BT6.1

#include<algorithm>

#include<vector>

using namespace std;

void nhap(long a[], long n) {

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

cin >> a[i];

}

}

void xuat(long a[], long n) {

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

cout << a[i] << ' ';

}

cout << endl;

}

void xuatLonHon5(long a[], long n) {

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

if (a[i] > 5)

cout << a[i] << " ";

}

}

void sapXep(long a[], long n) {

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

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

if (a[j] > a[j + 1])

swap(a[j], a[j + 1]);

}

}

xuat(a, n);

}

void them(long a[], long& n, long x, long gt) {

n++;

for (int i = n; i > x; i--) {

a[i] = a[i - 1];

}

a[x] = gt;

xuat(a, n);

}

void xoa(long a[], long& n, long x) {

for (int i = x; i < n; i++) {

a[i] = a[i + 1];

}

n--;

xuat(a, n);

}

void capNhat(long a[], long n, long x, long gt) {

a[x] = gt;

xuat(a, n);

}

int main() {

long a[100];

long n, chon, vt, gt;

do {

cin >> chon;

if (chon == 8)

break;

switch (chon) {

case 1:

cout << "Nhap so luong: ";

cin >> n;

nhap(a, n);

break;

case 2:

xuat(a, n);

break;

case 3:

xuatLonHon5(a, n);

break;

case 4:

sapXep(a, n);

break;

case 5:

cout << "Nhap vt: ";

cin >> vt;

cout << "Nhap gt: ";

cin >> gt;

them(a, n, vt, gt);

break;

case 6:

cout << "Nhap vt: ";

cin >> vt;

xoa(a, n, vt);

break;

case 7:

cout << "Nhap vt: ";

cin >> vt;

cout << "Nhap gt: ";

cin >> gt;

capNhat(a, n, vt, gt);

break;

system("pause");

system("cls");

} while (chon >= 1 && chon <= 7);

}

BTH 6.2

#include<string>

#include<fstream>

using namespace std;

struct SinhVien {

int MSSV;

string hoTen;

string gt;

string diaChi;

int namSinh;

double diemTB;

};

void nhapDs(SinhVien sv[], long n) {

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

cin.ignore(1);

cout << "Nhap MSSV: ";

cin >> sv[i].MSSV;

cin.ignore(1);

cout << "Nhap Ho ten: ";

getline(cin, sv[i].hoTen);

cout << "Nhap dia chi: ";

getline(cin, sv[i].diaChi);

cout << "Nhap nam sinh: ";

//getline(cin, sv[i].namSinh);

cout << "Nhap diem TB: ";

cin >> sv[i].diemTB;

cout << endl;

}

}

void nhapDsTuFile(SinhVien sv[], long &n) {

ifstream myFile("SinhVien.txt");

if (myFile.is\_open())

{

int i = 0;

while (!myFile.eof())

{

myFile >> sv[i].MSSV;

myFile.ignore(1);

getline(myFile, sv[i].hoTen, ',');

getline(myFile, sv[i].gt, ',');

getline(myFile, sv[i].diaChi, ',');

myFile >> sv[i].namSinh;

myFile.ignore(1);

myFile >> sv[i].diemTB;

i++;

}

n = i;

myFile.close();

}

else

cout << "Mo file khong thanh cong!\n";}

void xuatDs(SinhVien sv[], long n) {

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

cout << "MSSV: " << sv[i].MSSV << endl;

cout << "Ho ten: " << sv[i].hoTen << endl;

cout << "dia chi: " << sv[i].diaChi << endl;

cout << "nam sinh: " << sv[i].namSinh << endl;

cout << "diem TB: " << sv[i].diemTB << endl;

cout << endl;

}

}

void suaThongTin(SinhVien sv[],long n, long i) {

cout << "Nhap MSSV: ";

cin >> sv[i].MSSV;

cin.ignore(1);

cout << "Nhap Ho ten: ";

getline(cin, sv[i].hoTen);

cout << "Nhap dia chi: ";

getline(cin, sv[i].diaChi);

cout << "Nhap nam sinh: ";

cin >> sv[i].namSinh;

cout << "Nhap diem TB: ";

cin >> sv[i].diemTB;

xuatDs(sv, n);

}

void them(SinhVien sv[], long& n, long x) {

for (int i = n; i > x; i--) {

sv[i].MSSV = sv[i - 1].MSSV;

sv[i].hoTen = sv[i - 1].hoTen;

sv[i].diaChi = sv[i - 1].diaChi;

sv[i].namSinh = sv[i - 1].namSinh;

sv[i].diemTB = sv[i - 1].diemTB;

}

cout << "Nhap MSSV: ";

cin >> sv[x].MSSV;

cin.ignore(1);

cout << "Nhap Ho ten: ";

getline(cin, sv[x].hoTen);

cout << "Nhap dia chi: ";

getline(cin, sv[x].diaChi);

cout << "Nhap nam sinh: ";

cin >> sv[x].namSinh;

cout << "Nhap diem TB: ";

cin >> sv[x].diemTB;

n++;

xuatDs(sv, n);

}

void xoa(SinhVien sv[], long& n,long x) {

for (int i = x; i < n; i++) {

sv[i].MSSV = sv[i + 1].MSSV;

sv[i].hoTen = sv[i + 1].hoTen;

sv[i].diaChi = sv[i + 1].diaChi;

sv[i].namSinh = sv[i + 1].namSinh;

sv[i].diemTB = sv[i + 1].diemTB;

}

n--;

xuatDs(sv, n);}

void xemTTTheoMS(SinhVien sv[], long n, int ms) {

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

if (sv[i].MSSV == ms) {

cout << "MSSV: " << sv[i].MSSV << endl;

cout << "Ho ten: " << sv[i].hoTen << endl;

cout << "dia chi: " << sv[i].diaChi << endl;

cout << "nam sinh: " << sv[i].namSinh << endl;

cout << "diem TB: " << sv[i].diemTB << endl;

break;

}

}

}

void sapXepTheoDiemTB(SinhVien sv[], long n) {

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

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

if (sv[j].diemTB < sv[j + 1].diemTB)

swap(sv[j], sv[j + 1]);

}}

xuatDs(sv, n);}

int demSoSVDiemHon5(SinhVien sv[], long n) {

long dem = 0;

SinhVien sv1[100];

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

if (sv[i].diemTB >= 2) {

dem++;

}

}

return dem;

}

int main() {

SinhVien sv[100];

long n = 0, chon;

do {

cin >> chon;

if (chon == 9)

break;

switch (chon) {

case 1:

nhapDsTuFile(sv, n);

break;

case 2:

cout << "Nhap so luong: ";

cin >> n;

nhapDs(sv, n);

break;

case 3:

xuatDs(sv, n);

break;

case 4:

long vt;

cout << "Nhap vi tri: ";

cin >> vt;

them(sv, n, vt);

break;

case 5:

long vt1;

cout << "Nhap vi tri: ";

cin >> vt1;

xoa(sv, n, vt1);

break;

case 6:

long ms;

cout << "Nhap ms: ";

cin >> ms;

xemTTTheoMS(sv, n, ms);

break;

case 7:

cout << "Ds sau sap xep: " << endl;

sapXepTheoDiemTB(sv, n);

break;

case 8:

cout << "So luong sv: " << demSoSVDiemHon5(sv, n) <<endl;

break;

}

system("pause");

system("cls");

} while (chon >= 1 && chon <= 8);

system("pause");

}

BÃ i thá»±c hÃ nh 7

#include<string>

#include<algorithm>

struct SinhVien

{

string id, name, sex, address;

int date;

double avg;

SinhVien \*next;

};

void init(SinhVien\*& first)

{

first = NULL;

}

void nhapDS(SinhVien\*& first) {

SinhVien\* sv = new SinhVien;

if (first == NULL) {

first = sv;

}

else {

SinhVien \*temp = first;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = sv;

}

sv->next = NULL;

cin.ignore();

cout << "Nhap mssv: ";

getline(cin, sv->id);

cout << "Nhap Ho ten: ";

getline(cin, sv->name);

cout << "Nhap gt: ";

getline(cin, sv->sex);

cout << "Nhap dia chi: ";

getline(cin, sv->address);

cout << "Nhap nam sinh: ";

cin >> sv->date;

cout << "Nhap diem trung binh: ";

cin >> sv->avg;

cin.ignore();

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

}

void nhapDSNhieu(SinhVien\*& first, long n)

{

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

nhapDS(first);

}

}

void xuatDS(SinhVien\* first)

{

while (first != NULL) {

cout << "MSSV: " << first->id << endl;

cout << "Ho va Ten: " << first->name << endl;

cout << "Gioi tinh: " << first->sex << endl;

cout << "Nam sinh: " << first->date << endl;

cout << "Diem Trung Binh: " << first->avg << endl;

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

first = first->next;

}

}

void hieuChinhSV(SinhVien\*& sv, string ms) {

SinhVien \*temp = sv;

while (temp != NULL) {

if (temp->mssv == ms) {

cout << "Nhap mssv: ";

getline(cin, sv->id);

cout << "Nhap Ho ten: ";

getline(cin, sv->name);

cout << "Nhap gt: ";

getline(cin, sv->sex);

cout << "Nhap dia chi: ";

getline(cin, sv->address);

cout << "Nhap nam sinh: ";

cin >> sv->date;

cout << "Nhap diem trung binh: ";

cin >> sv->avg;

cin.ignore();

break;

}

temp = temp->next;

}

xuatDS(sv);

}

void xoaCuoi(SinhVien \*&sv) {

SinhVien \*temp = sv;

while (temp->next->next != NULL) {

temp = temp->next;

}

SinhVien \*x = temp->next;

delete x;

temp->next = NULL;

}

void xoaDau(SinhVien \*&sv) {

SinhVien \*temp = sv;

sv = sv->next;

delete temp;

}

void xoaSV(SinhVien \*&sv, string ms) {

SinhVien\* temp = sv;

while (temp != NULL) {

if (temp->mssv == ms) {

if (temp->next == NULL) {

xoaCuoi(sv);

}

else if (temp == sv){

xoaDau(sv);

}

else

{

SinhVien\* temp1 = sv;

while (temp1->next != temp)

{

temp1 = temp1->next;

}

temp1->next = temp->next;

delete temp;

}

break;

}

temp = temp->next;

}

cout << "Danh sach hien tai: " << endl;

xuatDS(sv);

}

void xemThongTin(SinhVien\* first, string ms) {

cout << "Thong tin sv co MS: " << ms << endl;

while (first != NULL)

{

if (first->id == ms)

break;

first = first->next;

}

cout << "MSSV: " << first->id << endl;

cout << "Ho va Ten: " << first->name << endl;

cout << "Gioi tinh: " << first->sex<< endl;

cout << "Nam sinh: " << first->date<< endl;

cout << "Diem Trung Binh: " << first->avg << endl;

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

}

void sapXepTheoTB(SinhVien \*&sv) {

SinhVien \*temp = sv;

SinhVien \*temp1 = sv;

long count = 1;

while (temp != NULL)

{

while (temp1->next != NULL)

{

if (temp1->avg < temp1->next->avg)

{

swap(temp1->avg, temp1->next->avg);

swap(temp1->name, temp1->next->name);

swap(temp1->address, temp1->next->address);

swap(temp1->sex, temp1->next->sex);

swap(temp1->id, temp1->next->id);

swap(temp1->date, temp1->next->date);

}

temp1 = temp1->next;

}

temp = temp->next;

temp1 = sv;

}

cout << "DS sau sap xep: " << endl;

xuatDS(sv);

}

long demSVCoAvgHon5(SinhVien \*sv)

{

long count = 0;

while (sv != NULL)

{

if (sv->avg >= 5)

{

count++;

}

sv = sv->next;

}

return count;

}

int main(){

SinhVien\* sv = new SinhVien;

long n, chon;

string ms;

do {

if (chon == 9)

break;

switch (chon) {

case 1:

init(sv);

break;

case 2:

cout << "Nhap so luon muon nhap: ";

cin >> n;

nhapDSNhieu(sv, n);

break;

case 3:

xuatDS(sv);

break;

case 4:

nhapDS(sv);

break;

case 5:

cout << "Nhap mssv: ";

getline(cin, ms);

xoaSV(sv, ms);

break;

case 6:

cout << "Nhap mssv: ";

getline(cin, ms);

xemThongTin(sv, ms);

break;

case 7:

sapXepTheoTB(sv);

break;

case 8:

cout << "So sv co TB > 5: " << demSVCoAvgHon5(sv) << endl;

break;

}

system("pause");

} while (chon >= 1 && chon <= 8);

}

BÃ i thá»±c hÃ nh 8

struct node

{

int info;

node\* left;

node\* right;

}\*root;

int MAX = 100;

void init() {

root = NULL;

}

void add(node\*& p, int x) {

p = new node;

if (p == NULL) {

p->info = x;

p->left = NULL;

p->right = NULL;

}

else {

if (p->info == x) {

return;

}

else if (p->info > x) {

return add(p->left, x);

}

else {

return add(p->right, x);

}

}

}

node\* timKiem(node\* p, int x) {

if (p != NULL) {

if (p->info == x) {

return p;

}

else

{

if (p->info < x) {

return TimKiem(p->right, x);

}

else

return TimKiem(p->left, x);

}

}

return NULL;

}

void searchStandFor(node\*& p, node\*& q) {

if (q->left == NULL) {

p->info = q->info;

p = q;

q = q->right;

}

else

searchStandFor(p, q->left);

}

int deleteNode(node\*& p, int x) {

if (p == NULL)

return 0;

if (p->info == x) {

node\* T = p;

if (p->left == NULL)

p = p->right;

else if (p->right == NULL)

p = p->left;

else

searchStandFor(T, p->right);

delete T;

return 1;

}

if (p->info < x)

return deleteNode(p->right, x);

if (p->info > x)

return deleteNode(p->left, x);

}

void duyetNLR(node\* p)

{

if (p != NULL)

{

cout << p->info << " ";

duyetNLR(p->left);

duyetNLR(p->right);

}

}

void duyetLRN(node\* p)

{

if (p != NULL)

{

duyetLRN(p->left);

duyetLRN(p->right);

cout << p->info << " ";

}

}

void duyetLNR(node\* p)

{

if (p != NULL)

{

duyetLNR(p->left);

cout << p->info << " ";

duyetLNR(p->right);

}

}

int demNut(node\* p){

if (p == NULL)

{

return 0;

}

int demL = demNut(p->left);

int demR = demNut(p->right);

return demL + demR + 1;

}

int aggregate(node\* p){

if (p != NULL)

{

int A = aggregate(p->left);

int B = aggregate(p->right);

return p->info + A + B;

}

return 0;

}

float avg(node\* p) {

return aggregate(p) / demNut(p);

}

void timMax(node\* t){

if (t != NULL)

{

if (t->info > MAX)

{

MAX= t->info;

}

timMax(t->left);

timMax(t->right);

}

}

bool kiemTraSNT(int x) {

if (x < 2)

{

return false;

}

else

{

if (x == 2) {

return true;

}

else{

if (x % 2 == 0) {

return false;

}

else {

for (int i = 2; i < x / 2; i++) {

if (x % i == 0)

return false;

}

}

}

}

return true;

}

int sLSNT(node\* t) {

int dem = 0;

if (t != NULL) {

if (kiemTraSNT(t->info)) {

dem++;

}

sLSNT(t->left);

sLSNT(t->right);

}

return dem;

}

int main() {

int chon;

bool in = false;

do {

cout << "\n";

cin >> chon;

switch (chon) {

case 1: { //Khoi Tao

init();

in = true;

break;}

case 2:{".Them Node \n"

if (in) {

int x;

cout << "Nhap gia tri can them: ";

cin >> x;

add(root, x);

}

else {

cout << "Vui long chon khoi tao\n";

}

break;

}

case 3: "3.Tim Phan tu\n"{

if (in) {

int x;

cout << "Nhap gia tri can tim: ";

cin >> x;

if (timKiem(root, x) != NULL) {

cout << "Gia tri " << x << " co ton tai trong cay nhi phan tai vi tri " << timKiem(root, x) << endl;

}

else

cout << "Gia tri " << x << " khon ton tai trong cay\n";

}

else {

cout << "Vui long chon khoi tao\n";

}

break;

}

case 4: "4. Xoa phan tu\n"

{

if (in) {

int x;

cout << "Nhap gia tri can xoa: ";

cin >> x;

if (deleteNode(root, x)) {

cout << "Xoa thanh cong\n";

}

else

cout << "Khong xoa thanh cong\n";

}

else {

cout << "Vui long chon khoi tao\n";

}

break;

}

case 5: "5. Duyet LNR\n"{

if (in) {

cout << "Duyet LNR: ";

duyetLNR(root);

}

else {

cout << "Vui long chon khoi tao" <<endl;

}

break;

}

case 6: "6. Duyet NLR\n"{

if (in) {

cout << "DuyetNLR:";

duyetNLR(root);

}

else {

cout << "Vui long chon khoi tao" <<endl;

}

break;

}

case 7: "7. Duyet LRN\n"

{

if (in) {

cout << "Duyet LRN:";

duyetLRN(root);

}

else {

cout << "Vui long chon khoi tao" <<endl;

}

break;

}

case 8: Viet thu tuc dem so phan tu

{

if (in) {

cout << "So phan tu trong cay: " << demNut(root);

}

else {

cout << "Vui long chon khoi tao" <<endl;

}

break;

}

case 9: Trung Binh Cong cac phan tu{

if (in) {

cout << "TB CONG CUA CAC PHAN TU TRONG CAY: " << avg(root);

}

else {

cout << "Vui long chon khoi tao" <<endl;

}

break;

}

case 10: Tim Max{

if (in) {

timMax(root);

cout << "Gia tri lon nhat trong cay: " << MAX;

}

else {

cout << "Vui long chon khoi tao" <<endl;

}

break;

}

case 11: Dem so phan tu la so nguyen to{

if (in) {

cout << "So phan tu la so nguyen to trong cay: " << sLSNT(root);}

else {

cout << "Vui long chon khoi tao" <<endl;

}

break;

}

default:

cout << "Byeee!" <<endl;

break;

}

} while (chon >= 0 && chon <= 11);