Assignment 9
Due: Wednesday, November 6
11 pts (+3 pts bonus)

1. (5 pts total) Write code to pack your suitcase. I've included a csv file that has our items from our Conceptualize activity, but now a weight is also associated with each item. Read in this file to pack your suitcase using a greedy approach. Your implementation should be written as a function which takes in as parameters the path/name of the csv file and the max allowed weight, *w*. The function will return the total weight and a list of the items packed.

A. (3 pts) Implement the fractional knapsack problem. (Code: LASTNAME_9.1A.py)

B. (2 pts) Implement the integer knapsack problem. (Code: LASTNAME_9.1B.py)

2. (2 pts) Let's say the various things you want to pack into your knapsack are partly divisible. That is, you can divide them at certain evenly spaced points (such as a candy bar divided into squares). The different items have different spacings between their breaking points. Could a greedy algorithm still work? Justify your answer. (Exercise 7-6) (Answer in Word/PDF)

Yeah it would work if the knapsack and the interments are compatible in that the knapsack is divisible by them. It would essentially be integer knapsack but with each spaced out part as it's own item.

3. (4 pts total) (3 pts) Let's assume that your suitcase has two compartments, each in equal size, such that each can fit *w*/2. Adapt your code from #1A and #1B to consider this partitioned suitcase. (Code: LASTNAME_9.3A.py and LASTNAME_9.3B.py)

(1 pt) Will the items packed in this partitioned suitcase necessarily be the same as in the unpartitioned suitcase? Why/ why not? Justify your answer. Which is likely to be able to pack more and under what scenarios? (Answer in Word/PDF)

It depends on the scenario but for this dataset it will pretty much be the same. The highest value items have a low weight so it usually won't make a difference if there is a middle partition. If the big items were higher value it would definitely make a difference. For example if the weight limit was 1000 and there was an item with a weight of 550 but it was worth like 10 points we would not be able to pack it due to the partition.

4. (3 pts) Quintessential to the greedy algorithm is making a choice and never looking back. Let's assume that we are able to backtrack to consider an alternative solution, but just one. In other words, let's say that we've packed 10 items in our backpack. We have space left over, but none of the items we could pack fit in this space. If we roll back one, i.e., choose a different item #10 say i_y rather than i_x, could we fit in an 11th item, i_z, such that value(i_y+i_z) > value(i_x)? (Code: LASTNAME_9.4.py)