

Spitter



Spitter es una red social de microblogging que permite a sus usuarios añadir publicaciones cortas (*spits*). Cada publicación puede llevar asociada una serie de etiquetas (*hashtags*). Para tener un primer prototipo de nuestra red, queremos implementar el TAD *Spitter* con las siguientes operaciones:

- `nueva_publicacion(id, texto, hashtags)`: Añade una nueva publicación a la red social con el identificador `id` y el texto dados (de tipo `int` y `string` respectivamente). El tercer parámetro es un vector<`string`> con la lista de *hashtags* asociados a la publicación que quiere añadirse. Puedes suponer que esta lista no contiene *hashtags* duplicados. Si ya existe una publicación con el mismo identificador, se lanza una excepción `std::domain_error` con el mensaje `Identificador duplicado`.
- `borrar_publicacion(id)`: Elimina del sistema la publicación con el identificador `id` dado. Si no existe ninguna publicación con tal identificador, la operación no tiene ningún efecto.
- `mas_recientes(n)`: Devuelve una lista con los textos de las `n` publicaciones más recientes, siendo `n` un número mayor o igual a 0. Si `n` es mayor que el número de publicaciones existentes, se devolverán todas. La lista resultante contiene las publicaciones ordenadas, desde la más reciente hasta la más antigua.
- `hashtags(id)`: Devuelve una lista con los *hashtags* asociados a la publicación con identificador `id`. La lista estará ordenada alfabéticamente. Si no existe una publicación con el identificador indicado, se lanza una excepción `std::domain_error` con el mensaje `Publicacion no existente`.
- `anyadir_hashtag(id, hashtag)`: Añade un nuevo *hashtag* a la publicación con identificador `id`. Si no existe una publicación con el identificador indicado, se lanza una excepción `std::domain_error` con el mensaje `Publicacion no existente`. Si el *hashtag* ya estaba asociado a la publicación `id`, la operación no hace nada.
- `popularidad(hashtag)`: Devuelve el número de publicaciones que contienen el *hashtag* indicado. Si no hay ninguna publicación que lo contenga, devuelve 0.

En este ejercicio se pide:

1. **Implementar** las operaciones descritas en el TAD. Ninguna de ellas debe realizar operaciones de E/S. El manejo de E/S debe hacerse en la función `tratar_caso()`.
2. Indicar el **coste** de cada operación. Las operaciones deben implementarse de la manera más eficiente posible desde el punto de vista del coste asintótico en tiempo.

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. En el caso de la operación `nueva_publicacion`, el nombre de dicha operación va seguida del identificador y del texto de la publicación a insertar. Los *hashtags* de esta publicación se encuentran en la siguiente línea, separados por espacios. Si la publicación no contiene *hashtags*, esta línea estará en blanco.

La palabra `FIN` en una línea indica el final de cada caso.

Salida

Las operaciones `nueva_publicacion` y `borrar_publicacion` no provocan salida, salvo que se produzca un error. Con respecto a las restantes:

- `mas_recientes` imprime una línea con el texto `Las N publicaciones mas recientes:`, siendo `N` el número de publicaciones devueltas por esta operación. A continuación imprime `N` líneas, cada una con el texto de cada publicación (desde la más reciente hasta la más antigua). Cada publicación va precedida por tres espacios.
- `hashtags` imprime una línea con los *hashtags* de la publicación seleccionada, separados por espacios.
- `popularidad` imprime un número con el resultado de la operación.

Cada caso termina con una línea con tres guiones (`---`). Si una operación produce un error, entonces se escribirá una línea con el mensaje `ERROR:`, seguido del error que devuelve la operación, y no se escribirá nada más para esa operación.

Entrada de ejemplo

```
nueva_publicacion 3 Primera publicacion
#pruebas #primeras_veces
nueva_publicacion 1 Ahora va otra
#pruebas #segundas_veces
nueva_publicacion 5 Y una tercera
#pruebas #me_canso
mas_recientes 10
hashtags 3
anyadir_hashtag 1 #mas
hashtags 1
popularidad #pruebas
borrar_publicacion 1
popularidad #pruebas
popularidad #segundas_veces
FIN
nueva_publicacion 3 Hola, mundo

nueva_publicacion 3 Duplicada

borrar_publicacion 1
hashtags 1
anyadir_hashtag 1 #vale
FIN
```

Salida de ejemplo

```
Las 3 publicaciones mas recientes:
  Y una tercera
  Ahora va otra
  Primera publicacion
#primeras_veces #pruebas
#mas #pruebas #segundas_veces
3
2
0
---
ERROR: Identificador duplicado
ERROR: Publicacion no existente
ERROR: Publicacion no existente
---
```

Créditos

Autor: Manuel Montenegro. **Revisores:** Alberto Verdejo, Isabel Pita.