

## Vectores superb

### El problema

Se considera que un vector es *superb* si (i) su longitud es estrictamente menor que 3; o (ii) el elemento central es igual a la resta (en valor absoluto) del elemento más grande de todos los que lo preceden (aparecen antes que él en el vector) menos el elemento más pequeño de todos los que lo suceden (aparecen después que él) y además ambas mitades (la formada por los elementos que preceden al central, y la formada por los elementos que lo suceden) son *superb*. Recuerda que el índice del elemento central es la suma de los índices de los extremos entre dos (división entera). Diseña e implementa un algoritmo recursivo que dado un vector devuelva si es *superb*. Debes, así mismo, determinar justificadamente el coste de este algoritmo, planteando y resolviendo las recurrencias apropiadas.

### Trabajo a realizar

Para realizar el control se proporciona un archivo `plantilla.cpp` que contiene un programa que lee por la entrada estándar casos de prueba y los ejecuta (llamando antes a la función `es_superb`). Cada caso de prueba consiste en dos líneas: La primera línea indica el tamaño  $n$  del vector ( $n \leq 10000$ ). La segunda línea contiene los  $n$  valores del vector, separados por espacios. La lista de casos de prueba termina con un -1. Por cada caso de prueba se escribirá una línea con `SUPERB` si el vector es *superb* y `NO SUPERB` en caso contrario.

A continuación, se muestra un ejemplo de entrada / salida.

Entrada	Salida
3	NO SUPERB
7 2 10	SUPERB
7	
5 10 15 11 12 4 8	
-1	

Tu trabajo consiste en:

- Diseñar el algoritmo, rellenando los comentarios incluidos en el archivo.
- Implementar este algoritmo.
- Entregar la solución a través del juez en línea de la asignatura.

### Importante:

- No modificar el código proporcionado. Únicamente deben responderse a los distintos apartados, en el interior de los comentarios, implementar la función `es_superb` y todas las funciones auxiliares que se consideren necesarias.