

Ariketa 1:

Zerrenda bateko elementu errepikakorrak ezabatzen dituen programa garatu ezazu.

- 1) Sortu datu egitura;
- 2) Dago: Zerrenda bat eta posizio bat emanda, posizio horretan dagoen elementua posizio hori baina lehenagoko beste posizio batean agertzen bada TRUE itzuliko du, FALSE bestela.
- 3) Ezkerretara_desplazatu: Zerrenda eta posizio bat emanda, posizio horretatik aurrerako elementu guztiak ezkerretara desplazatuko ditu.
- 4) Zerrenda_pantailaratu: Zerrenda emanda, zerrenda hori pantailaratuko du.
- 5) Errepikatuak_ezabatu: Zerrenda bat emanda, zerrenda horretako elementu errepikakorrak ezabatuko ditu.

ADIB:

Zerrenda: 1 1 3

Zerrenda irteera: 1 3

Zerrenda: 1 3 3 5 7 7 7 8

Zerrenda irteera: 1 3 5 7 8

Ariketa 2: Mamu bokalak

Karakterez osatutako zerrenda bat emanik, pantailaratu itzazu zerrendako hitzak (lerro bakoitzean bat) non bokalak ez diren agertuko:

Adibidez:

	k	a	i	x	o			n	i
--	---	---	---	---	---	--	--	---	---

Irteera:

kx

n

Erabili beharreko datu egitura:

type karaktereen_zerrenda is array (Integer range <>) of Character;

Garatu beharreko azpiprogramak:

```
function/procedure bokala_da (...) ... is
--aurre: karaktere bat izango du sarrera bezala
--post: karakterea bokala bada orduan TRUE, bestela FALSE
```

```
function/procedure hitza_salto (...) is
-- Aurre: Sarrera zerrenda bat eta indize bat. Zerrendak hizkiak eta
hutsuneak ditu -- eta 1 <= indizea <= Z'Last + 1
-- Indizea <= Z'Last bada orduan posizio horretan hizki bat egongo da.
-- Post: Zerrendako hizkiak salto egingo ditu hutsune bat bilatu arte
eta posizio hori itzuliko du. Ez bada hutsunerik bilatzen orduan indizea
= Z'Last + 1 izango da
```

```

function/procedure hutsuneak_salto (...) is
    -- Aurre: Sarrera zerrenda bat eta indize bat. Zerrendak hizkiak eta
    hutsuneak ditu eta 1 <= indizea <= Z'Last + 1
    -- Post: Zerrendako hutsuneak salto egingo ditu hizki bat bilatu arte
    eta posizio hori itzuliko du.
    --      Ez bada hizkirik bilatzen orduan indizea = Z'Last + 1 izango
    da

function/procedure idatzi_zerrenda (...) is
    -- aurre: sarrera zerrenda bat izango da.
    -- post: zerrendako hizki guztiak pantailaratzen dira, letik Zren
    luzera bitarte

function/procedure bokalik_gabe_idatzi (...) is
    -- Aurre: Sarrera zerrenda bat izango da. Zerrendak hizkiak eta
    hutsuneak ditu
    -- Post: Zerrendako hitzak idatzi dira bokalak kenduta

```

Ariketa 3: Roland Garros

Roland Garroseko partidu bat errepresentatzeko datu egitura bat sortzeko eskatzen digute. Tennis partidu batetan 2 jokalarik daude, eta jokalarik bakoitzeko jolastu diren set-etan zenbat joku egin dituen gorde beharko dugu. Roland Garros-en partidu bakoitzak gehienez 5 set izan ditzake, eta partidu bat irabazteko 3 set irabazi behar dira. Jokalarik batek lehenengo hiru set-ak irabaziko balitu, orduan ez dira jolastuko set gehiago jada garaile bat dagoelako. Bestela 4 set jolastuko dira, eta oraindik ez bata ez besteak ez balitu 3 set irabaziko 5. Bat jolastu beharko litzateke.

Datu egitura beteta egongo da eta irabazlea zein izan den bilatu nahi dugu.

Lehenik pentsatu datu egitura bat informazio hori gordetzeko.

1. Aukera:

zenbat_sets

3

sets

1	2	1	1		
2	6	6	6		

2. Aukera:

zenbat_set	3
jokalarik	
	211
	666

3. Aukera:

zenbat_set	<input type="text" value="3"/>					
Jokalari1	<table><tr><td><input type="text" value="2"/></td><td><input type="text" value="1"/></td><td><input type="text" value="1"/></td><td><input type="text"/></td><td><input type="text"/></td></tr></table>	<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text"/>	<input type="text"/>		
Jokalari2	<table><tr><td><input type="text" value="6"/></td><td><input type="text" value="6"/></td><td><input type="text" value="6"/></td><td><input type="text"/></td><td><input type="text"/></td></tr></table>	<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text"/>	<input type="text"/>		

4. Aukera:

zenbat_set	<input type="text" value="3"/>					
set-ak	<table><tr><td><input type="text" value="2"/></td><td><input type="text" value="1"/></td><td><input type="text" value="1"/></td><td><input type="text"/></td><td><input type="text"/></td></tr></table>	<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text"/>	<input type="text"/>		

zenbat_set	<input type="text" value="3"/>					
set-ak	<table><tr><td><input type="text" value="6"/></td><td><input type="text" value="6"/></td><td><input type="text" value="1"/></td><td><input type="text"/></td><td><input type="text"/></td></tr></table>	<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text" value="1"/>	<input type="text"/>	<input type="text"/>
<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text" value="1"/>	<input type="text"/>	<input type="text"/>		

Behin datu egitura pentsatu ondoren, bete datuak programa nagusian eta honako azpiprogramak garatu:

```
procedure/function bat_inprimatu(...) is
  ---pre: Gutxienez 3 sets. Sarrera bezala Partiduak eta jokalaria
  zebakia izango dira.
  ---post: jokalaria baten set-ak inprimatuko dira

procedure/function biak_inprimatu(...) is
  ---pre:Gutxienez 3 sets. Sarrera bezala partiduak jasoko dira
  ---post:Jokalari bakoitzerako irabazitako set bakoitza inprimatuko da
  --- jokalaria_baten_jokoak_inprimatu azpiprogramari deituko dio

procedure/function topatu_irabazlea (...) is
  ---pre: jokalaria batek 3 set irabazi ditu
  ---post: irabazi duen jokalariaren zenbakia (1 edo 2) itzuliko dugu
```