HOMEWORK 8

1. Getting information from a dictionary

Problem 1. Write a function named higher_than_average(d) that takes a dictionary d as input. In this dictionary d, the keys represent students enrolled in Chem 101, and the values represent their midterm scores. The function should return a list of students who scored above the average midterm score. For example, for

```
d = {
    "Alice": 85,
    "Bob": 78,
    "Charlie": 92,
    "Daisy": 88,
    "Ethan": 76}
higher_than_average(d)
```

the average score is 83.8. As a result, the function should return the list

```
["Alice", "Charlie", "Daisy"]
```

Problem 2. Write a function called total_price(quantity_dict, price_dict) that takes in quantity_dict and price_dict as input and returns the total cost of all items. Assume keys in both dictionaries are the same, and the values in quantity_dict are integers. For example,

```
total_price({"fries": 7, "hot dogs": 9, "soda": 9}, {"fries": 1.5,
    "hot dogs": 1, "soda": 1.1})
```

should return 29.4.

Problem 3. Write a function called duplicated_data(dictionar1, dictionary2) that takes in dictionary1 and dictionary2 as parameters and returns a dictionary with key-value pairs that are in both dictionaries. For example,

```
duplicated_data({"apple": 10, "banana" : 20, "cherry" : 30}, {"
   apple": 10, "banana" : 20, "mango" : 5})
```

should return

```
{"apple": 10, "banana": 20}.
```

Problem 4. Write a function update_inventory(inventory, new_shipment) that takes two dictionaries as input.

- inventory: A dictionary representing current stock (e.g., "apple": 10, "banana": 5, "orange": 7).
- new_shipment: A dictionary representing new items arriving (e.g., "banana": 10, "orange": 5, "mango": 3).

The function should update the inventory with the quantities from new_shipment. If an item in the shipment is not in the inventory, add it. The function should return the updated inventory. For example

```
inventory = {"apple": 10, "banana": 5, "orange": 7}
new_shipment = {"banana": 10, "orange": 5, "mango": 3}
update_inventory(inventory, new_shipment)
```

should return

```
{"apple": 10, "banana": 15, "orange": 12, "mango": 3}
```

2. Using dictionaries as a counter

Problem 5. Write a function called word_count(sentence) that takes a sentence as input. The function should return a dictionary where the keys are the words in the sentence and the values represent the count of how many times each word appears. For this problem, we do not treat uppercase and lowercase as identical. For example

```
sentence = "Whoever has learned how to listen to trees no longer
   wants to be a tree"
word_count(sentence)
```

should return

```
{"Whoever": 1,
    "has": 1,
    "learned": 1,
    "how": 1,
    "to": 3,
    "listen": 1,
    "trees": 1,
    "no": 1,
    "longer": 1,
    "wants": 1,
    "be": 1,
    "a": 1,
    "tree": 1
}
```

Problem 6. Write a function named mail_count(alist) that takes a list of email strings as input. Each email is a string with the following format:

```
"From stephen.marquard@uct.ac.za Sat Jan 7"
```

The function should return a dictionary where the keys are days of the week, and the values represent the number of emails sent on each day. For example

```
emails = [
    "From stephen.marquard@uct.ac.za Sat Jan 7",
    "From louis@media.berkeley.edu Fri Jan 5",
    "From zqian@umich.edu Fri Jan 5",
    "From rjlowe@iupui.edu Thu Jan 4",
    "From cwen@iupui.edu Sat Jan 7"
]
print(mail_count(emails))
```

should return

```
{"Sat": 2, "Fri": 2, "Thu": 1}
```

For this problem, you should use the split method to get the day of the week from an email.

Problem 7. Write a function called major_count(d) that takes a dictionary as input. In this dictionary, the keys represent student names, and the values indicate their respective majors. The function should return a new dictionary where the keys are the majors and the values are the corresponding counts of students enrolled in each major. For example

```
d = {
    "Alice": "Biology",
    "Bob": "Mathematics",
    "Charlie": "Biology",
    "David": "Computer Science",
    "Eva": "Mathematics",
    "Frank": "Computer Science",
}
major_count(d)
```

should return

```
{
    "Biology": 2,
    "Mathematics": 2,
    "Computer Science": 2
}
```

Problem 8. Write a function named most_frequent(alist) that accepts a list of integers as input and returns the number that occurs most frequently. If there are ties for the most frequent number, return the largest one among them.

```
alist = [1, 3, 2, 3, 4, 2, 5, 3, 2]
most_frequent(alist)
```

should return 3. Note that both 3 and 2 appear 3 times. We pick 3 because it is the bigger number.

Problem 9. Write a function named election_result(votes) that takes a list of candidate names as input and returns the winner of the election. If a candidate receives at least 50% of the votes, that candidate is declared the winner. If no candidate meets this requirement, the function should indicate that a reelection is necessary. For example

```
votes = ["Alice", "Bob", "Alice", "Charlie", "Bob", "Alice", "
   Alice"]
election_result(votes)
```

should return "Alice". On the other hand

```
votes = ["Alice", "Bob", "Alice", "Charlie", "Bob", "Alice", "
   Charlie"]
election_result(votes)
```

should return

```
"Reelection".
```

Problem 10. Write a function named count_digits(n) that takes an integer n as input and returns a dictionary where the keys are the digits (0-9) and the values are the counts of how many times each digit appears in n. For example

```
n = 1122334455
count_digit(n)
```

should return

```
{
    1: 2,
    2: 2,
    3: 2,
    4: 2,
    5: 2
}
```

For this problem, you might want to convert an integer to a string and vice versa. This is not absolutely necessary but it will make the problem a bit easier.

Problem 11. Write a function named product_sales(sales_data) that takes a list of dictionaries representing product sales, where each dictionary contains the product name and the quantity sold. The function should return a dictionary where the keys are product names and the values are the total quantities sold. For example

```
sales_data = [
    {"product": "apple", "quantity": 3},
    {"product": "banana", "quantity": 2},
    {"product": "apple", "quantity": 1},
    {"product": "orange", "quantity": 4},
    {"product": "banana", "quantity": 1}
]
product_sales(sales_data)
```

should return

```
{
    "apple": 4,
    "banana": 3,
    "orange": 4
}
```

3. Nested Dictionaries

Problem 12. Given the following nested dictionaries

```
grades = {
    "Alice": {
        "Math": 85,
        "English": 78,
        "Science": 92
    },
    "Bob": {
        "Math": 90,
        "English": 82,
        "Science": 88
    },
    "Charlie": {
        "Math": 95,
        "English": 97,
        "Science": 91
    }
}
```

(1) Create a dictionary where the keys are student names and the values are their average scores across Math, English, and Science.
(2) Find the list of students who have the highest scores in Math.
(3) Find the list of students whose Math score is higher than their English score.