

Bektore bat ordenatzeko definitu genituen bertsioetatik abiatuz aldatu azpiprograma horiek lista bat ordena dezaten. Hauek dira bektoreak eta lista definitzeko erazagupenak:

```
Max : constant Integer := 10;
type Osoko_Bektore is array (1..Max) of Integer;
```

```
Max : constant Integer := 15;
subtype Osoko0_Max is Integer range 0 .. Max;
subtype Osoko1_Max is Integer range 1 .. Max;
type Taula is array (Osoko1_Max) of Integer;
type Lista_Estatiko is record
    Info : Taula;
    Zenbat : Osoko0_Max;
end record;
```

Bektoreko ordenazioa, *Burbuila* deritzon algoritmoa jarraituz

Osoko-bektore bat emanda, bektoreko osagaiak *Burbuila* deritzon algoritmoa jarraituz ordenatuko dituen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

```
procedure Ordenatu_Burbuila (B: in out Osoko_Bektore) is
-- Aurrebaldintza:
-- Postbaldintza: Bko elementuak ordenatuta daude
-- txikienetik handienera
begin
    for I in reverse B'First+1 .. B'Last loop
        Korritu_Elementuak_Handienak_Jasotzen (B, B'First, I);
        -- B(I..N) bektoreko osagaiak ordenatuta daude
        -- eta B bektoreko handienak dira
    end loop ;
end Ordenatu_Burbuila ;

procedure Korritu_Elementuak_Handienak_Jasotzen
    (B: in out Osoko_Bektore;
     I_Hasiera, I_Bukaera: in Integer) is
begin
    for I in I_Hasiera .. I_Bukaera - 1 loop
        if B(I) > B(I+1) then Balioak_Trukatu (B(I), B(I+1)) ;
        end if ;
    end loop;
end Korritu_Elementuak_Handienak_Jasotzen ;
```

Bektoreko ordenazioa, Hautaketa algoritmoa jarraituz

N elementu dituen osoko-bektore bat emanda, bektoreko osagaiak Txertaketa deritzon algoritmoa jarraituz ordenatuko dituen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

```
procedure Ordenatu (B: in out Osoko_Bektore) is
-- Aurrebaldintza:
-- Postbaldintza: B bektoreko lehenengo N elementuak
--                   ordenatuta daude txikienetik handienara
begin
  for I in B'First .. B'Last loop
    Bilatu_Minimoaren_Posizioa (B, I, B'Last, Pos);
    Balioak_Trukatu (B(I), B(Pos)) ;
    -- B(1) ... B(I) bektoreko osagaiak ordenatuta daude
    -- eta B bektoreko txikienak dira
  end loop ;
end Ordenatu ;
```

```
procedure Bilatu_Minimoaren_Posizioa
  (B: in Osokoen_Bektore;
   I1, I2: in Integer;
   Minimoaren_Posizioa: out Integer) is
-- Aurrebaldintza:
-- I1 eta I2 B bektorearen indize posibleak dira.
-- Postbaldintza: B(I1) ... B(I2) balioen arteko txikiena
-- B(Minimoaren_Posizioa) da.

begin
  Minimoa :=Integer'Last ;
  for I in I1 .. I2 loop
    if B(I) < Minimoa then
      Minimoa := B(I);
      Minimoaren_Posizioa := I;
    end if ;
  end loop;
end Bilatu_Minimoaren_Posizioa ;
```

Bektoreko ordenazioa, *Burbuila* deritzon algoritmoa jarraituz

Osoko-bektore bat emanda, bektoreko osagaiak *Burbuila* deritzon algoritmoa jarraituz ordenatuko dituen algoritmoa espezifikatu eta egin. Azpiprograma modura implementatu.

```
procedure Ordenatu_Burbuila (L: in out Lista) is
-- Aurrebaldintza:
-- Postbaldintza: L listako elementuak ordenatuta daude
-- txikienetik handienara
begin
  for I in reverse 1 .. L.Zenbat loop
    Korritu_Elementuak_Handienak_Jasotzen (L.Info, 1, I);
    -- L.Info(I..N) bektoreko osagaiak ordenatuta daude
    -- eta L listako balio handienak dira
  end loop ;
end Ordenatu_Burbuila ;

procedure Korritu_Elementuak_Handienak_Jasotzen
(B: in out Taula;
 I_Hasiera, I_Bukaera: in Integer) is
begin
  for I in I_Hasiera .. I_Bukaera - 1 loop
    if B(I) > B(I+1) then Balioak_Trukatu (B(I), B(I+1)) ;
    end if ;
  end loop;
end Korritu_Elementuak_Handienak_Jasotzen ;
```

Bektoreko ordenazioa, Hautaketa algoritmoa jarraituz

N elementu dituen osoko-bektore bat emanda, bektoreko osagaiak Txertaketa deritzon algoritmoa jarraituz ordenatuko dituen algoritmoa espezifikatu eta egin. Azpiprograma modura implementatu.

```
procedure Ordenatu (L: in out Lista) is
-- Aurrebaldintza:
-- Postbaldintza: L listako elementuak
--                  ordenatuta daude txikienetik handienera
begin
  for I in 1 .. L.Zenbat-1 loop
    Bilatu_Minimoaren_Posizioa (L.Info, I, L.Zenbat, Pos);
    Balioak_Trukatu (L.Info(I), L.Info(Pos)) ;
    -- L.Info(1) ... L.Iinfo(I) bektoreko osagaiak ordenatuta daude
    -- eta L listako txikienak dira
  end loop ;
end Ordenatu ;
```

```
procedure Bilatu_Minimoaren_Posizioa
  (B: in Taula;
   I1, I2: in Integer;
   Minimoaren_Posizioa: out Integer) is
-- Aurrebaldintza:
-- I1 eta I2 B bektorearen indize posibleak dira.
-- Postbaldintza: B(I1) ... B(I2) balioen arteko txikiena
-- B(Minimoaren_Posizioa) da.
begin
  Minimoa :=Integer'Last ;
  for I in I1 .. I2 loop
    if B(I) < Minimoa then
      Minimoa := B(I);
      Minimoaren_Posizioa := I;
    end if ;
  end loop;
end Bilatu_Minimoaren_Posizioa ;
```