

## Sarrera\_irteera\_2.hs

```
module Sarrera_irteera_2 where
import Data.List
import Zerrenda_eraketa

-- F ataleko ariketen soluzioak
-----

--1

-- Erabiltzaileari osoa eta positiboa den zenbaki bat eskatu, zenbakia jaso eta
-- zenbaki horren zatitzaileen zerrenda aurkeztuko duen funtzioa.
-- Programaren exekuzioa bukatzen denean, programa bukatu dela esanez
-- mezu bat ere aurkeztuko da.
-- Erabiltzaileak teklatutako zenbakia positiboa ez bada, zenbakia ez dela
-- egokia esanez mezu bat aurkeztu eta exekuzioa bukatu egingo da.

zatitzaileak_sil :: IO ()
zatitzaileak_sil = do putStrLn "Positiboa den zenbaki oso bat idatzi:"
                     z <- getLine           -- z String motakoa da
                     let zenb = (read z :: Integer) -- zenb Integer motakoa da
                     if zenb > 0
                     then do putStr "Zatitzaileen zerrenda: "
                              print (zatizer_ze zenb)
                     else do putStrLn "Idatzitako zenbakia ez da positiboa."
                           putStrLn "Exekuzioaren bukaera."

-----

--2

-- Zenbaki positibo bat lortu arte zenbakia eskatzeko prozesua errepikatuko
-- duen funtzioa. Zenbaki positibo bat eskuratzen denean, zenbaki hori itzuliko
-- da beste funtzio batetik jasotzeko moduan.

positiboa_eskatu :: IO Integer
positiboa_eskatu = do putStrLn "Positiboa den zenbaki oso bat idatzi:"
                     z <- getLine
                     let zenb = (read z :: Integer)
                     if zenb > 0
                     then (return zenb)
                     else do putStrLn "Idatzitako zenbakia ez da positiboa."
                           positiboa_eskatu

-----

-- Erabiltzaileari osoa eta positiboa den zenbaki bat eskatu, zenbakia
-- jaso eta zenbaki hori eta zenbaki horren zatitzaileen zerrenda aurkeztuko
-- dituen funtzioa.
```

```

                                Sarrera_irteera_2.hs
-- Programaren exekuzioa bukatzen denean, programa bukatu dela esanez mezu
-- bat ere aurkeztuko da.
-- Zenbaki positiboa lortzeko "positiboa_eskatu" izeneko funtzioa erabiliko da.

zatitzaileak_si2 :: IO ()
zatitzaileak_si2 = do zenb <- positiboa_eskatu
                    let r = zatizer_ze zenb
                    putStr (show zenb)    -- Ez da lerroz aldatuko
                    putStr " zenbakiaren zatitzaileen zerrenda: "
                    print r                -- Zerrenda aurkeztu ondoren lerroz aldatuko da
                    putStrLn "Exekuzioaren bukaera."

-----

--3

-- Erabiltzaileari beste zenbakiren baten zatitzaileak nahi al dituen galdetzeko,
-- galdera burutu, erabiltzailearen erantzuna jaso eta erantzuna b edo e izan al den
-- ziurtatzeaz arduratuko den funtzioa.
-- Erabiltzaileak teklatutakoa b edo e ez bada, erantzuna egokia ez dela esanez
-- mezu bat aurkeztu eta berriro galdetuko da, b edo e lortu arte.

zat_gehiago :: IO Char
zat_gehiago = do putStrLn "Beste zenbakiren baten zatitzaileak nahi al dituzu? (b/e): "
                besterik <- getLine
                let erantzuna = head besterik
                if erantzuna `elem` "be"
                then (return erantzuna)
                else do
                    putStrLn "Erantzuna ez da egokia..."
                    zat_gehiago

-----

-- Erabiltzaileari osoa eta positiboa den zenbaki bat eskatu, zenbakia
-- jaso eta zenbaki hori eta zenbaki horren zatitzaileen zerrenda aurkeztu eta beste
-- zenbakiren baten zatitzaileen zerrenda kalkulatu nahi al den galdetuko duen
-- funtzioa.
-- Beste zenbakiren baten zatitzaileak kalkulatu nahi al diren galdetutakoan,
-- erabiltzaileak b edo e (bai edo ez) erantzun beharko du.
-- Erabiltzaileak e erantzuten badu, programaren exekuzioa bukatu dela esanez
-- mezu bat aurkeztuko da.
-- Erabiltzaileak b erantzuten badu, beste zenbaki positibo bat eskatu eta prozesu
-- osoa behin eta berriz burutuko da, erabiltzaileak bukatzea aukeratu arte.

-- Zenbaki positiboa lortzeko "positiboa_eskatu" izeneko funtzioa erabiliko da.
-- Beste zenbakiren baten zatitzaileak nahi al diren galdetzeko "zat_gehiago"

```

## Sarrera\_irteera\_2.hs

-- izeneko funtzioa erabiliko da.

```
zatitzaileak_si3 :: IO ()
zatitzaileak_si3 = do zenb <- positiboa_eskatu
                    let r = zatizer_ze zenb
                    putStr (show zenb)
                    putStr " zenbakiaren zatitzaileen zerrenda: "
                    print r
                    gh <- zat_gehiago
                    if gh == 'b'
                    then zatitzaileak_si3
                    else putStrLn "Exekuzioaren bukaera."
```

-----

--4

-- Erabiltzaileari zenbakiren baten zatitzaileak nahi al dituen galdetzeko,  
-- galdera burutu, erabiltzailearen erantzuna jaso eta erantzuna b edo e  
-- izan al den ziurtatzeaz arduratuko den funtzioa.  
-- Erabiltzaileak teklatutakoa b edo e ez bada, erantzuna egokia ez dela esanez  
-- mezu bat aurkeztu eta berriro galdetuko da, b edo e lortu arte.

```
zat_nahi :: IO Char
zat_nahi = do putStrLn "Zenbakiren baten zatitzaileen zerrenda nahi al duzu? (b/e): "
              q <- getLine
              let erantzuna = head q
              if erantzuna `elem` "be"
              then (return erantzuna)
              else do
                  putStrLn "Erantzuna ez da egokia..."
                  zat_nahi
```

-----

-- Erabiltzaileari zenbakiren baten zatitzaileen zerrenda kalkulatzeko  
-- nahi al duen galdetuz hasten den funtzioa. Zenbakiren  
-- baten zatitzaileen zerrenda kalkulatzeko nahi al duen galdetutakoan,  
-- erabiltzaileak b edo e (bai edo ez) erantzun beharko du.  
-- Erabiltzaileak e erantzuten badu, programaren exekuzioa bukatu dela esanez  
-- mezu bat aurkeztu beharko da.  
-- Erabiltzaileak b erantzuten badu, zenbaki positibo bat eskatu, zenbaki hori eta  
-- bere zatitzaileen zerrenda aurkeztu eta prozesu osoa behin eta berriz burutuko  
-- da, erabiltzaileak bukatzea aukeratu arte.

-- Zenbaki positiboa lortzeko "positiboa\_eskatu" izeneko funtzioa erabiliko da.  
-- Zenbakiren baten zatitzaileak nahi al diren galdetzeko "zat\_nahi"

## Sarrera\_irteera\_2.hs

-- izeneko funtzioa erabiliko da.

```
zatitzaileak_si4 :: IO ()
zatitzaileak_si4 = do qd <- zat_nahi
  if qd == 'b'
  then do
    zenb <- positiboa_eskatu
    let r = zatizer_ze zenb
    putStr (show zenb)
    putStr " zenbakiaren zatitzaileen zerrenda: "
    print r
    zatitzaileak_si4
  else putStrLn "Exekuzioaren bukaera."
```

-----