Dada una mochila con una capacidad máxima de peso P, y dispones de n objetos. Cada objeto tiene un peso w y un valor v El objetivo es seleccionar algunos objetos de manera que maximicen el valor total, sin superar la capacidad de la mochila, que es de 6 Se pide realizar una prueba de escritorio para fuerza bruta y para programación dinámica (valor máximo se encuentra en la celda dp[n][B], donde B es la capacidad máxima de la mochila)

**Datos de Entrada**

* **Capacidad máxima de la mochila (P):** 6
* **Número de objetos (n):** 3
* **Pesos (w) y valores (v) de los objetos:**
  + Objeto 1: w1=2*w*1​=2, v1=10*v*1​=10
  + Objeto 2: w2=3*w*2​=3, v2=15*v*2​=15
  + Objeto 3: w3=5*w*3​=5, v3=20*v*3​=20

**1. Fuerza Bruta**

El enfoque de **fuerza bruta** consiste en evaluar todas las combinaciones posibles de objetos y seleccionar la que maximiza el valor sin exceder la capacidad de la mochila.

**Combinaciones posibles:**

**Sin objetos:**

Valor total: 0

Peso total: 0

**Objeto 1:**

Valor total: 10

Peso total: 2

**Objeto 2:**

Valor total: 15

Peso total: 3

**Objeto 3:**

Valor total: 20

Peso total: 5

**Objeto 1 + Objeto 2:**

Valor total: 10 + 15 = 25

Peso total: 2 + 3 = 5

**Objeto 1 + Objeto 3:**

Valor total: 10 + 20 = 30

Peso total: 2 + 5 = 7 (supera la capacidad, se descarta)

**Objeto 2 + Objeto 3:**

Valor total: 15 + 20 = 35

Peso total: 3 + 5 = 8 (supera la capacidad, se descarta)

**Objeto 1 + Objeto 2 + Objeto 3:**

Valor total: 10 + 15 + 20 = 45

Peso total: 2 + 3 + 5 = 10 (supera la capacidad, se descarta)

La combinación óptima es **Objeto 1 + Objeto 2** con un valor total de **25** y un peso total de **5**.