CSE 231 **Fall 2020**

Computer Project #8

This assignment focuses on the design, implementation and testing of a Python program that uses Dictionary and Sets and Functions.

It is worth 55 points (5.5% of course grade) and must be completed no later than **11:59 PM on Monday, November 16, 2020).**

Assignment Overview

In this assignment, you will practice the use of dictionaries by creating dictionaries of dietary minerals such as calcium, copper, iron, etc. As each food is unique, our aim is to find which foods belong within certain dietary mineral categories to some criteria. In a dictionary, the keys are unique. So, dictionaries are a good choice in this case. Using the FOOD.txt file, create a dictionary with the **Dietary Minerals** as keys and the "**set of food**" as its values. A list could also be used instead of a set here, but then you would have to implement the union, intersection and other set operations all by yourself; operations that are provided with the **set** data structure.

Assignment Background

Diet is essential to our well-being as it does not only affect our physical state but also our mental state as well. When buying food, each food has specific dietary minerals that may be listed on the package, unless they are unprocessed. This project will give you some insight of dietary minerals we consume in typical everyday meals. What is an appropriate data structure for this assignment? Just like in mathematics, there are set operands (&, or |) to show intersection, and union.

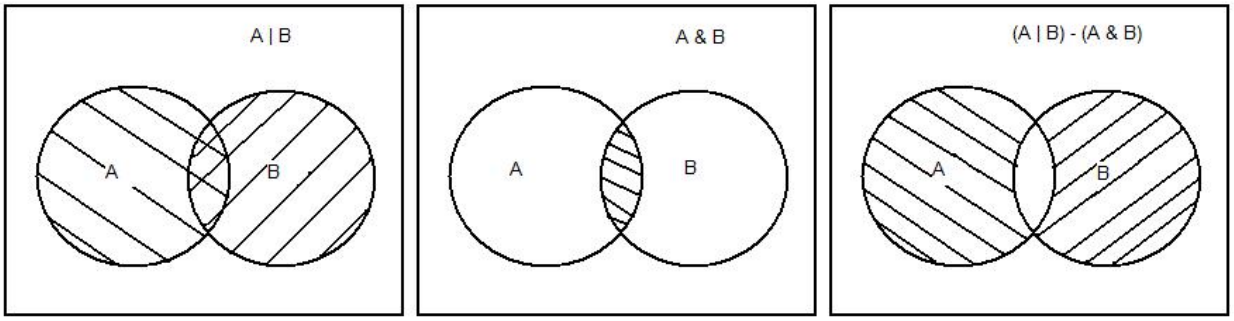
Assignment Specifications

You will develop a Python program that does the following tasks:

1) Given a dictionary of unique dietary minerals, each representing the set of foods with that specific dietary mineral:

   a) Find all the foods that contains one or more of the 3 minerals (A union B union C). This is also represented as the | operand (A | B | C)

   b) Find foods that have all 3 minerals (A intersection B intersection C). This is also represented as the & operand (A & B & C)



2) **Anemia** is a condition in which you lack enough healthy red blood cells to carry adequate oxygen to your body's tissues. Having **anemia** can make you feel tired and weak. Given the dietary mineral "iron", find all the foods that contain iron to promote Anti-Anemia.

`open_file () → fp:`

a. This function repeatedly prompts for a file name until it is opened successfully. If no name is entered, the default is "FOOD.txt".
b. Parameters: none
c. Output: prompt and optionally an error message
d. Returns: a file pointer (fp)

`read_file (fp) -> dictionary:`

a. This function opens and reads the "FOOD.txt" file and builds a mineral dictionary, **`minerals_D`** , with a dietary mineral as a key and a *set* of foods as the value. Although FOOD.txt is a txt file, feel free to use the `csv` module, to read the file. Initialize a mineral dictionary and then pass each line of the file to the function **`build_dictionary(minerals_D, line_list)`** where `line_list` is the list formed from one line of the file. Each line of the file has one food as the first item in the line followed by comma-separated minerals.
b. Parameters: file pointer
c. Output: none
d. Returns: dictionary

`food_and_minerals (minerals_D): → food_list, mineral_list:`

a. This function creates a food list and a nutrition list from **`minerals_D.`** Each list is sorted alphabetically. Note that the lists cannot have repeated items. **Hint: to get unique items in a list, collect items into a set first and then convert that set into a list.**
b. Parameters: dictionary
c. Output: none
d. Returns: two lists
a food list and a mineral list in the order; each list is sorted alphabetically

`build_dictionary (minerals_D,line_list):`

a. This adds a key-value entry to the dictionary `minerals_D` with dietary minerals as key, and a *set* of food as value. Hint: Dictionaries are mutable, if we do not need to return the dictionary.
b. Parameters: dictionary, list
c. Output: none
d. Returns: None

Hint: In python, you are always passing the object itself as input to a function, and the object's mutability determines whether or not it can be modified. Dictionaries are mutable objects. Therefore, you are passing the dictionary to the function, not a copy. Thus, when you modify it, you are also modifying the original copy.

`search (minerals_str,minerals_list,D) → set:`

a. This function takes in an input string , **`minerals_str`**, of three dietary minerals separated by one of two operators: `|` (or) or `&` (and).
   For example: `"sodium & selenium & zinc"`
   We process exactly three minerals to make the assignment easier than if we allowed a variable number of minerals. **Hint: use the `split()` method to split the parameter string by each operator.** Check that the resulting list has exactly three minerals (the **`minerals_list`** is handy for this check.) Then perform the specified operation on the sets of foods for the minerals.
b. If any items in **`minerals_str`** are not minerals or there are not exactly three minerals or the operators are incorrect, `None` is returned.
c. Parameters: string, list, dictionary
d. Output: none
e. Returns: set or `None`

`anti_anemia (minerals_D)→ set:`

a. This function returns a set foods with iron (can be done in one line).
b. Parameter: dictionary
c. Output: none
d. Returns: set

`main():`

This function opens and reads the file, and then displays the foods and minerals read from the file (both displayed *alphabetically*). Next it loops to prompt the user for an input query to find food with specific nutritional content—that is the string passed to the **`search`** function; see the function description above. Print the returned foods in alphabetical order. Continue prompting for input until 'q' is entered. Finally, call the `anti_anemia` function and print the foods returned in *alphabetical order*, comma separated on *one line*. Note that there should not be a comma at the end. **Hint: create a string to print and then slice off the trailing comma before printing. The join() method can help as well.**

Assignment Notes and Hints

1. The coding standard for CSE 231 is posted on the course website:

    http://www.cse.msu.edu/~cse231/General/coding.standard.html

Items 1-9 of the Coding Standard will be enforced for this project.

2. The program will produce reasonable and readable output, with appropriate labels for all values displayed.

3. We provide a `proj08.py` program for you to start with.

4. If you "hard code" answers, you will receive a grade of zero for the whole project. An example of hard coding is to simply print the approximate value of e rather than calculating it and then printing the calculated average.

Suggested Procedure

The last version of your solution is the program which will be graded by your TA.

*You should use the **Mimir** system to back up your partial solutions, especially if you are working close to the project deadline. That is the easiest way to ensure that you won't lose significant portions of your work if your machine fails or there are other last-minute problems.*

Assignment Deliverable

The deliverable for this assignment is the following file:

`proj08.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **Mimir system** before the project deadline.

Test 1

```
Please input a file to use: foo.txt
Invalid filename, please try again
Please input a file to use: Foo.txt
Invalid filename, please try again
Please input a file to use: file.txt
Invalid filename, please try again
Please input a file to use: FOOD.txt

We consider these foods:
Avocado
Bananas
Bell Peppers
Blackberries
Blueberries
Broccoli
Butter
Cabbage
Carrots
Cauliflower
Celery
Chicken Wings
Chickpeas
Cucumber
Eggplant
Eggs
Garlic
Grapes
Green Onions
Green Peas
Ground Beef
Ground Veal
Kale
Lentils
Milk
Mozzarella Cheese
Multi-grain Bread
Mushrooms
Onions
Oranges
Peaches
Plums
Porkleg
Potatoes
Quinoa
Raspberries
```

```
Rice
Spinach
Strawberry
Tomatoes
Turkey
Zucchini

We consider these minerals:
calcium
copper
iron
magnesium
manganese
phosphorus
potassium
salt
selenium
sodium
zinc

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): q

Foods that contain iron, please eat these foods if you are
anemic:
Avocado, Blackberries, Chicken Wings, Chickpeas, Eggs, Green
Onions, Green Peas, Ground Beef, Kale, Lentils, Multi-grain
Bread, Porkleg, Quinoa, Spinach
```

Test 2

```
Please input a file to use: FOOD.txt

We consider these foods:
Avocado
Bananas
Bell Peppers
Blackberries
Blueberries
Broccoli
Butter
Cabbage
Carrots
Cauliflower
Celery
Chicken Wings
```

```
Chickpeas
Cucumber
Eggplant
Eggs
Garlic
Grapes
Green Onions
Green Peas
Ground Beef
Ground Veal
Kale
Lentils
Milk
Mozzarella Cheese
Multi-grain Bread
Mushrooms
Onions
Oranges
Peaches
Plums
Porkleg
Potatoes
Quinoa
Raspberries
Rice
Spinach
Strawberry
Tomatoes
Turkey
Zucchini

We consider these minerals:
calcium
copper
iron
magnesium
manganese
phosphorus
potassium
salt
selenium
sodium
zinc

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): yeah
```

```
Error in input.

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): nah
Error in input.

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): idk
Error in input.

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): Q

Foods that contain iron, please eat these foods if you are
anemic:
Avocado, Blackberries, Chicken Wings, Chickpeas, Eggs, Green
Onions, Green Peas, Ground Beef, Kale, Lentils, Multi-grain
Bread, Porkleg, Quinoa, Spinach
```

Test 3

```
Please input a file to use: FOOD.txt

We consider these foods:
Avocado
Bananas
Bell Peppers
Blackberries
Blueberries
Broccoli
Butter
Cabbage
Carrots
Cauliflower
Celery
Chicken Wings
Chickpeas
Cucumber
Eggplant
Eggs
Garlic
Grapes
Green Onions
Green Peas
```

```
Ground Beef
Ground Veal
Kale
Lentils
Milk
Mozzarella Cheese
Multi-grain Bread
Mushrooms
Onions
Oranges
Peaches
Plums
Porkleg
Potatoes
Quinoa
Raspberries
Rice
Spinach
Strawberry
Tomatoes
Turkey
Zucchini

We consider these minerals:
calcium
copper
iron
magnesium
manganese
phosphorus
potassium
salt
selenium
sodium
zinc

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): manganese|IRON|POtasSIUM
Avocado
Bananas
Bell Peppers
Blackberries
Blueberries
Broccoli
Cabbage
Carrots
```

```
Cauliflower
Celery
Chicken Wings
Chickpeas
Cucumber
Eggplant
Eggs
Garlic
Grapes
Green Onions
Green Peas
Ground Beef
Ground Veal
Kale
Lentils
Milk
Multi-grain Bread
Mushrooms
Onions
Oranges
Peaches
Plums
Porkleg
Potatoes
Quinoa
Raspberries
Rice
Spinach
Strawberry
Tomatoes
Turkey
Zucchini

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): CAlcium|selenium|phOsphorus
Avocado
Bananas
Blackberries
Broccoli
Carrots
Cauliflower
Chicken Wings
Chickpeas
Eggs
Green Onions
Green Peas
```

```
Ground Beef
Ground Veal
Kale
Lentils
Milk
Mozzarella Cheese
Multi-grain Bread
Mushrooms
Onions
Oranges
Peaches
Porkleg
Potatoes
Quinoa
Raspberries
Rice
Spinach
Strawberry
Turkey
Zucchini

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): CAlcium&selenium&phOsphorus
Chickpeas
Milk
Mozzarella Cheese
Multi-grain Bread

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): CAlcium&selenium&phOsphorus
Chickpeas
Milk
Mozzarella Cheese
Multi-grain Bread

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): q

Foods that contain iron, please eat these foods if you are
anemic:
Avocado, Blackberries, Chicken Wings, Chickpeas, Eggs, Green
Onions, Green Peas, Ground Beef, Kale, Lentils, Multi-grain
Bread, Porkleg, Quinoa, Spinach
```

Test 4

```
Please input a file to use: foo.txt
Invalid filename, please try again
Please input a file to use: Foo.txt
Invalid filename, please try again
Please input a file to use: file.txt
Invalid filename, please try again
Please input a file to use: FOOD.txt

We consider these foods:
Avocado
Bananas
Bell Peppers
Blackberries
Blueberries
Broccoli
Butter
Cabbage
Carrots
Cauliflower
Celery
Chicken Wings
Chickpeas
Cucumber
Eggplant
Eggs
Garlic
Grapes
Green Onions
Green Peas
Ground Beef
Ground Veal
Kale
Lentils
Milk
Mozzarella Cheese
Multi-grain Bread
Mushrooms
Onions
Oranges
Peaches
Plums
Porkleg
Potatoes
Quinoa
Raspberries
```

```
Rice
Spinach
Strawberry
Tomatoes
Turkey
Zucchini

We consider these minerals:
calcium
copper
iron
magnesium
manganese
phosphorus
potassium
salt
selenium
sodium
zinc

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): copper &salt &iron
Chicken Wings
Ground Beef
Multi-grain Bread
Porkleg

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): sodium&salt& salt
Butter
Carrots
Celery
Chicken Wings
Eggs
Ground Beef
Milk
Mozzarella Cheese
Multi-grain Bread
Porkleg
Turkey

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): magnesiUM|smoke|Selenium
Error in input.
```

```
Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): manganese&copper|iron
Error in input.

Specify three types of minerals separated by &(and) or |(or)
Please enter 3 minerals using a single operand type (or q to
quit): q

Foods that contain iron, please eat these foods if you are
anemic:
Avocado, Blackberries, Chicken Wings, Chickpeas, Eggs, Green
Onions, Green Peas, Ground Beef, Kale, Lentils, Multi-grain
Bread, Porkleg, Quinoa, Spinach
```

Grading Rubric

```
Computer Project #08  Scoring Summary
General Requirements:
 ( 5 pts) Coding Standard 1-9
      (descriptive comments, function headers, mnemonic identifiers,
      format, etc...)

Implementation:
 ( 4 pts)  open_file function (no Mimir test)

 ( 5 pts)  read_file function

 ( 5 pts)  food_and_minerals function

 ( 7 pts)  search function

 ( 5 pts)  build_dictionary function

 ( 4 pts)  anti_anemia function

 ( 4 pts)  Test 1

 ( 4 pts)  Test 2

 ( 6 pts)  Test 3

 ( 6 pts)  Test 4


Note: hard coding an answer earns zero points for the whole project
-10 points for not using main()
```