Text, letter

Description automatically generatedBài 1:  
Text, letter

Description automatically generated

Text, letter

Description automatically generated

#include <iostream>

using namespace std;

struct SinhVien

{ string MaSV;

string HoDem;

string Ten;

int NamSinh;

float DiemTK;};

struct List{

int cous;

SinhVien sv[Max];};

void creat (List &L)

{L.cous = -1;}

int emply(List L)

{ return (L.cous == -1);}

int full(List L)

{ return (L.cous = Max - 1);}

void Nhap(SinhVien &sv)

{ cout << "Nhap ma sinh vien: ";

fflush(stdin);

getline(cin, sv.MaSV);

cout << "Nhap ho dem: ";

fflush(stdin);

getline(cin, sv.HoDem);

cout << "Nhap ten: ";

fflush(stdin);

getline(cin, sv.Ten);

cout << "Nhap nam sinh: ";

cin >> sv.NamSinh;

cout << "Nhap diem tong ket: ";

cin >> sv.DiemTK;}

void Xuat(SinhVien sv)

{ cout << sv.MaSV << setw(15) << sv.HoDem << setw(15) << sv.Ten << setw(15) << sv.NamSinh << setw(10) << sv.DiemTK << endl;}

void Nhap\_DS(List &L){

cout << "Nhap so phan tu danh sach can tao: ";

cin >> L.cous;

for(int i=0; i < L.cous ; i++)

{ cout << "Nhap thong tin sinh vien thu: " << i+1 << endl;

Nhap(L.sv[i]);}}

void Xuat\_DS(List &L)

{ cout << "MaSV" << setw(15) << "Ho Dem" << setw(15) << "Ten" << setw(15) << "NamSinh" << setw(10) << "DiemTK" << setw(15) << endl;

for (int i = 0; i < L.cous; i++)

{ Xuat(L.sv[i]); }}

int remove\_number(List L)

{ int k;

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

cin >> k;

if(!emply(L))

{ if (k <= L.cous + 1 && k >0)

{

for (int i =k; i <= L.cous; i++)

{

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

}

return 1;

}

else return 0;

}

}

int insertintheList(List &L)

{

CanBo x;

cout << "Nhap du lieu sinh vien can them\n";

Nhap(x);

int k;

cout << "\nNhap vi tri can them: ";

cin >> k;

int i;

if (k <= L.cunt+1 && k > 0 && full(L))

{

for (i = L.cunt; i >= k-1; i--)

{

L.A[i+1] = L.A[i];

}

L.A[i-1] = x;

L.cunt++;

return 1;

}

else return 0;

}

void Swap(SinhVien \*a, SinhVien \*b)

{

SinhVien temp = \*a;

\*a = \*b;

\*b = temp;

}

void SelectionSort(List L)

{

for (int i =0; i < L.cous; i++)

{

int m =i;

for (int j = i+1; j < L.cous; j++)

{

if (L.sv[j] < L.sv[m])

m =j;

}

if (m != i)

{

Swap(&arr[m], &arr[i]);

}

}

}

int main()

{

List L;

Nhap\_DS(L);

cout << "Danh sach cac sinh vien la " << endl;

Xuat\_DS(L);

cout << "Xoa phan tu " << endl;

remove\_number(L);

Xuat\_DS(L);

cout << "\nChen sinh vien" << endl;

insertintheList(L);

Xuat\_DS(L);

cout << "\nSap xep danh sach " << endl;

insertintheList(L);

Xuat\_DS(L);

return 0;

Bài 2:

}Text, letter

Description automatically generated

Text, letter

Description automatically generatedText, letter

Description automatically generated

#include <iostream>

#include <iomanip>

using namespace std;

#define Max 10

struct MaHang

{

int MaHang;

string TenHang;

string DVTinh;

long DonGia;

int SoLuong;

long ThanhTien;

};

struct List

{

int Count;

MaHang A[Max];

};

void creat (List &L)

{

L.Count = -1;

}

int emply(List L)

{

return (L.Count == -1);

}

int full(List L)

{

return (L.Count = Max - 1);

}

void Nhap(MaHang &cb)

{

cout << "Nhap ma hang: ";

cin >> cb.MaHang;

cout << "Nhap ten hang: ";

fflush(stdin);

getline(cin, cb.TenHang);

cout << "Nhap don vi tinh: ";

fflush(stdin);

getline(cin, cb.DVTinh);

cout << "Nhap don gia: ";

cin >> cb.DonGia;

cout << "Nhap so luong: ";

cin >> cb.SoLuong;

}

void Nhap\_ds(List &L)

{

cout << "Nhap so luong phan tu: ";

cin >> L.Count;

for (int i =0 ; i < L.Count ; i++)

{

cout << "\nNhap thong tin don hang thu: " << i +1 << endl;

Nhap(L.A[i]);

}

}

void Xuat(MaHang cb)

{

cb.ThanhTien = cb.DonGia \* cb.SoLuong;

cout << cb.MaHang << setw(15) << cb.TenHang<< setw(20) << cb.DVTinh << setw(10) << cb.DonGia << setw(10) << cb.SoLuong << setw(15) << cb.ThanhTien << endl;

}

void Xuat\_DS(List L)

{

cout << "MaHang " << setw(10) << "TenHang " << setw(10) << "DV Tinh " << setw(10) << "DonGia" << setw(10) << "SoLuong" << setw(10) << "ThanhTien " << endl;

for (int i =0; i < L.Count; i++)

{

Xuat(L.A[i]);

}

}

int remove\_number(List L)

{

int k;

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

cin >> k;

if(!emply(L))

{

if (k <= L.Count + 1 && k >0)

{

for (int i =k; i <= L.Count; i++)

{

L.A[i-1] = L.A[i];

}

return 1;

}

else return 0;

}

}

int insertintheList(List &L)

{

MaHang x;

cout << "Nhap du lieu hang hoa can them\n";

Nhap(x);

int k;

cout << "\nNhap vi tri can them: ";

cin >> k;

int i;

if (k <= L.Count+1 && k > 0 && full(L))

{

for (i = L.Count; i >= k-1; i--)

{

L.A[i+1] = L.A[i];

}

L.A[i-1] = x;

L.Count++;

return 1;

}

else return 0;

}

void Swap(MaHang \*a, MaHang \*b)

{

MaHang temp = \*a;

\*a = \*b;

\*b = temp;

}

void Sort(List L)

{

for (int i = 0; i < L.Count-1 ; i++)

{

for (int j = i + 1; j < L.Count; j++)

{

if (L.A[i].ThanhTien < L.A[j].ThanhTien)

{

Swap(&L.A[i], &L.A[j]);

}

}

}

Xuat\_DS(L);

}

int main()

{

List L;

Nhap\_ds(L);

cout << "\nXuat danh sach" << endl;

Xuat\_DS(L);

cout << "\nXoa phan tu" << endl;

remove\_number(L);

Xuat\_DS(L);

cout << "\nChen hang hoa" << endl;

insertintheList(L);

Xuat\_DS(L);

cout << "\nSap xep" << endl;

Sort(L);

Xuat\_DS(L);

return 0;

}

Bài 3:

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

#define MAX 7

struct CanBo

{

string MaCanBo;

string HoDem;

string Ten;

string GioiTinh;

int Tuoi;

float HeSoLuong;

long Luong;

};

struct List

{

int cunt;

CanBo A[MAX];

};

List L;

void creat (List &L)

{

L.cunt = -1;

}

int emply(List L)

{

return (L.cunt == -1);

}

int full(List L)

{

return (L.cunt = MAX - 1);

}

void Nhap(CanBo &cb)

{

cout << "Nhap ma can bo: ";

fflush(stdin);

getline(cin, cb.MaCanBo);

cout << "Nhap ho dem: ";

fflush(stdin);

getline(cin, cb.HoDem);

cout << "Nhap ten: ";

fflush(stdin);

getline(cin, cb.Ten);

cout << "Nhap gioi tinh: ";

fflush(stdin);

getline(cin, cb.GioiTinh);

cout << "Nhap tuoi: ";

cin >> cb.Tuoi;

cout << "Nhap he so luong: ";

cin >> cb.HeSoLuong;

}

void Nhap\_ds(List &L)

{

cout << "Nhap so luong phan tu: ";

cin >> L.cunt;

for (int i =0 ; i < L.cunt ; i++)

{

cout << "\nNhap thong tin can bo thu: " << i +1 << endl;

Nhap(L.A[i]);

}

}

void Xuat(CanBo cb)

{

cb.Luong = cb.HeSoLuong \* 1150000;

cout << cb.MaCanBo << setw(15) << cb.HoDem << setw(20) << cb.Ten << setw(10) << cb.GioiTinh << setw(10) << cb.Tuoi << setw(15) << cb.HeSoLuong << setw(15) << cb.Luong << endl;

}

void Xuat\_DS(List L)

{

cout << "Ma can bo " << setw(10) << "Ho dem " << setw(10) << "Ten " << setw(10) << "Gioi tinh" << setw(10) << "Tuoi" << setw(10) << "He so luong " << setw(10) << "Luong " << endl;

for (int i =0; i < L.cunt; i++)

{

Xuat(L.A[i]);

}

}

void ShowList(List L)

{

for (int i =0; i < L.cunt; i++)

{

if (L.A[i].Tuoi < 50 && L.A[i].GioiTinh == "Nu")

{

Xuat(L.A[i]);

}

}

}

void ShowSearch(List L)

{

int Count =0;

for (int i =0; i < L.cunt ; i++)

{

if (L.A[i].Ten == "Tung")

{

Count++;

}

}

for (int i = 0 ; i < L.cunt; i++)

{

if (L.A[i].Ten == "Tung" && Count == 1)

{

Xuat(L.A[i]);

break;

}

}

}

int insertintheList(List &L)

{

CanBo x;

cout << "Nhap du lieu can bo can them\n";

Nhap(x);

int k;

cout << "\nNhap vi tri can them: ";

cin >> k;

int i;

if (k <= L.cunt+1 && k > 0 && full(L))

{

for (i = L.cunt; i >= k-1; i--)

{

L.A[i+1] = L.A[i];

}

L.A[i-1] = x;

L.cunt++;

return 1;

}

else return 0;

}

int main()

{

List L;

Nhap\_ds(L);

Xuat\_DS(L);

cout << "Show List" << endl;

ShowList(L);

cout << "\nHien thi thong tin can bo can tim " << endl;

ShowSearch(L);

insertintheList(L);

return 0;

}

Bài 4:

#include <iostream>

#include <iomanip>

using namespace std;

#define Max 7

struct SinhVien

{

string MaSV;

string HoDem;

string Ten;

int NamSinh;

float DiemTK;

};

struct List{

int cous;

SinhVien sv[Max];

};

void creat (List &L)

{

L.cous = -1;

}

int emply(List L)

{

return (L.cous == -1);

}

int full(List L)

{

return (L.cous = Max - 1);

}

void Nhap(SinhVien &sv)

{

cout << "Nhap ma sinh vien: ";

fflush(stdin);

getline(cin, sv.MaSV);

cout << "Nhap ho dem: ";

fflush(stdin);

getline(cin, sv.HoDem);

cout << "Nhap ten: ";

fflush(stdin);

getline(cin, sv.Ten);

cout << "Nhap nam sinh: ";

cin >> sv.NamSinh;

cout << "Nhap diem tong ket: ";

cin >> sv.DiemTK;

}

void Xuat(SinhVien sv)

{

cout << sv.MaSV << setw(15) << sv.HoDem << setw(15) << sv.Ten << setw(15) << sv.NamSinh << setw(10) << sv.DiemTK << endl;

}

void Nhap\_DS(List &L)

{

cout << "Nhap so phan tu danh sach can tao: ";

cin >> L.cous;

for(int i=0; i < L.cous ; i++)

{

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

Nhap(L.sv[i]);

}

}

void Xuat\_DS(List &L)

{

cout << "MaSV" << setw(15) << "Ho Dem" << setw(15) << "Ten" << setw(15) << "NamSinh" << setw(10) << "DiemTK" << setw(15) << endl;

for (int i = 0; i < L.cous; i++)

{

Xuat(L.sv[i]);

}

}

int remove\_number(List L)

{

int k;

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

cin >> k;

int dem = L.cous;

if(!emply(L))

{

if (k <= L.cous + 1 && k >0)

{

for (int i =k; i <= L.cous; i++)

{

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

}

return 1;

}

else return 0;

}

if (L.cous < dem)

{

Xuat\_DS(L);

}

}

int Search(List L)

{

if (!emply(L))

{

for (int i=0; i < L.cous ;i++)

{

if (L.sv[i].Ten == "Doanh")

{

Xuat(L.sv[i]);

}

}

}

}

void Swap(SinhVien \*sv1, SinhVien \*sv2)

{

SinhVien \*temp = sv1;

sv1 = sv2;

sv2 = temp;

}

void Sort(List L)

{

for (int i = 0; i < L.cous-1 ; i++)

{

for (int j = i + 1; j < L.cous ; j++)

{

if (L.sv[i].DiemTK < L.sv[j].DiemTK)

{

SinhVien team = L.sv[i];

L.sv[i] = L.sv[j];

L.sv[j] = team;

}

}

}

Xuat\_DS(L);

}

int main()

{

List L;

Nhap\_DS(L);

Xuat\_DS(L);

cout << "\nXoa danh sach sinh vien" << endl;

remove\_number(L);

cout << "\nTim kiem sinh vien co ten 'Doanh' " << endl;

Search(L);

cout << "\nSap xep lai danh sach " << endl;

Sort(L);

return 0;

}