

# Reconstruir un árbol desde sus recorridos post-orden + in-orden

Un árbol binario se puede reconstruir a partir de sus recorridos en post-orden y en in-orden. Se pide diseñar una función recursiva que reciba dos listas (TAD *list* de C++) conteniendo los recorridos post-orden e in-orden de un árbol binario (sin elementos repetidos), y devuelva el árbol binario correspondiente (TAD *Arbin*).

## Entrada

La entrada consistirá en distintos casos de prueba, cada uno en dos líneas que representan los recorridos post-orden e in-orden respectivamente. Cada línea contiene una secuencia de números enteros no negativos y sin repeticiones separados por espacios (que se van introduciendo por detrás en la lista) y acabada en  $-1$  (que no se introduce en la lista). El programa acabará cuando se introduzcan dos listas vacías (dos  $-1$ 's).

## Salida

Para cada caso de prueba se imprimirá en una línea el recorrido por niveles del árbol reconstruido, con un espacio separando los elementos (y sin espacio al final).

## Entrada de ejemplo

```
7 2 11 4 13 1 4 8 5 -1
7 11 2 4 5 13 8 4 1 -1
2 3 1 -1
2 1 3 -1
3 2 1 -1
1 2 3 -1
3 2 1 -1
3 2 1 -1
5 -1
5 -1
-1
-1
```

## Salida de ejemplo

```
5 4 8 11 13 4 7 2 1
1 2 3
1 2 3
1 2 3
5
```

## Nota

El primer caso se corresponde con el árbol dibujado en el ejercicio 10 del tema 9.