



Nº matrícula: \_\_\_\_\_ Nombre: \_\_\_\_\_

Apellidos: \_\_\_\_\_

### Sesión 1.1. Divide y Vencerás

Consideremos un *array* de  $N$  enteros. Queremos conocer la suma de todos sus elementos positivos. Para ello deseamos implementar un algoritmo basado en divide y vencerás que resuelva el problema, cuya cabecera es la siguiente:

```
int sumaPositivos(int[] vector)
```

Ejemplo:

|    |   |   |   |    |    |     |     |    |
|----|---|---|---|----|----|-----|-----|----|
| 0  | 1 | 2 | 3 | 4  | 5  | 6   | 7   | 8  |
| -1 | 0 | 2 | 3 | 10 | 12 | -23 | -14 | -7 |

Para este vector, se devolvería el valor 27.

**Pregunta 1.** Implementa en Java un algoritmo basado divide y vencerás que resuelva el problema.

```
int sumaPositivos(int[] vector){
```

**Pregunta 2 (3 puntos).** Calcula la complejidad del algoritmo implementado (justifica la respuesta)

## Sesión 1.1. Divide y Vencerás

Consideremos un *array* de  $N$  enteros. Queremos conocer la suma de todos sus elementos positivos. Para ello deseamos implementar un algoritmo basado en divide y vencerás que resuelva el problema, cuya cabecera es la siguiente:

```
int sumaPositivos(int[] vector)
```

Ejemplo:

|    |   |   |   |    |    |     |     |    |
|----|---|---|---|----|----|-----|-----|----|
| 0  | 1 | 2 | 3 | 4  | 5  | 6   | 7   | 8  |
| -1 | 0 | 2 | 3 | 10 | 12 | -23 | -14 | -7 |

Para este vector, se devolvería el valor 27.

**Pregunta 1.** Implementa en Java un algoritmo basado divide y vencerás que resuelva el problema.

```
private static int sumaPositivosAux(int[] vector, int i0, int iN){
    if (i0==iN)
        if (vector[i0]>0) return vector[i0]; else return 0;
    else{
        int k=(i0+iN)/2;
        int s1=sumaPositivosAux(vector,i0,k);
        int s2=sumaPositivosAux(vector,k+1,iN);
        return s1+s2;
    }
}

public static int sumaPositivos1(int[] vector){
    return sumaPositivosAux(vector,0,vector.length-1);
}
```

**Pregunta 2 (3 puntos).** Calcula la complejidad del algoritmo implementado (justifica la respuesta)

El algoritmo implementado contempla siguiente ecuación de recurrencia:

$$T(N)=2 \cdot T(N/2) + \Theta(1) \text{ para } N>1$$

Aplicando el Teorema Maestro para el caso 1 (ya que  $\log_2 2=1>0$ ), la complejidad es

$$O(N)$$