**9.  Usted se escapara al campo luego del examen de IA, deberá llevar varios artículos que no ingresan a su mochila. ¿Cómo optimizaría este problema permitiendo llevar la mayor cantidad de articulos?**

**Respuesta**

Dado un conjunto de objetos, cada uno con un peso y un valor, y una mochila con una capacidad máxima, el objetivo es maximizar el valor total de los artículos que se pueden incluir en la mochila sin exceder la capacidad.

from operator import itemgetter

OBJETOS = (

    ("objeto 1", 9, 150), ("objeto 2", 9, 160), ("objeto 3", 153, 200), ("objeto 4", 50, 160),

    ("objeto 5", 15, 60), ("objeto 6", 66, 45), ("objeto 7", 27, 60), ("objeto 8", 39, 40),

    ("objeto 9", 230, 591), ("objeto 10", 520, 10), ("objeto 11", 110, 70), ("objeto 12", 32, 30))

PESOMAXIMO = 150

get\_peso = itemgetter(1)

get\_valor = itemgetter(2)

def total\_peso(objeto):

    return sum(get\_peso(x) for x in objeto)

def total\_valor(objeto):

    return sum(get\_valor(x) for x in objeto)

print(total\_peso(OBJETOS), total\_valor(OBJETOS))

(1260, 1576)

def combinaciones(objeto, peso\_maximo):

    paqs = [ p for p in objeto if get\_peso(p) <= peso\_maximo ]

    resultado = []

    for p in paqs:

        res = combinaciones([x for x in paqs if x!=p], peso\_maximo - get\_peso(p))

        if len(res) == 0:

            resultado.append([p])

        else:

            resultado.extend([[p]+x for x in res])

    return resultado

from pprint import pprint

sol = max(combinaciones(OBJETOS, PESOMAXIMO), key=total\_valor)

print("Peso {} Valor {}".format(total\_peso(sol), total\_valor(sol)))

pprint(sol)

