

Butacas en el teatro



En el teatro de mi barrio quieren informatizar el sistema de reserva de localidades. El dueño del teatro, que debe de ser informático, prefiere numerar las butacas de manera secuencial, como si todas ellas estuvieran dispuestas en una única fila. Además, quiere que las butacas empiecen a numerarse desde el 0.

Para ayudar al dueño, vamos a implementar un tipo abstracto de datos Teatro con dos operaciones:

- `bool nueva_reserva(int inicio, int fin, string nombre)`
Comprueba la disponibilidad de las butacas cuyos números están comprendidos entre `inicio` y `fin` (ambos incluidos). Si todas estas butacas están libres, se reservarán al nombre indicado y la operación devolverá `true`. Si, por el contrario, alguna de estas butacas ya estaba reservada a nombre de otra persona, no se efectuará la reserva y se devolverá `false`.
- `string nombre_reserva(int num_butaca)`
Devuelve el nombre de la persona que ha reservado la butaca número `num_butaca`. Si esa butaca no había sido reservada por nadie, devuelve la cadena vacía.

Para implementar este TAD utilizamos la siguiente representación, basada en un vector reservas de tamaño fijo.

```
const int MAX_RESERVAS = 50000;
```

```
class Teatro {  
public:  
    // ...  
  
private:  
    struct Reserva {  
        int inicio;  
        int fin;  
        string nombre;  
    };  
  
    vector<Reserva> reservas;  
    int num_reservas;  
};
```

El vector `reservas` contiene valores de tipo `Reserva`. Cada uno de estos valores tiene tres atributos, que indican el intervalo `[inicio..fin]` de butacas reservadas y el nombre de la persona que las ha reservado. El atributo `num_reservas` indica el número de reservas realizadas hasta el momento. Todas las reservas están contenidas en el segmento del vector `reservas` contenido entre las posiciones 0 y `num_reservas-1` (ambas incluidas).

El invariante de representación I_{rep} de Teatro es el siguiente, suponiendo que t denota una instancia de esta clase:

$$I_{rep}(t) \equiv 0 \leq t.num_reservas \leq MAX_RESERVAS \\ \wedge (\forall i : 0 \leq i < t.num_reservas \implies t.reservas[i].inicio \leq t.reservas[i].fin \\ \wedge t.reservas[i].nombre \neq "") \\ \wedge (\forall i : 1 \leq i < t.num_reservas \implies t.reservas[i-1].fin < t.reservas[i].inicio)$$

De estas tres condiciones se deduce que el vector reservas ha de contener las reservas en orden creciente según el intervalo de butacas reservado, y que no puede haber solapamiento de butacas entre dos reservas distintas.

En esta práctica se pide:

1. Implementar las operaciones `nueva_reserva` y `nombre_reserva` del TAD Teatro. En la operación `nueva_reserva` puedes suponer que $\text{inicio} \leq \text{fin}$ y que `nombre` no es una cadena vacía. Las operaciones han de preservar el invariante de representación indicado anteriormente.
2. Indicar el coste en tiempo de cada una de las operaciones anteriores. Este coste ha de expresarse con respecto al número de reservas existentes en el momento de realizar la operación.
3. Escribir un programa que, utilizando el TAD Teatro, lea de la entrada estándar una serie de reservas e indique cuáles se han podido efectuar y cuales no. Tras esto, deberá leer una lista de números de butacas, y para cada una de ellas deberá decir el nombre de la persona que la ha reservado. El formato de la entrada y la salida se define a continuación.

Entrada

La entrada consta de una serie de casos de prueba.

- Cada caso comienza con una línea que contiene dos números N y Q ($1 \leq N, Q \leq 50\,000$) que indican, respectivamente, el número de reservas y el número de consultas que se quieren realizar.
- Después vienen N líneas, cada una con un intento de reserva. En cada línea se indican dos números i y j que delimitan el rango $[i..j]$ de butacas a reservar y el nombre de la persona que las quiere reservar.
- Después vienen Q líneas, cada una de ellas con el número de butaca cuya disponibilidad se quiere consultar.

La entrada finaliza con un caso que contiene dos ceros, que no debe procesarse.

Salida

Para cada caso de prueba se ha de imprimir:

- Una línea con N caracteres indicando, para cada intento de reserva, si se ha podido efectuar o no. El carácter '.' indica que la reserva se ha realizado correctamente, y el carácter 'X' indica que la reserva no se ha podido llevar a cabo, debido a que alguna de las butacas que quería reservarse ya estaba reservada previamente.
- Q líneas, una por cada consulta. Cada línea debe contener el nombre de la persona que ha reservado la butaca indicada en la consulta. Si esa butaca estaba libre, ha de imprimirse la palabra LIBRE.

Al final de cada caso de prueba ha de imprimirse una línea con tres guiones (---).

Entrada de ejemplo

```
4 6
1 5 Diana
21 23 Ricardo
10 10 Nuria
22 24 David
22
7
9
10
11
5
3 3
5 5 Javier
1 8 Laura
1 4 Alberto
20
4
5
0 0
```

Salida de ejemplo

```
...X
Ricardo
LIBRE
LIBRE
Nuria
LIBRE
Diana
---
.X.
LIBRE
Alberto
Javier
---
```

Créditos

Manuel Montenegro