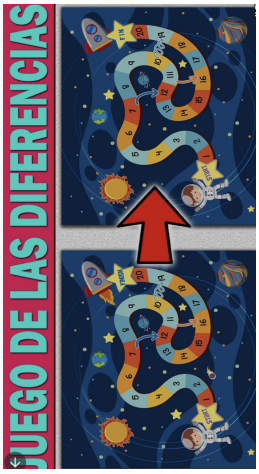
Comparando listados ordenados



Disponemos de dos listados ordenados (por ejemplo, con los alumnos matriculados en EDA y TPV respectivamente) y queremos averiguar qué elementos están en ambos listados y también qué elementos están en un listado y no en el otro.

Escribe un algoritmo eficiente que dados dos listados ordenados devuelva otros tres listados también ordenados, el primero con los elementos presentes en ambos listados, el segundo con los elementos presentes en el primero y no en el segundo, y el tercero con los presentes en el segundo y no en el primero. Indica y justifica la complejidad asintótica en tiempo del algoritmo. Hazlo en un comentario delante de la función que implementa el algoritmo.

Cuestión: Si los listados no viniesen ordenados, ¿Cuál sería el menor orden de complejidad posible del algoritmo correspondiente? Justifica la respuesta. Ten en cuenta que ordenar un vector con n elementos puede hacerse en complejidad $O(n * log(n))$

Entrada

Primero se solicitará el número de casos de prueba a procesar. Cada caso consistirá en 4 líneas: la primera con el número de elementos del primer listado, la segunda con los elementos (de tipo string) del segundo listado (separados por espacio), y la tercera y cuarta con lo análogo para el segundo listado.

Salida

Para cada caso de prueba se imprimirán tres líneas: la primera con los elementos presentes en ambos listados, la segunda con los elementos presentes en el primero y no en el segundo y la tercera con los presentes en el segundo y no en el primero (usando espacios para separar los elementos de un mismo listado). En este enlace tenéis una plantilla que incluye la implementación de la gestión de la entrada/salida.

Entrada de ejemplo

```
4
6
a b c d f h
6
a b d e g k
4
a b c d
3
c d e
2
d f
4
a b d e
2
a b
2
c d
```

Salida de ejemplo

```
a b d
c f h
e g k
c d
a b
e
d
f
a b e

a b
c d
```