Activities: Tuples, Sets, and Frozen Sets

1. Convert the following assignments in the following program into a tuple assignment.

```
sepal_length = 5.1
petal_length = 3.2
sepal_width = 1.7
petal_width = 0.9

print(sepal_length, sepal_width, petal_length, petal_width)
```

2. Write a function *count* that takes as its only parameter a file name and then returns the number of lines, the number of words, and the number of characters (without new lines). Then write a program that asks the user for a file name and then displays the result of *count* in a nice format. Here is an example of the latter:

```
Enter the name of a file: tuples.py
There are 39 lines, 117 words, and 762 characters in the file
tuples.py.
```

3. Tuple unpacking can be used to define functions with arbitrary number of arguments. Assume that we want to create a function multiply that returns the product of an arbitrary number of arguments. We can do so using tuple unpacking in the definition of the function:

```
def mult(*args):
     result = 1
     for x in args:
          result *= args
     return result
```

As you noticed, the asterisk in front of *args* means that *args* is really a tuple. Indeed, in the body of the function, we access the components *x* of *args* iteratively and multiply them to result. Using this construct, write functions with an arbitrary number of arguments that

(A) return the arithmetic mean $\frac{1}{n}(a_1 + a_2 + \ldots + a_n)$ of a number of floating point numbers.

(B) return the geometric average $\sqrt[n]{a_1 \cdot a_2 \cdot \ldots \cdot a_n}$ of a set of numbers.

(C) return the harmonic mean $\frac{n}{\frac{1}{a_1} + \frac{1}{a_2} + \ldots \frac{1}{a_n}}$ of a set of numbers.

4. Write a function that takes as arguments three variables: a left boundary *a*, a right boundary *b*, and a function *f*. The function returns the trapezoid approximation for the definite integral

$$\int_a^b f(x)dx \approx \left(\frac{1}{4}f(a) + \frac{1}{2}f(\frac{a+b}{2}) + \frac{1}{4}f(b)\right)(b-a).$$

Write a program that creates a tuple (*a*,*b*,*f*) and then uses tuple-unpacking when calling the function on it.