

Lucky Lucke en busca del culpable

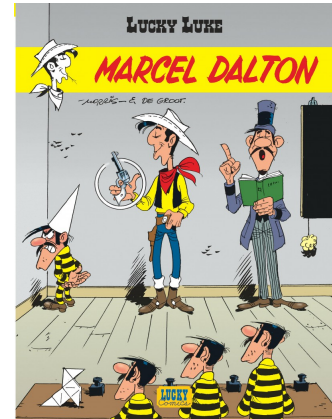
Lucky Lucke es el vaquero más rápido del oeste. Alguien ha robado el oro de la cámara blindada del banco y Lucky Lucke debe encontrar el mejor algoritmo posible para dar con el culpable. Lucky Lucke conoce la altura del bandido porque después de cometer el atraco dejó su figura marcada en la parte superior de la puerta del banco.

El sheriff le ha mandado por mail un fichero con la altura de todos los sospechosos del oeste ordenados de menor a mayor. ¿Puedes ayudarle a encontrar al culpable? .

Requisitos de implementación.

Se implementarán dos funciones recursivas diferentes: La primera *buscarIz* encuentra el primer sospechoso de la lista con la altura pedida, la segunda, *buscarDr* encuentra la posición en la lista del último sospechoso con la altura pedida.

La función *resuelveCaso*, llamará a la función *buscarIz* para calcular la primera posición en que aparece la altura pedida y llamará a la función *buscarDr* para obtener la última posición de la altura pedida en la lista.



Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con el número de sospechosos seguido de la altura del bandido. A continuación en una línea aparecen las alturas de todos los sospechosos ordenados de menor a mayor.

El número de sospechosos es un valor $0 \leq N \leq 1.000.000$ y las alturas de los sospechosos son valores enteros positivos menores de 250.

Salida

Para cada caso de prueba se muestra en una línea la posición que ocupa en la lista de sospechosos el primero con la altura buscada seguido de la posición del último sospechoso con la altura dada. Si sólo existe un sospechoso se mostrará solo su posición en la lista y si no existe ninguno se escribirá *NO EXISTE*.

Entrada de ejemplo

```
4 8
2 5 7 8
3 5
5 5 5
5 8
3 4 6 7 9
```

Salida de ejemplo

```
3
0 2
NO EXISTE
```