```
// arbori.cpp : Defines the entry point for the console application.
//

/*
creare arbore binar de cautare
stergere nod din arborele binar de cautare

*/

#include "stdafx.h"

struct bnode {
    int data;
    int cnt;
    bnode* left;
    bnode* right;
};
bnode* build_abc(bnode *r, int a);      // adauga un nod cu cheia a in arborele binar de cautare cu
    radacina r
void ldr(bnode* r);                     // parcurge in ordine arborele cu radacina r
bnode* del_node(bnode* r,int key);      // sterge nodul cu informatia key

bnode* root=NULL;

int _tmain(int argc, _TCHAR* argv[])
{
int key=-1;     // elementul curent de la tastatura
    while (key!=0)
    {
    // citeste elementele de la tastatura
    printf("Element nou = ");
    scanf("%d",&key);
    if (key!=0) root=build_abc(root,key);
    }
    // traverseaza arborele
    printf("Traversarea in ordine este:\n");
    ldr(root);
    printf("\n");

    printf("Elementul de sters: ");
    scanf("%d",&key);
    del_node(root,key);
    printf("Traversarea in ordine dupa stergere este:\n");
    ldr(root);
    printf("\n");

    return 0;
}

bnode* build_abc(bnode*r, int a)
{
    if (r==NULL)
    {
        r=new bnode;
        r->data=a;
        r->cnt=1;
        r->left=NULL;
        r->right=NULL;
    }
    else
        {
        if (a < r->data ) r->left=build_abc(r->left,a);
        if (a > r->data ) r->right=build_abc(r->right,a);
        if (a==r->data) r->cnt=r->cnt+1;
        }
```

```
    return r;
}

void ldr(bnode* r)
{
    if(r!=NULL)
    {
    ldr(r->left);
    if(r->cnt==1) printf("%d ", r->data);
    else printf("%d(%d) ", r->data,r->cnt);
    ldr(r->right);
    }
}

bnode* del_node(bnode* r,int key)
{
    bnode*v,*q;

    if (r==NULL) return NULL;
    else
    {
    if (key < r->data) r->left=del_node(r->left,key);
    else if (key > r->data) r->right=del_node(r->right,key);
    else
    {
    if (r->cnt == 1)               // stergere nod r
        {
            if(r->left==NULL && r->right==NULL) // nod frunza
            {
                v=r;
                r=NULL;
                delete v;
            }
            if(r->left==NULL && r->right!=NULL)     // nod cu 1 singur descendent (dreapta)
            {
                v=r;
                r=r->right;
                delete v;
            }
            if(r->right==NULL && r->left!=NULL)     // nod cu 1 singur descendent (stinga)
            {
                v=r;
                r=r->left;
                delete v;
            }
            if(r->right!=NULL && r->left!=NULL)     // nod cu 2 singur descendenti
            {
                v=r->left;
                while (v->right!=NULL) { q=v;v=v->right;}   // nodul cel mai din dreapta al subarborelui
    sting
                r->data=v->data;                            // se va sterge nodul v (care are 1 descendent
    pe stinga)
                                                            // in locul nodului r
                q->right=v->left;
                delete v;
            }
        }
    else r->cnt=r->cnt-1;       // decrementare contor de aparitii
    }
    return r;
    }
}
```