Autoescuela

Se desea diseñar un TAD para gestionar los alumnos de los distintos profesores de una autoescuela (tanto alumnos como profesores se identifican por su nombre, que es un `string`). Para ello se desea disponer de las siguientes operaciones:

- constructora: al comienzo ningún profesor tiene asignados alumnos.
- `alta(A, P)`: sirve tanto para dar de alta a un alumno como para cambiarle de profesor. Si el alumno no estaba matriculado en la autoescuela se le da una puntuación de cero. Si ha cambiado de profesor, se le da de alta con el nuevo, con la puntuación que tuviera, y se le da de baja con el anterior. La puntuación determinará quién se puede examinar.
- `es_alumno(A, P)`: comprueba si el alumno `A` está matriculado actualmente con el profesor `P`.
- `puntuacion(A)`: devuelve los puntos que tiene el alumno `A`. Si el alumno no está dado de alta con ningún profesor, entonces se lanza una excepción `std::domain_error` con mensaje `El alumno A no esta matriculado`.
- `actualizar(A, N)`: aumenta en una cantidad `N` la puntuación del alumno `A`. Si el alumno no está dado de alta, entonces se lanza una excepción `std::domain_error` con mensaje `El alumno A no esta matriculado`.
- `examen(P, N)`: obtiene una lista con los alumnos del profesor `P`, ordenados alfabéticamente, que se presentarán a examen por tener una puntuación mayor o igual a `N` puntos.
- `aprobar(A)`: el alumno `A` aprueba el examen y es borrado de la autoescuela, junto con toda la información que de él existiera. Si el alumno no está dado de alta, entonces se lanza una excepción `std::domain_error` con mensaje `El alumno A no esta matriculado`.

Seleccionar un tipo de datos adecuado para representar la información. 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.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso consiste en una serie de líneas donde se muestran las operaciones a realizar, sobre una autoescuela inicialmente vacía: el nombre de la operación seguido de sus argumentos, si los tiene. Cada caso termina con una línea con la palabra `FIN`. Los nombres de alumnos y profesores no tienen espacios en blanco.

Salida

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

- `es_alumno`, que debe escribir una línea con `A es alumno de P` o `A no es alumno de P`, según corresponda (donde `A` es el nombre del alumno y `P` el del profesor);
- `puntuacion`, que debe escribir una línea con `Puntuacion de A:` seguido de los puntos del alumno `A`;
- `examen`, que escribe una línea con `Alumnos de P a examen:`, seguida de una línea por cada alumno que se pueda presentar al examen, con su nombre, ordenados alfabéticamente.

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

Si una operación produce un error, entonces se escribirá una línea con `ERROR`, y no se escribirá nada más para esa operación.

Entrada de ejemplo

```
alta Luis PACO
alta Marta PACO
es_alumno Luis PACO
es_alumno Marta RAMON
examen PACO 0
actualizar Marta 10
examen PACO 10
examen JUAN 10
alta Marta JUAN
puntuacion Marta
actualizar Marta 10
examen JUAN 20
actualizar Miguel 5
aprobar Marta
examen JUAN 20
FIN
puntuacion Alejandro
alta Alejandro ISABEL
actualizar Alejandro 20
puntuacion Alejandro
aprobar Alejandro
puntuacion Alejandro
FIN
```

Salida de ejemplo

```
Luis es alumno de PACO
Marta no es alumno de RAMON
Alumnos de PACO a examen:
Luis
Marta
Alumnos de PACO a examen:
Marta
Alumnos de JUAN a examen:
Puntuacion de Marta: 10
Alumnos de JUAN a examen:
Marta
ERROR
Alumnos de JUAN a examen:
---
ERROR
Puntuacion de Alejandro: 20
ERROR
---
```

Autor

Alberto Verdejo