

1 (3 puntu)

Node eta LinkedList klaseak emanda (eranskin I), eta zerrendan gordeta dauden elementuak **Integer** datu motakoak izanda suposatuz, diseina eta inplementatu hurrengo metodoa LinkedListItr klasean:

```
public class LinkedList{
    LinkedList birkokatu(int zenb)
}
```

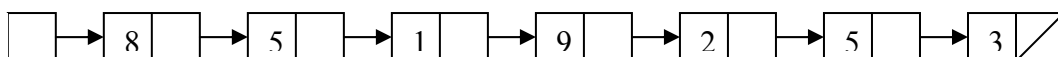
Metodo honek zerrendako zenbaki guztiak birkokatzen ditu zerrenda berria batean. Zerrenda berri honetan, *zenb* baino txikiagoak edo berdinak diren elementuak ezkerrean kokatuko dira eta *zenb* baino handiagoak direnak eskuinean. Prozesuaren bukaeran elementuen ordenak ez du axola, betiere aipatu den baldintza betetzen bada.

O(N) ordenako (non **N** hasierako zerrendako elementuen kopurua den) ebazpenak hobeto baloratuko dira puntuazioan.

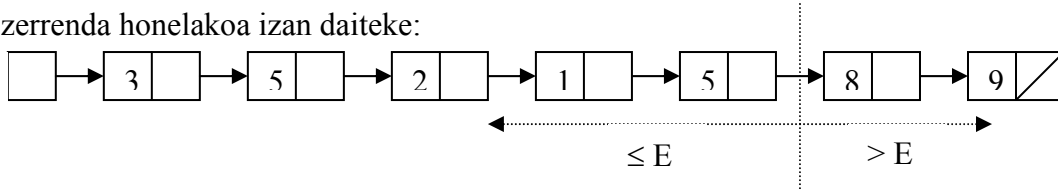
Adibidea:

Izan bitez

zenb = 7



Emaitza zerrenda honelakoa izan daiteke:



2 (3 puntu)

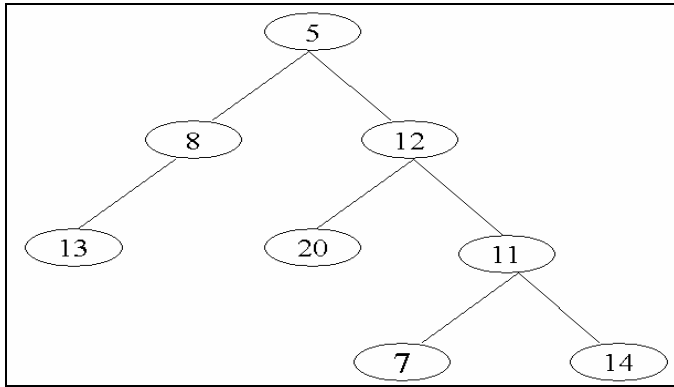
BTNode eta **BinTree** klaseak emanda (eranskina II), diseina eta inplementatu hurrengo emaitza itzultzen duen metodo bat **BinTreeItr** klasean:

- zuhaitzaren zenbaki lehen handiena (ez badauka zenbaki lehenik, 0 itzuliko du metodoak)
- zuhaitzak dauzkan zenbaki lehenen kopurua

Suposatu honako metodo hau inplementatuta daukagula **Integer** klasean:

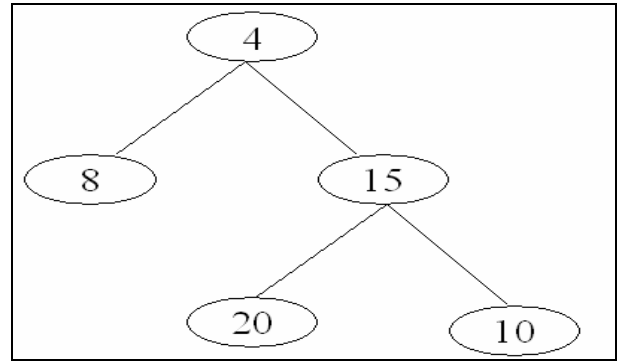
```
-- post: Egiazkoa, n zenbaki lehena bada, bestela, faltsua.
boolean islehena()
```

1. adibidea:



Lehenik handiena: 13
Lehen kopurua: 4

2. adibidea:



Lehenik handiena: 0
Lehen kopurua: 0

(4 puntu)

PrinterManager klaseak inprimatzera doazen fitxategien izenak (*String*) kudeatzeko erabiltzen da. Inprimatzeko sistemak hiru inprimagailu erabiltzen ditu (*int*). Fitxategi bakoitzak lehentasun bat dauka esleituta (*int 1tik 4ra*).

Inprimagailuen kudeatzailea ilaren matrize baten bidez dago implementatuta. Bi dimentsioko taula *Inprimagailu* motak eta *Lehentasun* motak indexatzen dute. Matrizeko elementu bakoitza ilara bat da, indizeek adierazten duten inprimagailuan eta lehentasunarekin inprimatuko diren fitxategien izenak gordetzen dituena.

	1	2	3	4
1	F1.ADS F7.ADB			F8.ADS
2	F2.ADB	F6.PAS		
3	F3.PAS	F4.ADS F5.ADB F9.ADB		

Honakoa eskatzen da:

a) Erazagutu Javan **PrinterManager** klasea. Erazagutu, halaber, behar dituzun osagarritzko klaseak.

b) Diseinatu eta inplementatu honako metodo hau:

```
void lehentasunaAldatu (String f, int lb)
```

lehentasunaAldatu metodoak *f* fitxategia *lb* lehentasuneko ilararen hasieran kokatzen du, zegoen inprimagailutik aldatu gabe.

f fitxategia lehendik ez bazegoen, "Fitxategi ez aurkitua" mezua idazten du, eta inprimagailuen kudeatzailea zegoen bezala uzten du.

f fitxategiaren aurreko lehentasuna eta *lb* lehentasun berria berdina badira, "Lehentasun berdina" mezua idazten du, eta inprimagailuen kudeatzailea zegoen bezala uzten du.

Aurreko adibidetik,

```
pm.lehentasunaAldatu ("F5.ADB", 1)
```

exekutatu balitz, honako emaitza lortuko litzateke:

	1	2	3	4
1	F1.ADS F7.ADB			F8.ADS
2	F2.ADB	F6.PAS		
3	F5.ADB F3.PAS	F4.ADS F9.ADB		

Eranskin I: NodeList eta LinkedList klaseen espezifikazioa

OHAR: Definititutako metodoak erabili daiteke soilik. Beste edozein metodo inplementatu beharko da..

```
class NodeList<T>
{
    T element;
    NodeList next;
}
//actual lehenengo posizioan kokatzen da
public void goFirst()
//actual hurrengo posizioan kokatzen da
public void goNext()
// True, elementu gehiago badaude eta
// false beste kasuan
public boolean hasMoreElements()
}

public class LinkedList<T> {
    NodeList<T> top;
    NodeList<T> actual;
    NodeList<T> previous;
}
```

Eranskin II: BTNode eta BinTree klaseen espezifikazioa

OHAR: Behar diren metodo guztiak inplementu beharko dira.

```
public class BTNode<T> {
    Object content;
    BTNode left;
    BTNode right;
}
public class BinTree<T> {
    BTNode<T> root;
}
```