**Cấu trúc cây nhị phân**

#include"stdio.h"

#include"conio.h"

#include"string.h"

typedef struct sv

{

char hoten[40];

int tuoi;

}sv;

typedef struct node

{

sv data;

node \*left;

node \*right;

}node;

typedef struct node \*TREE;

TREE root;

void taocay(TREE &root)

{

sv x;

printf("\n nhap thong tin sv:");

printf("\n nhap ho ten:");

fflush(stdin);

gets(x.hoten);

if(strcmp(x.hoten,"n")!=0)

{ printf("\n nhap tuoi:");

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

}

if(strcmp(x.hoten,"n")!=0)

{

root=new (node);

root->data=x;

printf("\n cay con trai cua:%s",x.hoten);

taocay(root->left);

printf("\n cay con phai cua:%s",x.hoten);

taocay(root->right);

}else

root=NULL;

}

void NLR(TREE root)

{

if(root!=NULL)

{

printf("\n ho ten:%s",root->data.hoten);

printf("\n tuoi:%d",root->data.tuoi);

NLR(root->left);

NLR(root->right);

}

}

void LRN(TREE root)

{

if(root!=NULL)

{

LRN(root->left);

LRN(root->right);

printf("\n ho ten:%s",root->data.hoten);

printf("\n tuoi:%d",root->data.tuoi);

}

}

void LNR(TREE root)

{

if(root!=NULL)

{

LNR(root->left);

printf("\n ho ten:%s",root->data.hoten);

printf("\n tuoi:%d",root->data.tuoi);

LNR(root->right);

}

}

int max(int a,int b)

{

if(a>b)

return a;

else

return b;

}

int chieucao(TREE root)

{

if(root==NULL)

return 0;

else

return max(chieucao(root->left),chieucao(root->right))+1;

}

int demla(TREE r)

{ int dem=0;

if(r==NULL)

return 0;

else

{ if((r->left==NULL)&&(r->right==NULL))

dem++;

return (demla(r->left)+demla(r->right)+dem);

}

}

int main()

{

TREE t;

taocay(t);

printf("\n duyet cay theo thu tu truoc:");

NLR(t);

printf("\n duyet cay theo thu tu sau:");

LRN(t);

printf("\n duyet cay theo thu tu giua:");

LNR(t);

printf("\n So nut la: %d",demla(t));

getch();

}