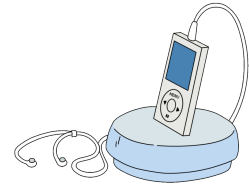


# iPud

Los ingenieros de la empresa Apel están actualmente diseñando el software para su nuevo reproductor de música *iPud*, el cual debe permitir añadir y eliminar canciones, añadir una canción existente a la lista de reproducción, avanzar a la siguiente canción, obtener el tiempo total de la lista de reproducción, y obtener una lista con las canciones escuchadas recientemente.



En particular, el TAD *iPud* debe contar con las siguientes operaciones:

- **addSong(S,A,D)**: Añade la canción **S** (un **string**) del artista **A** (un **string**) con duración **D** (un **int**) al *iPud*. Si ya existe una canción con el mismo nombre la operación dará error.
- **addToPlaylist(S)**: Añade la canción **S** al final de la lista de reproducción. Si la canción ya se encontraba en la lista entonces no se añade (es decir, la lista no tiene canciones repetidas). Si la canción no está en el *iPud* se devuelve error.
- **current()**: Devuelve la primera canción de la lista de reproducción. Si la lista de reproducción es vacía se devuelve error.
- **play()**: La primera canción de la lista de reproducción abandona la lista de reproducción y se registra como reproducida. Si la lista es vacía la acción no tiene efecto.
- **totalTime()**: Devuelve la suma de las duraciones de las canciones que integran la lista de reproducción actual. Si es vacía se devuelve 0.
- **recent(N)**: Obtiene la lista con las **N** (mayor que 0) últimas canciones que se han reproducido (mediante la operación **play**), de la más reciente a la más antigua. Si el número de canciones reproducidas es menor que **N** se devolverán todas. La lista no tiene repeticiones, de manera que si una canción se ha reproducido más de una vez solo figurará la reproducción más reciente.
- **deleteSong(S)**: Elimina todo rastro de la canción **S** del *iPud*. Si la canción no existe la operación no tiene efecto.

Se puede asumir que las canciones quedan unívocamente determinadas por su nombre (un **string**).

*Requisitos de implementación.*

Se elegirá un tipo representante que permita implementar todas las operaciones con coste constante, salvo **recent** que tendrá un coste lineal en su argumento **N**.

## Entrada

La entrada consiste en varios casos de prueba. Cada caso de prueba está formado por una lista de llamadas a operaciones del *iPud*, terminada por la palabra **FIN**. Cada una de estas llamadas comienza por el nombre de la operación, seguido de sus argumentos, en caso de tenerlos. Los nombres de canciones y artistas no contienen espacios.

## Salida

Las operaciones que producen salida son las siguientes:

- **play**: se escribirá **Sonando** y el nombre de la canción. Si la lista de reproducción es vacía se escribirá **No hay canciones en la lista**.
- **totalTime**: se escribirá **Tiempo total** y el tiempo de duración de la lista de reproducción.
- **recent**: se escribirá en una línea **Las R más recientes** (donde **R** es el número de canciones realmente mostradas) y en las líneas siguientes los nombres de las canciones, indentadas a la derecha cuatro espacios. Si la lista es vacía se escribirá **No hay canciones recientes** en su lugar.

Si una operación produce un error, entonces se escribirá **ERROR** seguido del nombre de la operación. Y no producirá más salida.

Cada caso se terminará con una línea con cuatro guiones (----).

## Entrada de ejemplo

```
addSong HumanTouch BruceSpringsteen 392
addSong BornToRun BruceSpringsteen 270
addToPlaylist BornToRun
addToPlaylist HumanTouch
play
totalTime
addSong LuckyTown BruceSpringsteen 207
addToPlaylist LuckyTown
play
play
deleteSong HumanTouch
recent 2
FIN
play
current
FIN
```

## Salida de ejemplo

```
Sonando BornToRun
Tiempo total 392
Sonando HumanTouch
Sonando LuckyTown
Las 2 mas recientes
    LuckyTown
    BornToRun
----
No hay canciones en la lista
ERROR current
----
```

**Autor:** Examen junio 2016.