

1. Print the number of Democratic states, Republican states, and neutral states in the list:

```
num_dem = 0
num_rep = 0
num_neutral = 0

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

print "Democratic States:", num_dem
print "Republican States:", num_rep
print "Neutral States:", num_neutral
```

2. Print the “most Democratic” state and the “most Republican” state:

```
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
```

3. (a) print data.keys()  
(b) print data.items()  
(c) print data["Gallup"].keys()  
(d) print data["RAND"].get("CA", None)

4. Write a function `get_results_for(data, state)` that returns a list of tuples `(pollster, edge)`:

```
def get_results_for(data, state):
    """
    Given a dictionary of pollsters mapped to rows, returns a list
    of tuples containing the pollster's name and it's corresponding
    edge for state. If there is no edge specified for state, stores None.
    """
    results = []

    for pollster in data:
        if state in data[pollster].keys():
            pollster_edge = (pollster, data[pollster][state])
        else:
            pollster_edge = (pollster, None)
        results.append(pollster_edge)

    return results
```

5. Write a function that returns the list of tuples for California "CA":

```
def california_results(data):
    return get_results_for(data, "CA")
```