

- Identifique qué parámetros debe tomar la función recursiva que resuelve el problema.
 - Describa con palabras **qué calcula** la misma, en función de sus argumentos.
 - Defina la función recursiva en notación matemática y opcionalmente en código.
 - Indique cuál es la llamada principal que obtiene el resultado pedido en el ejercicio.
7. En el problema de la mochila se buscaba el máximo valor alcanzable al seleccionar entre n objetos de valores v_1, \dots, v_n y pesos w_1, \dots, w_n , respectivamente, una combinación de ellos que quepa en una mochila de capacidad W . Si se tienen dos mochilas con capacidades W_1 y W_2 , ¿cuál es el valor máximo alcanzable al seleccionar objetos para cargar en ambas mochilas?

- **Parámetros de la función recursiva**

La función recursiva tomará **tres parámetros** para resolver el problema:

- c** , que indica la cantidad de objetos a guardar en alguna mochila. Cada objeto tiene un valor v_c y un peso w_c que lo identifica.
- $m1$** , que indica la capacidad de peso que hay disponible en la mochila 1.
- $m2$** , que indica la capacidad de peso que hay disponible en la mochila 2.

- **¿Qué calcula la función recursiva?**

Calcula el mayor valor alcanzable al seleccionar objetos, los cuales hay un total c para elegir, y guardarlos en las dos mochilas disponibles de las cuales la primera tiene capacidad $m1$ y la segunda tiene capacidad $m2$.

- **Función en notación matemática**

```
2mochilas(c,m1,m2) = ( si c = 0                                → 0
                      | si m1 = 0 ∧ m2 = 0                    → 0
                      | si c > 0 ∧ wc > m1 ∧ wc > m2        → 2mochilas(c-1,m1,m2)
                      | si c > 0 ∧ wc > m1 ∧ wc ≤ m2        → vc + 2mochilas(c-1,m1,m2-wc)
                                                                `max` 2mochilas(c-1,m1,m2)
                      | si c > 0 ∧ wc ≤ m1 ∧ wc > m2        → vc + 2mochilas(c-1,m1-wc,m2)
                                                                `max` 2mochilas(c-1,m1,m2)
                      | si c > 0 ∧ wc ≤ m1 ∧ wc ≤ m2        → vc + 2mochilas(c-1,m1-wc,m2)
                                                                `max` vc + 2mochilas(c-1,m1,m2-wc)
                                                                `max` 2mochilas(c-1,m1,m2)
                      )
```

Donde:

- Si no hay objetos ($c=0$), entonces no guardo nada y no obtengo ningún valor.
- Si ambas mochilas no tienen más espacio, no puedo guardar más objetos y por tanto no aumenta el valor.
- Si hay objetos pero el peso de tal objeto es superior a la capacidad de ambas mochilas, entonces descarto ese objeto.
- Si hay objetos pero solo puede entrar en la primera mochila, entonces elijo el máximo valor entre guardarlo en esa mochila o descartar el objeto.
- Si hay objetos pero solo puede entrar en la segunda mochila, entonces elijo el máximo valor entre guardarlo en esa mochila o descartar el objeto.
- Si hay objetos y al tomar un objeto este puede ser guardado en cualquier mochila, entonces elijo el máximo valor entre guardarlo en la primera mochila, guardarlo en la segunda mochila o descartar el objeto.

- **Llamada principal**

La llamada principal es **2mochilas($n, W1, W2$)**, donde n son todos los objetos a guardar en la mochila, $W1$ la capacidad inicial de la mochila 1 y $W2$ la capacidad inicial de la mochila 2, las cuales a medida que se van guardando objetos va disminuyendo.