Hàm trả về Node

B1: Duyệt tìm node cần xoá

B2: Xét trường hợp

+ Không tìm thấy return Null

+ Không có lá return NULL( nhớ free nốt cần xoá trước)

+ Trái có phải không, thì gán giá trị node cần xoá bằng node con trái, và quay về bài toán xoá nốt con trái

+ Phải có trái không, thì gán giá trị node cần xoá bằng node con phải , và quay về bài toán xoá nốt con phải

+ Có 2 lá

++Tìm giá trị nhỏ nhất tận cùng bên trái của node con phải (với Node cha là nốt cần xoá)

++ Nếu mà Node con phải không có node trái, thì gán giá trị node cần xoá=node con phải, và quay về bài toán xoá node con phải (chắc chắn là trường hợp có 1 lá nằm bên phải)

Node\* deleteNode(Node\*p,int val)

{

if(val>p->val)

{

p->Right=deleteNode(p->Right,val);

return p;

}

else

if(val<p->val)

{

p->Left=deleteNode(p->Left,val);

return p;

}

//Tim val trong node p , do xuong

if(p==NULL)//ko tim thay

return p;

if(p->Left==NULL && p->Right==NULL)

{

free(p);

return NULL;

}

if(p->Left==NULL)//Phai co, trai khong co

{

p->val=p->Right->val;

p->Right=deleteNode(p->Right,p->Right->val);

return p;

}

if(p->Right==NULL) //trai co, phai khong

{

p->val=p->Left->val;

p->Left=deleteNode(p->Left,p->Left->val);

return p;

}

Node\*pp=minval(p->Right);

if(pp==p->Right)

{

p->val=p->Right->val;

p->Right=deleteNode(p->Right,p->Right->val);

return p;

}

else

{

p->val=pp->val;

p->Right=deleteNode(p->Right,pp->val);

return p;

}

}