- Variables.py -

all the variables required for the linear program models. Read from xls and create dictionaties for PuLP to consume

- part1-dietmodel.py -
Part1 of the assignment which is :

Formulate an optimization model (a linear program) to find the cheapest diet that satisfies the maximum and minimum daily nutrition constraints, and solve it using PuLP. Turn in your code and the solution. (The optimal solution should be a diet of air-popped popcorn, poached eggs, oranges, raw iceberg lettuce, raw celery, and frozen broccoli. UGH!)

The objective function is of LpMinimize type, we are minimizing the cost of the diet plan.
Here we create constraints using PulP's lpSum function and use the constraints conditions provided in the xls to create our constraints

Output: The optimal solution is:

Portion_Celery,_Raw=52.64371
Portion_Frozen_Broccoli=0.25960653
Portion_Lettuce,Iceberg,Raw=63.988506
Portion_Oranges=2.2929389
Portion_Poached_Eggs=0.14184397
Portion_Popcorn,Air_Popped=13.869322

Total Cost of Ingredients per can = 4.337116797399999

-----***-----*****----------*****-----*****----------*****-----*****----------*****-----*****----------*****-----**

-part2-dietmodel.py -

Part2 of the assignment which is :
Please add to your model the following constraints (which might require adding more variables) and solve the new model:

a. If a food is selected, then a minimum of 1/10 serving must be chosen. (Hint: now you will need two variables for each food i: whether it is chosen, and how much is part of the diet. You’ll also need to write a constraint to link them.)

b. Many people dislike celery and frozen broccoli. So at most one, but not both, can be selected.

c. To get day-to-day variety in protein, at least 3 kinds of meat/poultry/fish/eggs must be selected. [If something is ambiguous (e.g., should bean-and-bacon soup be considered meat?), just call it whatever you think is appropriate – I want you to learn how to write

a. Here we create additional variables to hold the food chosen and the portion sizes and link the two using a constraint that limits the portion size to a minimum of 0.1 (1/10)

b. Using binary variables created above (step a, food_chosen) we make sure food_chosen['Frozen Broccoli']+food_chosen['Celery, Raw']<=1 either celery or broccoli can be picked not both to satisfy the condition i.e. celery=0 + broccoli=1 or celery=1 + broccoli=0

c. To add the protein constraint, we create a list of ingredients that are proteins and ensure via the constraint: prob += lpSum([food_chosen[p] for p in protein_choices]) >= 3.0 atleast 3 is selected, remember food_choosen is binary so atleast 3 need to be selected for it to satisfy the condition >=3.0

The rest is similar to part1, where we use PuLP's solve method to solve using PuLP's choice of solver.

'''
Output:

Status:Optimal

Chosen Ingredients:

Chosen_Celery,_Raw=1.0
Chosen_Kielbasa,Prk=1.0
Chosen_Lettuce,Iceberg,Raw=1.0
Chosen_Oranges=1.0
Chosen_Peanut_Butter=1.0
Chosen_Poached_Eggs=1.0
Chosen_Popcorn,Air_Popped=1.0
Chosen_Scrambled_Eggs=1.0

Portion sizes:

Portion_Celery,_Raw=42.399358
Portion_Kielbasa,Prk=0.1
Portion_Lettuce,Iceberg,Raw=82.802586
Portion_Oranges=3.0771841
Portion_Peanut_Butter=1.9429716
Portion_Poached_Eggs=0.1

Portion_Popcorn,Air_Popped=13.223294
Portion_Scrambled_Eggs=0.1

Total Cost:

Total Cost of food_items per can = 4.512543427000001