

```
module LKSA_2015_11_10 where
import Data.List
```

```
-----
-- MURGILKETA
-----
```

```
zatitu_lag :: Integer -> [Integer] -> [Integer] -> [[Integer]]
zatitu_lag x s a
    | (null s) = [a]
    | (x == (head s)) = a:(zatitu_lag x (tail s) [])
    | otherwise = (zatitu_lag x (tail s) (a++[head s]))
```

```
-----

zatitu :: Integer -> [Integer] -> [[Integer]]
zatitu x s = zatitu_lag x s []
```

```
-----
-- Bukaerako errekurtsibitatea
-----
```

```
nahastu :: [Integer] -> [Integer] -> [Integer]
nahastu r s
    | (null r) = s
    | (null s) = r
    | (head r) <= (head s) = (head r):(nahastu (tail r) s)
    | otherwise = (nahastu s r)
```

```
-----

nahastu_lag :: [Integer] -> [Integer] -> [Integer] -> [Integer]
nahastu_lag r s t
    | (null r) = t++s
    | (null s) = t++r
    | (head r) <= (head s) = (nahastu_lag (tail r) s (t++[(head r)]))
    | otherwise = (nahastu_lag s r t)
```

```
-----

nahastu_be :: [Integer] -> [Integer] -> [Integer]
nahastu_be r s = nahastu_lag r s []
```

```
-----
-- ZERRENDIA-ERAKETA
-----
```

```
hautatu :: Integer -> [Integer] -> Integer
hautatu i s
    | (i < 1) || (i > genericLength s) = error "Indizea ez da zuzena"
    | otherwise = head (genericDrop (i-1) s)
```

```
-----

ezabatu :: Integer -> [Integer] -> [Integer]
ezabatu i s
    | (i < 1) || (i > genericLength s) = error "Indizea ez da zuzena"
    | otherwise = (genericTake (i-1) s) ++ (genericDrop i s)
```

```
-----

errepikatua :: Integer -> [Integer] -> Bool
errepikatua i s
    | (i < 1) || (i > genericLength s) = error "Indizea ez da zuzena"
    | otherwise = (elem (hautatu i s) (ezabatu i s))
```

```
-----

errepikaturik_ez :: [Integer] -> Bool
errepikaturik_ez s = and [ not (errepikatua i s) | i <- [1..(genericLength s)] ]
```

```

aldata :: Integer -> Integer -> Integer -> [Integer]
aldata x b n
    | (x < 0) = error "Zenbakia negatiboa da"
    | (b < 2) = error "0inarria ez da zuzena"
    | (x > ((b^n)-1)) = error "Zenbakia handiegia da"
    | otherwise = [ (aldata_lag x b n i) | i <- [1..n] ]

aldata_lag :: Integer -> Integer -> Integer -> Integer -> Integer
aldata_lag x b n i
    | (x < 0) = error "Zenbakia negatiboa da"
    | (b < 2) = error "0inarria ez da zuzena"
    | (x > ((b^n)-1)) = error "Zenbakia handiegia da"
    | (i < 1) || (i > n) = error "Indizea ez da zuzena"
    | otherwise = (mod (div x (b^(n-i))) b)

-----

guztiak :: Integer -> [[Integer]]
guztiak b
    | (b < 2) = error "0inarria ez da zuzena"
    | otherwise = [ (aldata i b b) | i <- [0..((b^b)-1)] ]

-----

permutazioak :: Integer -> [[Integer]]
permutazioak n
    | (n < 2) = error "Elementu-kopurua ez da zuzena"
    | otherwise = [ x | x <- (guztiak n), (errepikaturik_ez x) ]

-----

aukeratu :: Integer -> Integer -> Integer -> [[Integer]]
aukeratu n i j
    | (n < 2) = error "Elementu-kopurua ez da zuzena"
    | (i < 0) || (j < 0) || (i >= n) || (j >= n) = error "Zenbakiak ez dira agertzen"
    | otherwise = [ s | s <- (permutazioak n), (elem j (dropWhile (/= i) s)) ]

-----

```