "Nam vao lam: ";

cin >> nv.namvaolam;

}

void thongtinnv(nhanvien nv)

{

cout << "Thong tin nhan vien: " << endl;

cout << "Ma nhan vien: " << nv.manv << endl;

cout << "Ho ten nhan vien: " << nv.hoten << endl;

cout << "Nam sinh nhan vien: " << nv.nam << endl;

cout << "Nam vao lam: " << nv.namvaolam << endl;

}

int tinhtuoinv(nhanvien nv)

{

time\_t now = time(0);//lay tg hien tai cua may tinh

tm \*tm = localtime(&now);//su dung doi tuong tm de co the truy xuat cac thuoc tinh

return (tm->tm\_year + 1900) - nv.nam;

}

int tinhthamnien(nhanvien nv)

{

time\_t now = time(0);//lay tg hien tai cua may tinh

tm \*tm = localtime(&now);//su dung doi tuong tm de co the truy xuat cac thuoc tinh

return (tm->tm\_year + 1900) - nv.namvaolam;

}

//Bài 3. Định nghĩa kiểu dữ liệu SinhVien gồm các thông tin :

//-Mã số sinh viên kiểu string

//- Họ tên kiểu string

//- Điểm toán kiểu double

//- Điểm văn kiểu double

//- Điểm ngoại ngữ kiểu double

//Viết các hàm :

//-Nhập vào thông tin một sinh viên

//- Nhập vào danh sách sinh viên trong một lớp học(sĩ số không quá 45).

//- In thông tin của sinh viên theo mã số sinh viên(do người dùng nhập)

//Bài tập thực hành Kỹ thuật lập trình

//- In danh sách sinh viên gồm các thông tin : Số thứ tự, mã số sinh viên, họ tên, điểm

//trung bình, xếp loại.Biết :

// -điểm trung bình = ((điểm toán + điểm văn) \* 2 + điểm ngoại ngữ) / 5

// - Xếp loại sinh viên :

//Nếu điểm trung bình từ 8 trở lên thì xếp loại là “Giỏi”

//Nếu điểm trung bình từ 6.5 đến < 8 thì xếp loại là “Khá”

// Nếu điểm trung bình từ 5 đến < 6.5 thì xếp loại là “TB”

// Nếu điểm dưới 5 thì xếp loại “Dưới TB”

// - Viết hàm main kiểm chứng chương trình.

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

#define MAX 45

struct Sinhvien {

string masv;

string hoten;

double toan;

double van;

double ngoaingu;

};

void nhapsv(Sinhvien &s);

void nhapdssv(Sinhvien arrSV[MAX], int n);

void thongtinsv(Sinhvien arrSV[MAX], int n, string maso);

void indssd(Sinhvien arrSV[MAX], int n);

double diemtrungbinh(Sinhvien sv);

string xeploaisv(Sinhvien sv);

int timsvtheomaso(Sinhvien arrSV[MAX], int n, string masv);

int main()

{

Sinhvien arrSV[MAX];

int n;

///\*cin.ignore();\*/

cout << "Nhap so sv: ";

cin >> n;

nhapdssv(arrSV, n);

indssd(arrSV, n);

cout << "Nhap ma so sinh vien can tim: ";

string masv;

cin >> masv;

thongtinsv(arrSV, n, masv);

return 0;

}

void nhapsv(Sinhvien &s)

{

cout << "ma so: ";

cin >> s.masv;

cin.ignore();

cout << "Ho ten: ";

getline(cin, s.hoten);

cout << "Diem toan: ";

cin >> s.toan;

cout << "Diem van: ";

cin >> s.van;

cout << "Diem ngoai ngu: ";

cin >> s.ngoaingu;

// do 1 luc nhap nhieu sv nen phai them dong igno

/\*cin.ignore;\*/

}

void nhapdssv(Sinhvien arrSV[MAX], int n)

{

for (int i = 0; i < n; i++)

{

cout << "Nhap thong tin sv thu: " << i + 1 << endl;

nhapsv(arrSV[i]);

}

}

void thongtinsv(Sinhvien arrSV[MAX], int n, string maso)

{

int pos = timsvtheomaso(arrSV, n, maso);

if (pos != -1)

{

cout << "Ma so: " << arrSV[pos].masv << endl;

cout << "Ho ten: " << arrSV[pos].hoten << endl;

cout << "Diem toan: " << arrSV[pos].toan << endl;

cout << "Diem van: " << arrSV[pos].van << endl;

cout << "Diem ngoai ngu: " << arrSV[pos].ngoaingu << endl;

}

else

{

cout << "Khong co thong tin" << endl;

}

}

void indssd(Sinhvien arrSV[MAX], int n)

{

cout << "STT\tMa so\tHo ten\t\t\tTrung binh\tXeploai" << endl;

for (int i = 0; i < n; i++)

{

cout << i + 1 <<"\t"<<arrSV[i].masv << "\t" << arrSV[i].hoten

<< setw(30 - arrSV[i].hoten.length()) << diemtrungbinh(arrSV[i])

<< "\t\t" << xeploaisv(arrSV[i]) << endl;

}

}

double diemtrungbinh(Sinhvien sv)

{

return ((sv.toan + sv.van) \* 2 + sv.ngoaingu) / 5;

}

string xeploaisv(Sinhvien sv)

{

double tb = diemtrungbinh(sv);

string loai;

if (tb >= 8) loai = "gioi";

else if (tb >= 6.5) loai = "kha";

else if (tb >= 5)loai = "trung binh";

else loai = "duoi tb";

return loai;

}

int timsvtheomaso(Sinhvien arrSV[MAX], int n, string masv)

{

for (int i = 0; i < n; i++)

if (arrSV[i].masv == masv)

return i;

return -1;

}

//Bài 4. Định nghĩa kiểu dữ liệu SanPham gồm các thông tin như sau :

//-Mã sản phẩm kiểu char(tối đa 10 ký tự)

//- Tên sản phẩm kiểu string.

//- Đơn giá kiểu double

//- Số lượng tồn kho kiểu int

//Viết chương trình dùng cấp phát động để :

//-Nhập thông tin cho tối đa 30 sản phẩm.

//- Xuất thông tin của các sản phẩm vừa nhập.

//- Sửa thông tin của một sản phẩm bất kỳ do người dùng yêu cầu.

//- Xóa thông tin của một sản phẩm bất kỳ do người dùng yêu cầu.

//- Sắp xếp lại toàn bộ sản phẩm theo trình tự số lượng tồn kho tăng dần.

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

struct Sanpham

{

char masp[10];

string tensp;

double dongia;

int ton;

};

void nhapdssp(Sanpham \* &arrsp, int n);

void indssp(Sanpham \*arrsp, int n);

void sapxepdssp(Sanpham \*arrsp, int n);

void suasp(Sanpham \*arrsp, int n, char masp[]);

void xoasp(Sanpham \*arrsp, int &n, char masp[]);

void xoasptheovitri(Sanpham arrsp, int &n, int vitri);

int main()

{

int n;

Sanpham \*arrsp;

cout << "Nhap so luong san pham: ";

cin >> n;

cin.ignore();

nhapdssp(arrsp, n);

indssp(arrsp, n);

cout << "Danh sach san pham sau khi ep tang dan theo so luong ton kho: \n";

sapxepdssp(arrsp, n);

indssp(arrsp, n);

cout << "Nhap ma san pham can sua: ";

char masp[10];

cin.getline(masp, 10);

suasp(arrsp, n, masp);

cout << "Nhap masp can xoa: ";

cin.getline(masp, 10);

xoasp(arrsp, n, masp);

return 0;

}

void nhapdssp(Sanpham \* &arrsp, int n)

{

arrsp = new Sanpham[n];

for (int i = 0; i < n; i++)

{

cout << "Nhap ma san pham: ";

cin.getline(arrsp[i].masp, 10);

cout << "Nhap ten san pham: ";

getline(cin, arrsp[i].tensp);

cout << "Don gia: ";

cin >> arrsp[i].dongia;

cout << "So luong ton: ";

cin >> arrsp[i].ton;

cin.ignore();

}

}

void indssp(Sanpham \*arrsp, int n)

{

cout << "Ma sp\tTen sp\t\tDon gia\t\tTon kho\n";

cout << resetiosflags(ios::fixed) << setprecision(2);

for (int i = 0; i < n; i++)

{

cout << arrsp[i].masp

<< "\t" << arrsp[i].tensp

<< "\t" << arrsp[i].dongia

<< "\t\t" << arrsp[i].ton << endl;

}

}

void sapxepdssp(Sanpham \*arrsp, int n)

{

Sanpham tmp;

for (int i = 0; i < n - 1; i++)

for (int j = 0; j < n; j++)

if (arrsp[i].ton > arrsp[j].ton)

{

tmp = arrsp[i];

arrsp[i] = arrsp[j];

arrsp[j] = tmp;

}

}

void suasp(Sanpham \*arrsp, int n, char masp[])

{

for (int i = 0; i < n; i++)

{

if (strcmp(arrsp[i].masp, masp) == 0)

{

cout << "Nhap ten moi: ";

getline(cin, arrsp[i].tensp);

cout << "Nhap don gia moi: ";

cin >> arrsp[i].dongia;

cout << "Nhap ton kho moi: ";

cin >> arrsp[i].ton;

cin.ignore();

cout << "cap nhat thanh cong\n";

indssp(arrsp, n);

return;

}

}

cout << "Khong tim thay san pham\n";

}

void xoasp(Sanpham \*arrsp, int &n, char masp[])

{

for (int i = 0; i < n; i++)

{

if (strcmp(arrsp[i].masp, masp) == 0)

{

xoasptheovitri(arrsp, n, i);

cout << "Xoa thanh cong! Danh sach sau khi xoa la: " << endl;

indssp(arrsp, n);

return;

}

}

cout << "Khong tim thay san pham\n";

}

void xoasptheovitri(Sanpham arrsp, int &n, int vitri)

{

for (int i = vitri + 1; i < n; i++)

arrsp[i - 1] = arrsp[i];

n--;

}

//Định nghĩa kiểu dũ liệu Matran gồm các thông tin sau :

//-Số dòng, số cột kiểu int

//- Con trỏ kiểu con trỏ đến mảng 2 chiều lưu trữ số nguyên

//Viết chương trình C++ thực hiện các việc sau :

//-Phát sinh ngẫu nhiên giá trị cho 1 biến kiểu Matran

//- Xuất giá trị đang lưu trữ trong mảng 2 chiều của biến kiểu Matran

//- Tính tổng các phần tử nào là số đối xứng trong mảng 2 chiều của biến kiểu Matran

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

struct Matran {

int n\_rows; // số dòng

int n\_cols; // số cột

int\*\* arr; // mảng 2 chiều lưu trữ số nguyên

// hàm khởi tạo mặc định

Matran() {

n\_rows = 0;

n\_cols = 0;

arr = NULL;

}

// hàm khởi tạo với số dòng và số cột

Matran(int rows, int cols)

{

n\_rows = rows;

n\_cols = cols;

arr = new int\*[rows];

for (int i = 0; i < rows; i++)

{

arr[i] = new int[cols];

}

}

// hàm phát sinh giá trị ngẫu nhiên cho mảng

void randomize()

{

srand(time(NULL));

for (int i = 0; i < n\_rows; i++)

{

for (int j = 0; j < n\_cols; j++)

{

arr[i][j] = rand() % 10;

}

}

}

// hàm xuất mảng

void print()

{

for (int i = 0; i < n\_rows; i++)

{

for (int j = 0; j < n\_cols; j++)

{

cout << arr[i][j] << " ";

}

cout << endl;

}

}

// hàm tính tổng các phần tử đối xứng (Đối xứng là tổng đường chéo của ma trận vuông)

int sum\_symmetric()

{

int sum = 0;

for (int i = 0; i < n\_rows; i++)

{

for (int j = 0; j < n\_cols; j++)

{

if (i <= j && arr[i][j] == arr[j][i])

{

sum += arr[i][j];

}

}

}

return sum;

}

};

int main()

{

Matran A(3,3); // tạo ma trận A có 3 dòng và 3 cột

A.randomize(); // phát sinh giá trị ngẫu nhiên cho A

cout << "Ma tran A:" << endl;

A.print(); // xuất giá trị của A

int sum = A.sum\_symmetric(); // tính tổng các phần tử đối xứng của A

cout << "Tong cac phan tu doi xung cua A la: " << sum << endl;

return 0;

}

Sử dụng cấu trúc (struct) viết ứng dụng quản lý một danh sách sinh viên với các thông tin như sau: mã số

sinh viên (làm khóa chính, không được trùng), Họ, Tên, Điểm môn 1, Điểm môn 2. Chương trình cho

phép thực hiện các chức năng sau:

a. Nhập danh sách sinh viên từ file text có cấu trúc như sau:

Trong đó, các trường dữ liệu cách nhau bởi ký tự ‘-‘, theo thứ tự MSSV-Họ-Tên-Điểm môn 1-

Điểm môn 2

b. Tìm kiếm, in thông tin sinh viên theo MSSV (MSSV, Họ, Tên, Điểm môn 1, Điểm môn 2,

Điểm trung bình

c. Thêm vào danh sách một sinh viên (có kiểm tra MSSV không được trùng)

d. Xóa sinh viên theo MSSV

e. In danh sách sinh viên theo Tên tăng dần

f. In danh sách sinh viên theo điểm trung bình giảm dần

3

g. In danh sách các sinh viên Đậu (TB >=5), Rớt (TB < 5)

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

using namespace std;

struct Student {

int studentID;

string firstName;

string lastName;

double grade1;

double grade2;

double average;

Student(int id, const string& first, const string& last, double g1, double g2)

: studentID(id), firstName(first), lastName(last), grade1(g1), grade2(g2), average((g1 + g2) / 2) {}

};

// Hàm nhập danh sách sinh viên từ file

void readStudentsFromFile(const string& filename, vector<Student>& students) {

ifstream inputFile(filename);

if (!inputFile) {

cout << "Khong mo duoc file " << filename << endl;

return;

}

students.clear();

int id;

string firstName;

string lastName;

double grade1;

double grade2;

char dash;

while (inputFile >> id >> dash >> firstName >> dash >> lastName >> dash >> grade1 >> dash >> grade2) {

students.push\_back(Student(id, firstName, lastName, grade1, grade2));

}

inputFile.close();

cout << "Da nhap " << students.size() << " sinh vien tu file " << filename << endl;

}

// Hàm in thông tin sinh viên

void outputStudent(const Student& student) {

cout << "MSSV: " << student.studentID << endl;

cout << "Ho va ten: " << student.lastName << " " << student.firstName << endl;

cout << "Diem mon 1: " << student.grade1 << endl;

cout << "Diem mon 2: " << student.grade2 << endl;

cout << "Diem trung binh: " << student.average << endl;

}

// Hàm tìm kiếm và in thông tin sinh viên theo MSSV

void searchStudentByID(const vector<Student>& students, int id) {

auto it = find\_if(students.begin(), students.end(), [&](const Student& student) {

return student.studentID == id;

});

if (it != students.end()) {

outputStudent(\*it);

} else {

cout << "Khong tim thay sinh vien voi MSSV " << id << endl;

}

}

// Hàm thêm sinh viên vào danh sách

void addStudent(vector<Student>& students) {

int id;

cout << "Nhap MSSV: ";

cin >> id;

auto it = find\_if(students.begin(), students.end(), [&](const Student& student) {

return student.studentID == id;

});

if (it != students.end()) {

cout << "MSSV da ton tai. Khong the them sinh vien." << endl;

return;

}

string firstName, lastName;

double grade1, grade2;

cout << "Nhap ho va ten: ";

cin.ignore();

getline(cin, lastName, ' ');

getline(cin, firstName);

cout << "Nhap diem mon 1: ";

cin >> grade1;

cout << "Nhap diem mon 2: ";

cin >> grade2;

students.push\_back(Student(id, firstName, lastName, grade1, grade2));

cout << "Them sinh vien thanh cong." << endl;

}

// Hàm xóa sinh viên theo MSSV

void deleteStudentByID(vector<Student>& students, int id) {

auto it = find\_if(students.begin(), students.end(), [&](const Student& student) {

return student.studentID == id;

});

if (it != students.end()) {

students.erase(it);

cout << "Xoa sinh vien thanh cong." << endl;

} else {

cout << "Khong tim thay sinh vien voi MSSV " << id << endl;

}

}

// Hàm in danh sách sinh viên theo tên tăng dần

void printStudentsByNameAscending(const vector<Student>& students) {

vector<Student> sortedStudents = students;

sort(sortedStudents.begin(), sortedStudents.end(), [](const Student& s1, const Student& s2) {

return s1.lastName < s2.lastName;

});

for (const Student& student : sortedStudents) {

outputStudent(student);

cout << "-------------------------" << endl;

}

}

// Hàm in danh sách sinh viên theo điểm trung bình giảm dần

void printStudentsByAverageDescending(const vector<Student>& students) {

vector<Student> sortedStudents = students;

sort(sortedStudents.begin(), sortedStudents.end(), [](const Student& s1, const Student& s2) {

return s1.average > s2.average;

});

for (const Student& student : sortedStudents) {

outputStudent(student);

cout << "-------------------------" << endl;

}

}

// Hàm in danh sách sinh viên đậu và rớt

void printStudentsPassedAndFailed(const vector<Student>& students) {

cout << "Danh sach sinh vien Dau:" << endl;

for (const Student& student : students) {

if (student.average >= 5.0) {

outputStudent(student);

cout << "-------------------------" << endl;

}

}

cout << "Danh sach sinh vien Rot:" << endl;

for (const Student& student : students) {

if (student.average < 5.0) {

outputStudent(student);

cout << "-------------------------" << endl;

}

}

}

int main() {

vector<Student> students;

string filename = "students.txt";

// Đọc danh sách sinh viên từ file

readStudentsFromFile(filename, students);

char choice;

do {

cout << "\n==== QUAN LY SINH VIEN ====" << endl;

cout << "1. Them sinh vien" << endl;

cout << "2. Xoa sinh vien theo MSSV" << endl;

cout << "3. Tim kiem va in thong tin sinh vien theo MSSV" << endl;

cout << "4. In danh sach sinh vien theo ten tang dan" << endl;

cout << "5. In danh sach sinh vien theo diem trung binh giam dan" << endl;

cout << "6. In danh sach sinh vien Dau va Rot" << endl;

cout << "0. Thoat" << endl;

cout << "Nhap lua chon: ";

cin >> choice;

switch (choice) {

case '1': {

addStudent(students);

break;

}

case '2': {

int id;

cout << "Nhap MSSV can xoa: ";

cin >> id;

deleteStudentByID(students, id);

break;

}

case '3': {

int id;

cout << "Nhap MSSV can tim kiem: ";

cin >> id;

searchStudentByID(students, id);

break;

}

case '4': {

cout << "Danh sach sinh vien theo ten tang dan:" << endl;

printStudentsByNameAscending(students);

break;

}

case '5': {

cout << "Danh sach sinh vien theo diem trung binh giam dan:" << endl;

printStudentsByAverageDescending(students);

break;

}

case '6': {

printStudentsPassedAndFailed(students);

break;

}

case '0': {

cout << "Thoat chuong trinh." << endl;

break;

}

default: {

cout << "Lua chon khong hop le. Vui long chon lai." << endl;

break;

}

}

} while (choice != '0');

return 0;