

5 Gaia. Datu egitura dinamikoak: erakusleak

eman la zabal zazu

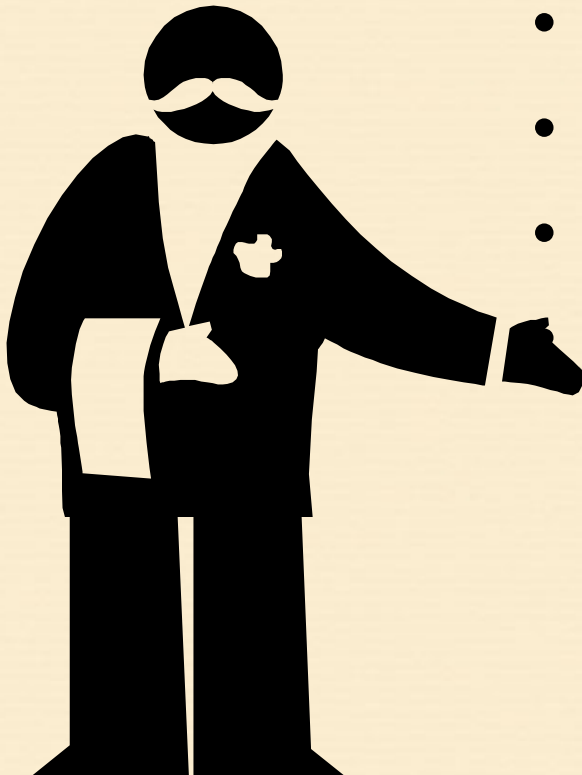


Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Aurkibidea

- **Gaiaren helburuak**
- Motibazioa
- Datu egitura dinamikoak: erakusleak
- Oinarrizko eragiketak
- Zerrenda estekatuak vs Arrayak
- Adibideak eta ariketak



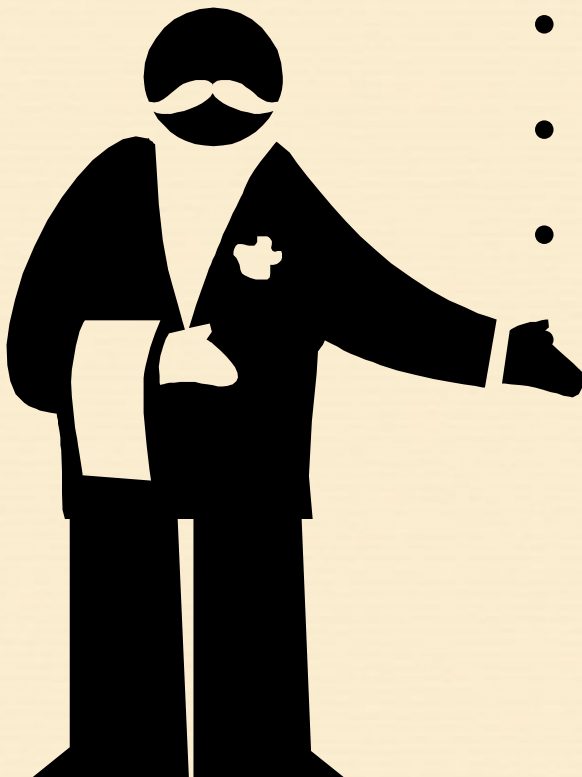
Gaiaren helburuak

- Datu egitura dinamikoak aurkeztea (erakusleak)
- Euren abantailak ezagutzea egitura estatikoen aurrean
- Oinarrizko eragiketetarako erabiltzea
- Array-en alternatiba gisa zerrenda estekatuak ezagutzea



Aurkibidea

- Gaiaren helburuak
- **Motibazioa**
- Datu egitura dinamikoak: erakusleak
- Oinarrizko eragiketak
- Zerrenda estekatuak vs Arrayak
- Adibideak eta ariketak



Motibazioa

- Memorian horrenbeste leku erreserbatu gabe berdina lor dezakegu?

Zenbat
3

```
type Osokoen_bektore is array (1..2000) of Integer;  
type Osokoen_zerrenda is record  
  Zenbat: Integer;  
  Balioak: Osokoen_bektore;  
end record;
```

1	2	3	4	5	...	199 7	199 8	199 9	200 0
10	7	2	¿?	¿?	...	¿?	¿?	¿?	¿?



Motibazioa

- Nola gehitu ahalko genuke elementu berri bat zerrendan?

Zenbat
2000



1	2	3	4	5	...	199 7	199 8	199 9	200 0
10	7	2	11	8	...	-2	0	4	9

Erreserba estatikoa vs dinamikoa

- Memoriaren erreserba estatikoa
 - Konpilazio unean egiten da programan erazagututako aldagaiarentzat
- Memoriaren erreserba dinamikoa
 - Exekuzio garaian egiten da programaren agindu baten ondorioz



Zerrenda estekatuak

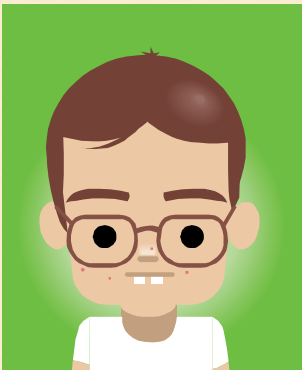
- Erakusleek memoria dinamikoki erreserbatzen dute
 - Definitutako zerrendek beharrezko elementuen tokia hartzen dute (ez gehio ez gutxiago)
 - Gainera, zerrenda hauek ez dute aurredefinitutako tamaina maximorik izango



Motibazioa

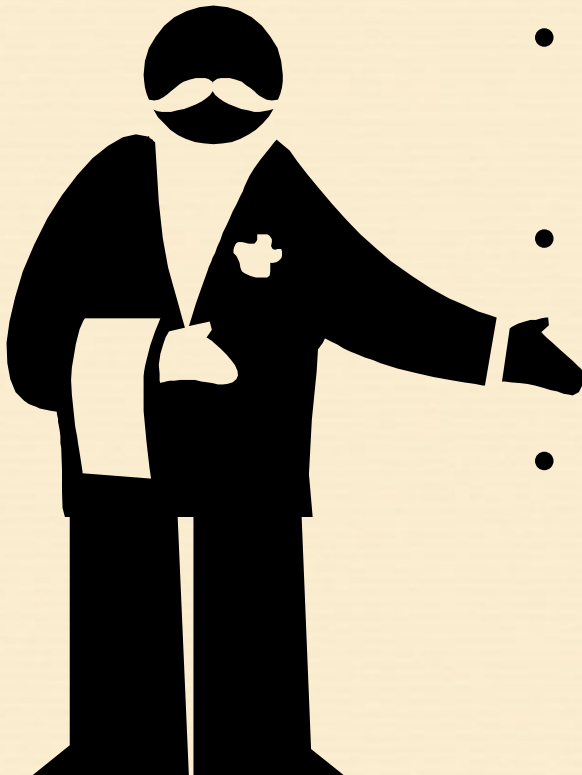
- Gai honetan ADA programazio lengoaia bakarrik landuko dugu
 - Pythonen zerrenda dinamikoak PMOO ikasgaian Javarekin erabiliko ditugun modu oso antzekoan erabiltzen dira

Kontzu, javako ArrayList-ak estatikoak dira, baina tamaina aldakorra dute oso modu gardenean



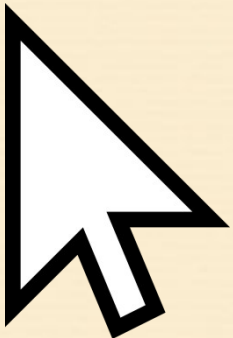
Aurkibidea

- Gaiaren helburuak
- Motibazioa
- **Datu egitura dinamikoak: erakusleak**
- Oinarrizko eragiketak
- Zerrenda estekatuak vs Arrayak
- Adibideak eta ariketak



Erakusleak

- access datu mota
 - erakusten (apuntatzen) duen memoria helbidea gordetzen du
 - null balio berezia ere jaso dezake, eta kasu horretan erakusleak ez du ezer erakusten



Motaren definizioa

- access hitz erreserbatua erabiltzen da eta jarraian erakutsi nahi den datu mota adierazten da

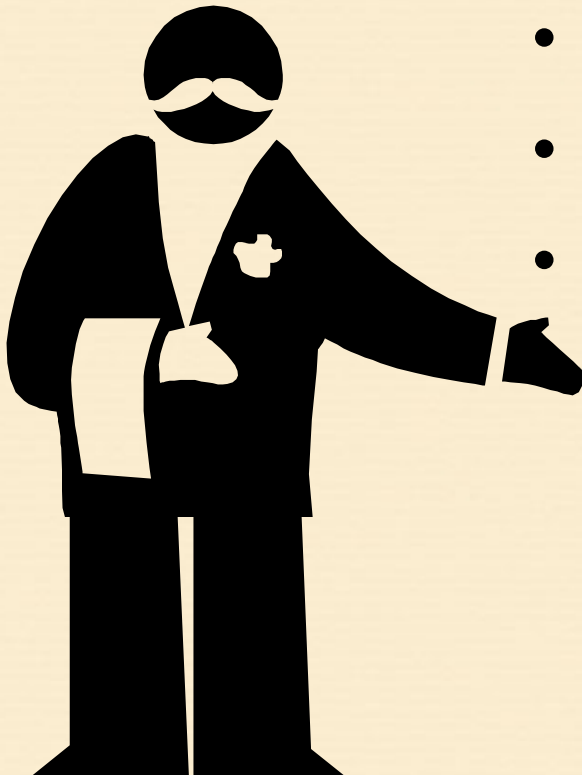
```
type Pertsona is record
  NAN: Integer;
  Hiria: String(1..10);
end record;

type Pertsona_erakuslea is access Pertsona;
```



Aurkibidea

- Gaiaren helburuak
- Motibazioa
- Datu egitura dinamikoak: erakusleak
- **Oinarrizko eragiketak**
- Zerrenda estekatuak vs Arrayak
- Adibideak eta ariketak



Aldagaien erazagupena

- Erakusle motako aldagai bat erazagutzean, gainerako erazagupenetan bezala, memori erreserbatzen da
 - Erakusleak defektuz, null-era hasieratzen dira

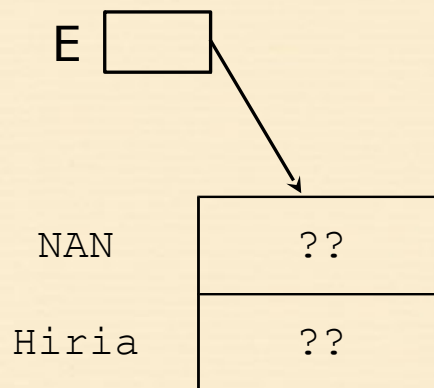
E: Pertsona_Erakuslea



Elementu berri baten esleipena

- *new* hitz erreserbatua erabiltzen da, eta honako funtzioak ditu:
 - erakusten duen elementu berriarentzat memoria erreserbatu modu dinamikoan
 - erreserbatutako memoriaren helbidea itzuli, zeina aldagaiari esleituko zaion

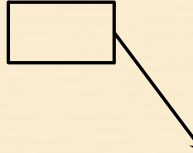
E: `Pertsona_Erakuslea`



Erakusten den elementua atzitzea

```
P1, P2 : Pertsona_Erakuslea;  
--erreserba estatikoa  
  
begin  
  
  P1 := new Pertsona;  
  --erreserba dinamikoa  
  
  P1.NAN := 12345;  
  P1.Hiria := "Bilbo    ";
```

P2 

P1 

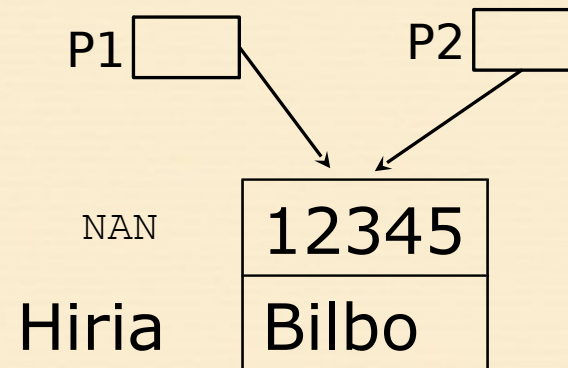
NAN	12345
Hiria	Bilbo



Erakusleen arteko esleipena

- P2-k P1-k erakusten duen elementu berdina erakusten du
 - P2-k aurretik erakusten zuen elementuaren atzipena galtzen da (ez bazeaio beste aldagai bati esleitu)

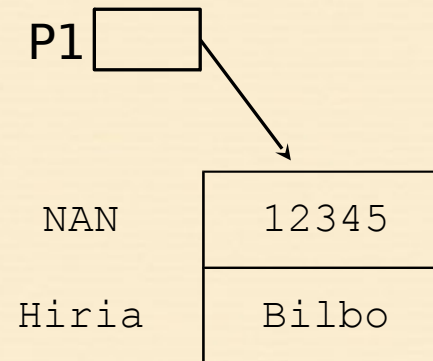
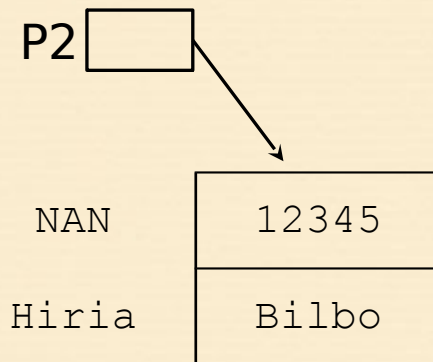
P2 := P1;



Kopia baten esleipena

- P2-k bere elementu propioa erakusten jarraitzen du, P1-en balioetatik kopiatu direnak

```
P2 := new Pertsona;  
--erreserba dinamikoa  
  
P2.all := P1;
```



Erakutsitako elementua eta balioa

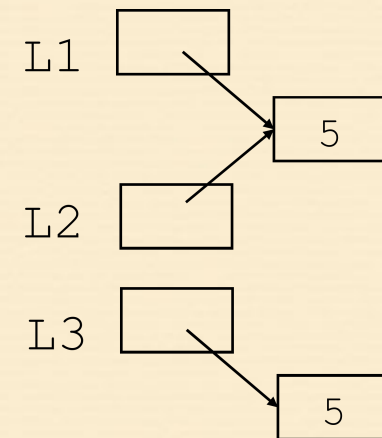
```
type Erakuslea_Osokora is access Integer;  
...
```

```
L1,L2,L3 : Erakuslea_Osokora;
```

```
L1 := new Integer;  
L1.all := 5;
```

```
L2 := L1;
```

```
L3 := new Integer;  
L3.all := 5;
```



- * $L1 = L2 \neq L3$
- * $L1.all = L2.all = L3.all = 5$

Garrantzitsua

- *Null* elementua erakusten duen esteka bat atzitzen denean, *Constraint Error* bat gertatuko da

```
type Erakuslea_Stringera is access String;  
S1: Erakuslea_Stringera;  
  
if (S1.all = ...) then  
    ...
```

S1 

Proposatutako ariketa

- Hurrengo ekintzen exekuzioaren emaitza adierazi ezazu (marraztu)

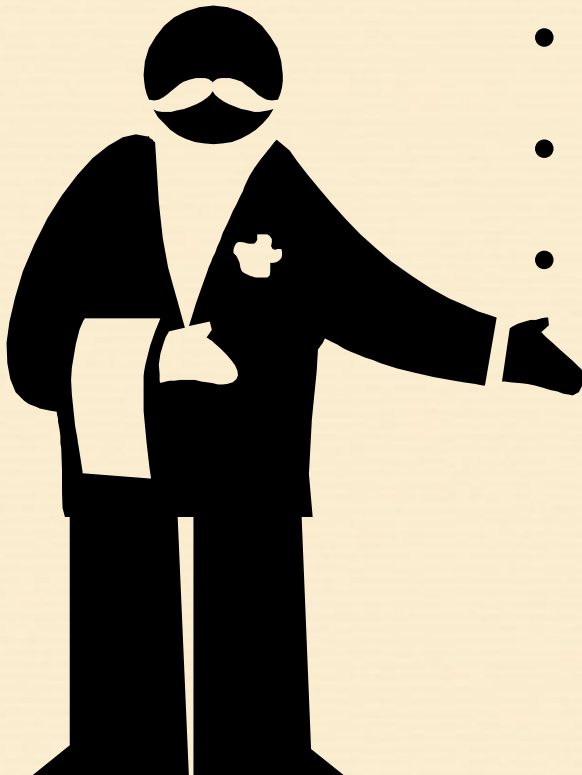
```
1. P, Q: Pertsona_Eraskuslea;  
   Elem : Pertsona;  
  
2. Elem.NAN := 3065;  
   Elem.Hiria := "Bilbo      ";  
   P := new Pertsona;  
  
3. P.NAN := 4567;  
   P.Hiria := "Donosti    ";  
   Q := P;
```

```
4. Q.NAN := 3075;  
   P.NAN := 1456;  
  
5. P.NAN := Elem.NAN;  
   P.Hiria := Elem.Hiria;  
  
6. Q := new Pertsona;  
   Q.NAN := 3076;  
   Q.Hiria := "Gasteiz    ";  
  
7. P := null;  
  
8. P := Q;
```



Aurkibidea

- Gaiaren helburuak
- Motibazioa
- Datu egitura dinamikoak: erakusleak
- Oinarrizko eragiketak
- **Zerrenda estekatuak vs Arrayak**
- Adibideak eta ariketak



Datu egitura estatikoak

- array-ek memoria modu estatikoan erreserbatzen dute elementu kopuru maximo batentzat
 - Konpilatu aurretik tamaina zehaztu beharko da, eta ezingo da exekuzioan zehar aldatu
 - Ez erabilitako tokia zabor elementuek beteko dute

Datu egitura dinamikoak

- Zerrenda estekatuak ez dute beharrezkoa baino gehiago okupatzen
- Ez dute aurrezehatutako tamaina finkorik
 - Exekuzioan zehar beharrezko tokia erreserbatzen da



Adibidea

- Zenbaki osokoen sekuentzia bat irakurri erabiltzaileak 0 sartzen duen arte, eta goranzko ordenean pantailaratu
 - type Osokoen_Bektorea is array (1..???) of Integer;

Zein tamaina maximo aukeratu beharko genuke?

100? --> Baliteke gutxiegia izatea
Integer'Last? --> Zabor elementuz
betetako memoria asko alferrik
erabiltzen egon gaitezke



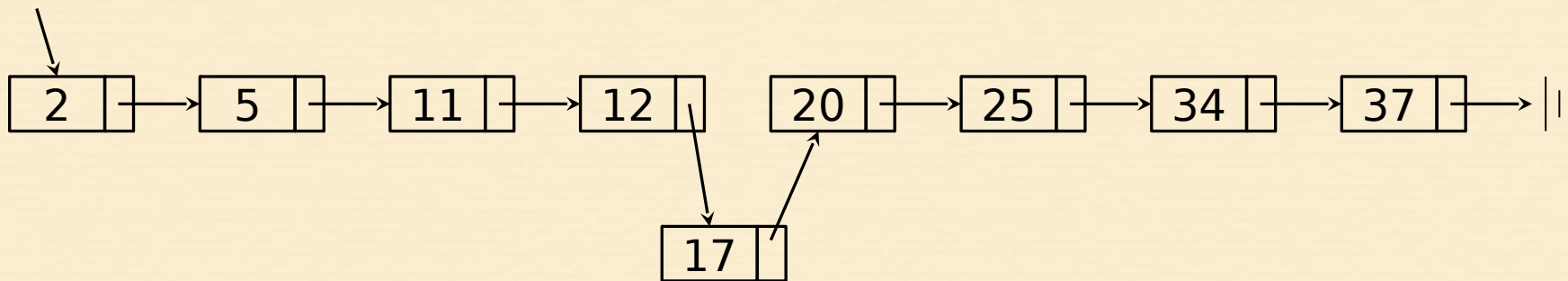
Adibidea

- Array bat erabiltzeak gainera, elementu bat txertatzean (edo ezabatzean) beste batzuk mugitu behar izaten da
 - Adiidez, 17 balioa ordenatuta sartu nahi da hurrengo array-an

1	2	3	4	5	6	7	8	...
2	5	11	12	20	25	34	37	

Adibidea

- Zerrenda estekatuek ez dute arazo hori
 - Elementu bat txertatzeko edo kentzeko ez da ezer mugitu beharko

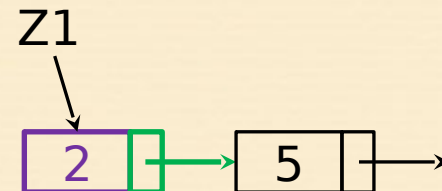


Zerrenda estekatuak

- Zerrenda estekatua erakusle moduan definituko da, eta eremuetako bat zerrenda estekatua bera izango da
 - Adibidea:

```
type Nodo; -- bukatugabeko definizioa  
  
type Zerrenda is access Nodo;  
  
type Nodo is record  
    Info: Integer;  
    Hurrengoa : Zerrenda;  
end record;
```

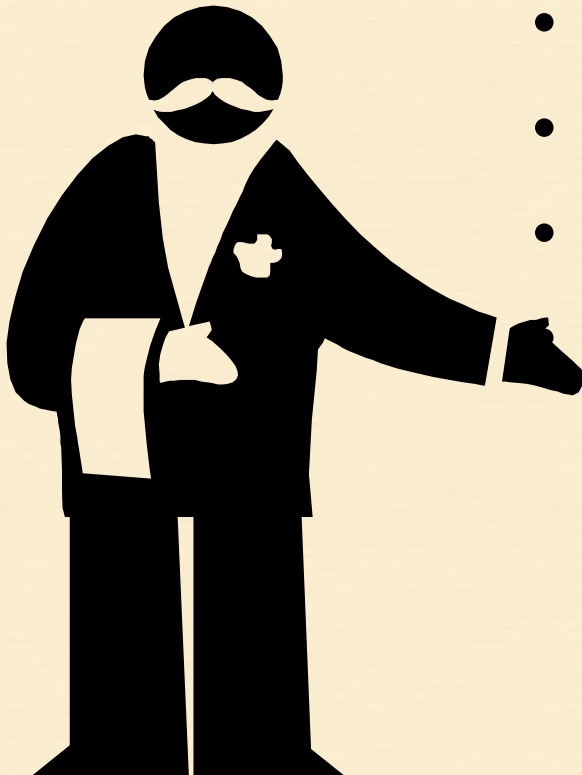
```
Z1 : Zerrenda;
```



Aurkibidea

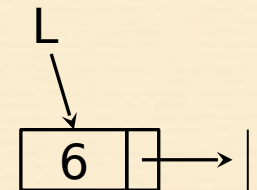
- Gaiaren helburuak
- Motibazioa
- Datu egitura dinamikoak: erakusleak
- Oinarrizko eragiketak
- Zerrenda estekatuak vs Arrayak

Adibideak eta ariketak



Zerrenda hutsa eta elementua gehitu

```
L : Lista:= null; -- Zerrenda hutsa  
  
...  
  
L:= new Nodo;  
L.Info := 6;  
L.Hurrengoa:= null; -- hautazkoa
```



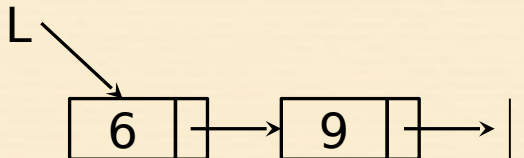
Bi elementutako zerrenda sortu

- Zein da ezberdintasuna?

```
Z, Berria: Zerrenda;  
...  
Z:=null;  
Berria := new Nodo;  
Berria.Info := 6;  
Berria.Hurrengoa := null;  
Z := Berria;  
  
Z.Hurrengoa := new Nodo;  
Z.Hurrengoa.Info := 9;  
Z.Hurrengoa.Hurrengoa := null;
```

```
Z, Berria: Zerrenda;  
...  
Z:=null;  
Berria := new Nodo;  
Berria.Info := 6;  
Berria.Hurrengoa := null;  
Z := Berria;
```

```
Beria := new Nodo;  
Berria.Info := 9;  
Berria.Hurrengoa := null;  
Z.Hurrengoa := Berria;
```



Edozein zerrenda sortu

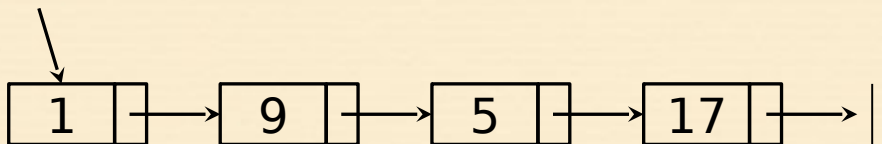
- Teklatutik elementuak eskatuko dira 0 irakurri arte

```
procedure zerrenda_orokorra_sortu (L: out Zerrenda) is
    Berria: Zerrenda:= null; --erreserba estatikoa
    Zen: Integer;
begin
    L:= null;
    get(Zen);
    loop exit when Zen = 0;
        Berria := new Nodo; --erreserba dinamikoa
        Berria.Info := Num;
        Berria.Hurrengoa := L; --txertatzea hasieran
        L := Berria;
        get(Zen);
    end loop;
end zerrenda_orokorra_sortu;
```



Zerrenda bat inprimatu

```
procedure zerrenda_inprimatu (Z: in Zerrenda) is
    Unekoa: Zerrenda;
begin
    Unekoa := Z;
    loop exit when Unekoa = null;
        put(Unekoa.Info);
        Unekoa := Unekoa.hurrengoa;
    end loop;
end zerrenda_inprimatu;
```



Bigarren elementua ezabatu

- Zergatik da Z parametroa *in* motakoa?

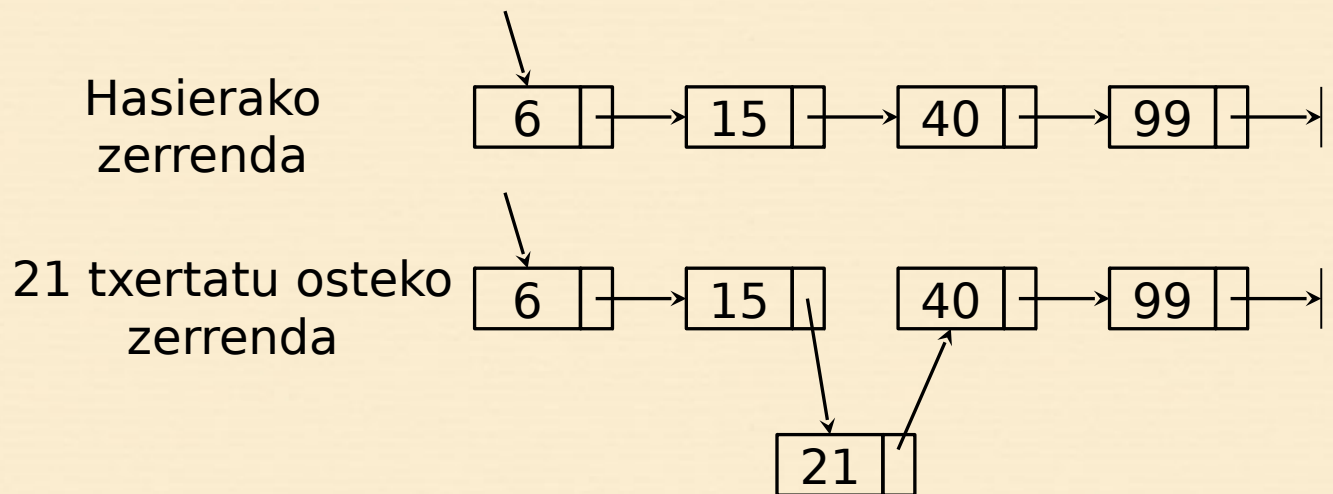
```
procedure bigarren_elementua_ezabatu (Z: in Zerrenda) is
begin
    Z.Hurrengoa := Z.Hurrengoa.Hurrengoa;
end bigarren_elementua_ezabatu;
```

```
procedure bigarren_elementua_ezabatu (Z: in Zerrenda) is
    Zaharra: Zerrenda;
begin
    Zaharra := Z.Hurrengoa;
    Z.Hurrengoa := Zaharra.Hurrengoa;
end bigarren_elementua_ezabatu;
```



Zerrenda ordenatuan txertatu

- Osokoen zerrenda ordenatu bat emanda (txikitik handienera) eta balio osoko bat, zerrendaren ordena mantentzeko toki egokian txertatuko da balio hori.



Soluzio posible bat

```
procedure ordenatuta_txertatu (Z: in out Zerrenda; N: in Integer) is
    Unekoa: Zerrenda:= Z;
begin
    while (Unekoa /= null) and then (Unekoa.Info < N) loop
        Unekoa:= Unekoa.Hurrengoa;
    end loop;
    ...
end;
```

- Honako kasuaren simulazioa egin dezagun:
txertatu 21 balioa { 6, 15, 40, 125} zerrendan
 - Zer gertatzen da?

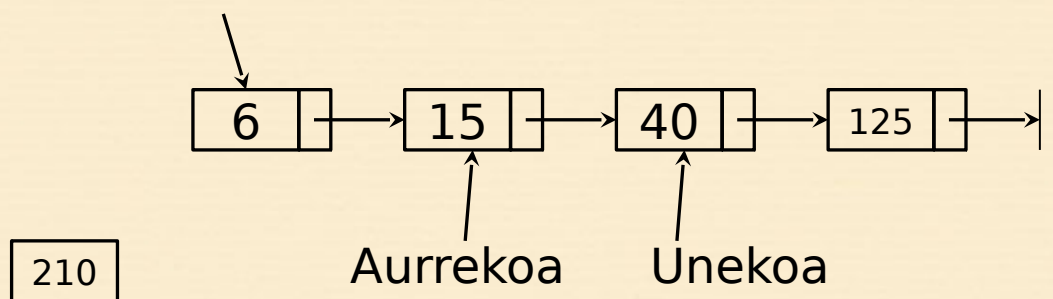
Beste soluzio posible bat

```
procedure ordenatuta_txertatu (Z: in out Zerrenda; N: in Integer) is
    Unekoa: Zerrenda:= Z;
begin
    while (Unekoa /= null) and then (Unekoa.Hurrengoa.Info < N) loop
        Unekoa:= Unekoa.Hurrengoa;
    end loop;
    ...
end;
```

- Beste honako kasuaren simulazioa egin dezagun:
txertatu 210 balioa { 6, 15, 40, 125} zerrendan
 - Zer gertatzen da orain?

Nola konpon dezakegu?

- Bi zerrenda laguntzaile erabiltzea da egokiena
 - Unekoa: hasierako zerrenda korritzeko erabiliko dugu (eta kokapena bilatu ahal izateko)
 - Aurrekoa: unekoaren aurreko elementua erakutsiko du (ez galtzeko, elementu berria horren ostean joango da eta)



Aurrekoa eta unekoa erabiliz

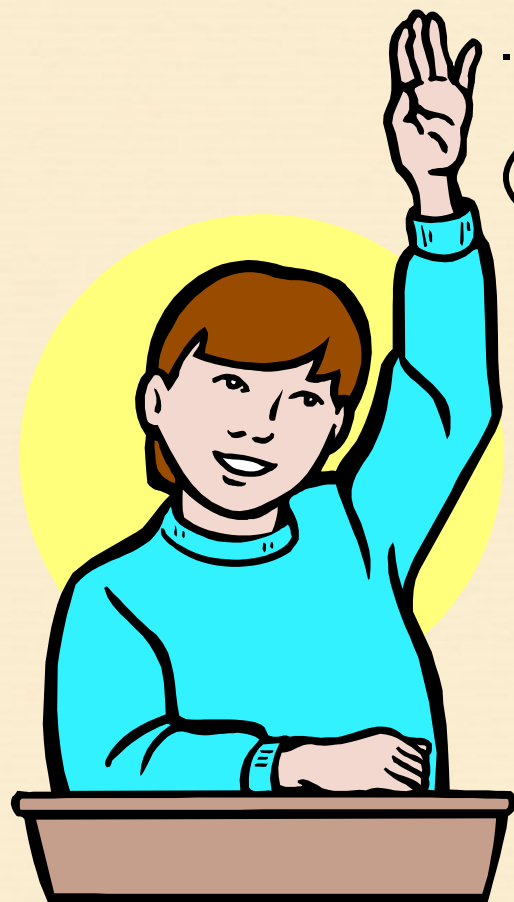
```
procedure ordenatuta_txertatu (L: in out Lista; N: in Integer) is
    Unekoa, Aurrekoa, Berria: Lista;
begin
    Aurrekoa:= null;
    Unekoa:= L;
    Berria:= new Nodo;
    Berria.Info:= N;
    while (Unekoa /= null) and then (Unekoa.Info < N) loop
        Aurrekoa:= Unekoa;
        Unekoa:= Unekoa.Hurrengoa;
    end loop;
    if Aurrekoa = null then - hasieran txertatzen da
        Berria.Hurrengoa:= L;
        L:=Berria;
    else -- Aurrekoa ez da null, Aurrekoaren eta Unekoaren artean txertatu
        Berria.Hurrengoa:= Unekoa;
        Aurrekoa.Hurrengoa:=Berria;
    end if;
end ordenatuta_txertatu;
```



Proposatutako ariketa

- Zerrenda estekatu bat emanda, bere elementuak handienetik txikienera ordenatuko dituen azpiprograma idatzi

```
procedure zerrenda_ordenatu (L: in out lista) is
--- Aurre:
--- Post: L-ren elementuak beherako ordenean daude
```



Galderarik?