#include "iostream";

#include "Math.h";

using namespace std;

//khai bao kieu du lieu tuong duong kieu int

typedef int typedata;

//khai bao cau truc node

typedef struct node

{

typedata data;

node\*link;

}NODE;

//khai bao kieu con tro node co ten la ptrNODE

typedef NODE\* ptrNODE;

typedef struct list

{

NODE \* first;

ptrNODE last;

}LIST;

//khoi tao danh sach rong

void Init(LIST &list)

{

list.first = list.last = NULL;

}

ptrNODE GetNode(typedata data)

{

ptrNODE p = (ptrNODE)malloc(sizeof(NODE));

//p = new NODE;

if (p == NULL)

exit;

p->data = data;

p->link = NULL;

return p;

}

//Ham AddFirst them mot phan tu vao dau danh sach

void AddFirst(LIST &list, ptrNODE new\_Node)

{

if (list.first == NULL)//danh sach rong

{

list.first = new\_Node;

list.last = list.first;

}

else // danh sach khac rong

{

new\_Node->link = list.first;

list.first = new\_Node;

}

}

void AddLast(LIST &list, ptrNODE new\_Node)

{

if (list.first == NULL)//danh sach rong

{

list.first = new\_Node;

list.last = list.first;

}

else//danh sach khac rong

{

list.last->link = new\_Node;

list.last = new\_Node;

}

}

void InsertList(LIST &list, typedata data,bool firstOption)

{

ptrNODE p;

p = GetNode(data);

if (firstOption)

AddFirst(list, p);

else

AddLast(list, p);

}

void CreateList(LIST &list,bool firstOption)

{

int i = 0;

Init(list);

while (true)

{

cout << "Nhap PT thu " << ++i<<": ";

typedata data;

cin >> data;

if (data != -1)

{

InsertList(list, data,firstOption);

}

else

{

break;

}

}

}

void InDanhSachDon(LIST &list)

{

ptrNODE p;

p = list.first;

while (p!=NULL)

{

//In data

cout << p->data<<", ";

p = p->link;

}

}

void TimKiemGiaTri(LIST &list,int x)

{

ptrNODE p;

p = list.first;

int count = 0;

while (p != NULL)

{

//In data

if (p->data == x)

{

cout << "co gia tri " << x << " trong danh sach";

count++;

}

//tang p

p = p->link;

}

if (count==0)

cout << "khong gia tri " << x << " trong danh sach";

}

void ThemVaoMotNodeSauQ(LIST &list, NODE\*q, ptrNODE new\_Node)

{

if (q != NULL)

{

new\_Node->link = q->link;

q->link = new\_Node;

if (q == list.last)

{

list.last = new\_Node;

}

}

}

ptrNODE Searching(LIST &list, typedata x)

{

ptrNODE p;

p = list.first;

while (p != NULL&&p->data==x)

{

p = p->link;

}

return p;

}

void ThemValueVaoSauGiaTriX(LIST &list, typedata value, typedata x)

{

//tiem kiem gia tri value;

ptrNODE q;

q = Searching(list, value);

//tao mot node moi co gia tri x;

ptrNODE p = GetNode(x);

if (q != NULL && p != NULL)

{

ThemVaoMotNodeSauQ(list, q, p);

}

else

{

cout << "khong ton tai nut q";

//InsertList(list, x, true);

}

}

void XoaDauDanhSach(LIST &list)

{

ptrNODE p = list.first;

if (list.first != NULL)

{

list.first = p->link;

if (list.first == NULL)

list.last = NULL;

free(p);

//delete p;

}

}

void XoaTatCaDanhSach(LIST&list)

{

ptrNODE p = list.first;

while (p!=NULL)

{

XoaDauDanhSach(list);

p = p->link;

}

}

void XoaNodeSauQ(LIST &list, ptrNODE q)

{

if (q != NULL&&q->link != NULL)

{

ptrNODE p;

p = q->link;

q->link = p->link;

if (p->link == NULL)

list.last = q;

free(p);

}

}

void XoaNutSauNutCoGiaTriX(LIST &list,typedata x)

{

ptrNODE q = Searching(list, x);

XoaNodeSauQ(list, q);

}

void XoaMotNodeCoGiaTriX(LIST &list, typedata x)

{

ptrNODE p = list.first;

ptrNODE q = p;

while (p != NULL)

{

if (p->data == x)

{

if (p == list.first)

{

XoaDauDanhSach(list);

break;

}

else{

XoaNodeSauQ(list, q);

break;

}

}

q = p;

p = p->link;

}

}

void main()

{

LIST list;

CreateList(list,false);

InDanhSachDon(list);

TimKiemGiaTri(list,4);

XoaMotNodeCoGiaTriX(list, 4);

InDanhSachDon(list);

system("pause");

}