**Task 4. DEAP Framework (08.11.2022, 15.11.2022 (2 practicals))**

1. Solve two problems (KP and TSP) using the DEAP Framework. The complete project code for both problems can be found [here](https://github.com/PacktPublishing/Hands-On-Genetic-Algorithms-with-Python/tree/master/Chapter04).   
   The found solution probably won’t be the best-known one, so try to figure out ways to improve the results. Experiment with changing the population size (POPULATION\_SIZE), number of generations (MAX\_GENERATIONS), the probabilities (P\_CROSSOVER, P\_MUTATION), and the tournament size (tournsize).   
   Add 5-6 screenshots for each task of the “Max/Min and average fitness over generations” graph. Improving the results means that on the graph the best solution is shown to be found in less than X generations. The smaller X the better.
2. For the knapsack problem (KP) create your own list of self.items ([line 25](https://github.com/PacktPublishing/Hands-On-Genetic-Algorithms-with-Python/blob/master/Chapter04/knapsack.py)). The table consists of name, weight and value for each item. The goal is to come up with a group of selected items that will provide the maximum total value, without exceeding the total weight capacity of the bag.  
   The knapsack problem consists of the following components:

             - A set of items, each of them associated with a certain value and a certain weight.

             - A bag/sack/container (the knapsack) of a certain weight capacity.

1. For the traveling salesman problem (TSP) the variants ([TSP\_NAME](https://github.com/PacktPublishing/Hands-On-Genetic-Algorithms-with-Python/blob/master/Chapter04/02-solve-tsp-first-attempt.py)) to choose from can be found [here](http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/) (\*.tsp).