

```
#include <stdlib.h>
#include <stdio.h>
typedef struct arvore{
    int info;
    struct arvore *esq;
    struct arvore *dir;
}Arvore;
Arvore* Ler_arvore_arq(FILE *arq){
    char c;
    int x;
    fscanf(arq,"%c",&c);
    fscanf(arq,"%d",&x);
    if(x==-1){
        fscanf(arq,"%c",&c);
        return NULL;
    }
    else{
        Arvore *no=(Arvore*)malloc(sizeof(Arvore));
        no->info=x;
        no->esq=Ler_arvore_arq(arq);
        no->dir=Ler_arvore_arq(arq);
        fscanf(arq,"%c",&c);
        return no;
    }
}
void imprimir_preordem (Arvore* a){
    if(a!=NULL)
    {
        printf("%d ",a->info);
        imprimir_preordem(a->esq);
        imprimir_preordem(a->dir);
    }
}
void imprimir_ordem (Arvore* a){
    if(a!=NULL)
    {
        imprimir_ordem(a->esq);
        printf("%d ",a->info);
        imprimir_ordem(a->dir);
    }
}
void imprimir_posordem (Arvore* a){
    if(a!=NULL)
    {
        imprimir_posordem(a->esq);
        imprimir_posordem(a->dir);
        printf("%d ",a->info);
    }
}
```

```

int altura(Arvore* a){
    if(a==NULL)
        return 0;
    else{
        int ae,ad;
        ae=altura(a->esq);
        ad=altura(a->dir);
        if(ae>ad)
            return ae+1;
        else
            return ad+1;
    }
}

int existe(Arvore *a,int y){
    if(a==NULL)
        return 0;
    else{
        if(a->info==y)
            return 1;
        else{
            if(existe(a->esq,y)==1)
                return 1;
            else
                return existe(a->dir,y);
        }
    }
}

int numero_nos(Arvore* a){
    if(a==NULL)
        return 0;
    else
        return 1+numero_nos(a->esq)+numero_nos(a->dir);
}

Arvore* altera_elemento(Arvore* a, int y){
    if (a==NULL)
        return a;
    if(a->info==y){
        int z;
        printf ("Digite o novo numero\n");
        scanf("%d",&z);
        a->info=z;
        return a;
    }
    else{
        a->esq=altera_elemento(a->esq,y);
        a->dir=altera_elemento(a->dir,y);
        return a;
    }
}

Arvore* remover_arvore(Arvore* a){
    if (a!=NULL){
        remover_arvore(a->esq);
        remover_arvore(a->dir);
    }
}

```

```

        free(a);
    }
    return NULL;
}

int main (void){
    FILE* arq;
    arq=fopen("arvore.txt","rt");
    if (arq==NULL){
        printf ("erro ao abrir o arquivo");
        exit (1);
    }
    int x=1,y=0,w=0;
    Arvore *a=NULL;
    while (x!=8){
        system("cls");
        printf ("Digite a opcao desejada\n\n");
        printf ("1 - Ler a arvore de um arquivo\n");
        printf ("2 - Imprimir a arvore\n");
        printf ("3 - Calcular a altura da arvore\n");
        printf ("4 - Verificar se um elemento existe na arvore\n");
        printf ("5 - Contar o numero nos na arvore\n");
        printf ("6 - Alterar um o conteudo de um no da arvore\n");
        printf ("7 - Sair\n");
        scanf ("%d",&x);
        system("cls");
        switch (x)
        {
            case 1:
            {
                a=Ler_arvore_arq(arq);
                printf ("A arvore foi lida com sucesso\n");
                system("pause");
                break;
            }
            case 2:{
                printf ("Digite a opcao desejada\n\n");
                printf ("1 - Para imprimir em pre-ordem\n");
                printf ("2 - Para imprimir em ordem\n");
                printf ("3 - Para imprimir em pos-ordem\n");
                scanf ("%d",&y);
                system("cls");
                switch (y){
                    case 1:{
                        printf ("Impressao em pre-ordem\n");
                        imprimir_preordem(a);
                        printf (" \n");
                        system("pause");
                        break;
                    }
                    case 2:{
                        printf ("Impressao em ordem\n");
                        imprimir_ordem(a);
                        printf (" \n");

```

```

        system("pause");
        break;
    }
    case 3:{
        printf ("Impressao em pos-ordem\n");
        imprimir_posordem(a);
        printf (" \n");
        system("pause");
        break;
    }
}
break;
}
case 3:{
    w=altura(a);
    printf("A altura da arvore eh %d\n", w);
    system("pause");
    break;
}
case 4:{
    printf ("Digite o elemento a ser verificado\n");
    scanf ("%d",&y);
    w=existe(a,y);
    if (w==1)
        printf ("Elemento existe\n");
    else
        printf ("Elemento nao existe\n");
    system("pause");
    break;
}
case 5:{
    w=numero_nos(a);
    printf ("O numero de nos da arvore eh %d\n",w);
    system("pause");
    break;
}
case 6:{
    printf ("Digite o elemento para ser alterado\n");
    scanf ("%d",&y);
    w=existe(a,y);
    if (w==1){
        printf ("Elemento encontrado\n");
        a=altera_elemento(a,y);

    }
    else
        printf ("Elemento nao existe\n");
    system("pause");

    break;
}
}

```

```
        case 7: {
            a=remover_arvore(a);
            x=8;
            break;
        }
    }
}
fclose(arq);
return 0;
}
```