CSE 160 Section 3 Problems

Question 1 - Part A

After the following lines of code are printed, what values
are stored in the set output_set?

```
        input_list = [3,1,4,1,5,9,2,6,5,3,5,9]
        output_set = set([])
        for i in input_list:
                output_set.add(i)
```

Question 2

In one line of code, print the set of all numbers that are
in both the sets. (i.e. - their intersection)

```
        set_one = {’a’, ’b’, ’c’, ’d’, ’e’, ’f’}
        set_two = {’a’, ’c’, ’d’,’g’}
```

Question 3

Given the following code:

```
weather = {
    ’Monday’:{’low’:45,’high’:62,’precipitation’:0.3},
    ’Tuesday’:{’low’:48,’high’:69,’precipitation’:0.2},
        ’Wednesday’:{’low’:42,’high’:58,’precipitation’:0.5}
}
```

What does the following code print:

```
    print weather[’Monday’][’high’]
    print weather[’Tuesday’][0]
    print weather[’Wednesday’]
```

Question 4

For each of the following questions, implement a function that returns the answer to the question. For each question, follow this procedure:

1. Identify a good name for the function.
2. Identify the return value.
3. Identify any necessary parameters.
4. Write the function definition.
5. Write the function's docstring.
6. Describe, on paper, in words, or in your head, the algorithm you'll use
to solve the problem.
7. Implement the function.

Effective programmers perform these steps for every program they write, and they solve them very quickly. Ineffective programmers often make the mistake of mixing steps 6 and 7, either skipping step 6 or performing 6 and 7 simultaneously. As mentioned in lecture, the most effective programmers start coding later, and finish earlier. This is because they understood the problem and its solution before ever writing code.

You may assume that every function takes as a parameter a data structure with the same structure as the following.

```
[
{'Company': 'Apple', 'Date': '01 Jan 2010', 'Purpose': 'payroll', 'Cost': 95},
{'Company': 'Microsoft', 'Date': '01 Feb 2010', 'Purpose': 'payroll', 'Cost': 1000},
{'Company': 'Microsoft', 'Date': '16 June 2010', 'Purpose': 'philanthropy', 'Cost': 250},
{'Company': 'Google', 'Date': '11 Dec 2010', 'Purpose': 'legal', 'Cost': 110},
{'Company': 'Apple', 'Date': '30 Aug 2010', 'Purpose': 'building renovations', 'Cost': 10000},
{'Company': 'Google', 'Date': '29 Dec 2010', 'Purpose': 'equipment', 'Cost': 25},
{'Company': 'Microsoft', 'Date': '01 Feb 2010', 'Purpose': 'payroll', 'Cost': 1000},
{'Company': 'Oracle', 'Date': '01 Jan 2010', 'Purpose': 'payroll', 'Cost': 30},
... More rows, with more companies, dates, purposes, and costs ...
]
```

You are allowed to use the solutions from previous problems to solve later problems.

1. For a given company, how much did that company spend in total?

2. For a given company, how much did that company spend on payroll?

3. What are all the companies that made expenditures?

4. Which company spent the least in total?

5. What are the evil companies, if any, that didn't have any payroll expenditures (and so clearly aren't paying their employees). When you've finished this function, do a Google search on "google motto".

```
CSE 140 Section 3 Solutions

Question 1 - Part A
        {1, 6, 3, 9, 5, 2, 4}
        * Remember sets are unordered, so the order of the values above
          has no significance



Question 2
        set_one & set_two
        The intersection is: {'a', 'c', 'd'}



Question 3
    62
    KeyError: 0
    {'low':42,'high':58,'precipitation':0.5}




Question 3


For a given company, how much did that company spend in total?

def total_costs(expenditures, company):
        '''
        Given a company's name (string) and a list of expenditure reports,
        returns the total expenditures for that company.
        '''

        # Keep a running sum of expenditures for the given company
        total = 0
        for item in expenditures:
                if item['Company'] == company:
                        total += item['Cost']
        return total


For a given company, how much did that company spend on payroll?

def payroll_costs(expenditures, company):
        '''
        Given a company's name (string) and a list of expenditures, returns the
        total payroll expenditures for that company.
        '''

        total = 0
        for item in expenditures:
                if item['Company'] == company and item['Purpose'] == 'payroll':
                        total = total + item['Cost']
        return total


What are all the companies that made expenditures?

def all_companies(expenditures):
        '''
        Given a list of expenditures, returns a set of all the companies that
        had expenditures.
        '''

        companies = set()
        for item in expenditures:
                companies.add(item['Company'])
        return companies
```

Which company spent the least in total?

```
def lowest_costs(expenditures):
        '''
        Given a list of expenditure reports, returns the name of the company
        that spent the least money.
        '''

        # Get a set of all companies
        companies = all_companies(expenditures)

        # Get a dictionary from a company to that company’s expenditures
        costs = {}
        for company in companies:
                costs[company] = total_costs(expenditures, company)

        # Keep a running min of the lowest expenditures
        min_name = None
        for company_name in costs:
                if min_name == None or costs[company_name] < costs[min_name]:
                        min_name = company_name
        return min_name
```

What are the evil companies, if any, that didn’t have any payroll expenditures (and so clearly aren’t paying their employees). When you’ve finished this function, do a Google search on "google motto".

```
def evil_companies(expenditures):
        '''
        Returns a set of evil companies (the ones that aren’t paying their
        employees, the scum-sucking corporate jerks).
        '''

        companies = all_companies(expenditures)

        payrolls = {}
        for company in companies:
                payrolls[company] = payroll_costs(expenditures, company)

        evil_companies = set()
        for company in payrolls:
                if payrolls[company] == 0:
                        evil_companies.add(company)

        return evil_companies
```