

- Given a dictionary of dictionaries, write a function `california_results` that returns a list of tuples where each tuple holds the name of the pollster as the first element and the edge corresponding to CA (california) as the second element. If that pollster does not have an edge for CA, store its value as `None`.

```
data = { "Gallup": { "WA": 7, "CA": 15, "UT": -30 },
        "SurveyUSA": { "CA": 14, "CO": 2, "CT": 13, "FL": 0, "KY": -14 },
        "Omniscient": { "AK": -14.0, "AL": -22.3, "CA": 20.9 },
        "RAND": { "NY": 11.2, "AZ": -9.8, "AR": -18.9 },
        ... }
```

Use the following function and list to help answer problems 2 & 3

```
def row_to_edge(row):
    '''
    Given an election result row or poll data row, returns the Democratic
    edge in that state.
    '''
    return float(row["Dem"]) - float(row["Rep"])

rows = [ { "State": "AK", "Dem": "41.3", "Rep": "55.3" },
         { "State": "AL", "Dem": "38.4", "Rep": "60.7" },
         { "State": "AR", "Dem": "36.9", "Rep": "60.5" },
         ... ]
```

- Given the function `row_to_edge` and a list of rows, print the number of Democratic states, Republican states, and neutral states in the list. If `row_to_edge` returns a positive float, then that state is considered to be a Democratic state, if it returns a negative float then the state is considered to be a Republican state, if it returns 0 the state is considered to be a neutral state.
- Print the "most Democratic" state and the "most Republican" state. Most Democratic is defined as the state with the lowest edge, and most Republican is defined as the state with the highest edge.
- Write one line of code that will execute the following commands,
 - Print a list of the keys in the dictionary `data`
 - Print a list of the values in the dictionary `data`
 - Print all of the key-value pairs in the dictionary `data` as a list of tuples
 - Print all of the keys in the dictionary associated to the pollster "Gallup"
 - Reassign the first element of the tuple, `t = ('CSE', 140)`, to 'CHEM'

1.

```
def californian_results(data):
    '''
    Given a dictionary of pollsters mapped to rows, returns a list
    of tuples containing the pollster's name and its corresponding
    edge for CA. If there is no edge specified for CA, stores None.
    '''

    ca_results = []
    for pollster in data:
        if('CA' in data[pollster].keys()):
            tup = (pollster, data[pollster]['CA'])
        else:
            tup = (pollster, None)
        ca_results.append(tup)
    return ca_results
```

2.

```
num_dem = 0
num_rep = 0
num_neutral = 0
for row in rows:
    if (row_to_edge(row) > 0):
        num_dem += 1
    elif (row_to_edge(row) < 0):
        num_rep += 1
    else:
        num_neutral += 1

print "Democratic States: " + str(num_dem)
print "Republican States: " + str(num_rep)
print "Neutral States: " + str(num_neutral)
```

3.

```
most_dem_state = None
most_rep_state = None
min_edge = 100.0
max_edge = -100.0
for row in rows:
    edge = row_to_edge(row)
    if (edge > max_edge):
        max_edge = edge
        most_dem_state = row['State']
    if (edge < min):
        min_edge = edge
        most_rep_state = row['State']

print "Most Democratic state: " + most_dem_state
print "Most Republican state: " + most_rep_state
```

4.

a.) print data.keys()
 b.) print data.values()
 c.) print data.items()
 d.) print data["Gallup"].keys()
 e.) Not possible! Tuples are immutable which means they cannot be changed once they have been created.