



Nº matrícula: _____ Nombre: _____

Apellidos: _____

Problema. Dado un vector, v , que contiene los valores de un conjunto de elementos, se desea comprobar si existen dos subconjuntos de elementos tal que la suma de valores de cada subconjunto sea exactamente igual a un valor predeterminado $vObjetivo$. Por ejemplo, dado el siguiente vector v :

0	1	2	3	4	5
4	2	5	5	1	8

Para $vObjetivo=10$, los **dos** subconjuntos de elementos $\{0,2,4\}$ y $\{1,5\}$ cumplen que $(v[0]+v[2]+v[4])=10$ y $(v[1]+v[5])=10$. Sin embargo, para $vObjetivo=11$ no existen **dos** subconjuntos en v tal que la suma de valores de cada subconjunto sea 11.

Implementar un algoritmo en Java, basado en el **esquema** de **Backtracking**, que valide si existen dos subconjuntos de elementos en el vector de valores cuya suma sea exactamente el valor objetivo:

`boolean dosSubconjuntos(int[] v, int vObjetivo)`

donde v es el vector de valores y $vObjetivo$ es el valor objetivo a conseguir. El método deberá devolver un valor *booleano* que indique si existen los dos subconjuntos que cumplan los requisitos o no. Se podrán implementar todos los métodos/clases adicionales que se consideren necesarios.

Aclaraciones: NO es necesario que todos los elementos del vector v formen parte de la solución. El vector v puede contener valores repetidos. Todos los valores de v son mayores que 0.

```
private static boolean dosSubconjuntos(int[] v, int vObjetivo){
    Booleano exito=new Booleano(false);
    int total=0;
    for (int i=0; i<v.length; i++) total=total+v[i];
    if ((total/2)>=vObjetivo)
        dosSubconjuntosAux(v, vObjetivo, 0, 0, 0, exito);
    return exito.getValor();
}
```

```

private static void dosSubconjuntosAux(int[] v, int vObjetivo, int obj,
                                       int acumA, int acumB, Booleano exito){
    if ((acumA==vObjetivo) && (acumB==vObjetivo)) exito.setValor(true);
    else if (obj<v.length){
        int c=0;
        // c=0 no se asigna v[obj] a ninguno de los subconjuntos
        // c=1 se intenta asignar v[obj] al subconjunto A
        // c=2 se intenta asignar v[obj] al subconjunto B
        while (c<3 && !exito.getValor()){
            if (c==0 || ((c==1) && acumA+v[obj] <=vObjetivo) ||
                ((c==2) && acumB+v[obj] <=vObjetivo)){
                if (c==1)    acumA = acumA + v[obj];
                else if (c==2)    acumB = acumB + v[obj];
                obj++;
                dosSubconjuntosAux(v, vObjetivo, obj, acumA, acumB, exito);
                if (!exito.getValor()){
                    obj--;
                    if (c==1)    acumA = acumA - v[obj];
                    else if (c==2)    acumB = acumB - v[obj];
                }
            }
            c++;
        }
    }
}

```

```

public class Booleano {
    private boolean valor;

    public Booleano(boolean valor) {
        this.valor = valor;
    }
    public boolean getValor() {
        return this.valor;
    }
    public void setValor(boolean valor) {
        this.valor = valor;
    }
}

```