CSE 160 Section 5 Problems

Given a dictionary of dictionaries with the following structure:

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

1. Write one line of code that will execute the following commands,
   a.) Print a list of the keys in the dictionary data
   b.) Print all of the key-value pairs in the dictionary data as a list of tuples
   c.) Print all of the keys in the dictionary associated to the pollster "Gallup"
   d.) Print the edge for "RAND" and ’CA’ or None if it does not exist.

2. Write a function get_results_for(data,state) that returns a list of tuples (pollster,edge). The first element is the name of the pollster and the second element the edge corresponding to the given state. If the pollster and state do not have an edge, store its value as None.

3. Write a function that returns the list of tuples for California (’CA’). You can use your function get_resutls_for(data,state) from Problem 2.

Use the following function and list to help answer problems 4 & 5

```
  def row_to_edge(row):
      ’’’
      Returns the difference between the "Dem" and "Rep" values of row.
      ’’’
      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" },
          ... ]
```

4. 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.

5. 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.

CSE 160 Section 5 Solutions

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

2. 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()):
               tup = (pollster, data[pollster][’CA’])
           else:
               tup = (pollster, None)
           results.append(tup)
       return results

3. def california_results(data):
      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

4. 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)


5.  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
```