```
module Zerrendak where

-----------------------------------------------------------------

import Ez_errek

{- Ez_errek.hs moduluan dauden "bikoitia" eta "bakoitia" funtzioak
   erabili nahi direlako "bikop" izeneko funtzioa definitzean.
-}

-----------------------------------------------------------------

-- "leh" funtzioa: Zerrenda bat emanda, zerrendako lehenengo
-- elementua itzuliko 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

-----------------------------------------------------------------

-- "hond" funtzioa: Zerrenda bat emanda, zerrendako lehenengo
-- elementua kenduz lortuko den zerrenda itzuliko duen funtzioa.
-- Zerrenda hutsa baldin bada, errore-mezua itzuliko du.

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

-----------------------------------------------------------------

-- "hutsa_da2 funtzioa: Zerrenda bat emanda, zerrenda hutsa al den
-- ala ez erabakiko duen funtzioa.

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

-----------------------------------------------------------------

-- "badago" funtzioa: Elementu bat eta zerrenda bat emanda,
-- elementua zerrendan agertzen al den ala ez erabakiko 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

-----------------------------------------------------------------

-- "luzera" funtzioa: Zerrenda bat emanda, zerrendako
-- elementu-kopurua kalkulatuko duen funtzioa.

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

-----------------------------------------------------------------

-- "bikop" funtzioa: Zerrenda bat emanda, zerrendako elementu
-- bikoitien kopurua kalkulatuko duen funtzioa.

bikop:: [Int] -> Int
bikop [] = 0
bikop (x:r)
```

```
        | bikoitia(x)   = 1 + bikop r
        | bakoitia(x)   = bikop r

-----------------------------------------------------------------

-- "tartekatu" funtzioa: Bi zerrenda emanda, zerrenda bietako
-- elementuak tartekatuz lortuko den zerrenda kalkulatuko duen funtzioa.
-- Hasteko lehenengo zerrendatik hartu behar da.
-- Zerrenda biek luzera bera ez badute, errore-mezua aurkeztuko du.

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

-----------------------------------------------------------------

-- "tartekatu2" funtzioa: Bi zerrenda emanda, zerrenda bietako
-- elementuak tartekatuz lortuko den zerrenda kalkulatuko duen funtzioa.
-- Hasteko bigarrengo zerrendatik hartu behar da.
-- Zerrenda biek luzera bera ez badute, errore-mezua aurkeztuko du.

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)))

-----------------------------------------------------------------

-- "pos_bik_kendu" funtzioa: Zerrenda bat emanda, posizio bikoitietako
-- elementuak kenduz lortuko den zerrenda itzuliko duen 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))

-----------------------------------------------------------------

-- "pos_bak_kendu" funtzioa: Zerrenda bat emanda, posizio bakoitietako
-- elementuak kenduz lortuko den zerrenda itzuliko duen 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))

-----------------------------------------------------------------
```