**Bài số 3: Danh sách liên kết**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>;

typedef struct sinhvien

{

char maso[10];

char hoten[50];

int tuoi;

}sinhvien;

typedef struct node

{

sinhvien data;

node \*pnext;

}node;

typedef struct list

{

node \*phead;

node \*ptail;

}list;

void khoitao(list &l)

{

l.phead=l.ptail=NULL;

}

node \*taonut()

{

node \*p;sinhvien sv;

p=new(node);

if(p!=NULL)

{

printf("\nNhap ma so:");

fflush(stdin);gets(sv.maso);

printf("\nNhap ho ten:");

fflush(stdin);gets(sv.hoten);

printf("\nNhap tuoi:");

scanf("%d",&sv.tuoi);

p->data=sv;

p->pnext=NULL;

return p;

}

return NULL;

}

void addtail(list &l,node \*new\_ele)

{

if(l.phead==NULL)

{

l.phead=new\_ele;

l.ptail=l.phead ;

}

else

{

l.ptail->pnext=new\_ele;

l.ptail=new\_ele;

}

}

void addfirst(list &l,node \*new\_ele)

{

if(l.phead==NULL)

{

l.phead=new\_ele;

l.ptail=l.phead ;

}

else

{

new\_ele->pnext=l.phead;

l.phead=new\_ele;

}

}

node \*timvitri(list l)

{

char ma[10];node \*q;

printf("\nNhap ma so can tim :");

fflush(stdin);gets(ma);

q=l.phead;

while(q!=NULL)

{

if(strcmp(ma,q->data.maso)==0)

return q;

else

q=q->pnext;

}

return NULL;

}

void addafter(list &l,node \*q,node \*new\_ele)

{

if(q!=NULL)

{

new\_ele->pnext=q->pnext;

q->pnext=new\_ele;

if(q==l.ptail)

l.ptail=new\_ele;

}

else

addfirst(l,new\_ele);

}

void taods(list &l)

{

int n;node \*p;

printf("\nnhap so phan tu:");

scanf("%d",&n);

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

{

p=taonut();

addfirst(l,p);

}

}

void duyetds(list l)

{

node \*p;p=l.phead;

printf("\nMa so || Ho ten ||Tuoi ");

while(p!=NULL)

{

printf("\n %s %s %d",p->data.maso,p->data.hoten,p->data.tuoi);

p=p->pnext;

}

}

sinhvien removehead(list &l)

{

node \*p;

sinhvien x;

if(l.phead!=NULL)

{

p=l.phead;x=p->data;

l.phead=l.phead->pnext;

delete p;

if(l.phead==NULL)

l.ptail=NULL;

}

return x;

}

void removeafter(list &l,node \*q)

{

node \*p;

if(q!=NULL)

{

p=q->pnext;

if(p!=NULL)

{

if(p==l.ptail)

l.ptail=q;

q->pnext=p->pnext;

delete p;

}

}

else

removehead(l);

}

void remove(list &l, char ma[])

{

node \*p,\*q;

p=l.phead;

while((p!=NULL)&&(strcmp(p->data.maso,ma)!=0))

{

q=p;

p=p->pnext;

}

if(p==NULL)

printf("\n Sinh vien khong co trong danh sach ");

else

if(p==l.phead)

{

l.phead=p->pnext;

delete(p);

}

else

{

q->pnext=p->pnext;

delete(p);

}

}

void Interchange(list &l)

{

for(node \*p=l.phead;p!=l.ptail;p=p->pnext)

for(node \*q=p->pnext;q!=NULL;q=q->pnext)

if(p->data.tuoi>q->data.tuoi)

{

sinhvien a;

a=p->data;

p->data=q->data;

q->data=a;

}

}

void noids(list &l,list &l2)

{

if(l2.phead==NULL)

return;

if(l.phead==NULL)

{

l.phead=l2.phead;

l.ptail=l2.ptail;

}

else

{

l.ptail->pnext=l2.phead;

l.ptail=l2.ptail;

}

khoitao(l2);

}

void quicksort(list &l)

{

node \*x,\*p;

list l1,l2;

if(l.phead==l.ptail)

return;

khoitao(l1);

khoitao(l2);

x=l.phead;

l.phead=x->pnext;

x->pnext=NULL;

while(l.phead!=NULL)

{

p=l.phead;

l.phead=p->pnext;

if(atoi(p->data.maso)<atoi(x->data.maso))

addtail(l1,p);

else

addtail(l2,p);

}

quicksort(l1);

quicksort(l2);

noids(l,l1);

addtail(l,x);

noids(l,l2);

}

int main()

{

list l;node \*q,\*new\_ele; char ma[10];

khoitao(l);

taods(l);

printf("\n Danh nhap la:");

duyetds(l);

/\*printf("\ndanh sach sau khi xoa la:");

fflush(stdin);

gets(ma);

remove(l,ma);

printf("\n Danh sau khi xoa la:");

duyetds(l);

/\*q=timvitri(l);

printf("\nnhap sinh vien moi:");

new\_ele=taonut();

addafter(l,q,new\_ele);

printf("\ndanh sach sau khi chen sinh vien moi:");

q=timvitri(l);

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

removeafter(l,q);

printf("\n Danh sach sau khi sap xep :\n");

Interchange(l);

quicksort(l);

duyetds(l);\*/

getch();

}