

**A) Zerrenden gaineko eragiketak**

Jarraian aipatzen diren eragiketak definitzen dituzten ekuazioak eman. Eragiketak [Int] eta [t] datu-motentzat dira:

**1) inkr -- #**

$$\text{inkr}:: ([\text{Int}]) \rightarrow [\text{Int}]$$

$$\text{inkr}([]) = [] \quad (\#1)$$

$$\text{inkr}(x:s) = (x + 1) : \text{inkr}(s) \quad (\#2)$$
**Adibidea:**

$\begin{aligned} \text{inkr}([4, 8, 5]) &= \\ &= \text{inkr}(4:8:5:[]) = (\#2) \\ &= (4 + 1) : \text{inkr}(8:5:[]) = (\#2) \\ &= 5 : (8 + 1) : \text{inkr}(5:[]) = (\#2) \\ &= 5 : 9 : (5 + 1) : \text{inkr}([]) = (\#1) \\ &= 5 : 9 : 6 : [] \\ &= [5, 9, 6] \end{aligned}$	$\begin{aligned} x &= 4, s = 8:5:[] \\ x &= 8, s = 5:[] \\ x &= 5, s = [] \end{aligned}$
--	--

Adibidea garatzerakoan, urrats bakoitzean zein ekuazio erabili den adierazten da (dagokion zenbakiaren bidez) eta gainera, ekuazioa formularen zein zatiri aplikatu zaion zehaztu da koloreak erabiliz. Bi kolore erabili dira irakurgarritasuna hobetzeko. Beste ariketeran ere irizpide bera jarraitu da.

**2) batu -- #**

$$\text{batu}:: ([\text{Int}]) \rightarrow \text{Int}$$

$$\text{batu}([]) = 0 \quad (\#1)$$

$$\text{batu}(x:s) = x + \text{batu}(s) \quad (\#2)$$
**Adibidea:**

$\begin{aligned} \text{batu}([4, 8, 5]) &= \\ &= \text{batu}(4:8:5:[]) = (\#2) \\ &= 4 + \text{batu}(8:5:[]) = (\#2) \\ &= 4 + 8 + \text{batu}(5:[]) = (\#2) \\ &= 4 + 8 + 5 + \text{batu}([]) = (\#1) \\ &= 4 + 8 + 5 + 0 \\ &= 17 \end{aligned}$	$\begin{aligned} x &= 4, s = 8:5:[] \\ x &= 8, s = 5:[] \\ x &= 5, s = [] \end{aligned}$
--	--

**3) bikoitirik -- #**

bikoitirik:: ([Int]) → Bool

bikoitirik([]) = False (#1)

bikoitirik(x:s)  
| bikoitia(x) = True (#2)

| bakoitia(x) = bikoitirik(s) (#3)

**1. adibidea:**

bikoitirik([7, 5, 9]) =  
= bikoitirik(7:5:9:[]) = (#3) x = 7, s = 5:9:[]  
= bikoitirik(5:9:[]) = (#3) x = 5, s = 9:[]  
= bikoitirik(9:[]) = (#3) x = 9, s = []  
= bikoitirik([]) = (#1)  
= False

**2. adibidea:**

bikoitirik([7, 5, 8, 9, 12]) =  
= bikoitirik(7:5:8:9:12:[]) = (#3) x = 7, s = 5:8:9:12:[]  
= bikoitirik(5:8:9:12:[]) = (#3) x = 5, s = 8:9:12:[]  
= bikoitirik(8:9:12:[]) = (#2) x = 8, s = 9:12:[]  
= True

**3. adibidea:**

bikoitirik([4, 5, 8, 9, 12]) =  
= bikoitirik(4:5:8:9:12:[]) = (#2) x = 4, s = 5:8:9:12:[]  
= True

**4) bik\_posi -- #**

bik\_posi:: ([Int], [Int]) → Bool

bik\_posi([], ℓ)  
| not hutsa\_da(ℓ) = error "Luzera desberdina" (#1)  
| otherwise = True (#2)

bik\_posi(x:s, ℓ)  
| luzera(x:s) ≠ luzera(ℓ) = error "Luzera desberdina" (#3)  
| (bikoitia(x) && bikoitia(leh(ℓ))) || (bakoitia(x) && bakoitia(leh(ℓ)))  
= bik\_posi(s, hond(ℓ)) (#4)  
| otherwise = False (#5)

**1. adibidea:**

```

bik_posi([7, 4, 9], [5, 10, 1]) =
= bik_posi(7:4:9:[], 5:10:1:[]) = (#4)
= bik_posi(4:9:[], 10:1:[]) = (#4)
= bik_posi(9:[], 1:[]) = (#4)
= bik_posi([], []) = (#2)
= True

```

```

x = 7, s = 4:9:[],
leh(ℓ) = 5, hond(ℓ) = 10:1:[]
x = 4, s = 9:[]
leh(ℓ) = 10, hond(ℓ) = 1:[]
x = 9, s = []
leh(ℓ) = 1, hond(ℓ) = []

```

**2. adibidea:**

```

bik_posi([7, 4, 8, 9], [5, 10, 1, 15]) =
= bik_posi(7:4:8:9:[], 5:10:1:15:[]) = (#4)
= bik_posi(4:8:9:[], 10:1:15:[]) = (#4)
= bik_posi(9:[], 1:[]) = (#5)
= False

```

```

x = 7, s = 4:8:9:[],
leh(ℓ) = 5, hond(ℓ) = 10:1:15:[]
x = 4, s = 8:9:[]
leh(ℓ) = 10, hond(ℓ) = 1:15:[]
x = 8, s = 9:[]
leh(ℓ) = 1, hond(ℓ) = 15:[]

```

**3. adibidea:**

```

= bik_posi(7:4:8:9:[], 2:10:1:15:[]) = (#5)
= False

```

```

x = 7, s = 4:8:9:[],
leh(ℓ) = 2, hond(ℓ) = 10:1:15:[]

```

**5) azkena -- #**

azkena:: ([t]) → t

azkena([]) = error "Zerrenda hutsa" (#1)

```

azkena(x:s)
  | hutsa_da(s)      = x          (#2)
  | otherwise        = azkena(s)  (#3)

```

**Adibidea:**

```

azkena([7, 5, 9]) =
= azkena(7:5:9:[]) = (#3)      x = 7, s = 5:9:[]
= azkena(5:9:[]) = (#3)       x = 5, s = 9:[]
= azkena(9:[]) = (#2)         x = 9, s = []
= 9

```

**6) azkena\_kendu -- #**

azkena\_kendu:: ([t]) → [t]

azkena\_kendu([]) = error "Zerrenda hutsa" (#1)

azkena\_kendu(x:s)  
| hutsa\_da(s) = [] (#2)

| otherwise = x : azkena\_kendu(s) (#3)

**Adibidea:**

azkena\_kendu([7, 5, 9]) =  
= azkena\_kendu(7:5:9:[]) = (#3) x = 7, s = 5:9:[]  
= 7 : azkena\_kendu(5:9:[]) = (#3) x = 5, s = 9:[]  
= 7 : 5 : azkena\_kendu(9:[]) = (#2) x = 9, s = []  
= 7 : 5 : []  
= [7, 5]

**7) alder -- #**

Bi eratara:

a) ++ erabiliz

alder:: ([t]) → [t]

alder([]) = [] (#1)

alder(x:s) = alder(s) ++ (x:[]) (#2)

**Adibidea:**

alder([2, 5, 9]) =  
= alder(2:5:9:[]) = (#2) x = 2, s = 5:9:[]  
= alder(5:9:[]) ++ (2:[]) = (#2) x = 5, s = 9:[]  
= alder(9:[]) ++ (5:[]) ++ (2:[]) = (#2) x = 9, s = []  
= alder([]) ++ (9:[]) ++ (5:[]) ++ (2:[]) = (#1)  
= [9, 5, 2]

b) azkena eta azkena\_kendu erabiliz

alder:: ([t]) → [t]

alder([]) = [] (#1)

alder(x:s) = azkena(x:s) : alder(azkena\_kendu(x:s)) (#2)

**Adibidea:**

```

alder([2, 5, 9]) =
= alder(2:5:9:[]) = (#2)           x = 2, s = 5:9:[]
           azkena(2:5:9:[]) = 9, azkena_kendu(2:5:9:[]) = 2:5:[]
= 9 : alder(2:5:[]) = (#2)           x = 2, s = 5:[]
           azkena(2:5:[]) = 5, azkena_kendu(2:5:[]) = 2:[]
= 9: 5: alder(2:[]) = (#2)           x = 2, s = []
           azkena(2:[]) = 2, azkena_kendu(2:[]) = []
= 9 : 5 : 2 : alder([]) = (#1)
= 9 : 5 : 2 : []
= [9, 5, 2]

```

## 8) alder\_dira -- #

Bi eratara:

### a) alder erabiliz

$\text{alder\_dira}:: ([t], [t]) \rightarrow \text{Bool}$

```

alder_dira(s, r)
  | s == alder(r)           = True           (#1)
  | otherwise               = False          (#2)

```

#### 1. adibidea:

```

alder_dira([2, 5, 9], [9, 5, 2]) =
= alder_dira(2:5:9:[], 9:5:2:[]) = (#1)           s = 2:5:9:[], r = 9:5:2:[]
= True

```

#### 2. adibidea:

```

alder_dira([2, 5, 9], [9, 8, 2]) =
= alder_dira(2:5:9:[], 9:5:2:[]) = (#2)           s = 2:5:9:[], r = 9:8:2:[]
= False

```

*alder\_dira* funtzioaren definizio honetan ez dago errekurtsibitaterik eta, horregatik, adibideak urrats bakar batean ebazten dira.

### b) azkena, azkena\_kendu, luzera eta hutsa\_da erabiliz

$\text{alder\_dira}:: ([t], [t]) \rightarrow \text{Bool}$

```

alder_dira([], ℓ)
  | not hutsa_da(ℓ)         = False           (#1)
  | otherwise               = True            (#2)

```

```

alder_dira(x:s, ℓ)
  | luzera(x:s) /= luzera(ℓ) = False          (#3)
  | x /= azkena(ℓ)          = False          (#4)
  | otherwise                = alder_dira(s, azkena_kendu(ℓ)) (#5)

```

**1. adibidea:**

```

alder_dira([2, 5, 9], [9, 5, 2]) =
= alder_dira(2:5:9:[], 9:5:2:[]) = (#5)      x = 2, s = 5:9:[], ℓ = 9:5:2:[]
      azkena(ℓ) = 2, azkena_kendu(ℓ) = 9:5:[]
= alder_dira(5:9:[], 9:5:[]) = (#5)          x = 5, s = 9:[], ℓ = 9:5:[]
      azkena(ℓ) = 5, azkena_kendu(ℓ) = 9:[]
= alder_dira(9:[], 9:[]) = (#5)              x = 9, s = [], ℓ = 9:[]
      azkena(ℓ) = 9, azkena_kendu(ℓ) = []
= alder_dira([], []) = (#2)
= True

```

**2. adibidea:**

```

alder_dira([2, 5, 9], [9, 8, 2]) =
= alder_dira(2:5:9:[], 9:8:2:[]) = (#5)      x = 2, s = 5:9:[], ℓ = 9:8:2:[]
      azkena(ℓ) = 2, azkena_kendu(ℓ) = 9:8:[]
= alder_dira(5:9:[], 9:8:[]) = (#4)          x = 5, s = 9:[], ℓ = 9:8:[]
      azkena(ℓ) = 8, azkena_kendu(ℓ) = 9:[]
= False

```

*alder\_dira* funtzioaren bigarren definizio honetan errekursibitatea erabili da eta, ondorioz, adibideak ebazteko orokorrean urrats bat baino gehiago behar dira.

**9) aldiz -- #**

$\text{aldiz} :: (t, [t]) \rightarrow \text{Int}$

$\text{aldiz}(x, []) = 0$  (#1)

$\text{aldiz}(x, y:s)$   
     |  $x == y$                        $= 1 + \text{aldiz}(x, s)$                       (#2)

    | otherwise                     $= \text{aldiz}(x, s)$                       (#3)

**Adibidea:**

```

aldiz(8, [7, 8, 5, 9, 8]) =
= aldiz(8, 7:8:5:9:8:[]) = (#3)      x = 8, y = 7, s = 8:5:9:8:[]
= aldiz(8, 8:5:9:8:[]) = (#2)      x = 8, y = 8, s = 5:9:8:[]
= 1 + aldiz(8, 5:9:8:[]) = (#3)      x = 8, y = 5, s = 9:8:[]
= 1 + aldiz(8, 9:8:[]) = (#3)      x = 8, y = 9, s = 8:[]
= 1 + aldiz(8, 8:[]) = (#2)      x = 8, y = 8, s = []
= 1 + 1 + aldiz(8, []) = (#1)      x = 8
= 1 + 1 + 0 = 2

```

**10)erre -- #**

erre:: ([t]) → Bool

erre([]) = False (#1)

erre(x:s)  
     | badago(x, s)           = True           (#2)  
     | otherwise            = erre(s)        (#3)

**1. adibidea:**

erre([7, 5, 8, 9, 8]) =  
 = erre(7:8:5:9:8:[]) = (#3)           x = 7, s = 5:8:9:8:[]  
 = erre(5:8:9:8:[]) = (#3)           x = 5, s = 8:9:8:[]  
 = erre(8:9:8:[]) = (#2)           x = 8, s = 9:8:[]  
 = True

**2. adibidea:**

erre([7, 5, 8, 9, 1]) =  
 = erre(7:8:5:9:1:[]) = (#3)           x = 7, s = 5:8:9:1:[]  
 = erre(5:8:9:1:[]) = (#3)           x = 5, s = 8:9:1:[]  
 = erre(8:9:1:[]) = (#3)           x = 8, s = 9:1:[]  
 = erre(9:1:[]) = (#3)           x = 9, s = 1:[]  
 = erre(1:[]) = (#3)           x = 1, s = []  
 = erre([]) = (#1)  
 = False

**11)kendu -- #**

kendu:: (t, [t]) → [t]

kendu(x, []) = [] (#1)

kendu(x, y:s)  
     | x == y               = kendu(x, s)           (#2)  
     | otherwise           = y: kendu(x, s)        (#3)

**Adibidea:**

kendu(8, [7, 5, 8, 9, 8]) =  
 = kendu(8, 7:5:8:9:8:[]) = (#3)           x = 8, y = 7, s = 5:8:9:8:[]  
 = 7:kendu(8, 5:8:9:8:[]) = (#3)           x = 8, y = 5, s = 8:9:8:[]  
 = 7:5:kendu(8, 8:9:8:[]) = (#2)           x = 8, y = 8, s = 9:8:[]  
 = 7:5:kendu(8, 9:8:[]) = (#3)           x = 8, y = 9, s = 8:[]  
 = 7:5:9:kendu(8, 8:[]) = (#2)           x = 8, y = 8, s = []  
 = 7:5:9:kendu(8, []) = (#1)           x = 8  
 = 7:5:9:[] = [7, 5, 9]

**12)bik\_kendu -- #**

bik\_kendu:: ([Int]) → [Int]

bik\_kendu([]) = [] (#1)

bik\_kendu(x:s)  
     | bikoitia(x)           = bik\_kendu(s)           (#2)  
     | bakoitia(x)          = x:bik\_kendu(s)        (#3)

**Adibidea:**

bik\_kendu([7, 5, 8, 9, 12]) =  
 = bik\_kendu(7:5:8:9:12:[]) = (#3)      x = 7, s = 5:8:9:12:[]  
 = 7:bik\_kendu(5:8:9:12:[]) = (#3)      x = 5, s = 8:9:12:[]  
 = 7:5:bik\_kendu(8:9:12:[]) = (#2)      x = 8, s = 9:12:[]  
 = 7:5:bik\_kendu(9:12:[]) = (#3)      x = 9, s = 12:[]  
 = 7:5:9:bik\_kendu(12:[]) = (#2)      x = 12, s = []  
 = 7:5:9:bik\_kendu([]) = (#1)  
 = 7:5:9:[] = [7, 5, 9]

**13)pos\_bik\_kendu -- #**

pos\_bik\_kendu:: ([t]) → [t]

pos\_bik\_kendu([]) = [] (#1)

pos\_bik\_kendu(x:s)  
     | hutsa\_da(s)           = x:s                   (#2)  
     | otherwise            = x:pos\_bik\_kendu(hond(s))    (#3)

**1. adibidea:**

pos\_bik\_kendu([7, 5, 8, 9, 12]) =  
 = pos\_bik\_kendu(7:5:8:9:12:[]) = (#3)      x = 7, s = 5:8:9:12:[], hond(s) = 8:9:12:[]  
 = 7:pos\_bik\_kendu(8:9:12:[]) = (#3)      x = 8, s = 9:12:[], hond(s) = 12:[]  
 = 7:8:pos\_bik\_kendu(12:[]) = (#2)      x = 8, s = []  
 = 7:8:12:[] = [7, 8, 12]

**2. adibidea:**

pos\_bik\_kendu([7, 5, 8, 9]) =  
 = pos\_bik\_kendu(7:5:8:9:[]) = (#3)      x = 7, s = 5:8:9:[], hond(s) = 8:9:[]  
 = 7:pos\_bik\_kendu(8:9:[]) = (#3)      x = 8, s = 9:[], hond(s) = []  
 = 7:8:pos\_bik\_kendu([]) = (#1)  
 = 7:8:[] = [7, 8]



**14)pos\_bak\_kendu -- #**

pos\_bak\_kendu:: ([t]) → [t]

pos\_bak\_kendu([]) = [] (#1)

pos\_bak\_kendu(x:s)  
     | hutsa\_da(s) = [] (#2)  
     | otherwise = leh(s):pos\_bak\_kendu(hond(s)) (#3)

**1. adibidea:**

pos\_bak\_kendu([7, 5, 8, 9, 12]) =  
 = pos\_bak\_kendu(7:5:8:9:12:[]) = (#3)  
     x = 7, s = 5:8:9:12:[], leh(s) = 5, hond(s) = 8:9:12:[]  
 = 5:pos\_bak\_kendu(8:9:12:[]) = (#3)  
     x = 8, s = 9:12:[], leh(s) = 9, hond(s) = 12:[]  
 = 5:9:pos\_bak\_kendu(12:[]) = (#2) x = 8, s = []  
 = 5:9:[] = [5, 9]

**2. adibidea:**

pos\_bak\_kendu([7, 5, 8, 9]) =  
 = pos\_bak\_kendu(7:5:8:9:[]) = (#3)  
     x = 7, s = 5:8:9:[], leh(s) = 5, hond(s) = 8:9:[]  
 = 5:pos\_bak\_kendu(8:9:[]) = (#3)  
     x = 8, s = 9:[], leh(s) = 9, hond(s) = []  
 = 5:9:pos\_bak\_kendu([]) = (#1)  
 = 5:9:[] = [5, 9]

**15)erre\_kendu -- #**

erre\_kendu:: ([t]) → [t]

erre\_kendu([]) = [] (#1)

erre\_kendu(x:s)  
     | badago(x, s) = x: erre\_kendu(kendu(x, s)) (#2)  
     | otherwise = x: erre\_kendu(s) (#3)

**Adibidea:**

erre\_kendu([7, 5, 8, 5, 2, 8, 8]) =  
 = erre\_kendu(7:5:8:5:2:8:8:[]) = (#3) x = 7, s = 5:8:5:2:8:8:[]  
 = 7:erre\_kendu(5:8:5:2:8:8:[]) = (#2) x = 5, s = 8:5:2:8:8:[]  
 = 7:5:erre\_kendu(8:2:8:8:[]) = (#2) x = 8, s = 2:8:8:[]  
 = 7:5:8:erre\_kendu(2:[]) = (#3) x = 2, s = []  
 = 7:5:8:2:erre\_kendu([]) = (#1)  
 = 7:5:8:2:[] = [7, 5, 8, 2]

**16)erre\_kendu2 -- #**

erre\_kendu2:: ([t]) → [t]

erre\_kendu2([]) = [] (#1)

erre\_kendu2(x:s)  
 | badago(x, s) = erre\_kendu2(s) (#2)  
 | otherwise = x:erre\_kendu2(s) (#3)

**Adibidea:**

erre\_kendu2([7, 5, 8, 5, 2, 8, 8]) =  
 = erre\_kendu2(7:5:8:5:2:8:8:[]) = (#3)  
 = 7:erre\_kendu2(5:8:5:2:8:8:[]) = (#2)  
 = 7:erre\_kendu2(8:5:2:8:8:[]) = (#2)  
 = 7:erre\_kendu2(5:2:8:8:[]) = (#3)  
 = 7:5:erre\_kendu2(2:8:8:[]) = (#3)  
 = 7:5:2:erre\_kendu2(8:8:[]) = (#2)  
 = 7:5:2:erre\_kendu2(8:[]) = (#3)  
 = 7:5:2:erre\_kendu2([]) = (#1)  
 = 7:5:2:8:[] = [7, 5, 2, 8]

x = 7, s = 5:8:5:2:8:8:[]  
 x = 5, s = 8:5:2:8:8:[]  
 x = 8, s = 5:2:8:8:[]  
 x = 5, s = 2:8:8:[]  
 x = 2, s = 8:8:[]  
 x = 8, s = 8:[]  
 x = 8, s = []

**17)bikote\_berdin -- #**

bikote\_berdin:: ([t]) → Bool

bikote\_berdin([]) = False (#1)

bikote\_berdin(x:s)  
 | hutsa\_da(s) = False (#2)  
 | x == leh(s) = True (#3)  
 | otherwise = bikote\_berdin(s) (#4)

**1. adibidea:**

bikote\_berdin([7, 5, 8, 8, 6, 5]) =  
 = bikote\_berdin(7:5:8:8:6:5:[]) = (#4)  
 = bikote\_berdin(5:8:8:6:5:[]) = (#4)  
 = bikote\_berdin(8:8:6:5:[]) = (#3)  
 = True

x = 7, s = 5:8:8:6:5:[] , leh(s) = 5  
 x = 5, s = 8:8:6:5:[] , leh(s) = 8  
 x = 8, s = 8:6:5:[] , leh(s) = 8

**2. adibidea:**

bikote\_berdin([7, 5, 8, 5]) =  
 = bikote\_berdin(7:5:8:5:[]) = (#4)  
 = bikote\_berdin(5:8:5:[]) = (#4)  
 = bikote\_berdin(8:5:[]) = (#4)  
 = bikote\_berdin(5:[]) = (#2)  
 = False

x = 7, s = 5:8:5:[] , leh(s) = 5  
 x = 5, s = 8:5:[] , leh(s) = 8  
 x = 8, s = 5:[] , leh(s) = 5  
 x = 5, s = []

**18)aurrizkia -- #**

aurrizkia:: ([t], [t]) → Bool

aurrizkia([], ℓ) = True (#1)

aurrizkia(x:s, ℓ)

hutsa_da(ℓ)	= False	(#2)
x /= leh(ℓ)	= False	(#3)
otherwise	= aurrizkia(s, hond(ℓ))	(#4)

**1. adibidea:**

aurrizkia([1, 2, 3], [1, 2, 5, 6]) =  
 = aurrizkia(1:2:3:[], 1:2:5:6:[]) = (#4)  
     x = 1, s = 2:3:[], ℓ = 1:2:5:6:[], leh(ℓ) = 1, hond(ℓ) = 2:5:6:[]  
 = aurrizkia(2:3:[], 2:5:6:[]) = (#4)  
     x = 2, s = 3:[], ℓ = 2:5:6:[], leh(ℓ) = 2, hond(ℓ) = 5:6:[]  
 = aurrizkia(3:[], 5:6:[]) = (#3)  
     x = 3, s = [], ℓ = 5:6:[], leh(ℓ) = 5, hond(ℓ) = 6:[]  
 = False

**2. adibidea:**

aurrizkia([1, 2, 3], [1, 2, 3, 5, 6]) =  
 = aurrizkia(1:2:3:[], 1:2:3:5:6:[]) = (#4)  
     x = 1, s = 2:3:[], ℓ = 1:2:3:5:6:[], leh(ℓ) = 1, hond(ℓ) = 2:3:5:6:[]  
 = aurrizkia(2:3:[], 2:3:5:6:[]) = (#4)  
     x = 2, s = 3:[], ℓ = 2:3:5:6:[], leh(ℓ) = 2, hond(ℓ) = 3:5:6:[]  
 = aurrizkia(3:[], 3:5:6:[]) = (#4)  
     x = 3, s = [], ℓ = 3:5:6:[], leh(ℓ) = 3, hond(ℓ) = 5:6:[]  
 = aurrizkia([], 5:6:[]) = (#1)      ℓ = 5:6:[]  
 = True

**3. adibidea:**

aurrizkia([1, 2, 3], [1, 2]) =  
 = aurrizkia(1:2:3:[], 1:2:[]) = (#4)  
     x = 1, s = 2:3:[], ℓ = 1:2:[], leh(ℓ) = 1, hond(ℓ) = 2:[]  
 = aurrizkia(2:3:[], 2:[]) = (#4)  
     x = 2, s = 3:[], ℓ = 2:[], leh(ℓ) = 2, hond(ℓ) = []  
 = aurrizkia(3:[], []) = (#2)      x = 3, s = [], ℓ = []  
 = True

**19)azpizer -- #**

azpizer:: ([t], [t]) → Bool

azpizer([], ℓ) = True (#1)

azpizer(x:s, ℓ)  
| hutsa\_da(ℓ) = False (#2)

| aurrizkia(x:s, ℓ) = True (#3)

| otherwise = azpizer(x:s, hond(ℓ)) (#4)

**1. adibidea:**

azpizer([4, 5], [1, 2, 5, 6]) =  
= azpizer(4:5:[], 1:2:5:6:[]) = (#4)  
x = 4, s = 5:[], ℓ = 1:2:5:6:[], hond(ℓ) = 2:5:6:[]  
= azpizer(4:5:[], 2:5:6:[]) = (#4)  
x = 4, s = 5:[], ℓ = 2:5:6:[], hond(ℓ) = 5:6:[]  
= azpizer(4:5:[], 5:6:[]) = (#4)  
x = 4, s = 5:[], ℓ = 5:6:[], hond(ℓ) = 6:[]  
= es\_sublista(4:5:[], 6:[]) = (#4)  
x = 4, s = 5:[], ℓ = 6:[], hond(ℓ) = []  
= es\_sublista(4:5:[], []) = (#2)  
x = 4, s = 5:[], ℓ = []  
= False

**2. adibidea:**

azpizer([4, 5], [1, 2, 4, 5, 6]) =  
= azpizer(4:5:[], 1:2:4:5:6:[]) = (#4)  
x = 4, s = 5:[], ℓ = 1:2:4:5:6:[], hond(ℓ) = 2:4:5:6:[]  
= azpizer(4:5:[], 2:4:5:6:[]) = (#4)  
x = 4, s = 5:[], ℓ = 2:4:5:6:[], hond(ℓ) = 4:5:6:[]  
= azpizer(4:5:[], 4:5:6:[]) = (#3)  
x = 4, s = 5:[], ℓ = 4:5:6:[]  
= True

**20)elem\_pos -- #**

elem\_pos:: (Int, [t]) → t

elem\_pos(pos, []) = error "Ez da egokia" (#1)

elem\_pos(pos, x:s)  
 | pos <= 0 || pos > luzera(x:s) = error "Ez da egokia" (#2)  
 | pos == 1 = x (#3)  
 | otherwise = elem\_pos(pos - 1, s) (#4)

**Adibidea:**

elem\_pos(3, [5, 9, 8, 6, 2]) =  
 = elem\_pos(3, 5:9:8:6:2:[]) = (#4) pos = 3, x = 5, s = 9:8:6:2:[]  
 = elem\_pos(2, 9:8:6:2:[]) = (#4) pos = 2, x = 9, s = 8:6:2:[]  
 = elem\_pos(1, 8:6:2:[]) = (#3) pos = 1, x = 8, s = 6:2:[]  
 = 8

**21)sartu -- #**

sartu:: (Int, t, [t]) → [t]

sartu(pos, x, [])  
 | pos /= 1 = error "Ez da egokia" (#1)  
 | pos == 1 = x:[] (#2)

sartu(pos, x, y:s)  
 | pos <= 0 || pos > (luzera(x:s) + 1) = error "Ez da egokia" (#3)  
 | pos == 1 = x:y:s (#4)  
 | otherwise = y:sartu(pos - 1, x, s) (#5)

**Adibidea:**

sartu(3, 7, [5, 9, 8, 6, 2]) =  
 = sartu(3, 7, 5:9:8:6:2:[]) = (#5) pos = 3, x = 7, y = 5, s = 9:8:6:2:[]  
 = 5:sartu(2, 7, 9:8:6:2:[]) = (#5) pos = 2, x = 7, y = 9, s = 8:6:2:[]  
 = 5:9:sartu(1, 7, 8:6:2:[]) = (#4) pos = 1, x = 7, y = 8, s = 6:2:[]  
 = 5:9:7:8:6:2:[]

**22)lehen\_bik\_pos -- #**

lehen\_bik\_pos:: (Int, [t]) → t

lehen\_bik\_pos([]) = 1 (#1)

lehen\_bik\_pos(x:s)  
 | x `mod` 2 == 0 = 1 (#2)

$$| \text{otherwise} \quad \quad \quad = 1 + \text{lehen\_bik\_pos}(s) \quad \quad \quad (\#3)$$

**1. adibidea:**

$$\begin{aligned} \text{lehen\_bik\_pos}([5, 9, 8, 6]) &= \\ &= \text{lehen\_bik\_pos}(5:9:8:6:[]) = (\#3) & x = 5, s = 9:8:6:[] \\ &= 1 + \text{lehen\_bik\_pos}(9:8:6:[]) = (\#3) & x = 9, s = 8:6:[] \\ &= 1 + 1 + \text{lehen\_bik\_pos}(8:6:[]) = (\#2) & x = 8, s = 6:[] \\ &= 1 + 1 + 1 = 3 \end{aligned}$$

**2. adibidea:**

$$\begin{aligned} \text{lehen\_bik\_pos}([5, 9]) &= \\ &= \text{lehen\_bik\_pos}(5:9:[]) = (\#3) & x = 5, s = 9:[] \\ &= 1 + \text{lehen\_bik\_pos}(9:[]) = (\#3) & x = 9, s = [] \\ &= 1 + 1 + \text{lehen\_bik\_pos}([]) = (\#1) \\ &= 1 + 1 + 1 = 3 \end{aligned}$$

**23) azken\_pos -- #**

azken\_pos:: (Int, [Int]) → Int

$$\text{azken\_pos}(x, []) = 0 \quad \quad \quad (\#1)$$

$$\begin{aligned} \text{azken\_pos}(x, y:s) & \\ | x \neq y \ \&\& \text{not badago}(x, s) & = 0 & (\#2) \\ | \text{otherwise} & = 1 + \text{azken\_pos}(x, s) & (\#3) \end{aligned}$$

**1. adibidea:**

$$\begin{aligned} \text{azken\_pos}(8, [8, 6, 8, 5]) &= \\ &= \text{azken\_pos}(8, 8:6:8:5:[]) = (\#3) & x = 8, y = 8, s = 6:8:5:[] \\ &= 1 + \text{azken\_pos}(8, 6:8:5:[]) = (\#3) & x = 8, y = 6, s = 8:5:[] \\ &= 1 + 1 + \text{azken\_pos}(8, 8:5:[]) = (\#3) & x = 8, y = 8, s = 5:[] \\ &= 1 + 1 + 1 + \text{azken\_pos}(8, []) = (\#1) & x = 8 \\ &= 1 + 1 + 1 + 0 = 3 \end{aligned}$$

**2. adibidea:**

$$\begin{aligned} \text{azken\_pos}(8, [3, 2]) &= \\ &= \text{azken\_pos}(8, 3:2:[]) = (\#2) & x = 8, y = 3, s = 2:[] \\ &= 0 \end{aligned}$$

**Funtzioa definitzeko beste aukera bat:**

azken\_pos:: (Int, [Int]) → Int

$$\text{azken\_pos}(x, []) = 0 \quad \quad \quad (\#1)$$

$$\begin{aligned} \text{azken\_pos}(x, y:s) & \\ | x == y & = 1 + \text{azken\_pos}(x, s) & (\#2) \\ | \text{badago}(x, s) & = 1 + \text{azken\_pos}(x, s) & (\#3) \\ | \text{otherwise} & = 0 & (\#4) \end{aligned}$$

**1. adibidea:**

```

azken_pos(8, [8, 6, 8, 5]) =
= azken_pos(8, 8:6:8:5:[]) = (#2)
= 1 + azken_pos(8, 6:8:5:[]) = (#3)
= 1 + 1 + azken_pos(8, 8:5:[]) = (#2)
= 1 + 1 + 1 + azken_pos(8, []) = (#1)
= 1 + 1 + 1 + 0 = 3

```

x = 8, y = 8, s = 6:8:5:[]  
x = 8, y = 6, s = 8:5:[]  
x = 8, y = 8, s = 5:[]  
x = 8

**2. adibidea:**

```

azken_pos(8, [3, 2]) =
= azken_pos(8, 3:2:[]) = (#4)
= 0

```

x = 8, y = 3, s = 2:[]

**24)hand -- #**

hand:: ([Int]) → Int

```
hand([]) = error "Zerrenda hutsa" (#1)
```

```

hand(x:s)
| hutsa_da(s)           = x (#2)
| x >= leh(s)           = hand(x:hond(s)) (#3)
| otherwise             = hand(s) (#4)

```

**Adibidea:**

```

hand([5, 3, 8, 7, 8]) =
= hand(5:3:8:7:8:[]) = (#3)
= hand(5:8:7:8:[]) = (#4)
= hand(8:7:8:[]) = (#3)
= hand(8:8:[]) = (#3)
= hand(8:[]) = (#2)
= 8

```

x = 5, s = 3:8:7:8:[], leh(s) = 3, hond(s) = 8:7:8:[]  
x = 5, s = 8:7:8:[], leh(s) = 8, hond(s) = 7:8:[]  
x = 8, s = 7:8:[], leh(s) = 7, hond(s) = 8:[]  
x = 8, s = 8:[], leh(s) = 8, hond(s) = []  
x = 8

**25)elkartu -- #**

a) Aurretik definituta dagoen beste funtziorik erabili gabe

$\text{elkartu}:: ([t], [t]) \rightarrow [t]$

$\text{elkartu}([], \ell) = \ell$  (#1)

$\text{elkartu}(x:s, \ell) = \text{elkartu}(s, x:\ell)$  (#2)

**Adibidea:**

$\text{elkartu}([5, 3, 8], [7, 9]) =$   
 $= \text{elkartu}(5:3:8:[], 7:9:[]) =$  (#2)  $x = 5, s = 3:8:[], \ell = 7:9:[]$   
 $= \text{elkartu}(3:8:[], 5:7:9:[]) =$  (#2)  $x = 3, s = 8:[], \ell = 5:7:9:[]$   
 $= \text{elkartu}(8:[], 3:5:7:9:[]) =$  (#2)  $x = 8, s = [], \ell = 3:5:7:9:[]$   
 $= \text{elkartu}([], 8:3:5:7:9:[]) =$  (#1)  $\ell = 8:3:5:7:9:[]$   
 $= 8:3:5:7:9:[] = [8,3,5,7,9]$

b) Aurretik definituta dauden beste funtzio hauek erabiliz: **alder** eta ++

$\text{elkartu}:: ([t], [t]) \rightarrow [t]$

$\text{elkartu}(\ell1, \ell2) = \text{alder}(\ell1) ++ \ell2$  (#1)

Kasu honetan *elkartu* funtzioaren definizioa ez da errekursiboa.

**Adibidea:**

$\text{elkartu}([5, 3, 8], [7, 9]) =$   
 $= \text{elkartu}(5:3:8:[], 7:9:[]) =$  (#1)  $\ell1 = 5:3:8:[], \ell2 = 7:9:[]$   
 $= 8:3:5:[] ++ 7:9:[]$   
 $= 8:3:5:7:9:[] = [8,3,5,7,9]$

**Beste aukera bat:**

$\text{elkartu}:: ([t], [t]) \rightarrow [t]$

$\text{elkartu}([], \ell) = \ell$  (#1)

$\text{elkartu}(x:s, \ell) = \text{alder}(s) ++ (x:\ell)$  (#2)

**Adibidea:**

$\text{elkartu}([5, 3, 8], [7, 9]) =$   
 $= \text{elkartu}(5:3:8:[], 7:9:[]) =$  (#2)  $x = 5, s = 3:8:[], \ell = 7:9:[]$   
 $= 8:3:[] ++ 5:7:9:[]$   
 $= 8:3:5:7:9:[] = [8,3,5,7,9]$



**26) ezabatu -- #**

ezabatu:: (Int, [t]) → [t]

ezabatu(zenbat, [])

| zenbat /= 0 = error "Ez da egokia" (#1)

| otherwise = [] (#2)

ezabatu(zenbat, x:s)

| zenbat < 0 || zenbat > luzera (x:s) = error "Ez da egokia" (#3)

| zenbat == 0 = x:s (#4)

| otherwise = ezabatu(zenbat – 1, s) (#5)

**Adibidea:**

ezabatu(3, [7, 6, 8, 5, 9]) =

= ezabatu(3, 7:6:8:5:9:[]) = (#5)

= ezabatu(2, 6:8:5:9:[]) = (#5)

= ezabatu(1, 8:5:9:[]) = (#5)

= ezabatu(0, 5:9:[]) = (#4)

= 5:9:[] = [5, 9]

zenbat = 3, x = 7, s = 6:8:5:9:[]

zenbat = 2, x = 6, s = 8:5:9:[]

zenbat = 1, x = 8, s = 5:9:[]

zenbat = 0, x = 5, s = 9:[]

**27) hartu -- #**

hartu:: (Int, [t]) → [t]

hartu(zenbat, [])

| zenbat /= 0 = error "Ez da egokia" (#1)

| otherwise = [] (#2)

hartu(zenbat, x:s)

| zenbat < 0 || zenbat > luzera (x:s) = error "Ez da egokia" (#3)

| zenbat == 0 = [] (#4)

| otherwise = x:hartu(zenbat – 1, s) (#5)

**Adibidea:**

hartu(3, [7, 6, 8, 5, 9]) =

= hartu(3, 7:6:8:5:9:[]) = (#5)

= 7:hartu(2, 6:8:5:9:[]) = (#5)

= 7:6:hartu(1, 8:5:9:[]) = (#5)

= 7:6:8:hartu(0, 5:9:[]) = (#4)

= 7:6:8:[] = [7, 6, 8]

zenbat = 3, x = 7, s = 6:8:5:9:[]

zenbat = 2, x = 6, s = 8:5:9:[]

zenbat = 1, x = 8, s = 5:9:[]

zenbat = 0, x = 5, s = 9:[]

**28)kolapsatu -- #****a)**

kolapsatu:: ([Int]) → [Int]

kolapsatu([]) = [] (#1)

kolapsatu(x:s)		
hutsa_da(s)	= x:s	(#2)
x == leh(s)	= kolapsatu(s)	(#3)
otherwise	= x:kolapsatu(s)	(#4)

**b)****Adibidea:**

kolapsatu([3, 9, 9, 3]) =	
= kolapsatu(3:9:9:3:[]) = (#4)	x = 3, s = 9:9:3:[], leh(s) = 9
= 3:kolapsatu(9:9:3:[]) = (#3)	x = 9, s = 9:3:[], leh(s) = 9
= 3:kolapsatu(9:3:[]) = (#4)	x = 9, s = 3:[], leh(s) = 3
= 3:9:kolapsatu(3:[]) = (#2)	x = 3, s = []
= 3:9:3:[] = [3, 9, 3]	

**29)hedatu -- #****a)**

hedatu:: ([Int]) → [Int]

hedatu([]) = [] (#1)

hedatu(x:s)		
hutsa_da(s)	= x:s	(#2)
x `mod` 2 == 0	= x:hedatu((x + leh(s)):hond(s))	(#3)
otherwise	= x:hedatu(s)	(#4)

**b)****Adibidea:**

hedatu([8, 7, 2]) =	
= hedatu(8:7:2:[]) = (#3)	x = 8, s = 7:2:[], leh(s) = 7, hond(s) = 2:[]
= 8:hedatu(15:2:[]) = (#4)	x = 15, s = 2:[], leh(s) = 2, hond(s) = []
= 8:15:hedatu(2:[]) = (#2)	x = 2, s = []
= 8:15:2:[] = [8, 15, 2]	

**30)zapaldu -- #****a)**

zapaldu:: ([Int]) → [Int]

zapaldu([]) = [] (#1)

zapaldu(x:s)

| hutsa\_da(s) = x:s (#2)

| x &gt;= leh(s) = x:zapaldu(x:hond(s)) (#3)

| otherwise = x:zapaldu(s) (#4)

**b)**

zapaldu([3, 9, 9, 3]) =

= zapaldu(3:9:9:3:[]) = (#4)

= 3:zapaldu(9:9:3:[]) = (#3)

= 3:9:zapaldu(9:3:[]) = (#3)

= 3:9:9:zapaldu(9:[]) = (#2)

= 3:9:9:9:[] = [3, 9, 9, 9]

x = 3, s = 9:9:3:[], leh(s) = 9, hond(s) = 9:3:[]

x = 9, s = 9:3:[], leh(s) = 9, hond(s) = 3:[]

x = 9, s = 3:[], leh(s) = 3, hond(s) = []

x = 9, s = []

**31)hondoratu -- #****a)**

hondoratu:: ([Int]) → [Int]

hondoratu([]) = [] (#1)

hondoratu(x:s)

| hutsa\_da(s) = x:s (#2)

| x &gt; leh(s) = leh(s):hondoratu(x:hond(s)) (#3)

| otherwise = x:s (#4)

**b)**

hondoratu([8, 3, 2, 9]) =

= hondoratu(8:3:2:9:[]) = (#3)

= 3:hondoratu(8:2:9:[]) = (#3)

= 3:2:hondoratu(8:9:[]) = (#4)

= 3:2:8:9:[] = [3, 2, 8, 9]

x = 8, s = 3:2:9:[], leh(s) = 3, hond(s) = 2:9:[]

x = 8, s = 2:9:[], leh(s) = 2, hond(s) = 9:[]

x = 8, s = 9:[], leh(s) = 9, hond(s) = []

**32)elem\_kendu -- #****a)**

elem\_kendu:: ([Int], Int) → [Int]

elem\_kendu([], pos) = error "Zerrenda hutsa" (#1)

elem_kendu(x:s, pos)		
pos <= 0    pos > luzera(x:s)	= error "Ez da egokia"	(#2)
pos == luzera(x:s)	= s	(#3)
otherwise	= x:elem_kendu(s, pos)	(#4)

**b)**

elem_kendu([1, 7, 5, 8], 2) =	
= elem_kendu(1:7:5:8:[], 2) = (#4)	x = 1, s = 7:5:8:[], pos = 2
= 1:elem_kendu(7:5:8:[], 2) = (#4)	x = 7, s = 5:8:[], pos = 2
= 1:7:elem_kendu(5:8:[], 2) = (#3)	x = 5, s = 8:[], pos = 2
= 1:7:8:[] = [1, 7, 8]	

**33)garbitu (2008ko apirila #1) -- #****a)**

garbitu:: ([Int]) → [Int]

garbitu([]) = error "Zerrenda hutsa" (#1)

garbitu(x:s)		
hutsa_da(s)	= x:s	(#2)
x > leh(s)	= garbitu(x:hond(s))	(#3)
otherwise	= leh(s):garbitu(x:hond(s))	(#4)

**b)**

garbitu([5, 8, 4, 1, 9]) =	
= garbitu(5:8:4:1:9:[]) = (#4)	x = 5, s = 8:4:1:9:[],
	leh(s) = 8, hond(s) = 4:1:9:[]
= 8:garbitu(5:4:1:9:[]) = (#3)	x = 5, s = 4:1:9:[],
	leh(s) = 4, hond(s) = 1:9:[]
= 8:garbitu(5:1:9:[]) = (#3)	x = 5, s = 1:9:[],
	leh(s) = 1, hond(s) = 9:[]
= 8:garbitu(5:9:[]) = (#4)	x = 5, s = 9:[], leh(s) = 9, hond(s) = []
= 8:9:garbitu(5:[]) = (#2)	x = 5, s = []
= 8:9:5:[] = [8, 9, 5]	

**34)zeharkatu (2008ko apirila #2) -- #****a)**

zeharkatu:: ([Int]) → [Int]

zeharkatu([]) = [] (#1)

zeharkatu(x:s)		
hutsa_da(s)	= x:s	(#2)
x == leh(s)	= 0:0:hond(s)	(#3)
otherwise	= leh(s):zeharkatu(x:hond(s))	(#4)

**b)**

zeharkatu([5, 8, 4]) =	
= zeharkatu(5:8:4:[]) = (#4)	x = 5, s = 8:4:[], leh(s) = 8, hond(s) = 4:[]
= 8:zeharkatu(5:4:[]) = (#4)	x = 5, s = 4:[], leh(s) = 4, hond(s) = []
= 8:4:zeharkatu(5:[]) = (#2)	x = 5, s = []
= 8:4:5:[] = [8, 4, 5]	

zeharkatu([5, 5, 4]) =	
= zeharkatu(5:5:4:[]) = (#3)	x = 5, s = 5:4:[], leh(s) = 5, hond(s) = 4:[]
= 0:0:4:[] = [0, 0, 4]	

**35)zatitu (2008ko ekaina) -- #****a)**

zatitu:: ([Int]) → [Int]

zatitu([]) = error "Zerrenda hutsa" (#1)

zatitu(x:s)		
x == 0	= error "Lehenengoa zero da"	(#2)
hutsa_da(s)	= s	(#3)
leh(s) `mod` x == 0	= (leh(s) `div` x):zatitu(x:hond(s))	(#4)
otherwise	= leh(s):zatitu(x:hond(s))	(#5)

**b)**

zatitu([5, 20, 6, 15]) =	
= zatitu(5:20:6:15:[]) = (#4)	x = 5, s = 20:6:15:[],
	leh(s) = 20, hond(s) = 6:15:[]
= 4:zatitu(5:6:15:[]) = (#5)	x = 5, s = 6:15:[], leh(s) = 6, hond(s) = 15:[]
= 4:6:zatitu(5:15:[]) = (#4)	x = 5, s = 15:[], leh(s) = 15, hond(s) = []
= 4:6:3:zatitu(5:[]) = (#3)	x = 5, s = []
= 4:6:3:[] = [4, 6, 3]	

**36)superbik (2008ko iraila) -- #****a)**

superbik:: ([Int]) → [Int]

superbik([]) = [] (#1)

superbik(x:s) (#2)

hutsa_da(s)	= x:s	(#2)
x `mod` 2 == 0 && leh(s) `mod` 2 == 0		

	= superbik((x + leh(s)):hond(s))	(#3)
--	----------------------------------	------

x `mod` 2 == 0	= leh(s):superbik(x:hond(s))	(#4)
----------------	------------------------------	------

otherwise	= x:superbik(s)	(#5)
-----------	-----------------	------

**b)**

superbik([3, 10, 8, 9]) =  
 = superbik(3:10:8:9:[]) = (#5)

= 3:superbik(10:8:9:[]) = (#3)

= 3:superbik(18:9:[]) = (#4)

= 3:9:superbik(18:[]) = (#2)

= 3:9:18:[] = [3, 9, 18]

x = 3, s = 10:8:9:[],

leh(s) = 10, hond(s) = 8:9:[]

x = 10, s = 8:9:[], leh(s) = 8, hond(s) = 9:[]

x = 18, s = 9:[], leh(s) = 9, hond(s) = []

x = 18, s = []

**37)metatu (2009ko apirila #1) -- #****a)**

metatu :: ([Int]) → ([Int])

metatu([]) = [] (#1)

metatu(x:s)

| hutsa\_da(s) = x:[] (#2)

| otherwise = x: metatu((x + leh(s)):hond(s)) (#3)

**b)**

metatu([10, 8, 15]) = metatu(10:8:15:[]) = (#3)

= 10:metatu(18:15:[]) = (#3)

= 10:18:metatu(33:[]) = (#2)

= 10:18:33:[]

Adibidea garatzerakoan, urrats bakoitzean lerroaren eskuineko ertzean zehazten den ekuazioa formularen zein zatiri aplikatu zaion garbiago ikusteko koloreak erabili dira. Irakurgarritasunagatik bi kolore desberdin erabili dira baina kolore desberdinek ez dute inolako esanahi berezirik.

**38)aurreratu (2009ko apirila #2) -- #****a)**

aurreratu :: ([Bool], [Int]) → [Int]

aurreratu([], r)

| luzera(r) ≠ 0 = error "Luzera desberdina" (#1)

| otherwise = [] (#2)

aurreratu(x:s, r)

| luzera(x:s) ≠ luzera(r) = error "Luzera desberdina" (#3)

| hutsa\_da(s) = r (#4)

| x == False = leh(r):aurreratu(s, hond(r)) (#5)

| x == True = leh(hond(r)):aurreratu(s, leh(r):hond(hond(r))) (#6)

**b)**

aurreratu([True, False, True, True, True], [8, 3, 9, 5, 2]) = (#6)

= 3:aurreratu([False, True, True, True], [8, 9, 5, 2]) = (#5)

= 3:8:aurreratu([True, True, True], [9, 5, 2]) = (#6)

= 3:8:5:aurreratu([True, True], [9, 2]) = (#6)

= 3:8:5:2:aurreratu([True], [9]) = (#4)

= 3:8:5:2:9:[] = [3, 8, 5, 2, 9]

**39)kendubikpos (2009ko ekaina) -- #****a)**

kendubikpos:: ([Int], Int) → [Int]

kendubikpos([], pos) = error "Zerrenda hutsa" (#1)

kendubikpos(x:s, pos)

| pos &lt; 1 || pos &gt; luzera(x:s) = error "Posizioa ez da egokia" (#2)

| pos == 1 &amp;&amp; x mod 2 == 0 = s (#3)

| pos == 1 &amp;&amp; x mod 2 /= 0 = x:s (#4)

| otherwise = x:kendubikpos(s, pos - 1) (#5)

**b)**

kendubikpos([8, 5, 16, 7, 10, 4], 3) =

= kendubikpos(8:5:16:7:10:4:[], 3) = (#5)

= 8:kendubikpos(5:16:7:10:4:[], 2) = (#5)

= 8: 5:kendubikpos(16:7:10:4:[], 1) = (#3)

= 8: 5: 7:10:4:[] =

= [8, 5, 7, 10, 4]

**40)azk\_lehena (2009ko iraila) -- #****a)**

azk\_lehena:: ([Int]) → Int

azk\_lehena([]) = -1 (#1)

azk\_lehena(x:s)

| not lehena\_da(x) = azk\_lehena(s) (#2)

| hutsa\_da(s) = x (#3)

| lehena\_da(leh(s)) = azk\_lehena (s) (#4)

| not lehena\_da(leh(s)) = azk\_lehena(x:hond(s)) (#5)

(#3), (#4) eta (#5) ekuazioetan *lehena\_da(x)* betetzen da. (#4) eta (#5) ekuazioetan *not hutsa\_da(s)* betetzen da. (#5) ekuazioan *otherwise* ipini daiteke *not lehena\_da(leh(s))* baldintzaren ordeaz.

**b)**

azk\_lehena([8, 11, -7, 9, 3, 6]) =

= azk\_lehena(8:11:-7:9:3:6:[]) = (#2)

= azk\_lehena(11:-7:9:3:6:[]) = (#5)

= azk\_lehena(11:9:3:6:[]) = (#5)

= azk\_lehena(11:3:6:[]) = (#4)

= azk\_lehena(3:6:[]) = (#5)

= azk\_lehena(3:[]) = (#3)

= 3



**41)azpiluzbikgehi (2010eko apirila #1) -- #****a)**

azpiluzbikgehi:: ([t]) → ([t])

azpiluzbikgehi([]) = [] (#1)

azpiluzbikgehi(x:s)

| hutsa\_da(s) = x:x:[] (#2)

| x == leh(s) = x:x:azpiluzbikgehi(hond(s)) (#3)

| x /= leh(s) = x:x: azpiluzbikgehi(s) (#4)

**b)**

```

azpiluzbikgehi([10, 10, 10, 8, 8, 15, 8]) =
= azpiluzbikgehi(10:10:10:8:8:15:8:[]) = (#3)
= 10:10:azpiluzbikgehi(10:8:8:15:8:[]) = (#4)
= 10:10:10:10:azpiluzbikgehi(8:8:15:8:[]) = (#3)
= 10:10:10:10:8:8:azpiluzbikgehi(15:8:[]) = (#4)
= 10:10:10:10:8:8:15:15:azpiluzbikgehi(8:[]) = (#2)
= 10:10:10:10:8:8:15:15:8:8:[] =
= [10, 10, 10, 10, 8, 8, 15, 15, 8, 8]

```

**42)handibik (2010eko apirila #2) -- #****a)**

handibik:: ([Int]) → ([Int])

handibik([]) = [] (#1)

handibik(x:s)

| luzera(x:s) mod 2 /= 0 = error "Luzera bakoitia" (#2)

| x &gt;= leh(s) = x: x: handibik(hond(s)) (#3)

| x &lt; leh(s) = leh(s):leh(s):handibik(hond(s)) (#4)

**b)**

```

handibik([10, 8, 5, 7, 7, 20, 5, 5]) =
= handibik (10:8:5:7:7:20:5:5:[]) = (#3)
= 10:10:handibik(5:7:7:20:5:5:[]) = (#4)
= 10:10:7:7:handibik(7:20:5:5:[]) = (#4)
= 10:10:7:7:20:20:handibik(5:5:[]) = (#3)
= 10:10:7:7:20:20:5:5:handibik([]) = (#1)
= 10:10:7:7:20:20:5:5:[] =
= [10, 10, 7, 7, 20, 20, 5, 5]

```

**43)kokatu (2010eko ekaina) -- #****a)**

kokatu:: ([Int], [Int]) → [Int]

kokatu([], ℓ) = [] (#1)

kokatu(x:s, ℓ)

| aldiz(0, x:s) &gt; luzera(ℓ) = error "Luzera ez da egokia" (#2)

| x == 0 = x:leh(ℓ):kokatu(s, hond(ℓ)) (#3)

| x /= 0 = x:kokatu(s, ℓ) (#4)

**b)**

kokatu([8, 0, 0, 7], [3, 20, 12, 28])

= kokatu(8:0:0:7:0:6:[], 3:20:12:28:[]) = (#4)

= 8:kokatu(0:0:7:[], 3:20:12:45:28:[]) = (#3)

= 8: 0: 3:kokatu(0:7:[], 20:12:45:28:[]) = (#3)

= 8: 0: 3: 0: 20:kokatu(7:[], 12:45:28:[]) = (#4)

= 8: 0: 3: 0: 20: 7:kokatu([], 12:45:28:[]) = (#1)

= 8: 0: 3: 0: 20: 7: [] = [8, 0, 3, 0, 20, 7]

**44)azpiluzbikken (2010eko iraila) -- #****a)**

azpiluzbikken:: ([t]) → ([t])

azpiluzbikken([]) = [] (#1)

azpiluzbikken(x:s)

| hutsa\_da(s) = [] (#2)

| x == leh(s) = x:x:azpiluzbikken(hond(s)) (#3)

| x /= leh(s) = azpiluzbikken(s) (#4)

**b)**

azpiluzbikken([10, 10, 10, 8, 8, 15, 8]) =

= azpiluzbikken(10:10:10:8:8:15:8:[]) = (#3)

= 10:10:azpiluzbikken(10:8:8:15:8:[]) = (#4)

= 10:10:azpiluzbikken(8:8:15:8:[]) = (#3)

= 10:10:8:8:azpiluzbikken(15:8:[]) = (#4)

= 10:10:8:8:azpiluzbikken(8:[]) = (#2)

= 10:10:8:8:[ ] =

= [10, 10, 8, 8]