

```

uses newdelay, crt;

const n=2;
type tree=^no;
    no=record
        Info:array[1..n*2+1] of integer;
        Pos:array[1..n*2+1] of integer;
        Lig:array[1..n*2+2] of tree;
        tl:integer;
    end;

procedure Cria_Caixa(var no:tree);
var i:integer;
begin
    new(no);
    no^.tl:=0;
    for i:=1 to 2*n+1 do
    begin
        no^.Info[i]:=0;
        no^.Pos[i]:=0;
        no^.Lig[i]:=nil;
    end;
    no^.Lig[2*n+2]:=nil;
end;

procedure navega_ate_folha(raiz:tree;info:integer;var p:tree);
var i:integer;
begin
    {navegar ate no folha}
    p:=raiz;
    while p^.Lig[1]<>nil do
    begin
        i:=1;
        while (i<=p^.tl) and (info>p^.Info[i]) do
            i:=i+1;
        p:=p^.Lig[i];
    end;
end;

function procura_posicao(no:tree;info:integer):integer;
var i:integer;
begin
    i:=1;
    while (i<=no^.tl) and (info>no^.Info[i]) do
        i:=i+1;
    procura_posicao:=i;
end;

procedure remaneja(no:tree;pos:integer);
var i:integer;
begin
    no^.Lig[no^.tl+2]:=no^.Lig[no^.tl+1];
    for i:=no^.tl+1 downto pos+1 do
    begin
        no^.Info[i]:=no^.Info[i-1];
        no^.Pos[i]:=no^.Pos[i-1];
        no^.Lig[i]:=no^.Lig[i-1];
    end;
end;

```

```

function localizaPai(raiz:tree;folha:tree;info:integer):tree;
var p,pai:tree;
    i:integer;
begin
    p:=raiz;
    pai:=p;
    while p<>folha do
    begin
        i:=1;
        while (i<=p^.tl) and (info>p^.Info[i]) do
            i:=i+1;
        pai:=p;
        p:=p^.Lig[i];
    end;
    localizaPai:=pai;
end;

procedure split(var raiz:tree; folha,pai:tree; info:integer);
var cx1,cx2:tree;
    i,pos:integer;
begin
    Cria_Caixa(cx1);
    Cria_Caixa(cx2);
    for i:=1 to n do
    begin
        cx1^.Info[i]:=folha^.Info[i];
        cx1^.Pos[i]:=folha^.Pos[i];
        cx1^.Lig[i]:=folha^.Lig[i];
    end;
    cx1^.Lig[n+1]:=folha^.Lig[n+1];
    cx1^.tl:=n;

    for i:=n+2 to 2*n+1 do
    begin
        cx2^.Info[i-(n+1)]:=folha^.Info[i];
        cx2^.Pos[i-(n+1)]:=folha^.Pos[i];
        cx2^.Lig[i-(n+1)]:=folha^.Lig[i];
    end;
    cx2^.Lig[n+1]:=folha^.Lig[2*n+2];
    cx2^.tl:=n;

    if pai=folha then
    begin
        folha^.Info[1]:=folha^.Info[n+1];
        folha^.Pos[1]:=folha^.Pos[n+1];
        folha^.Lig[1]:=cx1;
        folha^.Lig[2]:=cx2;
        folha^.tl:=1;
    end
    else
    begin
        info:=folha^.Info[n+1];
        pos:=procura_posicao(pai,info);
        remaneja(pai,pos);
        pai^.tl:=pai^.tl+1;
        pai^.Info[pos]:=folha^.Info[n+1];
        pai^.Pos[pos]:=folha^.Pos[n+1];
        pai^.Lig[pos]:=cx1;
        pai^.Lig[pos+1]:=cx2;
        dispose(folha);
        if (pai^.tl>2*n) then {novo split}
        begin

```

```

        folha:=pai;
        info:=folha^.Info[n+1];
        pai:=localizaPai(raiz,pai,info);
        split(raiz,folha,pai,info);
    end;
end;
end;

procedure insere_arvore(var raiz:tree; info,posArq:integer);
var folha,pai:tree;
    i,pos:integer;
begin
    if raiz=nil then
    begin
        Cria_Caixa(raiz);
        raiz^.tl:=1;
        raiz^.Info[1]:=info;
        raiz^.Pos[1]:=posArq;
    end
    else
    begin
        {navegar ate no folha}
        navega_ate_folha(raiz,info,folha);
        {...}
        pos:=procura_posicao(folha,info);
        remaneja(folha,pos);
        folha^.tl:=folha^.tl+1;
        folha^.Info[pos]:=info;
        folha^.Pos[pos]:=posArq;
        if (folha^.tl>2*n) then {split}
        begin
            pai:=localizaPai(raiz,folha,info);
            split(raiz,folha,pai,info);
        end;
    end;
end;

procedure In_Ordem(raiz:tree);
var i:integer;
begin
    if raiz<>nil then
    begin
        for i:=1 to raiz^.tl do
        begin
            In_Ordem(raiz^.Lig[i]);
            write(raiz^.Info[i], ' ');
        end;
        i:=i+1;
        In_Ordem(raiz^.Lig[i]);
    end;
end;

```