Danh Sách Mãng(Con Trỏ) L1

Gi Chú: Với đề thi có chử danh sách mãng L, Con Trỏ, Tên Func cũng là tên yêu cầu đời

#include <stdio.h>

#include <conio.h>

#include <malloc.h>

typedef float elementType;

typedef struct Node{

elementType element;

Node\* Next;

};

typedef Node\* Position;

typedef Position List;

void MakeNull\_List(List \*L){

(\*L)=(Node\*)malloc(sizeof(Node));

(\*L)->Next=NULL;

}

int EmptyList(List L){

return L->Next == NULL;

}

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

Position T;

T=(Node\*)malloc(sizeof(Node));

T->element = X;

T->Next = P->Next;

P->Next = T;

}

elementType Retrieve(Position P,List L){

if(P->Next != NULL)

return(P->Next->element);

}

void Nhap\_DS(int n,List \*L){

Position P; elementType X;

int i;

P = \*L;

for(i = 1; i <=n ; i++){

printf("Nhap Phan Tu X = ");

scanf("%f",&X);

Insert\_List(X,P,L);

P = P->Next;

}

}

void In\_DS(List L){

Position P;

P= L;

while(P->Next != NULL){

printf("\n%.2f",Retrieve(P,L));

P = P->Next;

}

}

void Nhap\_DS\_Nguoc(int n ,List \*L){

Position P; elementType X;

int i;

P = \*L;

for(i = 1; i <=n ; i++){

printf("Nhap Phan Tu X = ");

scanf("%f",&X);

Insert\_List(X,P,L);

}

}

float Tinh\_TB(List L){

Position P;

int Tong=0;

int Dem=0;

P = L;

while(P->Next != NULL){

Tong = Tong+ Retrieve(P,L);

Dem++;

P = P->Next;

}

return (float)Tong/Dem;

}

void XenPT(elementType X,List \*L,int k){

Position P;

P=\*L;

int vitri = 0;

while((P->Next!= NULL) && (vitri < k-1)){

P = P->Next;

vitri++;

}

if(vitri == k-1){

Insert\_List(X,P,L);

}else

printf("\nVi tri %d khong hop le",k);

}

Position Locate(elementType X,List L){

Position P;

int Found = 0;

P = L;

while((P->Next !=NULL) &&(Found ==0))

{

if(P->Next->element == X){

Found =1;

}else

P= P->Next;

}

if(Found = 1){

return P;

}else

return NULL;

}

void Delete\_List(Position P, List \*L){

Position Temp;

if(P->Next!=NULL){

Temp = P->Next;

P->Next = Temp->Next;

free(Temp);

}

}

void XoaPT(elementType X,List \*L){

Position P;

if(EmptyList(\*L)){

printf("Danh sach rong: Errors");

}else{

P = Locate(X,\*L);

if(P != NULL){

Delete\_List(P,L);

}else

printf("Khong co PT %.2f trong Ds",X);

}

}

void XoaPTTrungNhau(List \*L){

Position P,Q;

P = \*L;

if(EmptyList(\*L)){

printf("Danh sach rong : Errors");

}else

{

while(P->Next !=NULL){

Q = P->Next;

while(Q->Next != NULL){

if(Q->element == Q->Next->element)

Delete\_List(Q,L);

Q= Q->Next;

}

P=P->Next;

}

}

}

int main(){

int k,n; elementType X;

List L1,L2;

printf("Nhap n: ");scanf("%d",&n);

MakeNull\_List(&L1);

printf("Nhap Danh sach L1 theo thu tu nhap:\n");

Nhap\_DS(n,&L1);

printf("\nDanh sach L1 theo thu tu nhap la :");

In\_DS(L1);

/\*

MakeNull\_List(&L2);

printf("\nNhap Danh sach L2 Theo thu tu nguoc :\n");

Nhap\_DS\_Nguoc(n,&L2);

printf("\n Danh Sach L2 theo thu tu nguoc la :");

In\_DS(L2);

printf("\nGia tri trung binh cua cac phan tu trong L2 la : %.2f ",Tinh\_TB(L2));

printf("\n");

printf("Nhap X :");scanf("%f",&X);

printf("Nhap k :");scanf("%d",&k);

XenPT(X,&L1,k);

printf("\nDanh sach L1 sau khi xen la :");

In\_DS(L1);

XoaPT(X,&L1);

printf("\nDanh sach L1 sau khi xoa la :");

In\_DS(L1);

\*/

XoaPTTrungNhau(&L1);

printf("\nDanh sach sau khi xoa PT trung nhau la:");

In\_DS(L1);

getch();

return 0;

}