

```

#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include "mapa.h"
#include "arvore.h"

struct smapa {
    int chave;
    int dados;
    Mapa* esq;
    Mapa* dir;
};

static Mapa *cria_no (int c, int d) {
    Mapa *nn = (Mapa *)malloc(sizeof(Mapa));
    if (nn!=NULL) {
        nn->esq = nn->dir = NULL;
        nn->chave =c;
        nn->dados = d;
    }
    return nn;
}

Mapa* cria (void) {
    //Mapa *new = (Mapa *)malloc(sizeof(Mapa));
    //return new;
    return NULL;
}

int busca (Mapa *m, int chave) {
    while (m!=NULL) {
        if (chave < m->chave)
            m = m->esq;
        else if (chave > m->chave)
            m = m->dir;
        else
            return m->dados; /* achou */
    }
    return INT_MIN;
}

void destroi (Mapa *m) {
    if (m==NULL) return;
    destroi (m->esq);
    destroi (m->dir);
    free(m);
}

/*Mapa *insere (Mapa *m, int chave, int d) {
    if (m==NULL)
        m = cria_no(chave, d);
    else
        {
            if(m->chave > chave)
                return insere(m->esq,chave,d);
            else
                return insere(m->dir,chave,d);
        }
}*/

```

```

Mapa *retira (Mapa *m, int chave) {
    if (m==NULL)
        return NULL;
    return m;
}

Mapa* cria_raiz(int chave, int dados, Mapa* sae, Mapa* sad) {
    Mapa *r = cria_no(chave, dados);
    if (r != NULL) {
        r->esq = sae;
        r->dir = sad;
    }
    return r;
}

void mostra(Mapa* m) {
    printf("[");
    if (m != NULL) {
        printf("<%d %d> ", m->chave, m->dados);
        mostra(m->esq);
        mostra(m->dir);
    }
    printf("] ");
}

int num_nos (Mapa *m) {
    if(m == NULL)
        return 0;
    return 1 + num_nos(m->esq) + num_nos(m->dir);
}

int maior_chave (Mapa *m) {
    if(m == NULL)
        return INT_MIN;
    else if(m->dir == NULL)
        return m->chave;
    else
        return maior_chave(m->dir);
}

int num_maiores_que (Mapa *m, int n) {
    int nos = 0;
    if(m == NULL)
        return 0;
    else if(m->chave > n)
    {
        return 1 + num_maiores_que(m->dir,n) + num_maiores_que(m-
>esq,n);
    }
    else
        return num_maiores_que(m->dir,n);
    //return nos;
}

```