M6: Knapsack Algorithm Assignment

Instructions

Design and write an algorithm in Python to solve the Knapsack Problem. Develop your algorithm within the provided Python program, `knapsack-template.py`, according to these directions:

- Retrieve the assignment materials from the Assignment page; rename your file to `KnapsackDev.py`.
- Implement your algorithm in the Python program named `KnapsackDev.py` in the function named `loadKnapsack()`.
- Use `KnapsackDev.py` to test your algorithm for the 10 problems in the accompanying JSON data file, `knapsack.json`.

Guidance

Details on the `loadKnapsack()` function:

- Your task:
  - Program `loadKnapsack()` so that it creates a list of item ID numbers (integers) in the `items_to_pack` variable indicating which items are placed in the knapsack.
- Input arguments:
  - `items`: a dictionary of the items not yet in the knapsack, with:
    - keys representing the item ID numbers (integer);
    - values as 2-tuples where the 0th element is the volume of the item and the 1st element is the value of the item;
      - e.g., {0: (1,2), 1: (2,5)…}
    - The code in `KnapsackDev.py` already creates this dictionary from a JSON file and inputs it into `loadKnapsack()`.
  - `knapsack_cap`: This value represents the volume capacity of the knapsack. Its value is already set in the `KnapsackDev.py` program based on the JSON data in `knapsack.json`. You need not make any changes.
- Output arguments:
  - `myUsername`: your W&M username in order to identify the output as your work
  - `myNickname`: a string variable indicating how you would like to be identified on a leaderboard showing the scores of everybody in class, which will be visible to your classmates. If you choose not to be identified on the leaderboard, you may return an empty string, that is, `''`.
  - `items_to_pack`: the list containing the item ID numbers your algorithm will place in the knapsack

Submission

- Once you have completed developing your algorithm, then sumbit your `KnapsackDev.py` file via Canvas.