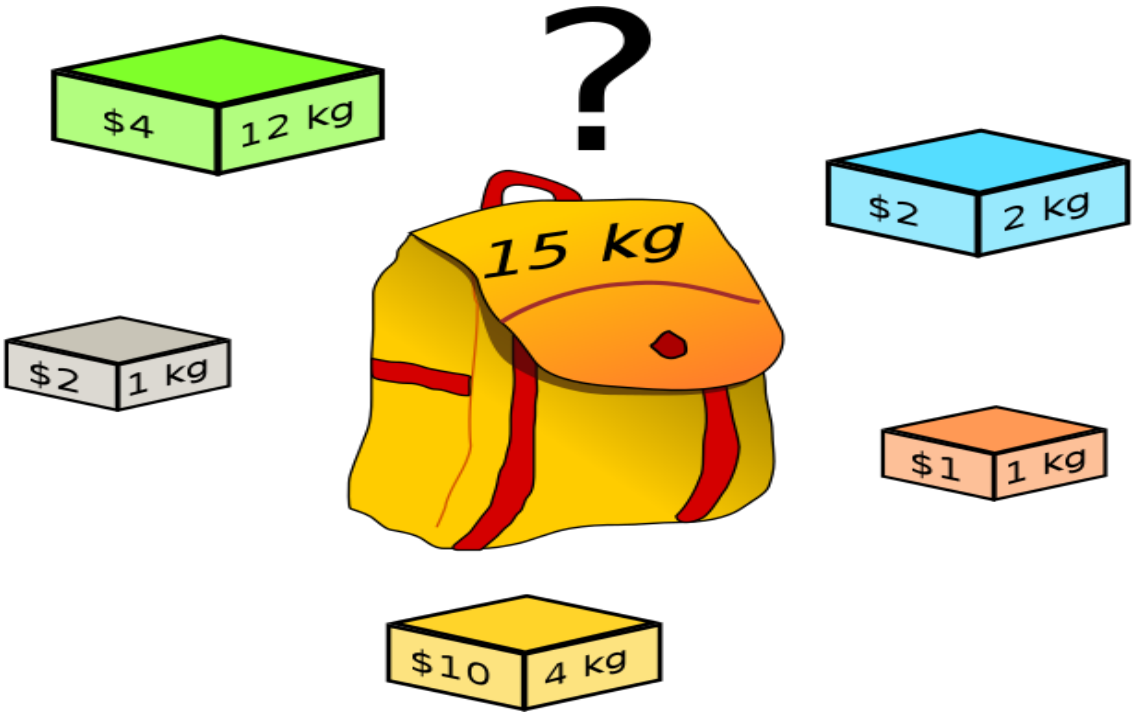
Knapsack Problem

The knapsack problem is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.



Code Provided:

Mainly 4 functions are already provided to you for your ease

1) **Generate Population**: Generate Population , given the size and backpack_Capacity

2) parent_selection(population) :

Select a parent from Population

Find 2 Fittest Individual to select parent Check

Total sum value of fittest individuals

3) **def apply_crossover:** Apply Crossover and Mutation on population, Given crossover probability and mutation probability

4) def apply_mutation(chromosome, backpack_capacity, mutation_probability):

Apply Mutation on chromosomes , given Mutation probability

------- Tasks To perform -----

1) def find_two_fittest_individuals(population):

Find Top 2 Fittest Individual from Population

2) def calculate_fitness(population, items, max_weight):

Calculate Fitness of population, given Items (weight,value) and max weight in action

Run The code Given at the end to check fitness values of your algorithm after every 100th generations.

Submission Guidelines:

- Lab must be submitted in the google classroom.
- Submission other than google classroom won't not be accepted.
- You are required to submit a python (version 3 compatible) file named after Your RollNo.