

## Part 2: .txt Files

For Part 1 functions, you will use the provided plants.txt file. Assume that the file will be formatted as follows:

```
plant_name
expected_lifetime
plant_size
$price

plant_name
expected_lifetime
plant_size
$price
...
```

There will be a line with the name of a household plant followed by a line with the expected lifetime (in years), the size of the plant, followed by the price.

An example file would be formatted as follows

```
Croton
39.4
Medium
$36.49

Lemon Lime Dracaena
37.5
Large
$49.99
```

## 2. Plant Finder

**Function name:** `plant_finder()`

**Parameters:** `old_file(str)`, `new_file(str)`, `plant_size(str)`

**Returns:** a dictionary

**Description:** You have decided to buy a bunch of plants and found many options while searching online. Write a function that takes in a .txt file name configured in the format above. Your function should also take in the parameters `plant_size` and `new_file`. Your function should identify all the plants of size `plant_size` and write these to a new file with the name being the parameter `new_file`. The new file should follow the

same format as the file passed in. Your function should then return a dictionary mapping the plant names to its corresponding lifespan.

**Test cases:**

```
>>> old_file = "plants.txt"
>>> new_file = "large_plants.txt"
>>> plant_size = "Large"
>>> print(plant_finder(old_file, new_file, plant_size))
{'Lemon Lime Dracaena': '37.5', 'Lucky Bamboo': '1.8',
'Dracaena Marginata': '9.0', 'Ponytail Palm': '6.2',
'Majesty Palm': '8.1'}
```

The generated file large\_plants.txt:

Lemon Lime Dracaena  
37.5  
Large  
\$49.99

Lucky Bamboo  
1.8  
Large  
\$17.00

Dracaena Marginata  
9.0  
Large  
\$109.95

Ponytail Palm  
6.2  
Large  
\$65.00

Majesty Palm  
8.1  
Large  
\$68.49

```
>>> old_file = "plants.txt"
>>> new_file = "medium_plants.txt"
>>> plant_size = "Medium"
>>> print(plant_finder(old_file, new_file, plant_size))
{'Croton': '39.4', 'Bunny Ear Cactus': '7.5', 'Golden
Pothos': '7.7', 'Peace Lily': '3.9'}
```

The generated file medium\_plants.txt:

```
Croton
39.4
Medium
$36.49
```

```
Bunny Ear Cactus
7.5
Medium
$14.85
```

```
Golden Pothos
7.7
Medium
$24.99
```

```
Peace Lily
3.9
Medium
$29.95
```

### 3. Plant Splitter

**Function name:** `plant_splitter()`

**Parameters:** `new_file` (str), `num_people` (int), `tax` (float)

**Returns:** a float

**Description:** Now that you have decided which plants you want to buy, you and your roommates have decided to go ahead and order them. Using the file `new_file` created in the previous question, write a function which calculates how much each person will have to pay for the plants if it's split between `num_people` evenly with tax. Round this

amount to 2 decimal places. You may assume the tax amount will be between 0 and 1, which is a percentage which must be added on top of the cost of the flowers.

**Test cases:**

```
>>> print(plant_splitter("large_plants.txt", 5, .2))  
74.5
```

```
>>> print(plant_splitter("medium_plants.txt", 7, .6))  
24.29
```

**Note:** large\_plants.txt and medium\_plants.txt were the two sample files generated in the previous problem.

### Part 3: .csv Files

For this part, you will use the provided food.csv file. Assume the file will be as follows:

item	price	rating	restaurant
item 1	price 1	rating 1	restaurant 1
item 2	price 2	rating 2	restaurant 2
...	...	...	...

There will be a header section at the top of the .csv file

## 4. Eating Out

**Function name:** `eating_out()`

**Parameters:** `file_name (str)`

**Returns:** a dictionary

**Description:** After shopping for all of those plants, you decide to go out to eat. However, since you spent so much on plants, you want to buy the cheapest thing possible on the menu. Write a function that takes in a csv file as a string and then returns a dictionary in the format `{restaurant: (str): (cheapest item (str), price (str) }`