

```
// IED-001 (Prof. Dr. Silvio do Lago Pereira)

// -----
// Exemplo 1
// -----

typedef int Item;
typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

// -----
// Exemplo 2
// -----

Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct arv));
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

// -----
// Exemplo 3
// -----

void exhibe(Arv A, int n) {
    if( A==NULL ) return;
    exhibe(A->dir, n+1);
    printf("%s%d\n", 3*n, "", A->item);
    exhibe(A->esq, n+1);
}

// -----
// Exemplo 4
// -----

void ins(Item x, Arv *A) {
    if( *A == NULL ) *A = arv(NULL, x, NULL);
    else if( x <= (*A)->item ) ins(x, &(*A)->esq);
    else ins(x, &(*A)->dir);
}

// -----
// Exercício 2
// -----

#include <stdio.h>
#include <stdlib.h>

...

int main(void) {
    Arv I = NULL;
    ins(7, &I);
    ins(4, &I);
    ins(6, &I);
    ins(8, &I);
    ins(2, &I);
    ins(5, &I);
    ins(3, &I);
}
```

```
    ins(1,&I);
    exibe(I,0);
    return 0;
}

// -----
// Exercício 3
// -----

#include <stdio.h>
#include <stdlib.h>

...

int main(void) {
    Arv I = NULL;
    Item x;
    puts("\nPara sair, digite um inteiro negativo!\n");
    while( 1 ) {
        printf("Item a ser inserido? ");
        scanf("%d",&x);
        if( x<0 ) break;
        ins(x,&I);
    }
    exibe(I,0);
    return 0;
}

// -----
// Exemplo 5
// -----

int busca(Item x, Arv A) {
    if( A == NULL ) return 0;
    if( x == A->item ) return 1;
    if( x < A->item ) return busca(x,A->esq);
    else return busca(x,A->dir);
}

// -----
// Exercício 4
// -----

#include <stdio.h>
#include <stdlib.h>

...

int main(void) {
    int v[9] = {71,43,64,92,80,27,58,35,16};
    Arv A = NULL;
    for(int i=0; i<9; i++) ins(v[i],&A);
    emordem(A);
    puts("\nPara sair, digite um inteiro negativo!");
    while( 1 ) {
        int x;
        printf("\nItem a ser buscado? ");
        scanf("%d",&x);
        if( x<0 ) break;
        if( busca(x,A) ) puts("Encontrado!");
        else puts("Inexistente!");
    }
    return 0;
}

// -----
```

```
// Exemplo 6
```

```
// -----  
  
Item remmax(Arv *A) {  
    if( *A == NULL ) abort();  
    if( (*A)->dir == NULL ) {  
        Arv n = *A;  
        Item x = n->item;  
        *A = n->esq;  
        free(n);  
        return x;  
    }  
    return remmax(&(*A)->dir);  
}
```

```
// -----  
// Exercício 5  
// -----
```

```
#include <stdio.h>  
#include <stdlib.h>
```

```
...
```

```
int main(void) {  
    int v[9] = {71,43,64,92,80,27,58,35,16};  
    Arv A = NULL;  
    for(int i=0; i<9; i++) ins(v[i],&A);  
    puts("Inicial");  
    exhibe(A,0);  
    for(int i=0; i<9; i++) {  
        puts("Depois de remover o maximo");  
        remmax(&A);  
        exhibe(A,0);  
        getchar();  
    }  
    return 0;  
}
```

```
// -----  
// Exemplo 7  
// -----
```

```
void remraiz(Arv *A) {  
    if( *A == NULL ) abort();  
    Arv n = *A;  
    if( n->esq == NULL ) *A = n->dir;  
    else if( n->dir == NULL ) *A = n->esq;  
    else n->item = remmax(&n->esq);  
    if( n != *A ) free(n);  
}
```

```
// -----  
// Exemplo 8  
// -----
```

```
void rem(Item x, Arv *A) {  
    if( *A == NULL ) return;  
    if( x == (*A)->item ) remraiz(A);  
    else if( x < (*A)->item ) rem(x,&(*A)->esq);  
    else rem(x,&(*A)->dir);  
}
```

```
// -----  
// Exercício 7
```

```
// -----  
  
#include <stdio.h>  
#include <stdlib.h>  
  
...  
  
int main(void) {  
    int v[9] = {71,43,64,92,80,27,58,35,16};  
    Arv A = NULL;  
    for(int i=0; i<9; i++) ins(v[i],&A);  
    puts("Inicial");  
    exhibe(A,0);  
    for(int i=0; i<9; i++) {  
        printf("Depois de remover o item %d\n",v[i]);  
        rem(v[i],&A);  
        exhibe(A,0);  
        getchar();  
    }  
    return 0;  
}
```