

Zerrendak.hs

```

module Zerrendak where
import Ez_errek

-- Zerrenda bat emanda, zerrendako lehenengo elementua itzultzen duen funtzioa.
-- Zerrenda hutsa baldin bada, errore-mezua itzuliko du.

leh:: [t] -> t
leh [] = error "Zerrenda hutsa. Ez dago lehenengo elementurik."
leh (x:s) = x

-- Zerrenda bat emanda, zerrendako lehenengo elementua kenduz lortzen den
zerrenda
-- itzultzen duen funtzioa.
-- Zerrenda hutsa baldin bada, errore-mezua itzuliko du.

hond:: [t] -> [t]
hond [] = error "Zerrenda hutsa. Ez dago hondarrik."
hond (x:s) = s

-- Zerrenda bat emanda, zerrenda hutsa al den ala ez erabakitzen duen funtzioa.

hutsa_da:: [t] -> Bool
hutsa_da [] = True
hutsa_da (x:s) = False

-- Elementu bat eta zerrenda bat emanda, elementua zerrendan agertzen al den ala
ez
-- erabakitzen duen funtzioa.
-- t motako elementuek konparagarriak izan behar dute (Eq t ==>)

badago:: Eq t => t -> [t] -> Bool
badago x [] = False
badago x (y:s)
    | x == y      = True
    | x /= y      = badago x s

-- Zerrenda bat emanda, zerrendako elementu-kopurua kalkulatzen duen funtzioa.

luzera:: [t] -> Int
luzera [] = 0
luzera (x:r) = 1 + luzera r

-- Zerrenda bat emanda, zerrendako elementu bikoitien kopurua kalkulatzen duen
funtzioa.

bikop:: [Int] -> Int
bikop [] = 0
bikop (x:r)
    | bikoitia(x) = 1 + bikop r
    | bakoitia(x) = bikop r

-- Bi zerrenda emanda, zerrenda bietako elementuak tartekatuz lortzen den
zerrenda
-- kalkulatzen duen funtzioa.
-- Hasteko lehenengo zerrendatik hartu behar da.
-- Zerrenda biek luzera bera ez badute, errore-mezua aurkeztuko da.

tartekatu:: [t] -> [t] -> [t]
tartekatu [] s
    | luzera s /= 0 = error "Luzera desberdineko zerrendak."
    | otherwise     = []
tartekatu (x:r) s

```

```

                                Zerrendak.hs
| luzera (x:r) /= luzera s      = error "Luzera desberdineko zerrendak."
| otherwise                     = x:((leh s): (tartekatu r (hond s)))

-- Bi zerrenda emanda, zerrenda bietako elementuak tartekatuz lortzen den
zerrenda
-- kalkulatzeko funtzioa.
-- Hasteko bigarrenko zerrendatik hartu behar da.
-- Zerrenda biek luzera bera ez badute, errore-mezua aurkeztuko da.

tartekatu2:: [t] -> [t] -> [t]
tartekatu2 [] s
| luzera s /= 0 = error "Luzera desberdineko zerrendak."
| otherwise    = []
tartekatu2 (x:r) s
| luzera (x:r) /= luzera s      = error "Luzera desberdineko zerrendak."
| otherwise                     = (leh s):(x:(tartekatu2 r (hond s)))

--Zerrenda bat emanda, posizio bikoitietako elementuak kentzen dituen funtzioa
pos_bik_kendu:: [t] -> [t]
pos_bik_kendu [] = []
pos_bik_kendu (x:s)
| hutsa_da s      = x:[]
| otherwise        = x:(pos_bik_kendu (hond s))

--Zerrenda bat emanda, posizio bakoitietako elementuak kentzen dituen funtzioa
pos_bak_kendu:: [t] -> [t]
pos_bak_kendu [] = []
pos_bak_kendu (x:s)
| hutsa_da s      = []
| otherwise        = (leh s):(pos_bak_kendu (hond s))

```