Ejercicio 3. Smart farming.

(3 puntos) El sector agrícola está empezando a aplicar técnicas de bigdata para mejorar la productividad. Vamos a diseñar un sistema que nos permita detectar diversas plagas para proceder a fumigar adecuadamente las zonas en las que se producen. La cooperativa agrícola ha dividido sus tierras en sectores, en los que ha situado unos robots capaces de diferenciar distintos tipos de plagas. Cuando detectan una plaga los robots envían al sistema la plaga detectada, el sector en el que está y el número de plantas que están afectadas. Un sector puede tener sus plantas afectadas por varios tipos de plagas. La cooperativa de vez en cuando alquila una avioneta y fumiga los sectores más afectados.

La implementación hará uso de los tipos `sector` y `plaga` que se representan con un `string`.

Las operaciones son las siguientes:

- `alta(id, n)`: da de alta un nuevo sector `id` con `n` plantas. Si el sector ya estaba dado de alta se incrementa su número de plantas con el valor recibido como parámetro.
- `datos(id,p,n)`: el sistema recibe notificación de una plaga `p` desde el sector `id` con `n` plantas afectadas. Las plantas afectadas por una plaga continuan afectadas hasta que se fumiga ese sector, por lo tanto si el robot manda varios avisos sobre la misma plaga en el mismo sector antes de que haya una fumigación cada aviso acumulará las plantas del aviso anterior (el número de plantas del aviso anterior para esa plaga en ese sector se desprecia porque está incluido en los nuevos datos).

  Si el `id` del sector no está dado de alta en el sistema se lanzará una excepción con el mensaje `Sector no existente`. Si la plaga no se encuentra registrada en ese sector se registrará y si el número de plantas afectadas es mayor que el número de plantas del sector se lanzará una excepción con el mensaje `Numero de plantas incorrecto`.
- `fumigar()`: la operación realiza dos acciones. Por un lado, devuelve un vector con los sectores que tienen un cuarto, o más, de sus plantas afectadas por una plaga en el momento en que se produce la notificación (infectadas >= plantas / 4.0). Por otro lado, realiza la fumigación de esos sectores.

  En el vector se indicará el sector afectado y la plaga de que se trata. Los sectores aparecen en el vector *en el orden en que fueron alcanzando el cuarto de sus plantas afectadas*. El vector no tendrá elementos repetidos, si un par (sector, plaga) ya se encuentra en el vector no se añadirá otra vez. Recuerda que las plantas solo sanan cuando se realiza esta operación, mientras tanto continuan infectadas.

  Queue.h

  Si no hay ningún sector en estas condiciones se devuelve el vector vacío.

  Los sectores fumigados contra una plaga no necesitan volver a fumigarse contra esa plaga hasta que vuelvan a tener un cuarto de sus plantas afectadas. Se supone que al fumigar todas las plantas quedan sanas.
- `plagas(id)`: Obtiene en un vector todas las plagas que ha sufrido el sector que se da cómo parámetro, desde que se dió de alta en la aplicación en orden alfabético, junto a cada plaga se indicará el número actual de plantas afectadas por la plaga que todavía no han sido tratadas. Si el `id` del sector no está dado de alta en el sistema se lanzará una excepción con el mensaje `Sector no existente`.

Requisitos de implementación.

Seleccionar los tipos de datos adecuados para representar la información y obtener una solución eficiente. En la cabecera de cada función debe indicarse el coste de la misma.

Los métodos del TAD no deben mostrar nada por pantalla. El manejo de la entrada y salida de datos se realizará en funciones externas al TAD.

Debe justificarse el tipo representante elegido y dar el coste de todas las operaciones

Entrada

La entrada consta de una serie de casos de prueba. Cada caso está formado por una serie de líneas, en las que se muestran las operaciones a llevar a cabo, una por cada línea: el nombre de la operación seguido de sus argumentos. La palabra FIN en una línea indica el final de cada caso.

Salida

Para cada caso de prueba se escribirán los datos que se piden. Las operaciones que generan datos de salida son:

- `fumigar`, que debe escribir una línea del tipo `Fumigar la plaga` *nombre de la plaga* `del sector` *nombre del sector* por cada plaga que vaya a fumigar. Debe dejarse un caracter blanco a ambos lados de los dos puntos.
- `plagas`, que debe escribir `Plagas del sector` *nombre del sector* `:` seguido de las plagas y su número de plantas afectadas separadas por un caracter blanco y en orden alfabético. Debe dejarse un caracter blanco a ambos lados de los dos puntos.

Cada caso termina con una línea con tres guiones (`---`).

Si alguna operación produce una excepción se mostrará el mensaje de la excepción como resultado de la operación.

Entrada de ejemplo

```
alta sector3 100
alta sector2 200
datos sector2 pulgon 100
datos sector2 cochinilla 20
datos sector3 cochinilla 50
fumigar
datos sector2 cochinilla 500
datos sector2 cochinilla 100
datos sector3 pulgon 50
fumigar
alta sector3 50
datos sector3 cochinilla 120
plagas sector3
plagas sector2
fumigar
FIN
```

Salida de ejemplo

```
Fumigar la plaga pulgon del sector sector2
Fumigar la plaga cochinilla del sector sector3
ERROR: Numero de plantas incorrecto
Fumigar la plaga cochinilla del sector sector2
Fumigar la plaga pulgon del sector sector3
Plagas del sector sector3 : cochinilla 120 pulgon 0
Plagas del sector sector2 : cochinilla 0 pulgon 0
Fumigar la plaga cochinilla del sector sector3
---
```

Autor: Isabel Pita