

EJERCICIO 2.- Cola de grupos (1,5 puntos)

En la cafetería de la EUITI, los alumnos hacen cola para hacer su pedido. Sin embargo, el sistema utilizado es distinto al de una cola normal. Cada alumno pertenece a un grupo y, cuando llega a la cola, si encuentra alguna persona de su grupo, se coloca al final de ese grupo. En caso de que no haya en la fila nadie perteneciente a su grupo, se colocará al final de la cola.

Por ejemplo, supongamos que tenemos estos 3 grupos (para simplificar, identificamos a cada alumno por un valor entero):

- Grupo A: 1, 5, 7
- Grupo B: 2, 4, 8, 10
- Grupo C: 3, 6, 11

Partiendo de una cola inicial vacía, esta sería una posible secuencia de acciones:

Evento	Cola	Acción
	vacía	
Entrada 2	2	
Entrada 6	2, 6	
Entrada 8	2, 8 , 6	
Entrada 5	2, 8, 6, 5	
Entrada 4	2, 8, 4 , 6, 5	
Entrada 1	2, 8, 4, 6, 5, 1	
Entrada 3	2, 8, 4, 6, 3 , 5, 1	
Salida	8, 4, 6, 3, 5, 1	devuelve 2
Salida	4, 6, 3, 5, 1	devuelve 8
Salida	6, 3, 5, 1	devuelve 4
Entrada 11	6, 3, 11 , 5, 1	
...		

Se pide **implementar** los siguientes subprogramas y **calcular el coste** de cada algoritmo:

```
public class Cafeteria {  
  
    Queue<Queue<Integer>> cola = null; // una cola de colas  
  
    public void insertar(Integer elem)  
        // Postcondición: se ha añadido "elem" en la cola de la cafetería  
  
    public Integer borrar()  
        // Precondición: se tiene al menos un elemento en la cola  
        // Postcondición: se ha eliminado el primer elemento de la cola y  
        // se devuelve su valor  
  
    public boolean mismoGrupo(int p1, int p2)  
        // Postcondición: true si p1 y p2 pertenecen al mismo grupo y false si no  
        // NO HAY QUE IMPLEMENTAR ESTE MÉTODO  
  
} // Cafeteria
```

ARIKETA 2.- Taldeen ilara (1,5 puntu)

Ingeniaritza eskolako kafetegian ikasleak ilaran egoten dira beraien eskaria egiteko. Baina ilara honen funtzionamendua ez da ilara arrunt baten berdina. Ikasle bakoitza talde batekoa da eta, ilarara iristen denean, bere taldeko kide bat aurkitzen baldin bada, talde horren bukaeran jartzen da. Ilara osoan ez balego bere taldeko inor, orduan ilararen bukaeran jarriko da.

Adibidez, demagun 3 talde hauek dauzkagula (adibidea errazteko, ikasle bakoitza zenbaki baten bidez izendatuko dugu):

- Talde A: 1, 5, 7

- Talde B: 2, 4, 8, 10

- Talde C: 3, 6, 11

Hasierako ilara hutsa baldin bada, hau izan liteke ekintza-sekuentzia bat:

Gertaera	Ilara	Ekintza
	hutsa	
Sarrera 2	2	
Sarrera 6	2, 6	
Sarrera 8	2, 8 , 6	
Sarrera 5	2, 8, 6, 5	
Sarrera 4	2, 8, 4 , 6, 5	
Sarrera 1	2, 8, 4, 6, 5, 1	
Sarrera 3	2, 8, 4, 6, 3 , 5, 1	
Irteera	8, 4, 6, 3, 5, 1	2 bueltatuko da
Irteera	4, 6, 3, 5, 1	8 bueltatuko da
Irteera	6, 3, 5, 1	4 bueltatuko da
Sarrera 11	6, 3, 11 , 5, 1	
...		

Ondorengo azpiprogramak **implementatu** behar dira, eta bakoitzaren **kostua kalkulatu**:

```
public class Kafetegia {  
  
    Queue<Queue<Integer>> ilara = null; // ilararen ilara  
  
    public void txertatu(Integer elem)  
    // Postbaldintza: "elem" gehitu da kafetegiko ilaran  
  
    public Integer ezabatu()  
    // Precondición: gutxienez pertsona bat dago ilaran  
    // Postbaldintza: ilarako lehen elementua kendu da eta bere balioa bueltatu da  
  
    public boolean taldeBera(int p1, int p2)  
    // Postbaldintza: true baldin p1 eta p2 talde berekoak badira eta false bestela  
    // EZ DA METODO HAU INPLEMENTATU BEHAR  
  
} // Kafetegia
```