**DANH SÁCH ĐẶC**

#include<stdio.h>

#define Maxlength 30

typedef int ElementType;

typedef int Position;

typedef struct {

ElementType Elements[Maxlength];

Position Last;

} List;

//Ket thuc khai bao - dinh nghia cac phep toan co ban

void MakeNull\_List(List \*L)

{ L->Last=0;

}

int Empty\_List(List L)

{ return (L.Last==0);

}

int Full\_List(List L)

{ return (L.Last==Maxlength);

}

Position FirstList(List L)

{ return 1;

}

Position EndList(List L)

{ return L.Last+1;

}

Position Next(Position P, List L)

{ return P+1;

}

Position Previous(Position P, List L)

{ return P-1;

}

ElementType Retrieve(Position P,List L)

{ return L.Elements[P-1];

}

void Insert\_List(ElementType X,Position P, List \*L)

{ int i=0;

if (L->Last==Maxlength)

printf("\n Danh sach day !!!");

else if ((P<1)||(P>L->Last+1))

printf("\n Vi tri khong hop le !!!");

else {

for (i=L->Last;i>=P;i--)

L->Elements[i]=L->Elements[i-1];

L->Last++;

L->Elements[P-1]=X;

}

}

// In ds L ra man hinh

void Print\_List(List L)

{ Position P;

P=FirstList(L);

printf("\n Bat dau in danh sach: ");

while (P!=EndList(L))

{ printf("%d ",Retrieve(P,L));

P=Next(P,L);

}

printf("\n Ket thuc in danh sach \n");

}

void Read\_List(List \*L)

{ int i,N;

ElementType X;

MakeNull\_List(L);

printf("\n Nhap vao so phan tu trong ds: ");

scanf("%d",&N);

for(i=1;i<=N;i++)

{ printf("\n Phan tu thu %d = ",i);

scanf("%d",&X);

Insert\_List(X,EndList(\*L),L);

}

}

// Vi tri cua phan tu X trong List L

Position locate(int X,List L){

Position P=FirstList(L);

Position E=EndList(L);

int found = 0;

while(P!=E){

if(Retrieve(P,L)==X){

return P;

}

else{

P = Next(P,L);

}

}

return EndList(L);

}

// Xóa phan tu tai vi trí P trong danh sách L

void deleteList(Position P,List \*L){

int i;

if(Empty\_List(\*L)){

printf(" Danh sach rong!");

}

else if(P<1||P>L->Last+1){

printf(" Vi tri P khong hop le!");

}

else{

for(i=P-1;i<L->Last-1;i++){

L->Elements[i]=L->Elements[i+1];

}

L->Last--;

}

}

// Sap xep tang dan cac phan tu trong danh sach L

void sapXepTangDan(List \*L){

int i,j;

for(i=0;i<L->Last-1;i++){

for(j=i+1;j<L->Last;j++){

if(L->Elements[i]>L->Elements[j]){

ElementType temp = L->Elements[i];

L->Elements[i]=L->Elements[j];

L->Elements[j]=temp;

}

}

}

}

// Xoa phan tu trung

void delete\_duplicate(List \*L)

{ Position p,q; //ki?u v? trí c?a các ph?n t? trong danh sách

p=FirstList(\*L); //v? trí ph?n t? d?u tiên trong danh sách

while (p!=EndList(\*L))

{ q=Next(p,\*L); //v? trí ph?n t? d?ng ngay sau ph?n t? p

while (q!=EndList(\*L))

{

if (Retrieve(p,\*L) == Retrieve(q,\*L))

deleteList(q,L); // xoa ph?n t?

else

q=Next(q,\*L);

}

p=Next(p,\*L);

}

}

int main()

{ List L;

Read\_List(&L);

Print\_List(L);

// Test ham locate(int X,List L)

printf("\n Nhap phan tu X can tim: ");

ElementType X;

scanf("%d",&X);

printf("\n Vi tri cua phan tu X trong List L la: %d",locate(X,L));

// Test ham deleteList(Position P,List \*L)

printf("\n\n Nhap vi tri cua phan tu can xoa = ");

Position P;

scanf("%d",&P);

deleteList(P,&L);

Print\_List(L);

printf("\n Nhap phan tu X can chen: ");

scanf("%d",&X);

printf("\n Nhap vi tri can chen: ");

scanf("%d",&P);

Insert\_List(X,P,&L);

printf("\nDanh sach sau khi chen la:\n");

Print\_List(L);

// Test ham sapXepTangDan

printf("\nDanh sach sau khi sap xep tang dan: \n");

sapXepTangDan(&L);

Print\_List(L);

// Test ham xoaPhanTuTrung

printf("\nDanh sach sau khi xoa phan tu trung: \n");

delete\_duplicate(&L);

Print\_List(L);

return 0;

while(1);

}