

Ejercicio 3

3 puntos

En la casa de subastas "*Barato, Barato*" se llevan a cabo subastas simultáneas de objetos de gran valor. Cada participante comienza con un saldo inicial y puede pujar por cada uno de los objetos en los que está interesado, entregando el importe pujado. Además, cada puja es secreta; ningún participante conoce la cantidad pujada por el resto. Esto provoca que varios participantes puedan pujar la misma cantidad para un objeto. Al cerrarse la subasta de un objeto determinado, este es ganado por aquel participante que haya pujado una cantidad mayor por él. En caso de haber varios participantes que hayan pujado dicha cantidad, el objeto es asignado al participante que primero pujó (en orden cronológico). El resto de los participantes que perdieron la puja recuperan el dinero pujado. Las pujas son simultáneas, por lo que los participantes pueden apostar por tantos objetos como deseen, pero un participante no puede pujar dos veces por el mismo objeto. En cualquier momento, el participante puede abandonar la casa de subastas, llevándose lo ganado hasta el momento y perdiendo el dinero apostado de las subastas pendientes de cerrarse.

Se pide implementar un TAD CasaDeSubastas con las siguientes operaciones:

- `void nuevo_participante(string part, int saldo_inicial)`

Añade un nuevo participante a la casa de subastas, con el saldo inicial pasado por parámetro. Lanza la excepción `std::domain_error("Participante ya existente")` si `part` ya está presente en la casa. Si el `saldo_inicial` no tiene un valor mayor que 0, lanza la excepción `std::domain_error("Saldo inicial no valido")`.

- `void nueva_subasta(string obj, int puja_min)`

Añade un nuevo objeto listo para ser subastado, con una puja mínima indicada por parámetro. Lanza la excepción `std::domain_error("Objeto no valido")` si el objeto `obj` ya está siendo subastado por la casa de apuestas o ya ha sido vendido. Si `puja_min` no tiene un valor mayor que 0, lanza la excepción `std::domain_error("Puja inicial no valida")`.

- `void nueva_puja(string part, string obj, int cantidad)`

Añade la puja del participante `part` para conseguir el objeto `obj` con un valor indicado por `cantidad`. Lanza las siguientes excepciones:

- `std::domain_error("Participante no existente")` si `part` no existe.
- `std::domain_error("Objeto no valido")` si `obj` no existe o ya ha sido vendido.
- `std::domain_error("Participante repetido")`, si `part` ya ha realizado una puja para `obj`.
- `std::domain_error("Cantidad no valida")` si la cantidad supera el saldo actual del participante o no supera la cantidad mínima requerida por el objeto `obj`.

- `list<string> subastas_ganadas(string part) const`

Devuelve en una lista los objetos ganados por el participante en orden alfabético. Si `part` no existe, se lanza la excepción `std::domain_error("Participante no existente")`.

- `void abandonar_casa(string part)`

El participante abandona la casa, eliminando toda la información relativa a él, incluidas las pujas de objetos cuya subasta está pendiente de cerrar. Si `part` no existe, se lanza la excepción `std::domain_error("Participante no existente")`.

- `string cerrar_subasta(string obj)`

Se otorga `obj` a quien pujó la cantidad más alta, y si hay varios, al primero que lo hizo y no haya abandonado la casa, devolviendo su nombre. Lanza la excepción `std::domain_error("Objeto no valido")` si `obj` no existe o ya ha sido vendido. Lanza la excepción `std::domain_error("Objeto no vendido")`, si nadie ha pujado por él.

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 participantes y de objetos subastados son cadenas de caracteres sin espacios en blanco.

Salida

Tras ejecutar las operaciones `nuevo_participante`, `nuevo_objeto`, `nueva_puja` y `abandonar_casa` debe imprimirse una línea con la palabra `OK`, salvo en caso de error.

Tras ejecutar la operación `cerrar_subasta` se debe imprimir, salvo en caso de error, una línea con el texto `X ha sido ganado por: Y`, donde `X` es el objeto cuya subasta se ha cerrado, e `Y` es el participante ganador de la subasta.

Tras ejecutar la operación `objetos_ganados` se debe imprimir, salvo en caso de error, una línea con el texto `X ha ganado:` (donde `X` es el participante), seguido por los objetos que el participante ha ganado, cada uno precedido por un espacio.

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

```
nuevo_participante juan 100
nuevo_participante ana 160
nueva_subasta funko_luffy 80
nueva_subasta funko_goku 120
nueva_subasta funko_sailor 10
nueva_puja juan funko_luffy 90
nueva_puja juan funko_sailor 10
cerrar_subasta funko_sailor
nueva_puja ana funko_luffy 85
nueva_puja ana funko_goku 130
cerrar_subasta funko_luffy
nueva_puja ana funko_goku 130
cerrar_subasta funko_goku
subastas_ganadas juan
FIN
nuevo_participante juan 100
nuevo_participante ana 160
nueva_puja juan funko_pop 20
nueva_subasta funko_pop -5
nueva_subasta funko_luffy 80
nueva_puja juan funko_luffy 90
nueva_puja ana funko_luffy 90
nueva_puja juan funko_luffy 10
abandonar_casa juan
cerrar_subasta funko_luffy
FIN
nuevo_participante juan 100
nuevo_participante ana 160
nueva_subasta funko_luffy 80
nueva_puja juan funko_luffy 90
nueva_puja ana funko_luffy 85
abandonar_casa juan
cerrar_subasta funko_luffy
FIN
```

Salida de ejemplo

```
OK
OK
OK
OK
OK
OK
OK
funko_sailor ha sido ganado por: juan
OK
ERROR: Cantidad no valida
funko_luffy ha sido ganado por: juan
OK
funko_goku ha sido ganado por: ana
juan ha ganado: funko_luffy funko_sailor
---
OK
OK
ERROR: Objeto no valido
ERROR: Puja inicial no valida
OK
OK
OK
ERROR: Participante repetido
OK
funko_luffy ha sido ganado por: ana
---
OK
OK
OK
OK
OK
OK
funko_luffy ha sido ganado por: ana
---
```