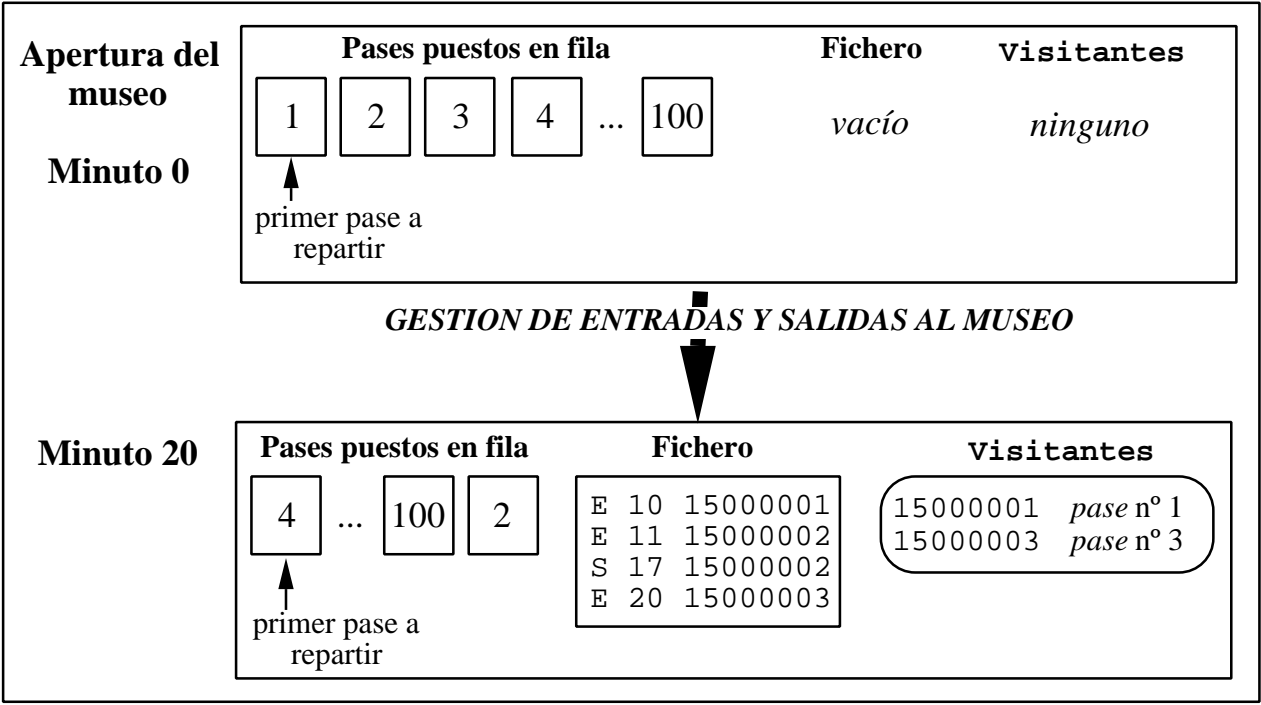


(S94) Para entrar al Museo de las joyas de la Castafiore es preciso tener un pase numerado que se recoge y entrega en la portería. El portero que gestiona los pases va creando un fichero de texto con los datos de entrada y salida de visitantes. La gestión de los pases consiste en lo siguiente:

- Todas las mañanas antes de abrir (minuto cero) se colocan todos los pases en fila y en orden creciente del 1 al 100.
- Cuando entra una persona se le entrega *el primer pase de la fila* y se añade al fichero el movimiento de entrada correspondiente. Por ejemplo el movimiento: **E 330 15999888** significa que la persona de DNI **15999888** entró ('E') al museo en el minuto **330** (integer) tras la apertura del museo (minuto 0).
- Cuando sale una persona se le recoge el pase, se coloca éste *al final de la fila de pases* y se escribe el movimiento de salida correspondiente. Por ejemplo: **S 360 15999888** significa que la persona de DNI **15999888** salió ('S') en el minuto **360**.



Una noche, ante la sospecha de un robo y una vez que el fichero F_MVTOS ha sido cerrado con todos los movimientos del día, el detective del museo quiere saber qué personas estaban visitando el museo al término de un minuto concreto y cuál era el número del pase que portaba cada una (esto último, aunque misterioso para nosotros, resulta vital para sus investigaciones, insiste el detective).

Se pide:

- a) **Escoger uno de los TAD conocidos** para representar la *fila de pases*.
Explicar por qué se elige y de qué tipo son sus elementos.
- b) Nos dan la especificación de la clase Visitantes:

Clase Visitantes
<pre>public class Visitantes extends Object {</pre>
public void apertura () Inicializa el <i>grupo de visitantes</i> a vacío
public void entrada(Long dni, Integer pase) Añade el visitante de <i>dni</i> y numero de <i>pase</i> dados al <i>grupo de visitantes</i> .
public void salida(Long dni) Elimina del <i>grupo de visitantes</i> a aquél que tiene el <i>dni</i> dado.
ver_primero(in t_visitantes; out t_dni; out t_pase) Da como salida el número de <i>dni</i> y el número de <i>pase</i> del visitante que fue introducido el primero respecto a los que están en este momento en el <i>grupo de visitantes</i> .
public Integer ver_pase(Long dni) Dado un <i>dni</i> de un visitante que se encuentra en el <i>grupo de visitantes</i> devuelve el número de pase que le fue asignado.
public boolean hay_alguien() Devuelve falso si el <i>grupo de visitantes</i> es vacío y cierto si hay algún visitante en ese grupo.

Utilizando el **TAD** elegido en el apartado a) y la clase Visitantes, **parametriza, diseña e implementa un método de la clase Visitantes** tal que dado un minuto y partiendo de la información del fichero F_MVTOS escriba en la salida estandar el DNI y el número de pase de las personas que se encontraban visitando el museo en el minuto dado.

(S96) Todas las mañanas, en la oficina de Correos de Villa García, se hace un reparto de la correspondencia entre los diez carteros que forman parte de la plantilla. Cada uno de ellos tiene asociado un distrito postal (desde el **10.001** hasta el **10.010**).

El reparto de las cartas entre los carteros se hace de la siguiente manera:

- La asignación de cartas se hace guardando el orden de llegada a la oficina de Correos.
- Cada cartero podrá tener asignadas como máximo **200** cartas.
- Todas las mañanas comienza el reparto de cartas a los diferentes carteros según su distrito. Es decir, si una carta tiene distrito 10.004 se le asigna al cartero que tiene asociado dicho distrito. Esto es así, siempre y cuando al cartero en cuestión se le hayan asignado menos de 200 cartas.
- Las cartas que no se pueden asignar en el proceso descrito anteriormente se distribuirán entre los carteros con arreglo a otro tipo de criterios.

La oficina de Correos quiere que desarrollemos un programa que realice el reparto descrito anteriormente y obtenga información útil para realizar el reparto de las cartas que en su caso queden por asignar.

Se pide:

a) Seleccionar la o las **estructuras de datos** que permitan el reparto de la correspondencia y la obtención de las salidas que nos piden a continuación. Tened en cuenta que se pueden usar los TADs conocidos.

b) Utilizando la estructura seleccionada, **diseñar e implementar** un programa tal que dado el fichero de texto **CARTAS.TXT** que contiene la información sobre las cartas a repartir, introducidas **por orden de llegada**, cree los siguientes ficheros de texto, donde la información aparece **agrupada por distrito** y entre las direcciones de un mismo distrito se **mantiene el orden de llegada**:

- **REPARTO.TXT** que contiene la información sobre las cartas que han sido asignadas a cada cartero.

- **RESTO.TXT** que contiene la información sobre las cartas que **no** han sido asignadas todavía.

En la cabecera de cada distrito aparece el **nº de cartas** que quedan por asignar de ese distrito (antes de escribir las direcciones). Si todas las cartas se han podido asignar ya al cartero, en su lugar figura el **nº de cartas** que se le podrían asignar a este hasta completar el total de 200 cartas asignables.

Formato de los ficheros

El formato del fichero de entrada **CARTAS.TXT** (direcciones en orden de llegada) es:

10002 Gran vía 3
10002 Bailén 20
10001 Sicilia 11
10002 Gran vía 3
...

El formato del fichero de salida **REPARTO.TXT** (direcciones agrupadas por distrito. Las direcciones de cada grupo mantienen entre sí el orden de llegada a la oficina) es:

Cartero 10001	
10001	Sicilia 11
10001	...
Cartero 10002	
10002	Gran vía 3
10002	Bailén 20
10002	...
Cartero 10003	
10003	...

El formato del fichero de salida **RESTO.TXT** (direcciones agrupadas por distrito. Las direcciones de cada grupo mantienen entre sí el orden de llegada a la oficina. Las cabeceras incluyen el número de direcciones que se escriben por detrás, o en su caso el número de cartas que se podrían asignar al cartero hasta completar las 200) es:

Distrito 10001: 2 10001 Ortega 7 10001 Velazquez 4	→	Quedan 2 cartas por repartir, ya que el cartero tiene asignadas las 200
Distrito 10002: -5	→	Todas las cartas repartidas y el cartero tiene asignadas 195
Distrito 10003: 0	→	Todas las cartas repartidas y el cartero tiene asignadas las 200
Distrito 10004: 4 10004 Gaudi 3 10004 Urdaneta 20 10004 Ramon y Cajal 30 10004 Cervantes 7		
Distrito 10005: -10 ...		

(J97) En una vieja mansión escocesa vive una familia de fantasmas (cada uno de ellos tiene su DNI). Los duros tiempos de crisis han hecho que los dueños de la morada hayan decidido convertirla en hotel, disponiendo de **25** habitaciones. Ante la perspectiva de propinar sustos a los confiados huéspedes, los fantasmas se han organizado por habitaciones, de forma que:

- Cada uno de los fantasmas tiene asignada una sola habitación.
- Cada habitación puede tener asignados varios fantasmas diferentes.
- Una habitación recibe exactamente un susto por noche siempre y cuando haya algún fantasma asignado a ella.

Los fantasmas trabajan por riguroso orden: cuando una habitación está ocupada una noche, un fantasma asociado a esa habitación da un susto, aguardando para dar otro a que el resto de sus compañeros lo hayan hecho.

Todos los fantasmas tienen el mismo cupo de sustos: **10**. Cuando alguno consigue dar los 10 sustos, pasa a mejor vida y deja el hotel.

La oficina del hotel, en el fichero de texto **INICIO.TXT**, tiene información de la distribución inicial de los fantasmas en las habitaciones, junto con el número de sustos que lleva ya dados cada uno de ellos. Los primeros fantasmas que aparecen son los que antes pueden dar un susto. Además disponen de otro fichero de texto denominado **RESERVAS.TXT**, donde aparecen todas las reservas de las habitaciones durante una temporada.

Se pide:

- a) Selecciona la o las **estructuras de datos** necesarias para la obtención de la salida. Ten en cuenta que puedes utilizar los TAD conocidos. Por cada TAD utilizado indica el tipo de sus elementos.
- b) Utilizando la o las estructuras seleccionadas, desarrolla el programa que nos piden tomando los datos de los ficheros de texto **INICIO.TXT** y **RESERVAS.TXT**, y devolviendo por la **salida estándar** (pantalla):
- La relación de los DNI de los fantasmas que han pasado a mejor vida en esa temporada.
 - El número de la habitación donde más fantasmas han pasado a mejor vida en esa temporada (si hubiera varias, una cualquiera de ellas).

Formato de los ficheros:

El fichero **INICIO.TXT** contiene una línea por cada fantasma, en *orden de prioridad para dar un susto*. La información de cada línea consta del DNI del fantasma, el número de sustos dado hasta el momento por el mismo y el número de habitación que tiene asignada.

12554231	2	14
13455904	1	25
15667231	8	1
...		
19563211	9	14

DNI
del fantasma

nº de sustos dados

nº de habitación

El fichero **RESERVAS.TXT** contiene una línea por reserva de habitación en *orden de llegada de los huéspedes*. Cada línea consta del número de habitación y el número de noches de la reserva.

25	3
12	5
10	1
...	
5	4

nº de habitación

nº de noches reservadas

Por la **salida estándar** se escribirá una línea por fantasma que pase a mejor vida (10 sustos en su haber), y la habitación que más fantasmas han pasado a mejor vida en esa temporada:

```
Fantasmas que han pasado a mejor vida:
13455904
19563211
...
habitación con más fantasmas hacia mejor vida:
14
```

(S97) Un centro instala una nueva centralita telefónica que le permite detectar cada número de teléfono al que se hace una llamada, la extensión desde la que se produce, además de medir el tiempo de cada llamada.

Nos piden que desarrollemos un programa para conocer cuál es **el teléfono al que más tiempo se llama** considerando el tiempo total de todas las llamadas realizadas al mismo. también interesa **conocer la información de todas las llamadas realizadas a ese teléfono**, esto es la extensión desde la que se llamaba y la duración de cada conversación, en el *orden en el que se produjeron*.

Se pide:

a) Selecciona la o las **estructuras de datos** necesarias para la obtención de la salida *si los teléfonos fueran un máximo de 100* y además fuesen *consecutivos* (ejemplo desde el teléfono 943000000 al 943000100).

Trata de utilizar alguno de los TAD conocidos. Por cada TAD utilizado indica el tipo de sus elementos.

b) Utilizando la o las estructuras seleccionadas en el apartado **a)**, desarrolla el programa (todos los pasos hasta la implementación en Java) que nos piden tomando los datos de las llamadas realizadas del fichero de texto **LLAMADAS.TXT** y creando el fichero de texto de salida **TIEMPO_MAX.TXT**.

c) Què estructura o estructuras elegirías *si los teléfonos NO fueran un máximo de 100*.

Trata de utilizar alguno de los TAD conocidos. Por cada TAD utilizado indica el tipo de sus elementos.

Formato de los ficheros

El fichero **LLAMADAS.TXT** contiene una línea por cada llamada realizada y en el *orden en el que se produjeron*. La información de cada línea consta del teléfono al que se llama (9 cifras), extensión de la que parte la llamada (3 cifras) y duración en minutos. En cada línea los números vienen separados entre sí por un espacio en blanco.

```
948750000 314 2
943218000 235 10
948750000 314 3
943007789 135 3
948750000 093 5
944700988 314 12
943218000 211 7
948750000 122 30
...
```

El fichero **TIEMPO_MAX.TXT** contiene información de todas las llamadas realizadas al número al que más se llama basándose en el tiempo total de conversación. La información de las llamadas aparece en el *orden en el que se produjeron*. La cabecera del fichero incluye el número del teléfono al que se dirigen las llamadas y el tiempo total. Cada línea incluye la extensión desde la que se llama y el tiempo de esa conversación.

```
número: 948750000  
tiempo total: 2000
```

```
Lista de llamadas:
```

```
314 2  
314 3  
093 5  
122 30  
...
```