

# Ejercicio opcional de EDA: Navidades 2014/15

## La moneda chungu

### ENTREGA

Entra en <http://bantu.fdi.ucm.es/domjudge/team> con tu nombre de grupo y contraseña. Sube un único .cpp como solución al problema (el .cpp debe compilar por sí mismo, sin #includes a excepción de <iostream>) al sistema.

### Enunciado

Tienes  $N$  monedas. 1 de ellas pesa distinto que todas las demás, pero son indistinguibles sólo mirándolas. Encuentra la moneda usando una balanza de las antiguas, y con el mínimo número de pesadas.

### La entrada

La entrada se leerá por entrada estándar (cin), y consistirá en parejas de líneas, la primera con un entero  $n$  ( $3 \leq n < 10000$ ) que especifica la longitud del array, seguido por los  $n$  enteros positivos que lo constituyen. Entre los enteros habrá a lo sumo uno que sea menor ó mayor que el resto; los demás serán todos iguales. La entrada terminará con un '-1', que no deberás procesar.

### La salida

Se deberá escribir por "cout" una única línea por cada caso de entrada: el índice que ocupaba, en el array original, la moneda que es distinta del resto; precedido por el símbolo "-" (si era más ligera que las demás) o el símbolo "+" (si era más pesada). Los índices empiezan en 0.

### Entrada y salida de ejemplo

3	+2
5 5 7	-1
4	+4
7 6 7 7	
7	
9 9 9 9 10 9 9	
-1	

### IMPORTANTE

Tu programa **sólo se considerará válido** si

- **sólo usa recursión para solucionar cada caso:** no hay bucles fuera del main(), y el main() sólo lee casos de prueba y muestra el resultado de llamar, para cada uno, a tus funciones que **no** tienen bucles (la función `pesa()` no cuenta para esta restricción)
- **sólo usa la función `pesa()` para comparar elementos** (ver página siguiente); queda prohibido *leer* del array en cualquier otro lugar
- tiene **coste  $O(\log n)$** .

**Más opcional:** puedes sumar hasta un 15/10 si añades un comentario en cada función (exceptuando el "main()", pero incluyendo a "pesa()") con la pre-condición y post-condición correspondientes.

```
/**
 * compara v[ia] + ... + v[ia+n-1] con v[ib] + ... + v[ib+n-1]
 * (platillos A y B respectivamente), y devuelve
 * 1 si A>B; 0 si A==B, -1 si A<B
 */
int pesa(int v[], int ia, int ib, int n){
    int t=0;
    for (int i=0; i<n; i++) {
        t += (v[ib+i] - v[ia+i]);
    }
    return t ? (t<0?-1:1) : 0;
}
```

*Función pesa() - inclúyela en tu código, y no leas del array fuera de ella*