

```

unit Drzewo_DW;

interface

uses
    Crt;

type
    TDane = integer;
    PDrzewoDW = ^TDrzewoDW;
    TDrzewoDW = record
        dane : TDane;
        dane2 : integer;
        wskaznik_l : PDrzewoDW;
        wskaznik_p : PDrzewoDW
    end;

    Ttablica = array[1..5000] of TDane;

var
    skladnik_biezacy, pierwszy_skladnik : PDrzewoDW;
    t : Ttablica;

    procedure CreateTreeDW(l, p:integer; var t:Ttablica; var
skladnik_biezacy:PDrzewoDW;
        zmp:integer; var pierwszy_skladnik:PDrzewoDW);

    procedure DeleteTreeDW(l, p:integer; var skladnik_biezacy:PDrzewoDW);

    procedure PrintTreeDW(l, p:integer; skladnik_biezacy:PDrzewoDW; var zmp:integer;
zmp2:integer);

    procedure SearchComponentTreeDW(dane:TDane; var skladnik_biezacy:PDrzewoDW;
        pierwszy_skladnik:PDrzewoDW);

implementation

    procedure CreateTreeDW(l, p:integer; var t:Ttablica; var
skladnik_biezacy:PDrzewoDW;
zmp:integer; var pierwszy_skladnik:PDrzewoDW);
var
    i : integer;
begin
    if l <= p then

```

```

begin
  Inc(zmp);
  i := (l + p) div 2;
  New(skladnik_biezacy);
  if zmp = 1 then
    pierwszy_skladnik := skladnik_biezacy;
    skladnik_biezacy^.dane := t[i];
    skladnik_biezacy^.dane2 := i;
    CreateTreeDW(l, i - 1, t, skladnik_biezacy^.wskaznik_l, zmp, pierwszy_skladnik);
    CreateTreeDW(i + 1, p, t,
skladnik_biezacy^.wskaznik_p, zmp, pierwszy_skladnik);
  end
  else
    skladnik_biezacy := nil
  end;

  procedure DeleteTreeDW(l, p:integer; var skladnik_biezacy:PDrzewoDW);
var
  i : integer;
begin
  if l <= p then
    begin
      i := (l + p) div 2;

      DeleteTreeDW(l, i - 1, skladnik_biezacy^.wskaznik_l);
      DeleteTreeDW(i + 1, p, skladnik_biezacy^.wskaznik_p);

      Dispose(skladnik_biezacy)
    end
  end;

  procedure PrintTreeDW(l, p:integer; skladnik_biezacy:PDrzewoDW; var zmp:integer;
zmp2:integer);
var
  i : integer;
begin
  if l <= p then
    begin
      Inc(zmp);

      i := (l + p) div 2;

      if zmp = 25 then
        begin
          ClrScr;
          zmp := 1
        end;
      GotoXY(zmp2, zmp);
      WriteLn(skladnik_biezacy^.dane);
      PrintTreeDW(l, i - 1, skladnik_biezacy^.wskaznik_l, zmp, zmp2 + 1);
      PrintTreeDW(i + 1, p, skladnik_biezacy^.wskaznik_p, zmp, zmp2 + 1)
    end
  end;

```

end;

procedure SearchComponentTreeDW(dane:TDane; var skladnik_biezacy:PDrzewoDW;

pierwszy_skladnik:PDrzewoDW);

begin

end;

begin

 skladnik_biezacy := nil;

 pierwszy_skladnik := nil

end.