¿A qué hora pasa el próximo tren?

Todo el que ha estado alguna vez esperando que llegue el tren se ha planteado lo que podría haber hecho si hubiese sabido a qué hora pasaría. Es entonces cuando envidiamos a los usuarios que lo utilizan con frecuencia y son capaces de llegar a la estación con solo un minuto de antelación. ¿Por qué siempre confiamos en la suerte y no comprobamos a qué hora pasará antes de vernos ya en el andén?

Hoy, al llegar a la estación y ver que de nuevo tendrás que esperar, has decidido hacer una aplicación que te permita saber en cualquier momento a qué hora pasará el próximo tren. Para empezar, en el tiempo que has estado esperando ya has conseguido el horario de los trenes de las estaciones que utilizas a menudo. Ahora, ya será muy fácil calcular cuándo pasará el próximo tren.

Requisitos de implementación

Se debe utilizar una clase `horas` con la representación que se considere más oportuna. Como mínimo se debe implementar la sobrecarga del operador < como función miembro de la clase. La sobrecarga de los operadores de inserción (<<) y extracción (>>) de las horas se hará con funciones externas a la clase. El constructor de la clase debe comprobar que los datos son correctos ($0 \leq$ *horas* ≤ 23; $0 \leq$ *minutos, segundos* ≤ 59) y lanzar una excepción si no lo son.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea con dos valores: el número de trenes que sale de la estación y el número de horas que se van a querer consultar. A continuación se muestra en una línea el horario de trenes de la estación en orden creciente. Para cada tren se dice la hora, minutos y segundos en que saldrá de la estación (en el formato `HH:MM:SS`). A continuación se listan las horas que se van a consultar, una en cada línea y no necesariamente ordenadas.

El número de trenes que sale de la estación es siempre mayor que cero y menor que 1000. Se garantiza que las fechas del horario de trenes son correctas y están en orden creciente; sin embargo, las fechas que se consultan pueden ser incorrectas.

La entrada termina con `0 0`.

Salida

Para cada caso de prueba se escribirá una línea por cada hora consultada, indicando la hora de salida (con el mismo formato que en la entrada) del primer tren con salida posterior o igual a la hora consultada, si existe . Si no existe, se escribirá `NO`. Si la fecha de entrada es incorrecta se escribirá `ERROR`. Cada caso termina con tres guiones (`---`).

Entrada de ejemplo

```
4 2
06:40:30 12:50:00 19:20:00 21:25:00
10:20:00
22:00:00
6 3
00:00:00 09:30:00 16:40:30 17:00:00 20:10:40 22:35:00
20:10:40
08:62:30
08:40:30
0 0
```

Salida de ejemplo

```
12:50:00
NO
---
20:10:40
ERROR
09:30:00
---
```

Autores: Isabel Pita y Alberto Verdejo.