

CSE 160 Section 5 Solutions

1. You have a file named `students.txt` where each line contains a student's name and age separated by a comma, like this:

Alice,20
Bob,22

Write a program that reads the file and prints each student's name and age separately.

Example output:

Name: Alice, Age: 20

```
students_file = open("students.txt")
for line in students_file:
    parts = line.split(",")
    name = parts[0]
    age = parts[1]
    print("Name:", name, "Age:", age)
students_file.close()
```

2. Write a function called `count_words(filename)` that reads the file and returns the total number of words across all lines. The file is formatted so that each line is a sentence.

For example, if you had a file `quotes.txt` that contains:

Hi my name is Annie.
Let's learn Python!

The returned value of `count_words("quotes.txt")` would be:

```
def count_words(filename):  
    word_count = 0  
    my_file = open(filename)  
    for line in my_file:  
        words = line.split()  
        word_count += len(words)  
    my_file.close()  
    return word_count
```

- 3.. Create `get_squares (number_list)` that accepts a list of numbers and returns a dictionary mapping each number in the list to its square.

Ex: `get_squares([1, 4, 4])` should return `{1:1, 4:16}`

```
def get_squares(nums):  
    squares = {}  
    for num in nums:  
        squares[num] = num ** 2  
    return squares
```

4. Write `coldest_city (city_temperatures)` that takes in a dictionary and return the city (key) with the lowest temperature (value).

Ex: `city_temperatures = {'Seattle': 36, 'Cupertino': 39, 'New York': 57}` `coldest_city(city_temperatures)` returns `"Seattle"`

```

def coldest_city(city_temps):
    low_temp = None
    city_return = ""
    for city in city_temps:
        temp = city_temps[city]
        if (low_temp is None) or (temp < low_temp):
            low_temp = temp
            city_return = city
    return city_return

```

5. (a) Write `pokemon_types` (`pokemon_dict`) that takes parameter `pokemon_dict` and returns a new dictionary mapping each type of pokemon to the number pokemon in `pokemon_dict` with that type.

Ex: when `pokemon_dict = {"pikachu": "electric", "charmander": "fire", "charizard": "fire"}`
then.... `pokemon_types(pokemon_dict)` returns `{"electric" : 1, "fire":2}`

```

def pokemon_types(pokemon_dict):
    freq_dict = {}
    for pokemon, pok_type in pokemon_dict.items():
        if pok_type not in freq_dict:
            freq_dict[pok_type] = 0
        freq_dict[pok_type] += 1
    return freq_dict

```

(b) Returns a new dictionary mapping each type of pokemon to the list of pokemon with that type.

For example, when `pokemon_dict=`

```
{"pikachu": "electric", "charmander": "fire", "charizard": "fire"}
```

Then.... `pokemon_types(pokemon_dict)` returns `{'electric': ['pikachu'], 'fire': ['charmander', 'charizard']}`

```
def get_num_types(pokemon_dict):  
    '''takes a dictionary pokemon_types mapping pokemon  
        to their type and returns a new dictionary  
        mapping pokemon type to the a list of the  
        pokemon in pokemon_type with that type.  
    ...  
  
    result = {}  
  
    for pokemon, pok_type in pokemon_dict.items():  
        if pok_type not in result:  
            result[pok_type] = []  
        result[pok_type].append(pokemon)  
  
    return result
```