

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

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

**Módulo 5. Voraces.** Diseñar e implementar el método *boolean hayDivision(int[] valores)* en Java basado en el esquema voraz, que permita determinar si, dado un conjunto de números enteros positivos mayores que 0, existen dos subconjuntos que sumen exactamente el mismo valor, y que cada uno contenga la mitad de los números del vector *valores*. Por ejemplo, dado el vector *valores* = {5, 8, 7, 1, 3, 2}, el algoritmo debería devolver *true*, dado que  $8+3+2 = 5+7+1$ ; dado el vector *valores* = {1, 6, 8, 1}, el algoritmo debería devolver *false*.

```
/* Se trata del problema del reparto equitativo con la restricción adicional de
que el número de bienes asignado a cada heredero debe ser exactamente
el mismo */
boolean hayDivision(int[] valores) {
    if ((valores.length%2)!=0) return false;
    else {
        ArrayList<Integer> copiaValores = new ArrayList<Integer>();
        for (int i = 0; i < valores.length; i++)
            copiaValores.add(valores[i]);
        int[] totalA_B = {0, 0};
        int[] cantidadA_B = {0,0};
        int c;

        while (!copiaValores.isEmpty()) {
            c = seleccionarCandidato(copiaValores);
            copiaValores.remove(new Integer(c));
            // el valor se asigna al subconjunto con menor
            // valor acumulado
            if (totalA_B[0]>totalA_B[1]) {
                totalA_B[1] = totalA_B[1]+c;
                cantidadA_B[1]++;
            }
            else{
                totalA_B[0] = totalA_B[0]+c;
                cantidadA_B[0]++;
            }
        }
        return ((totalA_B[0]==totalA_B[1]) &&
            (cantidadA_B[0]==cantidadA_B[1]));
    }
}
```

```
int seleccionarCandidato(ArrayList<Integer> candidatos){  
    int c = candidatos.get(0);  
    for (int i=1; i<candidatos.size();i++)  
        if (c < candidatos.get(i)) c=candidatos.get(i);  
    return c;  
}
```

// Otra posible implementación se basa en asignar de manera alternativa estricta los valores a los subconjuntos A y B (con la misma función voraz), aunque este enfoque sólo encuentra una solución si los valores están duplicados.