THE KNAPSACK PROBLEM

The goal of the Knapsack problem is to pack items into a bag without exceeding a threshold of 120kg, the primary objective is to have the knapsack filled with the most valuable items without repetition.

**As a GENETIC PROBLEM:**

We need to create candidate solutions to the search problem, the candidate solutions go through a natural selection process with the purpose of evolving the population via crossover and mutation to arrive at a solution.

An individual is defined by the collection of items in a knapsack.
Examples of individuals look like this:

**---- This is an individual ----**
**Fitness:** 6.5 **Rank:** 1

Item_1; Weight: 20 Value: 6
Item_6; Weight: 70 Value: 9

The chromosome of an individual is the combination of items in the knapsack.
In the case of the example individual: Item_1 and Item_6, represent a gene.

**Successor functions:** Mutation and Crossover

**Mutation:** The mutation operation in this problem, randomly selects a gene in a chromosome and negates its' current state.
for example, mutating the individual above could turn on or turn off the two item genes or turn on an item that's not activated already.

**Crossover:** The crossover operation takes two individuals and conjoins their genes from any random location forming a new individual with the mix of both parents genes. for example;

parent_1_genes: Item_1 and Item_3
parent_2_genes: Item_4 and Item_6
offspring_genes: Item_1, Item_3 and Item_4, Item_6

**<u>The fitness function</u>** is a heuristic function that tells the algorithm how close or far it is from the goal. In our case, we still consider individuals with chromosomes that exceed the weight limit, all we do is assign those individuals very low scores in comparison to individuals with weights below the limit.
The algorithm assigns individuals above the weight threshold a fitness that never exceeds 5, no matter how valuable that individual may be.
Individuals below the threshold however can get up to the max fitness of 10. But with the added consideration of their value; this consideration is addressed

in the code by normalizing the total value of an individual and summing this value with 5. This strategy enables us to determine the fittest individuals below the weight threshold.

The **genetic algorithm** is guided by a generation limit and the solution test is the fitness function.

**GENOME FOR THE PROBLEM:**
As stated already, the genome for the problem is a combination of items in a knapsack.
i.e;     offspring_genes: Item_1, Item_3 and Item_4, Item_6

**FRINGE OPERATIONS:** Crossover and Mutation
POPULATION CULLED BY 50%

Appendix --
- The global search space was formed by generating all combinations of
  a set items from the total 7 items.
  Global space: 7C1 + 7C2 + ... + 7C7

- Out of all these combinations we select 40 of them to begin the search space.

- It is worth noting the solution to this problem exists in a combination of three
  out of the seven items. Thus, the initial global search space of this problem
  excludes the combination: 7C3. Only to make things more interesting and prove the
  validity of our algorithm.

- Due to the technical challenge of crossover, genes in an individual are stored
  in a list of keys (1 or 0), that lets us know what genes are on. Thus;
  Individual key: [0, 1, 1, 0, 0, 0, 0]
  Individual genes: [Item_2, Item_3]

-Mutation happens to 30% of the population. Crossover repopulates a culled population.