

Take Home Questions:

1. Python exercise: Create a python function `ascii_density_histogram()` that takes a list of numbers as input and returns (or prints out) a density histogram plotting the distribution of the numbers in the list. Use only base python functions (i.e. the word *import* must not appear in your code).

Here is an example of the desired behaviour of the function:

```
histogram = ascii_density_histogram(  
    value_list = [-14,0,2,0,7,-5,0,0,18,11,-2,10,5,6,-6,-3,5,10,-11,16]  
    , n_bins = 6  
)  
  
print(histogram)  
✓ 0.7s
```

```
[ -14.0,  -8.67) oooooooooooooooooooooo  
[  -8.67,  -3.33) oooooooooooooooooooooo  
[  -3.33,   2.0) oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo  
[   2.0,   7.33) oooooooooooooooooooooooooooooooooooooooooooooooooooooo  
[   7.33,  12.67) oooooooooooooooooooooooooooooo  
[  12.67,  18.0] oooooooooooooooooooooo
```

The output of your function needn't look exactly like this, this is just an example. You are welcome to be creative with your design.

2. SQL exercise: Given a table of **customer order counts** and a table containing **item category information**, write a SQL query which returns a count of orders per customer per item category. You can expect the **order counts** table to have a few million rows. For illustration, here is an example using small toy tables:

Customer order counts:

```
select * from #orders;
```

	customerID	itemID
1	A	1
2	A	1
3	A	3
4	A	5
5	B	2
6	B	3
7	B	4
8	C	2
9	C	3
10	C	5

Item category information:

```
select * from #item_info;
```

	itemID	item_category
1	1	clothing
2	2	blankets
3	3	blankets
4	4	appliances
5	5	clothing

Desired output of SQL query:

```
select * from #customer_order_cnts_per_item_category;
```

	customerID	item_category	n_orders
1	A	appliances	0
2	A	blankets	1
3	A	clothing	3
4	B	appliances	1
5	B	blankets	2
6	B	clothing	0
7	C	appliances	0
8	C	blankets	2
9	C	clothing	1