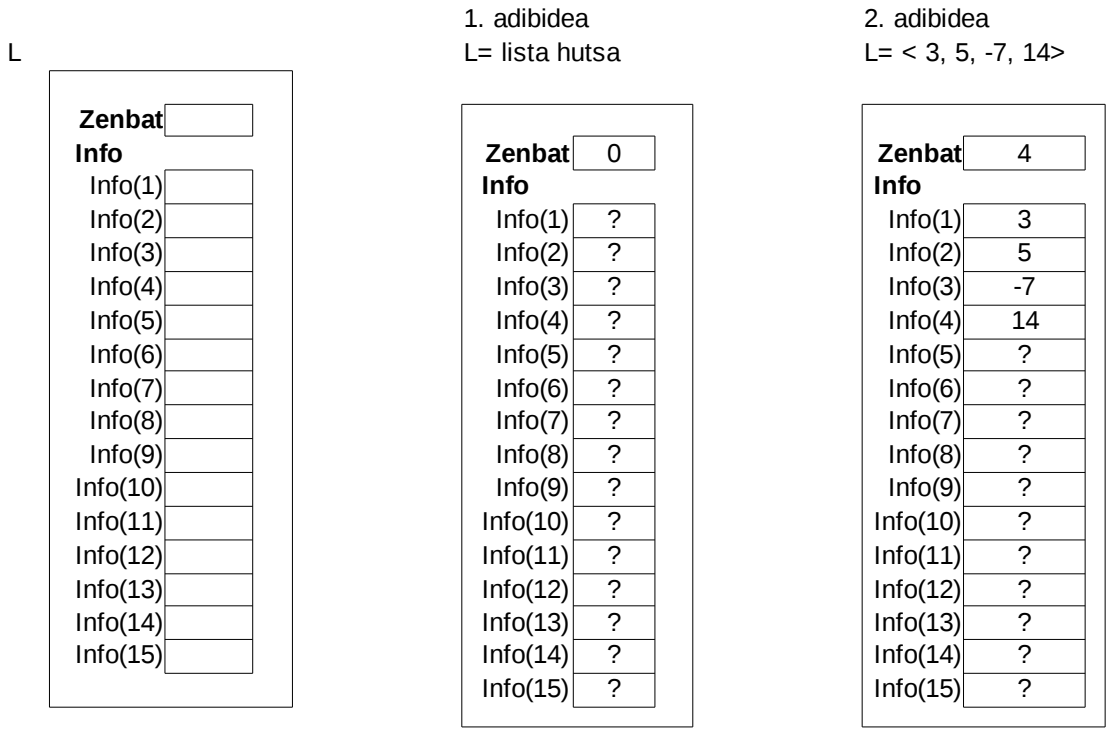
Lista datu-mota (IV. GAIA)

Listetan bektoreetan bezala mota bereko hainbat osagai biltzen ditugu. Baina listetan bektoreetan ez bezala osagai kopurua alda daiteke programa martxan dugunean. Exekuzio garaian listan elementu berriak sartuko ditugu, edo elementuak kendu ere. Lista errepresentatzeko modu bat honako eazagupenekin egin daiteke:

```
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 is record
   Info : Taula;
   Zenbat : Osoko0_Max;
end record;
```



Programatzaileak erne ibiliko beharko da, eta berak kontrolatu beharko ditu zenbat osagai dauden une bakoitzean (L.Zenbat eremua), eta listan dauden osagaiak taulako (L.Info bektoreko) lehen posizioetan kokatu, ordena egokian. L.Zenbat posiziotik aurrera egongo diren balioak ez dira kontuan hartzen, edozein izan ahal izango dira.

1. Lista idatzi

Elementu gisa osokoak dituen zerrenda bat emanda, zerrendako elementuak inprimatuko dituen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

```
procedure Lista_Idatzi (L: in Lista) is
-- Aurrebaldintza: B1 bektorean Osagai_kop osagai daude
-- Postbaldintza: Sek (Osokoen Sekuentzia
                  non B1 bektorearen osagaiak dauden.
begin
   for I in 1 .. L.Zenbat loop
      Idatzi_Osokoa (L.Info(I)) ;
   end loop;
end Lista_Idatzi;
```

2. Listako elementu minimoa eta bere posizioa

L lista emanda, zenbaki minimoa eta bere posizioa bilatzeko algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

```
procedure Bilatu_Maximoa (L            : in Lista ;
                          Maximoa, Pos : out Integer)
--Aurre: listak gutxienez elementu bat dauka
--Post: Maximoa parametroak L listako balio maximoa dauka.
--      Pos delakoak L listako maximoaren posizioa adierazten du.
begin
   Maximoa := L.info(1);
   Pos:= 1;
   for I in  2..L.Zenbat loop
      if  L.Info(I)> Maximoa  then
         Maximoa :=  L.info(I);
         Pos := I;
      end if;
   end loop;
 end  Bilatu_Maximoa;
```

3. Zenbakia bilatu lista ez-ordenatuan

L lista ez-ordenatu bat emanda eta Z zenbakia, Z bilatzeko algoritmoa espezifikatu eta egin. Zenbakia listan badago, lehenengo agerpenaren posizioa itzuli beharko da; eta bestela, ez dagoenean, zero itzuli beharko da. Azpiprograma modura inplementatu.

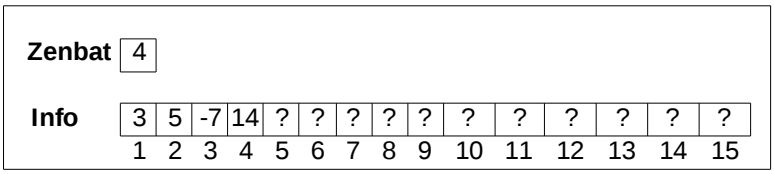
```
function Bilatu_Posizioa (L : in Lista ;
                          X : in Osokoa)  return  integer is
       -- Post: Emaitza-k X zenbakia L listako
       --       zenbatgarren posizioan dagoen adierazten du;
       --        X zenbakia  N osagaien artean ez badago,0 itzuliko da
begin
   I := 1;
   Aurkitua := False ;
   while  I<= L.Zenbat and not Aurkitua loop
      if  X = L.Info(I) then
         Aurkitua := True;
      else
          I := I + 1;
      end if;
   end loop;
   -- (Aurkitua =False)
   --  edo (Aurkitua = True eta X=B(I))
   if Aurkitua then
      Emaitza :=  I;
   else  Emaitza :=  0;
   end if;
   return Emaitza;
end  Bilatu_Posizioa;
```

4. Txertatu zerrendaren hasieran

Elementu gisa osokoak dituen zerrenda bat eta elementua emanda, zerrendako lehenengo zenbakiaren aurretik emandako elementua gehituko duen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

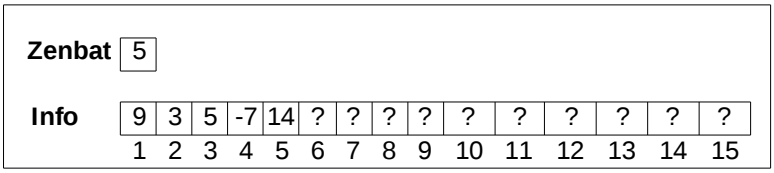
Adibidez:

L= < 3, 5, -7, 14>




```
Txertatu_Hasieran (L, 9);
```

L= < 9, 3, 5, -7, 14>



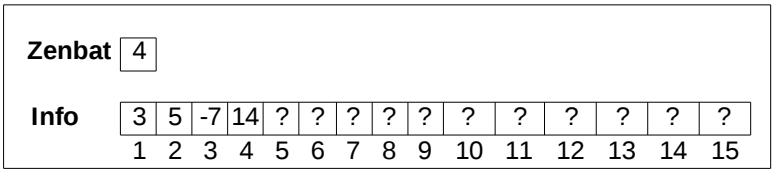

```
procedure Txertatu_Hasieran (L : in out Lista ;
                             X : in Osokoa) is
  -- Post: L listako hasieran X osagaia sartu da
begin
   Desplazatu_Eskuinaldera (1, L.Zenbat);
   L.Zenbat := L.Zenbat + 1;
   L.Info(1) := X;
end  Txertatu_Hasieran ;

   procedure Desplazatu_Eskuinaldera  (L    : in out Lista;
                                       I1,I2 : in     integer) is
     --Aurre:
     --Post : LL listako [I1..I2] indize-tarteko osagaiak posizio bat
     --       desplazatu dira eskuinaldera [I1+1..I2+1] tarteko osagaietara.
   begin
      for I in reverse I1..I2 loop
         L.Info(I+1) := L.Info(I);
      end loop;
   end Desplazatu_Eskuinaldera;
```

5. Txertatu zerrendaren bukaeran

Osoko bat eta elementu gisa osokoak dituen zerrenda bat emanda, zerrendako azkeneko zenbakiaren atzetik emandako osokoa gehituko duen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

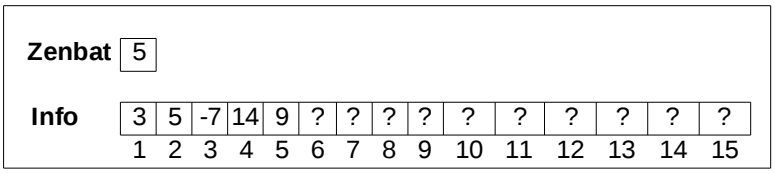
Adibidez:

L= < 3, 5, -7, 14>




```
Txertatu_Bukaeran (L, 9);
```

L= < 3, 5, -7, 14, 9>



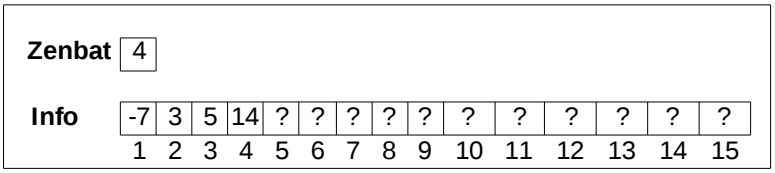

```
procedure Txertatu_Bukaeran (L : in out Lista ;
                             X : in Osokoa) is
  -- Post: L listako hasieran X osagaia sartu da
hasiera
   L.Zenbat := L.Zenbat + 1;
   L.Info(L.Zenbat) := X;
end  Txertatu_Bukaeran ;
```

6. Txertatu zerrenda ordenatuan

Elementu gisa osokoak dituen zerrenda ordenatu bat eta osoko bat emanda, zenbakia zerrendako dagokion tokian (ordenaren arabera) txertatuko duen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

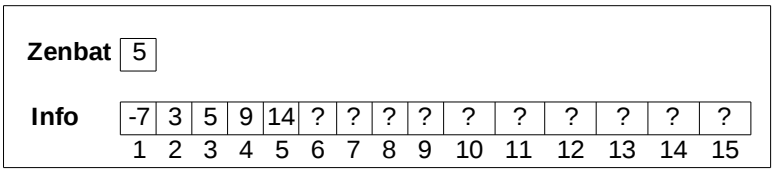
Adibidez:

L= < -7. 3, 5, 14>




```
Txertatu_Ordenatuan (L, 9);
```

L= < -7, 3, 5, 9, 14>




```
procedure  Txertatu_Ordenatuan  (L      : in out Lista;
                                 X      : in     Integer) is
   --Aurre : L listako osagaiak ordenatuta daude txikienetik handienera.
   --Post:   L listan X zenbakia gehitu da dagokion tokian, horrela orain ere
   --        L listako osagaiak kodearen arabera ordenatuta daude
    Posizioa: Osoko0_Max;
 begin
    Bilatu_Posizioa_Ordenatuan (L, X, Posizioa);
    Desplazatu_Eskuinaldera (L, Posizioa, L.Zenbat);
    L.Info(Posizioa) := L;
    L.Zenbat := L.Zenbat +1;
 end Txertatu_Ordenatuan;

 procedure  Bilatu_Posizioa_Ordenatuan  (L      : in     Lista;
                                         X      : in     Integer;
                                         Pos    :    out Integer) is
    -- Post: Aurkitua=True
    --        eta X < L.Info(Pos)
    --        eta [1..L.Pos] tarteko i guztietarako X >L.Info(i)
    --       Aurkitua=False
    --        eta Pos = L.Zenbat+1
    --        eta listako zenbaki guztiak X baino txikiagoak dira
    I : Integer;
    Aurkitua: Boolean;
 begin
    I := 1;
    Aurkitua := False;
    while   I<= L.Zenbat and not Aurkitua loop
```

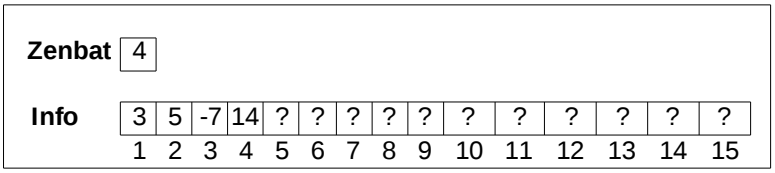
```
      if X < L.Info(I) then
         Aurkitua := True ;
      else
         I := I + 1;
      end if;
   end loop;
   -- (Aurkitua =True eta X < L.Info(I))
   --  edo (Aurkitua = False
   --        eta listako zenbaki guztiak X baino txikiagoak dira)
   if Aurkitua then
      Pos := I;
   else
      Pos := L.Zenbat + 1;
   end if;
end Bilatu_Posizioa_Ordenatuan;
```

7. Ezabatu lehenengo agerpena

Elementu gisa osokoak dituen zerrenda bat eta elementu bat emanda, elementu hori listatik ezabatuko duen algoritmoa espezifikatu eta egin. Parametroa batean itzuliko da ea elementua kendu den. Azpiprograma modura inplementatu. Suposatu zerrendan ez direla balioak errepikatzen.

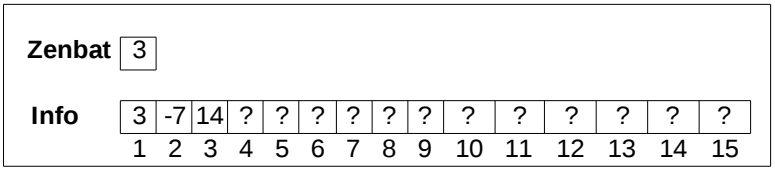
Adibidez:

L= < 3, 5, -7, 14>




```
Ezabatu (L, 5);
```

L= < 3, -7, 14>




```
procedure  Ezabatu (L       : in out Lista;
                    X       : in     Integer
                    Kenduta:    out Boolean) is
   --Aurre : L listako osagaiak ordenatuta daude txikienetik handienera.
   --Post:   L listatik X zenbakia ezabatu da eta Kenduta=True
   --        edo X ez zegoen listan eta Kenduta=False
    I: Osoko1_Max;
begin
    I := 1;
    Aurkitua := False;
    while   I<= L.Zenbat and not Aurkitua loop
       if X = L.Info(I) then
          Aurkitua := True;
       else
          I := I + 1;
       end if;
    end loop;
   -- (Aurkitua =True eta X = L.Info(I))
   --  edo (Aurkitua = False eta X ez dago listan)
    if Aurkitua then
       Desplazatu_Ezkerraldera (L, I+1, L.Zenbat);
       L.Zenbat := L.Zenbat -1;
       Kenduta := True;
    else
        Kenduta := False;
    end if;

end Ezabatu;
```

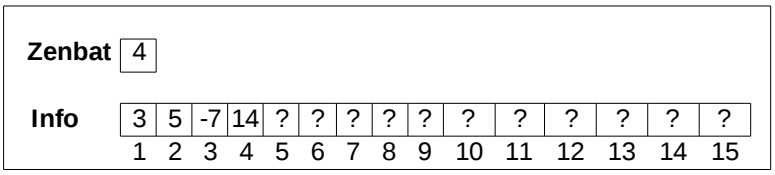
```
   procedure Desplazatu_Ezkerraldera  (L     : in out Lista;
                                       I1,I2 : in     Integer) is
      --Aurre:
      --Post : L listako [I1..I2] indize-tarteko osagaiak posizio bat
      --       desplazatu dira ezkerraldera [I-1..I2-1] tarteko
osagaietara.
   begin
      for I in I1..I2 loop
         L.Info(I-1) := L.Info(I);
      end loop;
   end Desplazatu_Ezkerraldera;
```

8. Ezabatu lehenengo osagaia

Elementu gisa osokoak dituen zerrenda bat emanda, lehenengo elementua zerrendatik ezabatuko duen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu. Suposatu zerrendan gutxienez osagai bat dagoela.

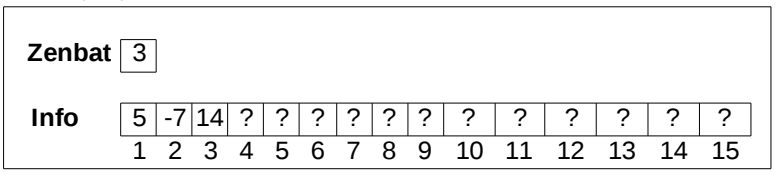
Adibidez:

L= < 3, 5, -7, 14>



Ezabatu_Lehenengoa (L, 5);

L= < 3, -7, 14>



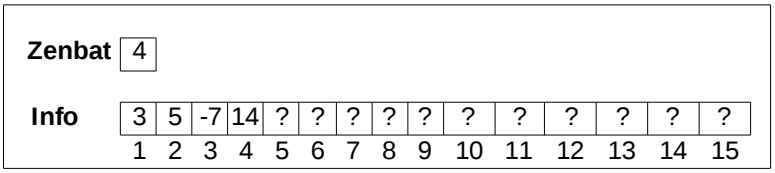

```
   procedure  Ezabatu_Lehenengoa  (L : in out Lista) is
      --Aurre : L listan gutxienez zenbaki bat dago.
      --Post:   L listatik lehen osagaia ezabatu da.
       I: Osoko1_Max;
   begin
      Desplazatu_Ezkerraldera (L, 2, L.Zenbat);
      L.Zenbat := L.Zenbat -1;
   end Ezabatu_Lehenengoa;
```

9. Ezabatu azkeneko osagaia

Elementu gisa osokoak dituen zerrenda bat emanda, azkeneko elementua zerrendatik ezabatuko duen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu. Suposatu zerrendan gutxienez osagai bat dagoela.

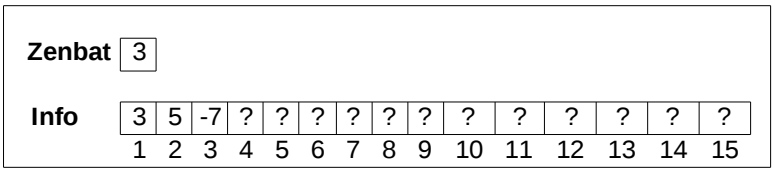
Adibidez:

L= < 3, 5, -7, 14>




```
Ezabatu_Azkenekoa (L, 9);
```

L= < 3, 5, -7>



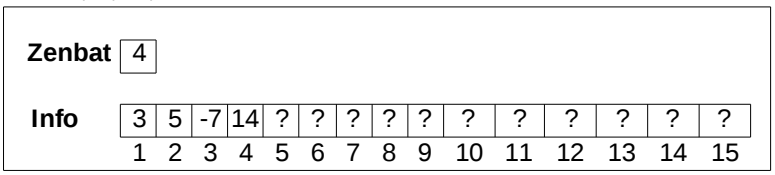

```
procedure  Ezabatu_Azkenekoa  (L : in out Lista) is
   --Aurre : L listan gutxienez zenbaki bat dago.
   --Post:   L listatik azken osagaia ezabatu da.
    I: Osoko1_Max;
begin
   L.Zenbat := L.Zenbat -1;
end Ezabatu_Azkenekoa;
```

10. Listaren ordenazioa

L lista bat emanda, bektoreko osagaiak Burbuila deritzon algoritmoa jarraituz ordenatuko dituen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

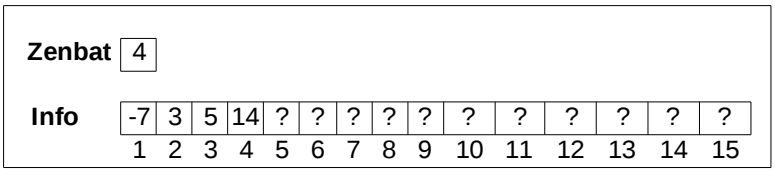
Adibidez:

L= < 3, 5, -7, 14>




```
Ordenatu(L);
```

L= < -7, 3, 5, 14>




```
procedure Ordenatu  (L: in out Lista) is
   -- Aurrebaldintza:
   -- Postbaldintza:  L listako elementuak
   --                ordenatuta daude txikienetik handienera
   Pos : Integer;
begin
   for I in  1.. L.Zenbat-1 loop
      Bilatu_Kode_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_Kodez ;



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) osagaien artekoen txikiena
   --                  B(Minimoaren_Posizioa)  da.
   Minimoa: Integer;
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 ;


procedure Balioak_Trukatu (X1, X2 : in out Integer) is
   Lag : Integer;
begin
   Lag := X1;
   X1 := X2;
   X2 := Lag;
end Balioak_Trukatu;
```