A
El Grande

El Grande es un juego de mesa ambientado en la España medieval. En este juego, los jugadores tienen que colocar estratégicamente a sus caballeros por las distintas regiones del país. Decimos que un jugador *predomina* en una región cuando tiene **estrictamente** más caballeros en dicha región que el resto de jugadores. Por otro lado, decimos que una región está *en disputa* si ningún jugador predomina en la misma.

En este ejercicio trataremos de implementar un TAD que almacene la información correspondiente a una partida de este juego. Las operaciones a implementar son:

- `anyadir_jugador(jugador)`. Añade un jugador a la partida. Si el `jugador` (de tipo `string`) ya estaba inscrito en ella, se lanza una excepción `domain_error` con el mensaje `Jugador existente`.
- `colocar_caballero(jugador, region)`. Indica que el `jugador` (de tipo `string`) coloca un caballero en la `region` (de tipo `string`) indicada. Si el `jugador` no se encuentra inscrito en la partida, se lanza la excepción `domain_error` con el mensaje `Jugador no existente`.
- `puntuacion(jugador)`. Devuelve el número de regiones en las que predomina el `jugador` pasado como parámetro. Si el jugador no se encuentra inscrito en la partida, se lanza la excepción `domain_error` con el mensaje `Jugador no existente`.
- `regiones_en_disputa()`. Devuelve un `vector<string>` con la lista de regiones que, teniendo al menos un caballero de algún jugador, están en disputa. La lista ha de estar ordenada ascendentemente, en orden alfabético según el nombre de la región.
- `expulsar_caballeros(region)`. Elimina a todos los caballeros de la `region` pasada como parámetro. Si la región no tenía ningún caballero, se lanza la excepción `domain_error` con el mensaje `Region vacia`.

La implementación de las operaciones debe ser lo más eficiente posible. Por tanto, debes elegir una representación adecuada para el TAD, implementar las operaciones y justificar la complejidad resultante.

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.

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.

Los nombres de jugadores y regiones son cadenas de caracteres sin espacios en blanco.

Salida

Las operaciones que generan salida son:

- `puntuacion J`, que debe escribir una línea con el mensaje `Puntuación de J: X`, donde `J` es el jugador del cual quiere obtenerse la puntuación, y `X` es la puntuación del mismo.
- `regiones_en_disputa`, que debe escribir una línea con el mensaje `Regiones en disputa:`, seguida de los nombres de las regiones que están en disputa, uno por línea.

Si una operación produce un error, entonces se escribirá una línea con el mensaje `ERROR:`, seguido del mensaje de la excepción que lanza la operación, y no se escribirá nada más para esa operación.

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

Entrada de ejemplo

```
anyadir_jugador jug1
anyadir_jugador jug2
colocar_caballero jug1 Granada
colocar_caballero jug2 Granada
colocar_caballero jug1 Granada
puntuacion jug1
puntuacion jug2
colocar_caballero jug2 Granada
puntuacion jug1
puntuacion jug2
colocar_caballero jug2 Aragon
colocar_caballero jug1 Aragon
regiones_en_disputa
FIN
colocar_caballero jug1 Sevilla
anyadir_jugador jug1
colocar_caballero jug1 Valencia
colocar_caballero jug1 Galicia
puntuacion jug1
expulsar_caballeros Valencia
puntuacion jug1
expulsar_caballeros Valencia
FIN
```

Salida de ejemplo

```
Puntuacion de jug1: 1
Puntuacion de jug2: 0
Puntuacion de jug1: 0
Puntuacion de jug2: 0
Regiones en disputa:
Aragon
Granada
---
ERROR: Jugador no existente
Puntuacion de jug1: 2
Puntuacion de jug1: 1
ERROR: Region vacia
---
```