GCIS 123
Practicum 3

Problem 1: insertion_sort_backwards (10 points)

Open the `1_sort.py` module and complete the `insertion_sort_backwards` function. This new version of the insertion sort should work as follows:

- Begin by dividing the array into two partitions: an unsorted partition and a sorted partition.
- At the beginning of each iteration, the rightmost element in the unsorted partition is moved to the leftmost position in the sorted partition.
- If the new element is out of place, it is swapped with its neighbor to the right.
- This swapping continues until the element is in the correct position.
- Repeat until the sorted partition is the entire list.
- Hint: You may want to define a helper function named `shift_backwards`.

In the main function, test your `insertion_sort_backwards` with various arrays. Using array [5, 3, 7, 4, 1] as an example, print the array at the end of each iteration of the for loop. Your program output should be:

[5, 3, 7, 1, 4]
[5, 3, 1, 4, 7]
[5, 1, 3, 4, 7]
[1, 3, 4, 5, 7]

Problem 2: List comprehensions (10 points)

Open the `2_list.py` module. Write the `build_table2` function that works the same way as `build_table1` except the new function must use list comprehensions. In order to get full credit, the function body should have only one line of code.

Problem 3: Sets and Dictionaries (10 points)

Take a look at the two text files in the data directory named “points_small.txt" and “points.txt". Each file contains x- and y-coordinates of points in the plane. Open the `3_dictionaries.py` module and write the following functions:

1. The `build_x_to_y` function reads the file and builds a dictionary that maps an x-coordinate to a list of y-coordinates of the points with the same x-coordinate.
2. The `build_y_to_x` function declares a parameter for the dictionary you built in the previous question and builds a new dictionary that maps a y-coordinate to a *unique* set of x-coordinates of the points with the same y-coordinate.

Example: Using the following file (“points_small.txt”) as an example,

```
10 20
15 20
10 30
15 25
15 20
```

you will create two dictionaries:

```
x_to_y = {10: [20, 30], 15: [20, 25, 20]}
y_to_x = {20: {10, 15}, 30: {10}, 25: {15}}
```